AD-A236 341

ARO Report 91-1

TRANSACTIONS OF THE EIGHTH ARMY

CONFERENCE ON APPLIED MATHEMATICS AND COMPUTING

DTIC
ELECTE
JUN 07 1991
S D
C



Approved for public release; distribution unlimited.
The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

Sponsored by

The Army Mathematics Steering Committee

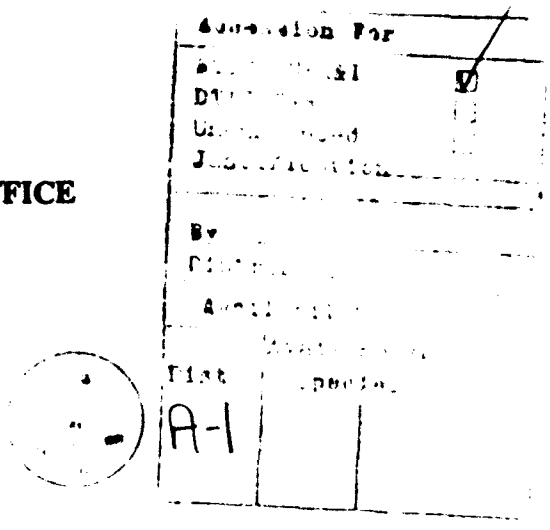
on behalf of

THE ASSISTANT SECRETARY OF THE ARMY FOR

RESEARCH, DEVELOPMENT, AND ACQUISITION

91-00598

91 5 28 051



U.S. ARMY RESEARCH OFFICE

Report No. 91-1

February 1991

TRANSACTIONS OF THE EIGHTH ARMY CONFERENCE ON APPLIED MATHEMATICS AND COMPUTING

Sponsored by the Army Mathematics Steering Committee

HOST

Cornell University
Ithaca, New York
19-22 June 1990

Approved for public release; distributions unlimited.
The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

FOREWORD

The Eighth Army Conference on Applied Mathematics and Computing was held on 19-22 June 1990, at Cornell University, Ithaca, New York. Four years earlier, the fourth in this series of conferences was also held at Cornell University. At that time the Mathematical Sciences Institute was established. At each of these meetings, there were several invited speakers that addressed vital research areas. At the 1990 conference, invited talks covered topics such as new ideas about turbulence, nonlinear dynamical systems, symbolic methods, domain decomposition methods for partial differential equations, and variational methods for free boundary problems with discontinuities and interfaces. Special Sessions were organized on wavelet transforms for image analysis, geometric modeling, symbolic methods, and adaptive methods for high performance architectures. In the eleven Technical Sessions and one Poster Session, more than sixty papers were contributed. An informal discussion session was held to discuss research issues in geometric modeling for vulnerability analysis.

The subcommittee of the Army Mathematics Steering Committee that oversees these conferences was very pleased with the high scientific quality of the contributed papers. Many of these papers provided the attendees a chance to see scientific developments taking place in the Army laboratories. Through these meetings techniques developed at one installation are brought to the attention of scientists at other places, thereby reducing duplication of effort. Another important phase of these meetings is presenting the members of the audience an opportunity to hear nationally known scientists discuss recent developments in their own field. This year the invited speakers, together with the titles of their talks, are listed below.

SPEAKER AND AFFILIATION	TITLE OF ADDRESS
Professor T. Brooke Benjamin **University of Oxford** **Oxford, England**	**New Ideas about the Origins of Turbulence**
Professor Ivar Ekeland **University of Paris IX Ceremade** **Paris, France**	**Variational Methods and Dynamical Systems**
Professor James H. Bramble **Cornell University** **Ithaca, New York**	**On the Analysis of Domain Decomposition Methods for Elliptic Partial Differential Equations**
Professor Sanjoy Mitter **Massachusetts Institute of Technology** **Cambridge, Massachusetts**	**Variational Problems with Free Discontinuities and Nonlinear Diffusions**

SPEAKER AND AFFILIATION (cont)	TITLE OF ADDRESS
Professor Bruno Buchberger **Johannes Kepler University** **Linz, Austria**	**Symbolic Computation:** **Theory and Practice**
Professor George R. Sell **University of Minnesota** **Minneapolis, Minnesota**	**Approximation Dynamics for** **the Navier-Stokes Equations**

The benefits derived from these conferences depend a great deal on the host's Chairman on Local Arrangements. The attendees at this meeting were fortunate to have Professor Anil Nerode, Director of Mathematical Sciences Institute, serving in this capacity. He, together with members of his capable staff, provided all those amenities such as coffee, projection equipment, travel information, etc., needed for an enjoyable and profitable symposium.

TABLE OF CONTENTS

TITLE	PAGE
Foreword	iii
Table of Contents	v
Agenda	ix
GLOBAL EXISTENCE, REGULARITY, AND BOUNDEDNESS FOR THE KURAMOTO-SIVASHINSKY EQUATION IN THIN 2D DOMAINS George R. Sell and Mario Taboada	1
A MATHEMATICAL CARTOON FOR THE DYNAMICS OF FINE STRUCTURE Philip J. Holmes and Pieter J. Swart	11
BREAKUP OF A VISCOUS LIQUID JET S. P. Lin and E. A. Ibrahim	23
COMMUNICATION AND CONTROL IN SPMD PARALLEL NUMERICAL COMPUTATIONS Dan C. Marinescu, John R. Rice, and Emmanuel A. Vavalis	37
RECURSIVE LLSP IN DISTRIBUTED-MEMORY *MULTIPROCESSORS* Jaeyoung Choi and Adam W. Bojanczyk	71
THE HYPERBOLIC SINGULAR VALUE DECOMPOSITION AND APPLICATIONS Ruth Onn, Allan O. Steinhardt, and Adam Bojanczyk	93
TAYLOR SERIES AND THE OVERALL PROPERTIES OF COMPOSITES Oscar P. Bruno	109
DYNAMIC SHEAR BAND DEVELOPMENT IN PLANE STRAIN COMPRESSION OF A BIMETALLIC BODY R. C. Batra and Z. G. Zhu	127
AN ANALYSIS OF THE TORSION SPECIMEN USED IN CONSTITUTIVE MODELING Charles S. White	137
WRAPABILITY OF CURVES ON SURFACES Royce W. Soanes	149
OPTIMAL CONTROL OF DISTRIBUTED PARAMETER SYSTEMS UNDER RESONANT AND UNSTABLE LOADING Iradj G. Tadjbakhsh and Yuan-an Su	165
A CENTRAL LIMIT THEOREM FOR INTEGRAL FUNCTIONALS OF A STATIONARY GAUSSIAN PROCESS Simeon M. Berman	189

THE VOROMOI DIAGRAM FOR THE EUCLIDEAN TRAVELING SALESMAN PROBLEM IS PIECEMEAL QUARTIC AND HYPERBOLIC
T. M. Cronin..... 195

OPTIMAL DIGITAL REDESIGN OF CONTINUOUS-TIME SYSTEMS
Leang S. Shieh and Jian L. Zhang..... 225

EFFECTIVENESS OF A CLASS OF SMART MUNITIONS: A STOCHASTIC MODEL
B. D. Sivazlian and K. Gakis..... 243

ELECTROMAGNETISM AND GRAVITY
Richard A. Weiss..... 265

QUANTUM MECHANICS AND THE BROKEN SYMMETRY OF SPACE AND TIME
Richard A. Weiss..... 319

GAUGE THEORY OF TIME
Richard A. Weiss..... 367

THERMAL RADIATION OF HIGH-T_c SUPERCONDUCTORS
Richard A. Weiss..... 399

SOME RESULTS ON NUMERICAL SOLUTION OF PARTIAL INTEGRO-DIFFERENTIAL EQUATIONS
Lars B. Wahlbin..... 413

ON SOLVING CAUCHY SINGULAR INTEGRAL EQUATIONS BY USING GENERAL QUADRATURE-COLLOCATION NODES
R. P. Srivastav and Fenggang Zhang..... 421

ON A HYPERBOLIC TANGENT QUADRATURE RULE FOR SOLVING SINGULAR INTEGRAL EQUATIONS WITH HADAMARD FINITE PART INTEGRALS
Fenggang Zhang..... 437

DOMAIN DECOMPOSITION METHODS FOR NONSELFADJOINT OPERATORS
Zbigniew Leyk..... 455

INELASTIC MICROSTRUCTURE IN RAPID GRANULAR FLOWS OF SMOOTH DISKS
Mark A. Hopkins and Michel Y. Louge..... 469

CURVE DESIGN AND ANALYSIS USING SPLINES AND WAVELETS
Charles K. Chui..... 471

MULTIPLE FAMILIES OF ENGINEERING ANALYSES INTERROGATING A SINGLE GEOMETRIC MODEL
Michael John Muuss..... 483

PARALLEL ALGORITHMS FOR FLUID INTERFACE PROBLEMS
Yuefan Deng, James Glimm, Yi Wang, and Qiang Zhang..... 497

INTERIOR PRESSURE DISCONTINUITIES IN COMPRESSIBLE VISCOUS STEADY STATE FLOW
Senhuei E. Chen and R. B. Kellogg..... 511

AN EXTENSION OF MESH EQUIDISTRIBUTION TO TIME-DEPENDENT PARTIAL DIFFERENTIAL EQUATIONS
J. M. Coyle..... 521

ADAPTIVE METHODS AND PARALLEL COMPUTATION FOR PARTIAL DIFFERENTIAL EQUATIONS
Rupak Biswas, Messaoud Benantar, and Joseph E. Flaherty... 531

SYNTHESIS OF REAL TIME ACCEPTORS
Amr Fawzy Fahmy and Alan W. Biermann..... 553

A DEDUCTIVE SYSTEM FOR THEORIES OF NONMONOTONIC REASONING
Frank M. Brown and Carlos L. Araya..... 561

A LOGIC PROGRAMMING APPROACH TO NETWORK FLOW ALGORITHMS
Andrew W. Harrell..... 577

DERIVE AS A RESEARCH TOOL
David C. Arney and Jeffrey L. Misner..... 611

COMPUTER ALGEBRA SYSTEMS: CAPABILITIES, APPLICATIONS, AND IMPACT ON EDUCATION
David C. Arney, James P. Cummings, and Lee S. Dewald..... 617

DOMAIN DECOMPOSITION METHODS FOR PROBLEMS WITH UNIFORM LOCAL REFINEMENT IN TWO DIMENSIONS
James H. Bramble, Richard E. Ewing, Rossen R. Parashkevov, and Joseph E. Pasciak..... 625

APPLICATIONS OF ALGEBRAIC LOGIC TO RECURSIVE QUERY OPTIMIZATION
Paul Broome..... 637

TIMES OF THE SIGNS
Moss E. Sweedler..... 649

GEOMETRIC MODELS FOR DEVELOPABLE AND MINIMAL SURFACES
T. F. Chen, G. J. Fix, and R. Kannan..... 657

CONSTRAINT-BASED SPATIAL PROBLEM SOLVING OF TACTICAL FORCE INTERACTIONS USING THE FUNCTIONAL BINARY DECOMPOSITION SPATIAL REPRESENTATION
Douglas Walter J. Chubb..... 679

GEOMETRIC REASONING FOR RECOGNITION OF THREE DIMENSIONAL OBJECT FEATURES
M. Marefat and R. L. Kashyap..... 705

A NEW CLASS OF SCHEMES FOR THE TIME-DEPENDENT STOKES EQUATIONS
John C. Strikwerda..... 733

DIAGONAL IMPLICIT MULTIGRID SOLUTION OF COMPRESSIBLE TURBULENT FLOWS
R. R. Varma and D. A. Caughey..... 741

MULTIGRID DIAGONAL IMPLICIT ALGORITHM FOR COMPRESSIBLE LAMINAR FLOWS
Thomas L. Tysinger and David A. Caughey..... 757

EXTREMUM CONTROL: THE EFFECTS OF ARTIFICIAL VISCOSITY
Culbert B. Laney and David A. Caughey..... 775

CRITICAL TIME-STEP OF VARIOUS NUMERICAL SCHEMES FOR TRANSIENT HEAT CONDUCTION
Rao Yalamanchili and S. Yalamanchili..... 783

SYMBOLIC COMPUTATION: PURE COMPUTER MATHEMATICS
Bruno Buchberger..... 795

PHASE TRANSITIONS AND MAXIMALLY DISSIPATIVE DYNAMIC SOLUTIONS IN THE RIEMANN PROBLEM FOR IMPACT
Thomas J. Pence..... 817

AN INVERSE RIEMANN PROBLEM FOR IMPACT INVOLVING PHASE TRANSITIONS
Thomas J. Pence..... 833

THE INFLATION AND DEFLATION OF A THICK WALLED VISCO-HYPERELASTIC SPHERE
A. R. Johnson, C. J. Quigley, K. D. Weight, and C. Cavallaro..... 847

EVALUATION OF DRAG LOADING MODELS IN OVERTURNING RESPONSE CODES
Aaron Das Gupta..... 859

ELASTIC-PLASTIC ANALYSIS OF A STEEL PRESSURE VESSEL WRAPPED WITH MULTILAYERED COMPOSITES
Peter C.T. Chen..... 871

EXPRESSION SWELL ANALYSIS OF THE COMPUTATION OF MATRIX CHARACTERISTIC POLYNOMIALS
Michael Wester..... 887

A VARIATIONAL METHOD FOR FINDING HOMOCLINIC ORBITS IN THE LARGE
Ivar Ekeland..... 917

EIGHTH ARMY CONFERENCE ON APPLIED MATHEMATICS AND COMPUTING

Cornell University, Ithaca, New York

19-22 June 1990

AGENDA

Tuesday, 19 June 1990

0800 - 1600 **Registration** - Warren Hall, Room 231

0830 - 0900 **Opening Remarks** - Warren Hall, Room 231

0900 - 1000 **General Session I** -

Chairperson: Benjamin E. Cummings, U.S. Army Human Engineering Laboratory, Aberdeen Proving Ground, Maryland

NEW IDEAS ABOUT THE ORIGINS OF TURBULENCE
T. Brooke Benjamin, University of Oxford, Oxford, England

1000 - 1030 **Break**

1030 - 1210 **Technical Session 1 - Dynamical Systems** - Warren Hall, Room 231

Chairperson: Norman Coleman, U.S. Army Armament R&D Center, Picatinny Arsenal, New Jersey

LOCAL DISSIPATIVITY AND ATTRACTORS FOR THE KURAMOTO - SIVASHINSKY EQUATION IN 2D
Mario Taboada, Cornell University, Ithaca, New York, and George Sell, University of Minnesota, Minneapolis, MN

NEW BIFURCATION PROCESSES FOR NONLINEAR PERIODICALLY FORCED EVOLUTION EQUATIONS
M. S. Berger, University of Massachusetts, Amherst, MA

A MATHEMATICAL CARTOON FOR THE DYNAMICS OF FINE STRUCTURE
Philip Holmes, Cornell University, Ithaca, New York

THE DYNAMICS OF THE CHAOTIC MIXING OF RAYLEIGH-TAYLOR UNSTABLE INTERFACES
Qiang Zhang, State University of New York at Stony Brook, New York

Tuesday (Continued)

INSTABILITY OF A VISCOUS LIQUID JET SURROUNDED BY A VISCOUS GAS IN A VERTICAL PIPE
S. P. Lin and E. A. Ibrahim, Clarkson University, Potsdam, New York

******

1030 - 1210 **Technical Session 2 - Parallel and High Speed Computation -** Warren Hall, Room 245

Chairperson: David Arney, United States Military Academy, West Point, New York

COMMUNICATION AND CONTROL IN SPMD PARALLEL NUMERICAL COMPUTATIONS - THE E/T MODEL
John R. Rice and Dan C. Marinescu, Purdue University, West Lafayette, Indiana

DEVELOPING LEVEL-3 BLAS FOR DISTRIBUTED MEMORY SYSTEMS
Anne C. Elster, Cornell University, Ithaca, New York

RLS PROBLEMS ON DISTRIBUTED MEMORY MULTIPROCESSORS
A. W. Bojanczyk and J-Y Choi, Cornell University, Ithaca, New York

THE HYPERBOLIC SINGULAR VALUE DECOMPOSITION AND APPLICATIONS
Allan O. Steinhardt, Adam Bojanczyk, and Ruth Onn, Cornell University, Ithaca, New York

THE ARITHMETIC FOURIER TRANSFORM: A SET OF QUADRATURE RULES WHICH LEAD TO A HIGH PERFORMANCE ARCHITECTURE
Donald W. Tufts, University of Rhode Island, Kingston, RI

1210 - 1330 **Lunch**

1330 - 1530 **Technical Session 3 - Mathematical Models of Materials I -** Warren Hall, Room 245

Chairperson: Douglas Walter J. Chubb, U.S. Army Signal Warfare Laboratory, Warrenton, Virginia

TAYLOR EXPANSIONS AND BOUNDS FOR THE EFFECTIVE CONDUCTIVITY AND THE EFFECTIVE ELASTIC MODULI OF MULTICOMPONENT COMPOSITES AND POLYCRYSTALS
Oscar Bruno, University of Minnesota, Minneapolis, Minnesota

A COMPUTATIONALLY EFFICIENT INVERSION ALGORITHM FOR A THREE-DIMENSIONAL FLAW EMBEDDED IN ADVANCED COMPOSITE MATERIALS
David Sabbagh, Sina Barkeshli, and Harold A. Sabbagh, Sabbagh Associates, Inc., Bloomington, Indiana

A GLOBAL PENALTY-CONSTRAINT FINITE ELEMENT FORMULATION FOR EFFECTIVE STRAIN AND STRESS RECOVERY
A. Tessler and C. Freese, U.S. Army Materials Technology Laboratory, Watertown, Massachusetts

SHEAR BAND DEVELOPMENT IN DYNAMIC PLANE STRAIN PROBLEMS
R. C. Batra, Z. G. Zhu, and Xiang-Tong Zhang, University of Missouri, Rolla, Missouri

AN ANALYSIS OF THE TORSION SPECIMEN USED IN CONSTITUTIVE MODELING
Charles S. White, U.S. Army Materials Technology Laboratory, Watertown, Massachusetts

A LIFTING FUNCTION FOR CURVES ON SURFACES
R. W. Soanes, Benet Weapons Laboratory, Watervliet, New York

******

1330 - 1530 **Technical Session 4 - Optimization and Stochastic Modeling** - Warren Hall, Room 231

Chairperson: Herbert Cohen, U.S. Army Materiel Systems Analysis Activity, Aberdeen Proving Ground, MD

OPTIMAL CONTROL OF DISTRIBUTED PARAMETER SYSTEMS UNDER RESONANT AND UNSTABLE LOADING
Iradj Tadjbakhsh and Yuan-An Su, Rensselaer Polytechnic Institute, Troy, New York

ANALYSIS OF STOCHASTIC ROBUSTNESS IN LINEAR SYSTEMS
Robert F. Stengel, Princeton University, Princeton, NJ

LIMIT THEOREMS FOR FUNCTIONALS OF STATIONARY GAUSSIAN PROCESSES BASED ON A SPECTRAL METHOD
Simeon M. Berman, Courant Institute of Mathematical Sciences, New York University, New York, NY

THE VORONOI DIAGRAM FOR THE EUCLIDEAN TRAVELING SALESMAN PROBLEM IS PIECEMEAL HYPERBOLIC
Terence M. Cronin, U.S. Army Signal Warfare Laboratory, Warrenton, Virginia

OPTIMAL DIGITAL REDESIGN OF CONTINUOUS-TIME SYSTEMS
L. S. Shieh and Jian L. Zhang, University of Houston, Houston, Texas, and Norman Coleman, U.S. Army Armament R&D Center, Picatinny Arsenal, New Jersey

EFFECTIVENESS OF A CLASS OF SMART MUNITIONS: A STOCHASTIC MODEL
B. D. Sivazlian, University of Florida, Gainesville, Florida

1530 - 1600 **Break**

1600 - 1700 **Technical Session 5 - Theoretical and Computational Physics** - Warren Hall, Room 231

Chairperson: Royce Soanes, Benet Weapons Laboratory, Watervliet, New York

MULTILEVEL ALGORITHMS FOR FEYNMAN PATH INTEGRALS
Dov Bai, Cornell University, Ithaca, New York

AN IMPLICIT TIME-STEPPING CONJUGATE GRADIENT SCHEME FOR COMPUTING TRANSIENT ELECTROMAGNETIC FIELDS COUPLING TO A METALLIC ENCLOSURE
Sina Barkeshli, Harold A. Sabbagh, and Denis J. Radecki, Sabbagh Associates, Inc., Bloomington, Indiana

Paper 1: ELECTROMAGNETISM AND GRAVITY
Paper 2: QUANTUM MECHANICS AND THE BROKEN SYMMETRY OF SPACE AND TIME
Richard A. Weiss, U.S. Army Engineer Waterways Experiment Station, Vicksburg, Mississippi

******

1600 - 1700 **Technical Session 6 - Numerical Methods for Integral and Integro-differential Equations** - Warren Hall, Room 245

Chairperson: Jeffrey Misner, United States Military Academy, West Point, New York

SOME RESULTS ON NUMERICAL SOLUTION OF PARTIAL INTEGRO-DIFFERENTIAL EQUATIONS
Lars B. Wahlbin, Cornell University, Ithaca, New York

ON SOLVING CAUCHY SINGULAR INTEGRAL EQUATIONS BY USING GENERAL QUADRATURE-COLLOCATION NODES
Ram P. Srivastav and Fenggang Zhang, State University of New York at Stony Brook, New York

ON A HYPERBOLIC TANGENT QUADRATURE RULE FOR SOLVING SINGULAR INTEGRAL EQUATIONS WITH HADAMARD FINITE PART INTEGRALS
Fenggang Zhang, State University of New York at Stony Brook, New York

******

1600 - 1700 **Poster Session - The three papers for the Poster Session are listed below** - Warren Hall, Room 101

DOMAIN DECOMPOSITION METHOD FOR NONSELF ADJOINT OPERATORS
Zbigniew Leyk, Cornell University, Ithaca, New York

INELASTIC MICROSTRUCTURE IN RAPID GRANULAR FLOWS OF SMOOTH DISKS
Michel Louge, Cornell University, Ithaca, New York and
Mark A. Hopkins, Dartmouth College, Hanover, New Hampshire

COMPUTATION OF NONEQUILIBRIUM ELECTRON SWARM PARAMETERS IN AIR
William T. Wyatt and C. S. Kenyon, U.S. Army Harry Diamond Laboratory, Adelphi, Maryland

------

Wednesday, 20 June 1990

0800 - 1600 **Registration** - Warren Hall, Room 231

0830 - 1030 **Special Session 1 - Geometric Modeling** - Warren Hall, Room 231

Chairperson: Paul Stay, U.S. Army Ballistic Research Laboratory, Aberdeen Proving Ground, Maryland

A PARALLEL ALGORITHM FOR AUTOMATIC MESHING FROM SOLID MODELS BASED ON RECURSIVE SPATIAL DECOMPOSITION
Renato Perucchio, University of Rochester, Rochester, New York

COMPUTATIONAL ASPECTS OF POLYNOMIAL INTERPOLATION IN SEVERAL VARIABLES
Carl de Boor, University of Wisconsin, Madison, Wisconsin

CURVE DESIGN AND ANALYSIS USING SPLINES AND WAVELETS
Charles K. Chui, Texas A&M University, College Station, TX

DRIVING MULTIPLE FAMILIES OF ENGINEERING ANALYSIS BY INTERROGATION OF A SINGLE GEOMETRIC MODEL
Michael John Muuss, U.S. Army Ballistic Research Laboratory, Aberdeen Proving Ground, Maryland

******

0830 - 1030 **Technical Session 7 - Linear and Nonlinear Waves** - Warren Hall, Room 245

Chairperson: A. Tessler, U.S. Army Materials Technology Laboratory, Watertown, Massachusetts

ELASTIC WAVES AND TOTAL REFLECTIONS IN ANISOTROPIC MEDIA
William W. Hager and Rouben Rostamian, University of Maryland, Baltimore, Maryland

VARIOUS SCENARIOS OF DETONATION INITIATION
A. K. Kapila, Rensselaer Polytechnic Institute, Troy, New York

FRONT TRACKING FOR THREE DIMENSIONS
Y. Deng, J. Glimm, Y. Wang, and Q. Zhang, State University of New York at Stony Brook, New York

INTERIOR PRESSURE DISCONTINUITIES IN COMPRESSIBLE VISCOUS FLOW
Bruce Kellogg, University of Maryland, College Park, MD

THE SIMULATION OF SHOCK ACCELERATED FLUID INTERFACES
John W. Grove, State University of New York at Stony Brook, New York

AN EXTENSION OF MESH EQUIDISTRIBUTION TO TIME-DEPENDENT PARTIAL DIFFERENTIAL EQUATIONS
J. M. Coyle, Benet Weapons Laboratory, Watervliet, New York

1030 - 1100 **Break**

1100 - 1200 **General Session II - Warren Hall, Room 231**

Chairperson: John Vasilakis, Benet Weapons Laboratory, Watervliet, New York

VARIATIONAL METHODS AND DYNAMICAL SYSTEMS
Ivar Ekeland, University of Paris IX Ceremade, Paris, France

1200 - 1330 **Lunch**

1330 - 1530 **Special Session 2 - Adaptive Methods for High Performance Architecture - Warren Hall, Room 245**

Chairperson: David A. Caughey, Cornell University, Ithaca, New York

ADAPTIVE METHODS AND PARALLEL COMPUTATION FOR PARTIAL DIFFERENTIAL EQUATIONS
Joseph E. Flaherty, Rensselaer Polytechnic Institute, Troy, New York

PARALLEL DOMAIN DECOMPOSITION WITH LOCAL MESH REFINEMENT
William D. Gropp, Argonne National Laboratory, Argonne, Illinois, and David E. Keyes, Yale University, New Haven, CT

CONTACT-IMPACT BY THE PINBALL ALGORITHM
T. Belytschko, M. O. Neal, and H.-Y. Chiang, Northwestern University, Evanston, Illinois

CHALLENGES FOR ADAPTIVE METHODS IN COMPUTATIONAL BALLISTICS
John Walter, U.S. Army Ballistic Research Laboratory, Aberdeen Proving Ground, Maryland

******

Wednesday (Continued)

1330 - 1530 **Technical Session 8 - Symbolic and Logic Computation** - Warren Hall, Room 231

Chairperson: J. M. Coyle, Benet Weapons Laboratory, Watervliet, New York

SYNTHESIS OF REAL TIME PROGRAMS
Amr F. Fahmy and Alan W. Biermann, Duke University, Durham, North Carolina

A DEDUCTIVE SYSTEM FOR THEORIES OF NONMONOTONIC REASONING
Frank M. Brown and Carlos L. Araya, University of Kansas, Lawrence, Kansas

A LOGIC PROGRAMMING APPROACH TO NETWORK FLOW ALGORITHMS
Andrew Harrell, U.S. Army Engineer Waterways Experiment Station, Vicksburg, Mississippi

DERIVE AS A RESEARCH TOOL
David C. Arney and Jeffrey Misner, United States Military Academy, West Point, New York

COMPUTER ALGEBRA SYSTEMS: CAPABILITIES, APPLICATIONS, AND IMPACT ON EDUCATION
David C. Arney, James Cummings, and Lee S. Dewald, Sr., United States Military Academy, West Point, New York

1530 - 1600 **Break**

1600 - 1700 **General Session III** - Warren Hall, Room 231

Chairperson: Terence M. Cronin, U.S. Army Signal Warfare Laboratory, Warrenton, Virginia

ON THE ANALYSIS OF DOMAIN DECOMPOSITION METHODS FOR ELLIPTIC PARTIAL DIFFERENTIAL EQUATIONS
James H. Bramble, Cornell University, Ithaca, New York

1730 - 1900 **MSI - Hosted Reception** - 700 Clark Hall

Thursday, 21 June 1990

0800 - 1600 **Registration** - Warren Hall, Room 231

0830 - 1030 **Special Session 3A - Symbolic Computation and Applications** - Warren Hall, Room 231

Chairperson: Julian J. Wu, U.S. Army Research Office, Research Triangle Park, North Carolina

HIGH RESOLUTION REAL-TIME VEHICLE DYNAMIC SIMULATION SUPPORT OF A CREW STATION/MOTION BASE SIMULATOR ON AN AD100 COMPUTER
Roger A. Wehage, U.S. Army Tank-Automotive Command, Warren, Michigan

APPLICATIONS OF ALGEBRAIC LOGIC TO RECURSIVE QUERY OPTIMIZATION
Paul Broome, U.S. Army Ballistic Research Laboratory, Aberdeen Proving Ground, Maryland

LEFT RIGHT UP DOWN: HOW SIGNS DETERMINE POSITION AND CONSISTENCY
Moss Sweedler, Cornell University, Ithaca, New York

ADVANCES IN INTEGRATING SYMBOLIC, NUMERIC AND GRAPHICS COMPUTING
Paul S. Wang, Kent State University, Kent, Ohio

******

0830 - 1030 **Technical Session 9 - Computational Geometry** - Warren Hall, Room 245

Chairperson: John Walter, U.S. Army Ballistic Research Laboratory, Aberdeen Proving Ground, Maryland

GEOMETRIC MODELS FOR DEVELOPABLE AND MINIMAL SURFACES
T. F. Chen, George J. Fix, and T. Kannan, University of Texas at Arlington, Texas

ALGEBRO-GEOMETRIC AND DIFFERENTIAL GEOMETRIC METHODS APPLIED TO SPLINES AND SYSTEMS OF PARTIAL DIFFERENTIAL EQUATIONS
Peter F. Stiller, Texas A&M University, College Station, TX

DISTANCE COMPUTATIONS BETWEEN PARAMETRICALLY AND IMPLICITLY REPRESENTED SUB-MANIFOLDS OF THE N-DIMENSIONAL EUCLIDEAN SPACE
Franz-Erich Wolter, Massachusetts Institute of Technology, Cambridge, Massachusetts

CONSTRAINT-BASED SPATIAL PROBLEM SOLVING OF TACTICAL FORCE INTERACTIONS USING THE FUNCTIONAL BINARY DECOMPOSITION SPATIAL REPRESENTATION
Douglas Walter J. Chubb, U.S. Army CECOM Center for Signals Warfare, Warrenton, Virginia

GEOMETRICAL REASONING FOR RECOGNITION OF THREE DIMENSIONAL OBJECT FEATURES
R. L. Kashyap and M. Marefat, Purdue University, West Lafayette, Indiana

THE DISTRIBUTION OF THE NUMBER AND LENGTH OF SHADOWS
Shelemyahu Zacks, State University of New York at Binghamton, New York

1030 - 1100 **Break**

1100 - 1200 **General Session IV - Warren Hall, Room 231**

Chairperson: Joseph E. Flaherty, Rensselaer Polytechnic Institute, Troy, New York

VARIATIONAL PROBLEMS WITH FREE DISCONTINUITIES AND NONLINEAR DIFFUSIONS
Sanjoy Mitter, Massachusetts Institute of Technology, Cambridge, Massachusetts

1200 - 1330 **Lunch**

1330 - 1530 **Special Session 3B - Symbolic Computation and Applications - Warren Hall, Room 231**

Chairperson: Julian J. Wu, U.S. Army Research Office, Research Triangle Park, North Carolina

POLYHEDRAL METHODS IN COMPUTATIONAL ALGEBRA
Bernd Sturmfels, Cornell University, Ithaca, New York

SYMBOLIC COMPUTATION FOR DERIVATION OF NONLINEAR MATERIAL MATRICES IN FINITE ELEMENT ANALYSIS
T. Y. P. Chang, University of Akron, Akron, Ohio and H. Q. Tan and S. M. Arnold, NASA Lewis Research Center, Cleveland, Ohio

TITLE TO BE ANNOUNCED
David Wood, University of Delaware, Newark, Delaware

ON THE APPLICATION OF COMPUTER ALGEBRA TO A PROBLEM OF FORCED NONLINEAR OSCILLATIONS
Julian J. Wu, U.S. Army Research Office, Research Triangle Park, North Carolina, M. A. Hussain, General Electric Corporation, Schenectady, New York, and Ben Noble, Cumbria, England

******

1330 - 1530 **Technical Session 10 - Computational Methods for Nonlinear Partial Differential Equations** - Warren Hall, Room 245

Chairperson: Peter C.T. Chen, Benet Weapons Laboratory, Watervliet, New York

A SECOND-ORDER ACCURATE SCHEME FOR THE INCOMPRESSIBLE NAVIER-STOKES EQUATIONS
John C. Strikwerda, University of Wisconsin at Madison, WI

DIAGONAL IMPLICIT MULTIGRID SOLUTION OF COMPRESSIBLE TURBULENT FLOWS
R. R. Varma and David A. Caughey, Cornell University, Ithaca, New York

MULTIGRID DIAGONAL IMPLICIT ALGORITHM FOR COMPRESSIBLE LAMINAR FLOWS
Thomas L. Tysinger and David A. Caughey, Cornell University, Ithaca, New York

DIRECT EXTREMUM CONTROL AND TOTAL VARIATION DIMINISHING CONDITIONS
Culbert B. Laney and David A. Caughey, Cornell University, Ithaca, New York

BLOCK MULTIGRID IMPLICIT SOLUTION OF THE THREE-DIMENSIONAL EULER EQUATIONS OF COMPRESSIBLE FLUID FLOW
Yoram Yadlin and David A. Caughey, Cornell University, Ithaca, New York

Thursday (Continued)

CRITICAL TIME-STEP OF VARIOUS NUMERICAL SCHEMES FOR TRANSIENT HEAT CONDUCTION
Rao Yalamanchili, U.S. Army Armament R&D Center, Picatinny Arsenal, New Jersey

1530 - 1600 **Break**

1600 - 1700 **General Session V - Warren Hall, Room 231**

Chairperson: Roger A. Wehage, U.S. Army Tank-Automotive Command, Warren, Michigan

SYMBOLIC COMPUTATION: THEORY AND PRACTICE
Bruno Buchberger, Johannes Kepler University, Linz, Austria

Friday, 22 June 1990

0800 - 1000 **Registration - Warren Hall, Room 231**

0830 - 1030 **Special Session 4 - Wavelets - Warren Hall, Room 231**

Chairperson: Gerald R. Andersen, U.S. Army Research Office, Research Triangle Park, North Carolina

RECONSTRUCTION OF FUNCTIONS FROM THE WAVELET TRANSFORM LOCAL MAXIMA
Stephane Mallat and Sifen Zhong, Courant Institute of Mathematical Sciences, New York, New York

ACTIVE PERCEPTION USING WAVELET REPRESENTATION
Ruzena Bajcsy, University of Pennsylvania, Philadelphia, PA

******

0830 - 1030 **Technical Session 11 - Mathematical Methods of Materials II - Warren Hall, Room 245**

Chairperson: Paul Broome, U.S. Army Ballistic Research Laboratory, Aberdeen Proving Ground, Maryland

AN INVERSE RIEMANN PROBLEM FOR IMPACT INVOLVING PHASE TRANSITIONS
Thomas J. Pence, Michigan State University, East Lansing, MI

THE INFLATION AND DEFLATION OF A THICK WALLED VISCO-HYPERELASTIC SPHERE
A. R. Johnson, C. J. Quigley, K. D. Weight, and C. Cavallaro, U.S. Army Materials Technology Laboratory, Watertown, Massachusetts, and D. L. Cox, Naval Underwater Systems Center, New London, Connecticut

FRUSTRATION IN FERROMAGNETIC MATERIALS
Richard James and David Kinderlehrer, University of Minnesota, Minneapolis, Minnesota

MODELING PLASTIC ANISOTROPY FOR PLANAR POLYCRYSTALLINE AGGREGATES USING ORIENTATION TENSORS
Vincent C. Prantil, Paul R. Dawson, and James T. Jenkins, Cornell University, Ithaca, New York

DYNAMIC OVERTURNING RESPONSE OF ARMY VEHICLES SUBJECTED TO SIDE-ON BLAST OVERPRESSURES
Aaron Das Gupta, U.S. Army Ballistic Research Laboratory, Aberdeen Proving Ground, Maryland

ELASTIC-PLASTIC ANALYSIS OF A STEEL PRESSURE VESSEL WRAPPED WITH MULTI-LAYERED COMPOSITES
Peter C.T. Chen, Benet Weapons Laboratory, Watervliet, NY

1030 - 1100 **Break**

1100 - 1200 **General Session VI - Warren Hall, Room 231**

Chairperson: Jagdish Chandra, U.S. Army Research Office, Research Triangle Park, North Carolina

APPROXIMATION DYNAMICS FOR THE NAVIER-STOKES EQUATIONS
George R. Sell, University of Minnesota, Minneapolis, MN

1200 - 1215 **ADJOURNMENT**

Global Existence, regularity, and boundedness for the Kuramoto-Sivashinsky equation in thin 2D domains*

George R. Sell[1] and Mario Taboada[2,3]

[1] School of Mathematics, University of Minnesota, Minneapolis, MN 55455
[2] Mathematical Sciences Institute, Cornell University, Ithaca NY 14853
[3] Department of Mathematics, University of Southern California, Los Angeles, CA 90089

* Supported in part by the Army Research Office through the Mathematical Sciences Institute, Cornell University, and by the Applied and Computational Mathematics Program/DARPA

Abstract

We study the global existence, regularity and boundedness of solutions of the two-dimensional periodic Kuramoto-Sivashinsky equation with a thin periodicity rectangle $\Omega_\varepsilon = [0,2\pi]\times[0,2\pi\varepsilon]$. The main result is that for a large set of initial conditions, the solution exists and is uniformly bounded. This implies the existence of a local compact attractor with a basin of attraction which expands to the whole space as $\varepsilon \to 0$. We state various theorems and give only brief indications of the proofs. A full treatment will be published elsewhere.

1. Introduction

Our goal in this paper is to present some results on the global asymptotic behavior of the Kuramoto-Sivashinsky (K-S) equation

(1.1)

$$u_t + \nu \Delta^2 u + \Delta u + \frac{1}{2} |\nabla u|^2 = 0 ,$$

in spatial dimension two, where $u = u(y,t) = u(y_1,y_2,t)$ satisfies the periodic boundary condition

(1.2) $u(y_1 + 2\pi, y_2,t) = u(y_1,y_2 + 2\pi\varepsilon,t) = u(y_1,y_2,t)$ for all y in R^2 and $t \geq 0$

and the periodic initial condition

(1.3) $u(y,0) = u_0(y)$.

Here $0 < \varepsilon \leq 1$ is a small parameter, so that the basic periodicity cell $[0,2\pi] \times [0,2\pi\varepsilon] = \Omega_\varepsilon$ is a thin domain.The dissipativity of the general two (and higher) dimensional problem has been open for some time, the essential difficulty being the lack of a proof of the existence of an absorbing set. In fact, if such a set exists, then there exists a global attractor, and one can prove very precise regularity results (cf.[NST2], [Te]).

The approach we shall adopt is based on the intuitive idea that (1.1)-(1.3) should be close to one-dimensional, and is close to the methods introduced by [HR1,2] and [RS]. The odd-periodic case was studied by [NST1,2],[FNST].The same authors treat

[NST1] a Neumann type of problem without symmetry, and the rigid Dirichlet problem is studied by [Ta], again without any symmetry assumptions.The general periodic one-dimensional problem has only recently been proved to be dissipative [I].

We start by changing (1.1) into a system by means of the hodograph transformation

$$D_{y_i} u = U_i \ , \ i = 1,2 \qquad \text{and } U = (U_1, U_2) \ .$$

Equality of the mixed partials requires the condition curl U = 0. Notice also that the average of U over the periodicity cell is zero. After this transformation we obtain

(1.4)

$$U_t + \nu \Delta^2 U + \Delta U + (U.\nabla)U = 0 \ .$$

Functional Setting

A point in $\Omega_\varepsilon = [0,2\pi] \times [0, 2\pi\varepsilon]$ will be denoted by $y = (y_1,y_2)$. Let us define $x = (x_1,x_2)$ by $x_1 = y_1$ and $x_2 = \varepsilon^{-1} y_2$.This maps Ω_ε onto the square $Q_2 = [0,2\pi]^2$. Define also the rescaled operators

$$\nabla_\varepsilon = (D_{x_1}, \varepsilon^{-1} D_{x_2}) \ , \Delta_\varepsilon = D^2_{x_1} + \varepsilon^{-2} D^2_{x_2} \ .$$

Note that these become singular when ε is small. Finally, we define a new function u = u(x) by u(x) = U(y), where x and y are related as above. In the sequel, we write (.,.) and | . | for the standard inner product and norm in $L^2(Q_2)$.

Given $u \in L^2(Q_2)$ we define the *projection operator* M as follows:

$$v = Mu \ , \text{ where } v = v(x_1) = \frac{1}{2\pi} \int_0^{2\pi} u(x)\, dx_2 \ \text{ and } \ w = (I - M)u.$$

This averaging operation maps $L^2(Q_2)$ onto the closed subspace formed by functions of x_1 alone and it is an orthogonal projection.The complementary projection I - M defines w = (I - M)u. Notice that Mw = 0, so that w has zero average with respect to x_2. Also, M and I - M commute with the Laplacian on its domain , and they preserve periodicity. After rescaling, Eq. (1.4) becomes

$$\begin{aligned} &u_t + \nu \Delta_\varepsilon^2 u + \Delta_\varepsilon u + (u \,.\, \nabla_\varepsilon)u = 0 \\ &u = u(x,t) \quad 2\pi \text{ periodic in } x_1, x_2 \ , \\ &u(x,0) = u_0(x) \ , \text{ also } 2\pi \text{ periodic.} \end{aligned} \qquad (1.5)$$

Without loss of generality, we shall set $\nu = 1$.

Let us now apply the projections M and I - M to (1.5). This gives

(1.6)

$$v_t + \Delta_\varepsilon^2 v + \Delta_\varepsilon v + M [(u . \nabla_\varepsilon) u] = 0$$

$$w_t + \Delta_\varepsilon^2 w + \Delta_\varepsilon w + (I - M) [(u . \nabla_\varepsilon) u] = 0 .$$

We wish to associate with (1.6) a simpler problem which, for small ε, will turn out to govern to a large extent the dynamics of the system. We thus define the *reduced problem* to be the one obtained from (1.6) by setting $(v,w) = (\bar{v},0)$ in (1.6) and taking as initial condition the projection of u_0 onto the v-space.This gives

(1.7)

$$\bar{v}_t + \Delta_\varepsilon^2 \bar{v} + \Delta_\varepsilon \bar{v} + (\bar{v} . \nabla_\varepsilon) \bar{v} = 0$$

$$\bar{v} (x,0) = v_0 = Mu_0 .$$

Let us write $\bar{v}_i$, i = 1,2 for the components of $\bar{v}$. Then these satisfy

$$\bar{v}_{1,t} + \Delta_\varepsilon^2 \bar{v}_1 + \Delta_\varepsilon \bar{v}_1 + \bar{v}_1 D_{x_1} \bar{v}_1 = 0$$

$$\bar{v}_{2,t} + \Delta_\varepsilon^2 \bar{v}_2 + \Delta_\varepsilon \bar{v}_2 + \bar{v}_1 D_{x_1} \bar{v}_2 = 0 .$$

Notice that the first component satisfies the one-dimensional K-S equation (the dependence of the equations on ε is illusory), while the second one satisfies a *linear* equation. By the results of [I], [NST2], if the initial condition $u_0 \in H^k$, then there exists an absorbing set in H^s for $0 \le s \le k$. Moreover, the same property holds, *even for dimensions 2 and 3* , as soon as a solution is uniformly bounded in time in L^2.

The above dissipativity result for the 1D problem, together with general theorems on attractors of asymptotically smooth nonlinear semigroups [Ha] imply the existence of a global attractor of finite Hausdorff and fractal dimensions [I], [NST2].

The evolutionary equation

In what follows, we will always assume that the initial condition has zero average over the periodicity cell; it is then easy to prove that the solutions have the same property for all times for which they exist. Let us define the unbounded linear operator

A_ε by $A_\varepsilon u = \Delta_\varepsilon^2 u$ on the closed subspace X of $L^2(Q_2)$ consisting of functions with zero average and whose (ε-rescaled) curl is zero. The operator defined above is self-adjoint, has a dense domain and, when restricted to the subspace of periodic functions with zero average, is positive definite. It is also well-known that its resolvent is compact. These properties imply that the fractional powers of A_ε are well-defined. In particular, one has

$$A_\varepsilon^{\frac{1}{2}} = -\Delta_\varepsilon \text{ , and } |A_\varepsilon^{\frac{1}{4}} u|^2 = |\nabla_\varepsilon u|^2 .$$

We will also introduce rescaled versions of the classical bilinear and trilinear forms associated with the nonlinear term in our equation ([Te] , [Li]). We thus consider

$$B_\varepsilon(u,v) = (u.\ \nabla_\varepsilon)\, v \text{ , and } b_\varepsilon(u,v,w) = (B_\varepsilon(u,v),\ w)$$

The trilinear form satisfies the following inequality, where C is independent of ε:

(1.8)

$$|b_\varepsilon(u,v,w)| \le C\, |u|^{\frac{1}{2}} |A_\varepsilon^{\frac{1}{4}} u|^{\frac{1}{2}} |A_\varepsilon^{\frac{1}{4}} v|\, |w|^{\frac{1}{2}} |A_\varepsilon^{\frac{1}{4}} w|^{\frac{1}{2}}$$

In particular, when u = v this gives

(1.9)

$$|b_\varepsilon(u,u,w)| \le C\, |u|^{\frac{1}{2}} |A_\varepsilon^{\frac{1}{4}} u|^{\frac{3}{2}} |w|^{\frac{1}{2}} |A_\varepsilon^{\frac{1}{4}} w|^{\frac{1}{2}} .$$

With the above definitions, equation (1.5) can be written in evolutionary form as follows

$$u' + A_\varepsilon u - A_\varepsilon^{\frac{1}{2}} u + B_\varepsilon(u,u) = 0$$

and the projections v and w satisfy the system

(1.10)

$$v' + A_\varepsilon v - A_\varepsilon^{\frac{1}{2}} v + M\, B_\varepsilon(u,u) = 0.$$

$$w' + A_\varepsilon w - A_\varepsilon^{\frac{1}{2}} w + (I - M) B_\varepsilon(u,u) = 0.$$

2. Regularity results for a thin domain

We now turn to the problem of global existence and regularity of solutions of the K-S equation on a thin domain (local existence and uniqueness are a consequence of classical theorems for sectorial evolutionary equations [P]). Our strategy is to prove existence and regularity for solutions with initial conditions in a large set, over intervals whose length is independent of the initial condition chosen in this set. This enables us to patch up local solutions and thus construct a global solution . In order to do this we shall need some results on the 1D equation, as well as on the reduced 2D equations.

The dynamics of the reduced 2D problem

We recall the reduced 2D Kuramoto-Sivashinsky equations

(2.1)

$$\bar{v}_{1,t} + \Delta_\varepsilon^2 \bar{v}_1 + \Delta_\varepsilon \bar{v}_1 + \bar{v}_1 D_{x_1} \bar{v}_1 = 0$$

$$\bar{v}_{2,t} + \Delta_\varepsilon^2 \bar{v}_2 + \Delta_\varepsilon \bar{v}_2 + \bar{v}_1 D_{x_1} \bar{v}_2 = 0 .$$

We observe that (2.1) is independent of ε and $\bar{v}_1$, $\bar{v}_2$ are independent of x_2. Notice that $\bar{v}_1$ satisfies the 1D K-S equation. In order to study the dynamics of the system, we first solve the equation for $\bar{v}_1$. We note that the absorbing property holds for $\bar{v}_1$ in $L^2[0,2\pi]$. Also, since the ε-curl of $\bar{v}$ is zero, $\bar{v}_2$ is also independent of x_1, hence this function depends only on time. It then follows from the equation satisfied by $\bar{v}_2$ that it must be constant . However, its average is zero, hence $\bar{v}_2$ is identically zero. Therefore, the reduced 2D K-S equation has a global attractor, namely $A\times\{0\}$.

Growth Estimates for the reduced equation

Let us recall some results regarding the one-dimensional K-S equation

(2.2)

$$u_t + \Delta^2 u + \Delta u + uu_x = 0.$$

We know ([I], [NST1,2]) that (2.2) has a global attractor A in X, and that there exist constants ρ_0 and ρ_1 such that

$$\limsup_{t\to\infty} |S(t)u_0|^2 \le \rho_0^2$$

$$\limsup_{t\to\infty} |A^{\frac{1}{4}} S(t)u_0|^2 \le \rho_1^2 ,$$

where S(t) is the solution semigroup of (2.2). Moreover, there exist absorbing balls of radii $2\rho_0$ and $2\rho_1$ in X and $X^{1/4} = D(A^{1/4})$. The next results are of a technical nature, and make more precise the above absorbing properties.

Lemma 2.1

Consider Eq. (2.2) in 1D with a periodic boundary condition and initial condition. Then there exist absolute positive constants γ and L, and a function D, real analytic in $|u_0|$, such that the solution semigroup satisfies the estimate

$$|A^{1/4}S(t)u_0|^2 \le e^{-2\gamma t} D + L \text{ for } t \ge 0.$$

This result is an easy consequence of the dissipativity estimates for (2.2) [NST1,2].

Lemma 2.2

Given k such that $0 < k < 1$, there exist absolute positive constants b_i, $i = 1,2$, such that for all $u_o \in X^{1/4}$ one has

$$|A^{1/4}S(t)u_0|^2 \le L + k\,|A^{1/4}u_0|^2 \text{ for } t \ge T_0, \text{ where L is a constant and}$$

$$T_0 = b_1 \exp(b_2 |A^{1/4}u_0|^4).$$

Lemma 2.3

Assume that $B_1 > L$, with L as above. Then there exists a $K_0 \ge 1$ such that for all η such that $0 \le \eta \le 1$ and for all $u_0 \in D(A^{1/4})$ satisfying $L < |A^{1/4}u_0|^2 \le B_1^2$ the following holds for $t \ge 0$:

$$|A^{1/4}S(t)u_0|^2 \le K_0\,|A^{1/4}u_0|^2\,\eta^{*-2} \text{ where } \eta^{*-2} = \exp(a_1 \exp(a_2 B_1^4 \eta^{-4}))$$

and a_1, a_2 are constants.

The proofs of these results are very similar to those in [RS].

The growth condition "G"

We will often consider functions $\eta = \eta(\varepsilon)$ for which η^{-1} blows up at a certain "sufficiently slow" rate as $\varepsilon \to 0^+$. Instead of giving a list of the requirements on η, we note that a function that satisfies these growth conditions is, for example,

$$\eta(\varepsilon) = \{A + B \log\log\log\log(C\,\varepsilon^{-1})\}^{-\frac{1}{4}} \text{ for appropriate constants A,B,C.}$$

Our first result is a lower bound on the blow-up time of solutions. Let us define, for a given initial condition u_0, the blow-up time in $D(A_\varepsilon^{1/4})$:

$$T^* = T^*(u_0) = \sup \{ t : \sup_{0 \le s \le t} | A_\varepsilon^{\frac{1}{4}} u(s) | < \infty \} .$$

Then, by a fairly crude estimate, one has

$$T^* \ge \frac{K}{| A_\varepsilon^{\frac{1}{4}} u_0 |^2} .$$

where K is a constant, independent of the initial conditions. As an easy consequence we have :

Lemma 2.4

Let R_0 and $N > 1$ be given . Then, for any $u_0 \in D(A^{1/4})$ such that $|A^{1/4}u_0|^2 \le R_0$ the following holds:

$$|A^{\frac{1}{4}} S(t)u_0|^2 \le NR_0 \quad \text{for } 0 \le t \le \frac{N-1}{N} \cdot \frac{K}{R_0} \text{ where K is a constant, independent of } u_0.$$

For simplicity, we shall henceforth assume that $K = 1$.

In the following Lemma, we give a short-time result which will be essential in the proof of the main global existence theorem.

Lemma 2.5

Let $B_0 > 0$, $C_0 > 0$ be given, and consider a function η satisfying the growth hypothesis (G) indicated above. Then there exist ε_0 , $0 < \varepsilon_0 \le 1$, $B_1 \ge B_0$, $C_1 > C_0$ and $T_1 = T_1(\varepsilon) > 0$ such that the following hold for all ε such that $0 < \varepsilon < \varepsilon_0$:

$$\text{If } | A_\varepsilon^{\frac{1}{4}} v_0|^2 \le B_0^2 \eta^{-2} \quad \text{and } |A_\varepsilon^{\frac{1}{4}} w_0|^2 \le C_0^2 \varepsilon^{-1}\eta^{-1} \text{ , then}$$

$$|A_\varepsilon^{\frac{1}{4}} v(T_1)|^2 \le B_1^2 \eta^{-2} \quad \text{and } |A_\varepsilon^{\frac{1}{4}} w(T_1)|^2 \le C_1^2 \varepsilon^2 \eta^{-2} .$$

The quantities ε_0, B_1, C_1, depend on B_0, C_0, *but not on* ε, and $T_1(\varepsilon) \to 0$ as $\varepsilon \to 0^+$. The proof follows by taking the inner product of equations (1.6) for v and w with $A^{1/2}v$ and $A^{1/2}w$, respectively. The growth rate of η plays a particularly important role in the estimates.

As a direct consequence of Lemma 2.5 we have the following result:

Lemma 2.6

Let $B_0 > 0$ and $C_0 > 0$ be given, and let η satisfy hypothesis (G). Then there exist ε_0, $0 < \varepsilon_0 < 1$, $B_1 > B_0$, $C_1 > C_0$ and $T_1 > 0$ such that the following holds for all ε satisfying $0 < \varepsilon < \varepsilon_0$:

$$\text{If } |A_\varepsilon^{\frac{1}{4}} v_0|^2 \le B_0^2 \eta^{-2} \text{ and } |A_\varepsilon^{\frac{1}{4}} w_0|^2 \le C_0^2 \varepsilon^{-1} \eta^{-1} \text{ then}$$

$$|A_\varepsilon^{\frac{1}{4}} v(T_1)|^2 \le B_1^2 \eta^{-2} \text{ and } |A_\varepsilon^{\frac{1}{4}} w(T_1)|^2 \le C_1^2 \varepsilon.$$

Lemma 2.7

Fix B_1 and C_1 and let $u_0 = (v_0, w_0)$ be chosen such that

$$|A_\varepsilon^{\frac{1}{4}} v_0|^2 \le B_1^2 \eta^{-2} \quad \text{and} \quad |A_\varepsilon^{\frac{1}{4}} w_0|^2 \le C_1^2 \varepsilon \ .$$

Let $K_0 \ge 1$ be given by Lemmas 2.2 and 2.3 with, say, $k = 1/8$ and set $N = 4K_0$, and define $T_0 = T_0(\varepsilon) = b_1 \exp(b_2 B_1^4 \eta^{-4})$ so that one has

$$|A^{\frac{1}{4}} v(t)|^2 \le L + \frac{1}{8} |A^{\frac{1}{4}} v_0|^2 \ , t \ge T_0 \ .$$

Next, define τ_N by

$$\tau_N = \sup \{ \tau > 0 : |A_\varepsilon^{\frac{1}{4}} u(t)|^2 \le N D_4^2 \eta^{*-2} \eta^{-2} \text{ for } 0 \le t \le \tau \}$$

where $D_4^2 = B_1^2 + C_1^2$.

Then there exists ε_0, $0 < \varepsilon_0 < 1$, such that for all ε satisfying $0 < \varepsilon < \varepsilon_0$ the following hold

$$T_0 \le \tau_N$$

$$|A_\varepsilon^{\frac{1}{4}} v(T_0)|^2 \le \frac{3}{4} B_1^2 \eta^{-2}$$

$$|A_\varepsilon^{\frac{1}{4}} w(T_0)|^2 \le C_1^2 \varepsilon \ .$$

The estimate for w is straightforward, while the estimate for v relies heavily on a comparison between v and the corresponding solution of the reduced problem.

The existence and ultimate boundedness of solutions are now an easy consequence of this Lemma.

Theorem 2.8

Consider the Kuramoto-Sivashinsky equation with periodic boundary conditions given by the thin periodicity cell $[0,2\pi]\times[0,2\pi\varepsilon]$ *and a periodic initial condition.Then*

there exists ε_0, $0 < \varepsilon_0 < 1$ *and a constant* B_0 *and on this range there are real-valued functions* $R(\varepsilon)$ *and* $K(\varepsilon)$ *such that* $R(\varepsilon) > 0$, $K(\varepsilon) \geq 1$ *and* $R(\varepsilon) \to \infty$ *as* $\varepsilon \to 0^+$ *and such that for all* $U_0 \in D(A^{1/4})$ *such that* $|U_0|_{1/4} \leq R(\varepsilon)$ *one has* $U(t) \in D(A^{1/4})$ *for all* $t \geq 0$ *and*

$$|U(t)|_{\frac{1}{4}} \leq K(\varepsilon)\, R(\varepsilon) \quad ,$$

$$\text{limsup}_{t \to \infty} |U(t)|_{\frac{1}{4}} \leq B_0$$

for all $t \geq 0$.

This result follows by repeatedly applying Lemma 2.8. Notice that we are stating the result in terms of the original function U.

Existence of a local attractor

We recall that a set **A** is a *local attractor* for a nonlinear semigroup $\{S(t)\}$ if it is compact, invariant, and there exists a bounded neighborhood B of **A** such that **A** attracts B. Let us also recall the following result :

Lemma 2.10 ([Ha], Lemma 3.2.1)

Let $\{S(t), t \geq 0\}$ be an asymptotically smooth semigroup in a Banach space X and let B be a nonempty set in X such that the semiorbit $\gamma^+(B)$ is bounded. Then $\omega(B)$ is nonempty, compact, invariant, and it attracts B. If B is connected, so is $\omega(B)$.

By using this result for $B = B_\varepsilon = \{U : |U|_{1/4} \leq R(\varepsilon)\}$ we see that $\mathbf{A}_\varepsilon = \omega(B_\varepsilon)$ is a local attractor whose basin of attraction contains at least the set B_ε.

References

[Ha] J.K. Hale (1988), *Asymptotic Behavior of Dissipative Systems*, AMS Math. Surveys and Monographs, # 25.

[HR1] J.K. Hale and G. Raugel (1989), *Reaction Diffusion equations on thin domains*, Preprint, Georgia Tech.

[HR1] J.K. Hale and G. Raugel (1989), *A damped hyperbolic equation on thin domains* Preprint, Georgia Tech.

[I] Ju. Il'yashenko (1990), *Global analysis of the phase portrait for the Kuramoto-Sivashinsky equation*, IMA Preprints Series, # 665

[Li] J.L. Lions(1969), *Quelques Methodes de Resolution des Problemes aux Limites non lineaires*, Gauthier-Villars, Paris.

[NST1] B. Nicolaenko, B. Scheurer and R. Témam (1985), *Some global dynamical properties of the Kuramoto-Sivashinsky equations: nonlinear stability and attractors*, Physica D 16 , 155-183.

[NST2] B. Nicolaenko, B. Scheurer and R. Témam (1987) , *Some global dynamical properties of a class of pattern formation equations*, IMA Preprints Series, # 381, Univ. of Minnesota.

[P] A. Pazy (1983), *Semigroups of linear operators and applications to partial differential equations*, Applied Math. Sci. 44, Springer-Verlag.

[RS] G. Raugel and G. Sell (1990) , *Navier Stokes in thin 3D domains: global regularity of solutions,I* IMA Preprints Series, #662, Univ. of Minnesota.

[Ta] M. Taboada (1990), *Finite-dimensional asymptotic behavior for the Swift-Hohenberg model of convection* , Nonlinear Analysis, TMA,**14**, 1, pp.43-54.

[Te] R. Témam (1988) *Infinite Dimensional Dynamical Systems in Mechanics and Physics*, Springer-Verlag, New York.

A Mathematical Cartoon for the Dynamics of Fine Structure*

Philip J. Holmes and Pieter J. Swart
Mathematical Sciences Institute and
Departments of Theoretical and Applied Mechanics
and Mathematics, Cornell University,
Ithaca, NY 14853.

Abstract

We summarize results on a dissipative infinite dimensional evolution equation having an associated Energy (Liapunov) function which possesses no classical minimizers. The equation has a countable set of equilibria, all unstable, which form a minimizing sequence, and our main result implies that a large, dense set of solutions explore this sequence. In doing so, energy approaches its global minimum via an escape to arbitrarily high wavenumbers in Fourier space. We describe the asymptotics of this process and illustrate it with numerical computations. This equation exhibits a remarkably subtle dependence on initial data very different from that in classical finite dimensional "chaos".

Introduction

This study concerns the behavior of the equation

$$u_{tt} = (\|u_x\|^2 - 1)u_{xx} - \alpha u + \beta u_{xxt}, \tag{1}$$

which was motivated by the simple model of a one-dimensional nonlinear viscoelastic bar bonded to a rigid substrate, namely

$$u_{tt} = (u_x^3 - u_x + \beta u_{xt})_x - \alpha u. \tag{2}$$

Further details on the background to these problems can be found in [1] and full details are in [2], from which most of the following is adapted. Here the displacement $u = u(x,t)$ is defined on $x \in (0,\pi)$ with Dirichlet boundary conditions $u(0,t) = u(\pi,t) = 0$ and $\|u\|^2 = \int_0^\pi u^2 dx$. The term αu penalizes large displacements and tends to promote the formation of microstructure. The βu_{xxt} term represents viscoelastic damping. The more tractable model (1) is obtained by replacing the nonlinear term $u_x^2 u_{xx}$ in (2) by the spatially

*Supported by the U.S. Army Research Office under ARO DAAG 29-85-C0018 (Mathematical Sciences Institute) and NSF under DMS 87-03656. The computations were performed using the Cornell National Supercomputer Facility, which receives major funding from NSF and the IBM Corporation.

averaged term $\|u_x\|^2 u_{xx}$. We remark that such non-local terms do arise, for example in ferromagnetism [3].

We henceforth restrict our attention to (1). This model can also be interpreted as the Euler-Lagrange equation corresponding to the nonlocal "strain" energy

$$E[u, u_t] = \frac{1}{2}\|u_t\|^2 + \frac{1}{4}(\|u_x\|^2 - 1)^2 + \frac{\alpha}{2}\|u\|^2 \tag{3}$$

to which has been added viscoelastic dissipation, which constantly bleeds of energy at the rate

$$\frac{dE}{dt} = -\beta\|u_{xt}\|^2. \tag{4}$$

The minimum of this energy cannot however be attained by any classical solution, since the conflicting requirements $u = 0$ and $\|u_x\|^2 = 1$ cannot be met. In other similar problems with nonconvex energies, the equilibrium solutions are characterized by a severe loss of uniqueness. It is by studying the long term *dynamical* behavior of such systems that we hope to shed some light on their equilibrium solutions.

Equilibrium States

Expanding $u(x, t)$ in the Fourier sine series

$$u(x, t) = \sum_{k=1}^{\infty} a_k(t)\sqrt{\frac{2}{\pi}}\sin kx \tag{5}$$

we obtain the infinite set of ODE's

$$\ddot{a}_k + \beta k^2 \dot{a}_k + k^2(\frac{\alpha}{k^2} - 1 + \sum_{j=1}^{\infty} j^2 a_j^2)a_k = 0, \quad k = 1, 2, \ldots \tag{6}$$

In addition to the trivial solution $u = u_0 = 0$ the equilibria of (1) are easily seen to occur in "pure-mode" pairs

$$a_k^{\pm} = \pm\frac{1}{k}\sqrt{1 - \frac{\alpha}{k^2}}, \quad a_j = 0,\ j \neq k,\ k^2 > \alpha. \tag{7}$$

We therefore have the countable set of equilibria

$$u_0^{\pm} = 0, \quad u_k^{\pm} = \pm\frac{1}{k}\sqrt{1 - \frac{\alpha}{k^2}}\sin kx, \quad k = K, K = 1, \ldots \tag{8}$$

where $K = K(\alpha) = \min\{k | k^2 > \alpha\}$. Since $E[u_k^{\pm}, 0] = \frac{\alpha}{2k^2}[1 - \frac{\alpha}{2k^2}] \searrow 0$ as $t \to \infty$, it follows that $\{u_k^{\pm}, 0\}_{k=K}^{\infty}$ is a minimizing sequence for this Liapunov function.

A local stability analysis shows that every equilibrium $u_k^{\pm}$ is exponentially unstable, albeit increasingly weakly as the wavenumber k is increased. Moreover, if $k_2 > k_1 > K$, then $u_{k_2}^{\pm}$ lie in the unstable manifolds of $u_{k_1}^{\pm}$. To see this, note that each 2N-dimensional subspace of the form $X_N = \{(u, u_t) | (u, u_t) = \sum_{j=1}^{N}(a_j, \dot{a}_j)\sin jx\}$ is invariant for (6) (and (1)). This fact is used in the proof of Theorem 1, below.

Asymptotic behavior

We establish a dichotomy which implies that solutions behave either in a "finite dimensional" fashion, essentially involving only a finite set of Fourier modes, or that *all* Fourier modes are active and that energy cascades out to infinity in wave number space. Since typical initial data contains arbitrary high Fourier wavenumbers, almost all solutions will contain "unstable" Fourier components and hence realize the second alternative.

Theorem 1 (J.M.Ball) *Let* $(u, u_t) \in X = H_0^1 \times L^2$ *solve* (1). *Then as* $t \to \infty$, *either* $(u, u_t) \to (u_k^{\pm}, 0)$ *strongly for some equilibrium* $u_k^{\pm}$ *and* $E(t) \to \frac{\alpha}{2k^2}[1 - \frac{\alpha}{2k^2}]$, *or*

$$\|u_t\| \to 0, \quad u \to 0 \text{ weakly in } H_0^1$$
$$\|u_x\| \to 1 \text{ and } E(t) \to 0.$$

The first alternative is realized for initial conditions in a set of first category and dense in the phase space, whilst the second alternative is realized for all initial conditions in its complement, a set of second category and also dense in the phase space.

This is a striking result. Arbitrary initial data can be arbitrarily close to orbits realizing either alternative, implying a sensitive dependence on initial data that is quite different from that in chaotic dynamical systems, being truly infinite dimensional and without any recurrence.

Energy Transport to higher wavenumbers

Since "most" solutions do minimize the energy by developing some form of microstructure we now describe how this happens. It is most conveniently stated in terms of the Fourier components of the "strain" $u_x = \sum_{k=1}^{\infty} b_k \sqrt{\frac{2}{\pi}} \cos kx$, $c_k = \dot{b}_k$.

Theorem 2 *Assume that the second alternative of Theorem 1 holds and pick any* $\nu > 0$ *and* $K < \infty$. *Then there exists a time* $T = T(\nu, K, \alpha, \beta) < \infty$ *such that, for all* $t \geq T$ *and* $k \leq K$ *the solutions of* (1) *satisfy*

$$|(b_k, c_k)(t)| \leq \nu \text{ and } |1 - \sum_{k=1}^{\infty} b_k^2| \leq \nu^2. \tag{9}$$

Moreover, for all $k \neq l$ *with* $k, l > K$ *and* $t \geq T$, *the modal ratio* $\rho_{kl} = b_k/b_l$ *satisfies*

$$\rho_{kl} = e^{\frac{\alpha}{\beta}(\frac{1}{l^2} - \frac{1}{k^2})\mu_{k,l}(t-T)} \rho_{kl}(T) \tag{10}$$

where $\mu_{k,l}(s) = s(1 + O(1/K^2) + O(l^2/k^2K^2))$.

The first assertion follows from Theorem 2. The key to the proof of the second result is the realization that, for each large k, (6) is a singularly perturbed second order ODE.

Noting that $b_k = ka_k$, we may rewrite (6) as

$$\dot{b}_k = c_k, \tag{11}$$

$$\dot{c}_k = k^2((1 - \frac{\alpha}{k^2} - \sum_{j=1}^{\infty} b_j^2)b_k - \beta c_k), \tag{12}$$

solutions of which rapidly enter and thereafter remain near the slow manifold $c_k = (1 - \frac{\alpha}{k^2} - \sum_{j=1}^{\infty} b_j^2)b_k/\beta$. Substitution of this into the first component of (11) yields

$$\dot{b}_k = (1 - \frac{\alpha}{k^2} - \sum_{j=1}^{\infty} b_j^2)b_k/\beta \tag{13}$$

and differentiation of $\rho_{k,l}$ and use of (12) yields (10) with $\mu_{k,l} \equiv s$. Justification of these formal manipulations and derivation of the error estimates may be found in [2].

This shows that any specific Fourier mode b_l eventually dies and it describes how the energy escapes to $k = \infty$. In fact, for $k \geq l > K$ this shows that high modes grow exponentially at the expense of low modes and that every mode eventually decays at an exponential rate. We can use this to illustrate the delicate influence of initial data on modal dynamics.

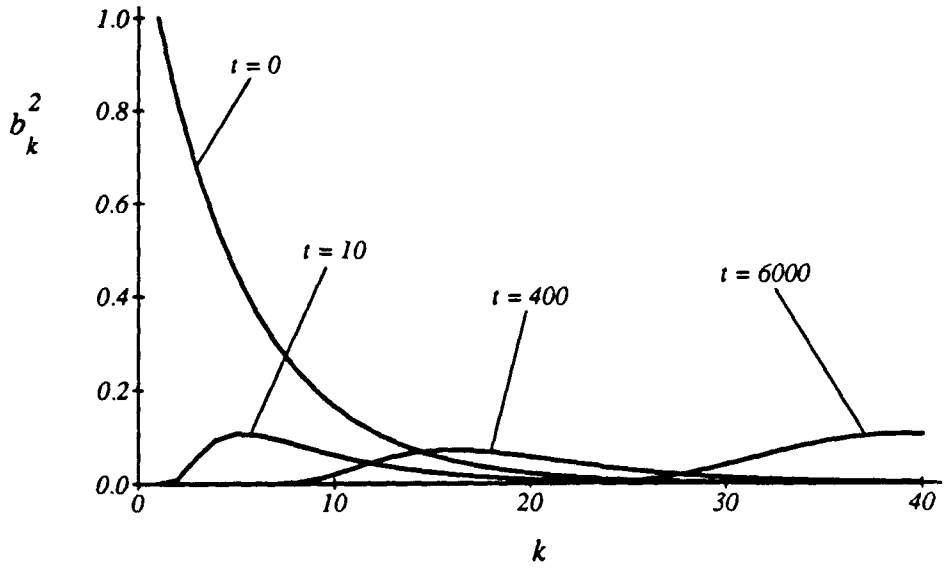


Figure 1: Typical evolution of the modal energy.

Suppose first that the initial data is analytic, namely that $b_l(T) = Ae^{-cl}$ for some $A, c > 0$. Then, as $t \to \infty$, the peak of the energy in wavenumber space can be shown to slowly move out to $k = \infty$ at a rate $\sim t^{\frac{1}{3}}/c^{\frac{1}{3}}$ and the "bump" spreads out with half bandwidth increasing as $t^{\frac{1}{6}}/c^{\frac{2}{3}}$. This behavior is illustrated in Figure 1.

Secondly, suppose that $b_l(T) = Al^{-r}$, so that $u \in C^{r-1}$. In this case it follows from (10) that the peak of the energy in wavenumber space moves out to $k = \infty$ at a rate $\sim t^{\frac{1}{2}}/r^{\frac{1}{2}}$ and the "bump" spreads out with halfbandwidth increasing as $t^{\frac{1}{2}}/r$ as $t \to \infty$. Numerical experiments (performed over large finite times and restricted to a large but finite number of Fourier modes) showed these estimates to be very accurate [2]. The system (6), truncated to N modes, becomes increasingly stiff as $\|u_x\|^2 - 1$ decreases in size and was numerically integrated using the backwards differentiation algorithm DDEBDF from the SLATEC subroutine library.

Numerical Examples

We now illustrate the sensitive dependence on initial conditions with some simple examples. In order to avoid the rather costly numerical integration of the full system (6), we assume that the solution is already on the slow manifold and that the reduced system (13) therefore provides a good description of the dynamics. The exact solution to (13) is given by

$$b_k(t) = A(t)e^{-\frac{\alpha t}{\beta k^2}} b_k(0), \quad k = 1, 2, \ldots \tag{14}$$

where

$$A(t) = e^{\frac{t}{\beta}} \left[1 + \sum_{j=1}^{\infty} b_j^2(0) \frac{\left[e^{\frac{2t}{\beta}(1-\frac{\alpha}{j^2})} - 1 \right]}{(1 - \frac{\alpha}{j^2})} \right]^{-\frac{1}{2}} \tag{15}$$

$$= \left[\sum_{j=1}^{\infty} \frac{b_j^2(0) e^{-\frac{2\alpha t}{\beta j^2}}}{(1 - \frac{\alpha}{j^2})} \right]^{-\frac{1}{2}} \left(1 + \mathcal{O}(e^{-\frac{2t}{\beta}}) \right). \tag{16}$$

Note that the $e^{-\frac{\alpha t}{\beta k^2}} b_k(0)$ term represents the solution to the "truly backwards" heat equation $\beta u_{xxt} = \alpha u$ and that the $A(t)$ term acts as a uniform scaling so as to achieve $\|u_x\| \approx 1$. The modal ratio $\rho_{kl} = b_k/b_l$ for (14) now satisfies (10) but without any error terms. Sensitive dependence on initial conditions is best illustrated using the long-term approximation given by (16). A further simplification follows by assuming that the first N modes have already decayed to zero, and choosing $\alpha \ll N^2$ while keeping α/β fixed, so as to obtain

$$A(t) = \left[\sum_{j=1}^{\infty} b_j^2(0) e^{-\frac{2\alpha t}{\beta j^2}} \right]^{-\frac{1}{2}} \left(1 + \mathcal{O}(\frac{\alpha}{N^2}) + \mathcal{O}(e^{-\frac{2t}{\beta}}) \right). \tag{17}$$

which provides an accurate description of the long-time evolution of (1) for initial data containing at least one nonzero high frequency component. We now use this approximation with 1024 Fourier modes and $\alpha/\beta = 1$ to generate some simple numerical examples displaying sensitive dependence on initial conditions.

In our first example we illustrate the effect of ignoring the high-frequency components that are present in generic initial data. In Figure 2 is shown the evolution of analytical

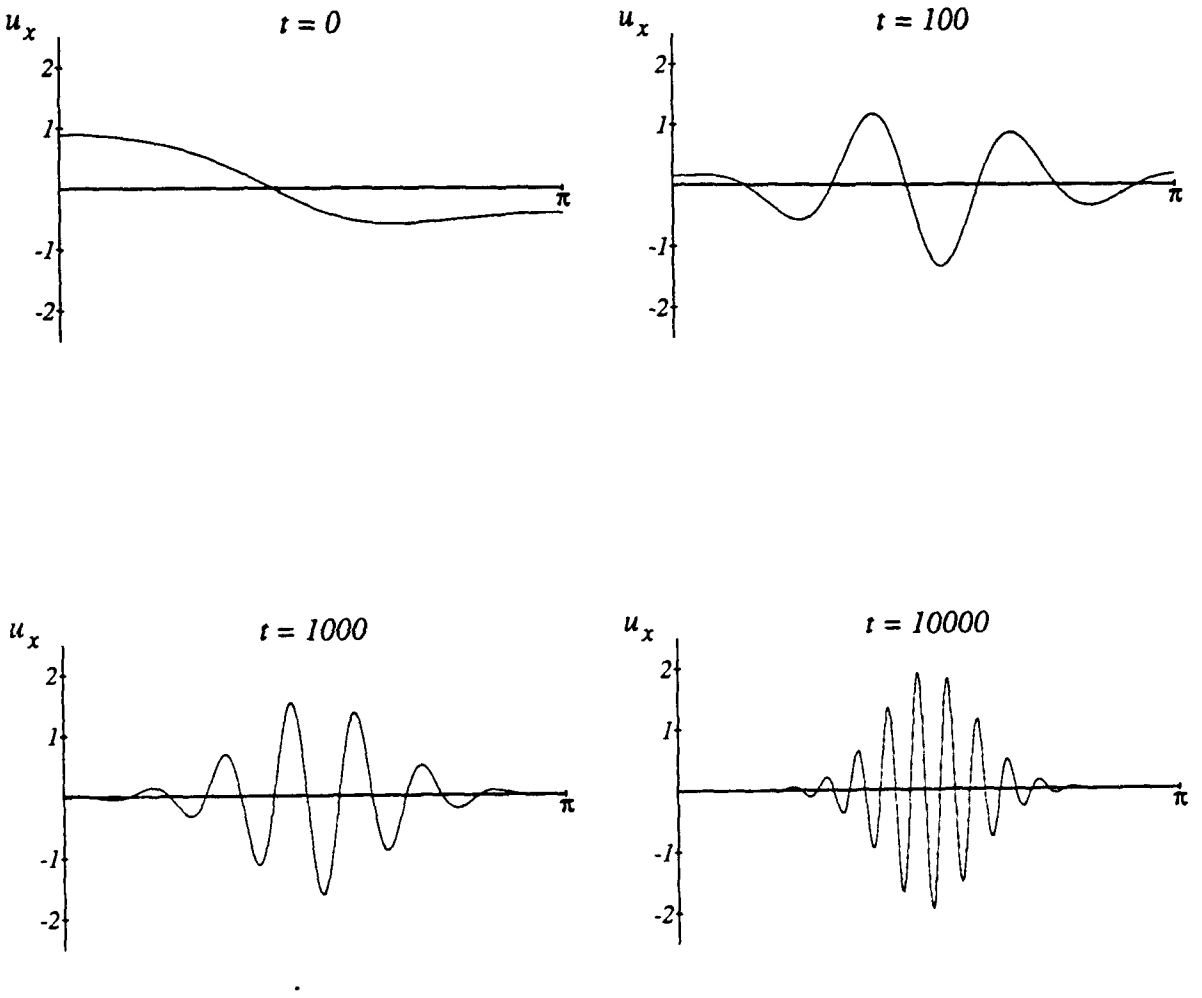


Figure 2: Approximate numerical solution of (1) using 1024 nonzero Fourier modes, obtained via (14) as described in text. Note that u_x is plotted vs x in this and in Figures 3, 4 and 5.

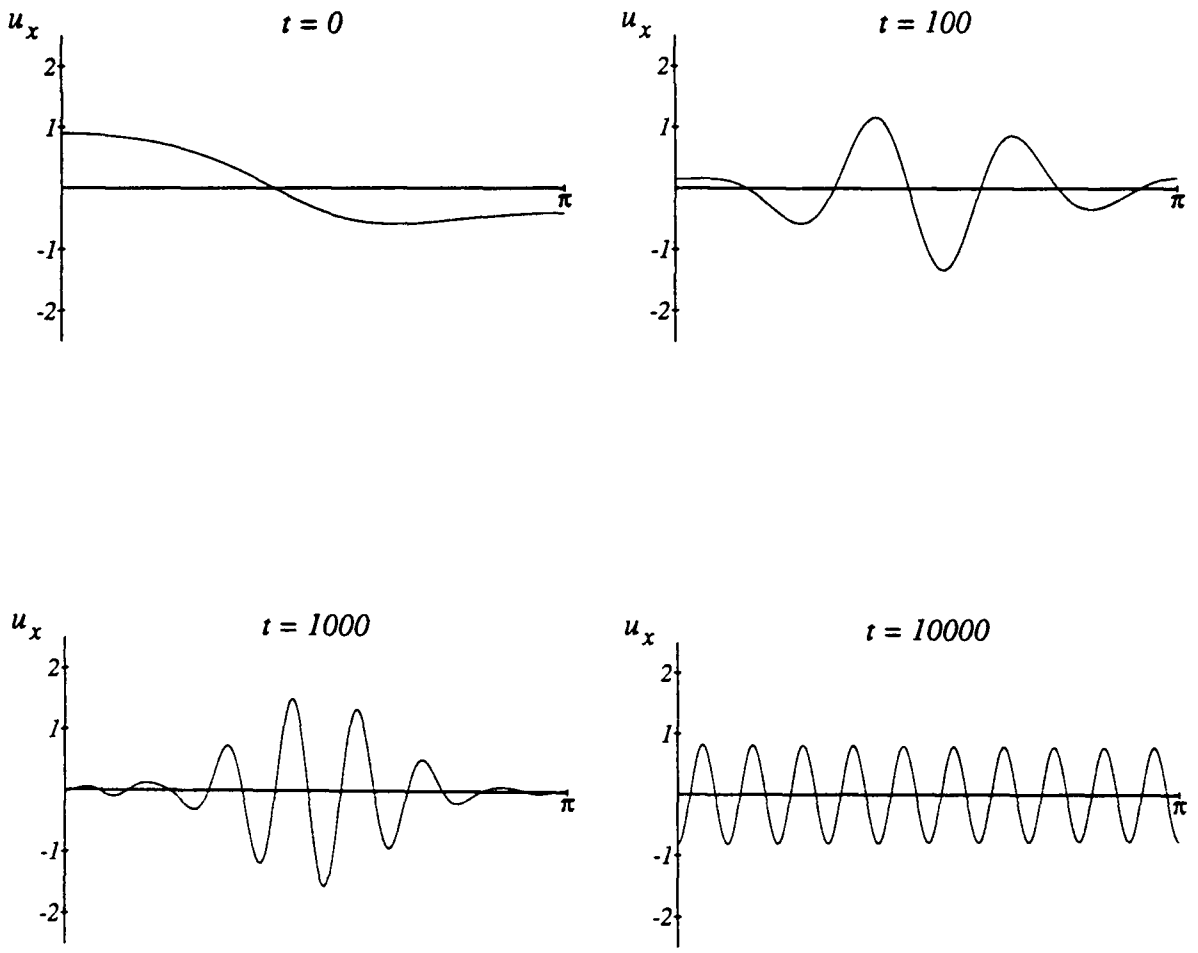


Figure 3: Approximate numerical solution of (1) using 20 nonzero Fourier modes, obtained via (14) as described in text.

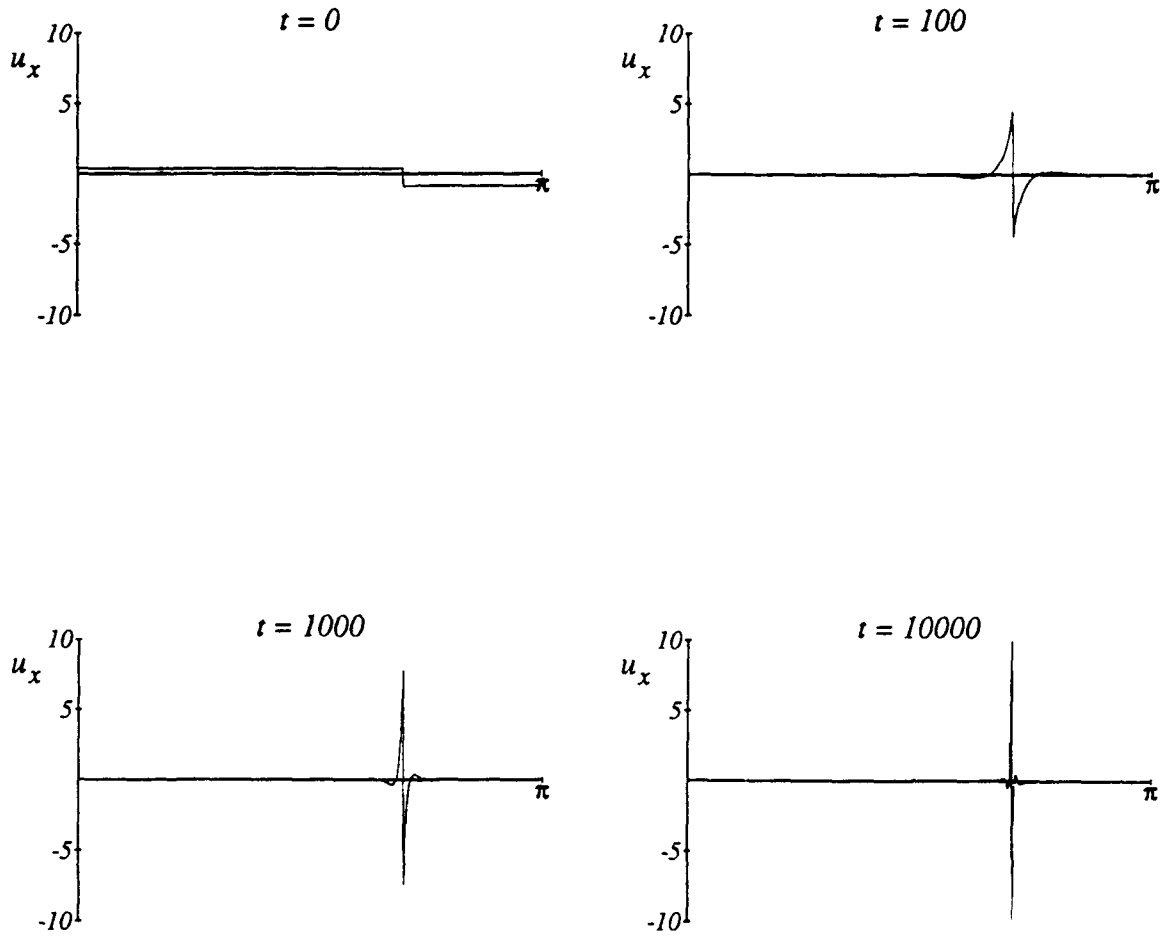


Figure 4: Approximate numerical solution of (1) using 1024 nonzero Fourier modes, obtained via (14) as described in text. Initial condition is given by $u_x(x,0) = -\tilde{H}(x - 0.7\pi)$.

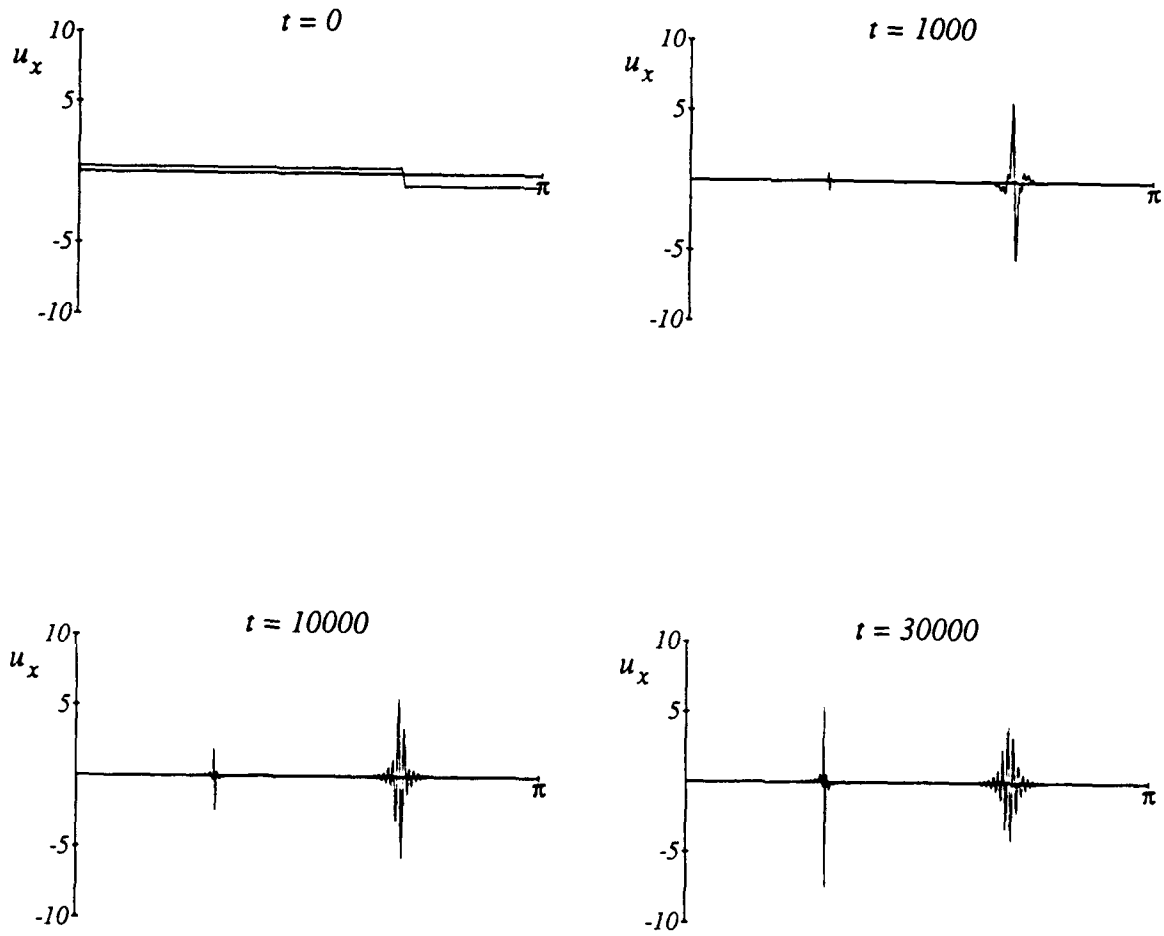


Figure 5: Approximate numerical solution of (1) using 1024 nonzero Fourier modes obtained via (14) as described in text. Initial condition is given by $u_x(x,0) = -\tilde{H}_{0,200}(x - 0.7\pi) - 0.1\tilde{H}_{200,1024}(x - 0.3\pi)$.

initial data $u_x(\cdot,0)$ with Fourier components $b_k(0) = Cs_k e^{-k}$, where the "resonant" coefficients $s_k = \{1,1,-1,-1,1,1,-1,-1,...\}$ are chosen to force concentration in the interior of the interval $(0,\pi)$ and C is chosen such that $\|u_x(\cdot,0)\|^2 = 1$. These initial conditions illustrate how solutions of (1) can focus spatially and form localized fine structure. Note that the increasingly finer oscillations observed as $t \to \infty$ is not a numerical artifact, but characteristic of a solution u that converges weakly to zero in H_0^1 while forced to satisfy $\|u_x(\cdot,0)\|^2 \approx 1$ (cf. Theorem 1). The first 20 modes of such initial data provide an excellent approximation at $t = 0$ and in Figure 3 is shown the evolution of this "truncated" problem. If there are initially only a finite number N of active Fourier modes, then (14) implies that, after a time $t \sim \mathcal{O}(\beta N^2/\alpha)$, all the modes except b_N will be decaying exponentially, with the highest active mode $b_N(t) \to \pm\sqrt{1-\alpha/N^2}$ and preserving the sign of $b_N(0)$. This is clearly displayed by the large time behavior shown in Figure 3. Note that the solutions only diverge after $t \sim 1000$, as expected.

In our next example we show how the solutions of the reduced system (14) can be made to display strong spatial concentrations at arbitrarily chosen points and after arbitrarily long times. In Figure 4 is shown the solution corresponding to $u_x(x,0) = -\tilde{H}(x-0.7\pi)$, where $\tilde{H}$ is the Heaviside step function, shifted to be of zero average and scaled such that $\|u_x(\cdot,0)\|^2 = 1$. This example also illustrates the "persistance of strain discontinuities" (cf. [2]) — discontinuities in u_x cannot be destroyed or created in finite time. Let $\tilde{H}_{k,l}$ be the function obtained by considering only the contribution of the k'th through to the l'th Fourier modes of $\tilde{H}$ (rescaled to be of unit L^2 norm). $\tilde{H}_{k,\infty}(x-x_0)$ represents a "spike" at x_0 that can be localized by choosing k large. In Figure 5 is shown the evolution of the initial data $u_x(x,0) = -\tilde{H}_{0,200}(x-0.7\pi) - 0.1\tilde{H}_{200,1024}(x-0.3\pi)$. The first term dominates initially, but decays after $t \approx 30000$. The second term, although initially an almost unnoticable little "spike" at $x = 0.3\pi$, displays its presence after the first term has begun to decay.

We can therefore construct initial data of the form $\sum_j \tilde{H}_{k_j,l_j}(x-x_j)$, with $k_j < l_j \ll k_{j+1} < l_{j+1}$ and arbitrary x_j, for which the process observed in Figure 5 can be repeated as often as we wish, causing the slow and successive appearance and disappearance of spatial concentrations at $x = x_1, x_2, ...$ Typical solutions of (1) therefore need not stabilize pointwise and can display the formation of concentrations in a seemingly haphazard fashion. Even when adding stronger dissipation to the system, e.g. by adding a capillarity term $-\gamma u_{xxxx}$ to the right hand side of (1), this process can still continue for extremely long times before the solution eventually settles down to an equilibrium.

Conclusion

While only of indirect physical interest, we believe the model discussed here provides a significant example of behavior characteristic of certain infinite dimensional evolution equations. In spite of the strong dissipation and the fact that the energy E decreases monotonically along solutions, excluding any chaotic or time-periodic motions, the initial data exerts a remarkable influence on the dynamical behavior and approach to equilibrium. In "simple"

dissipative systems possessing Liapunov functions, such as the Chafee-Infante problem:

$$u_t = u_{xx} + \lambda f(u) \quad (e.g. f(u) = u - u^3)$$

or the damped nonlinear wave equation:

$$u_{tt} = u_{xx} - \beta u_t + \lambda f(u),$$

one expects almost all solutions to approach a classical equilibrium corresponding to a local minimum in energy. For models (1) and (2), however, the forward orbits $\{u(t)|t > t_0\}$ cannot be shown to lie in a compact set and the usual methods fail, cf. [1, 2]. For (1) almost all solutions minimize energy with the nonlocal nonlinear energy term allowing new zeroes to appear in u_x without appreciable kinetic energy expenditure. However, our asymptotic results show that the rate at which and manner in which the "modal strain energy" $\|u_x\|^2$ escapes to arbitrary high Fourier wavenumbers is controlled by the *smoothness* of the initial data. This rather delicate influence of the initial data suggests that there may be problems in the interpretation of numerically determined equilibrium states of nonlinear elastic continua with non-convex strain energies by means of "dynamic relaxation" methods. Such methods usually ignore the inherent dynamics of the problem and identify the asymptotic equilibrium states with the minima of a nonconvex energy functional. Inertia and dissipation terms are then added in order to be able to apply dynamic relaxation, and the resulting initial value problem is then numerically solved for various initial data until the kinetic energy has numerically stabilized. If, as in model (1), initial data can so acutely affect either the fineness of the resulting equilibria or the rate at which fine structure develops, then such a dynamical process run for finite times from specific (sets) of initial data might yield results of doubtful significance. Secondly, the fact that such a process appears to have stabilized may merely be the consequence of a long period of extremely slow evolution and cannot rule out interesting surprises in the distant future.

References

[1] J. M. Ball [1990] Dynamics and minimizing sequences. *Problems Involving Change of Type* (K. Kirchgässner, Editor.) Springer Lecture Notes in Physics 359, 3-16, Springer Verlag, New York, Heidelberg, Berlin.

[2] J. M. Ball, P. J. Holmes, R. D. James, R. L. Pego and P. J. Swart [1990] (in preparation). On the dynamics of fine structure.

[3] R. D. James and D. Kinderlehrer [1990] Frustration in ferromagnetic materials. *Continuum Mechanics and Thermodynamics.* (in press)

BREAKUP OF A VISCOUS LIQUID JET*

S.P. LIN AND E.A. IBRAHIM

Department of Mechanical and Industrial Engineering
Clarkson University
Potsdam, NY 13676
Supported by Army Research Office

ABSTRACT. The instability of a cylindrical liquid jet encapsulated by a viscous gas in a pipe is analysed in a parameter space spanned by the Reynolds number, the Froude number, the Weber number, the density ratio, the viscosity ratio, and the diameter ratio. A convergent solution of the problem is constructed by a Galerkin projection with two orthogonal sets of functions. Two distinctively different modes of instability are obtained. The first is the Rayleigh mode which tends to break up the jet into drops of diameter comparable with the jet diameter. The Taylor mode instability is due to the pressure and the other mode tends to produce droplets of diameters much smaller than that of the jet. It is shown that the former mode appears when the Weber number is much larger than the gas to liquid density ratio. When this ratio is of order one, the instability can be due to either modes depending on the values of the rest of the parameters. When the density ratio is much larger than the Weber number, Taylor's atomization mode replaces the Rayleigh mode.

INTRODUCTION. The instability of an inviscid liquid jet with respect to temporally growing disturbances in the absence of gravity and ambient gas was analyzed by Rayleigh (1879). He showed that the disturbance possessing the maximum amplification rate could cause the jet to break up to form droplets comparable in size with the jet diameter. Chandrasekhar (1961) showed that the neglected liquid viscosity can only reduce the amplification rate of disturbances but cannot supress the instability caused by capillary pinching. The convective and absolute instability of a liquid jet was investigated by Keller et al. (1972), Leib and Goldstein (1986), and Lin and Lian (1989). Taylor (1963), Lin and Kang (1987) and Lin and Lian (1990) showed that when the gas to liquid density ratio, Q, is much greater than the Weber number, a viscous jet of radius R_1 may actually become unstable with respect to disturbances of wave length $\lambda << R_1$. Lin and Creighton (1990) found that while the mechanism of Rayleigh's instability is capillary pinching, the mechanism of Taylor's mode is the interfacial pressure fluctuation. However, the effects of the interfacial shear on the Rayleigh and the Taylor modes of the jet instability remain unknown, since the gas viscosity is neglected in all of the above mentioned theories. The effect of the viscosity of a motionless surrounding fluid on the breakup of a motionless viscous cylindrical thread was investigated theoretically by Tomotika (1934). The breakup mechanism for this case remains capillary pinching.

Joseph et al. (1984) investigated the instability of two immiscible liquids of the same density but of different viscosities in a pipe. The interfacial tension was neglected. The effects of surface tension and density stratification in the absence of gravity were later included in the investigations of Preziosi et al. (1989) and Hu and Joseph (1989). Smith (1989) investigated the instability of two immiscible fluids of the same viscosity but of different densities in a vertical pipe. These works are of fundamental importance, because they isolate the effects of the density and viscosity discontinuities at the interface. However, they cannot be applied to

*Supported by U.S. Army Research Office

infer the coupled effects of surface tension, interfacial shear gravitational acceleration and pressure fluctuation on the Rayleigh and the Taylor modes of instability.

FORMULATION. Consider the stability of a cylindrical liquid jet of radius R_1. The jet is surrounded by a viscous gas enclosed in a vertical circular pipe of radius R_2 which is concentric to the jet. For the jet to maintain a constant radius the pressure gradient in the steady liquid- and the gas-flows must remain the same constant. This will allow the pressure force difference across the liquid-gas interface to be exactly balanced by the surface tension force as required. Such coaxial flows of liquid and gas, in the presence of gravity, which satisfy exactly the Navier-Stokes equations are given by

$$W_1(r) = -1 + \frac{Nr^2}{[N-(1-\ell^2)]}\{1 - \frac{(1-Q)}{4N}\frac{Re}{Fr}[2\ln l + (1-l^2)]\},$$

$$W_2(r) = -\frac{(\ell^2-r^2)}{[N-(1-\ell^2)]}\{1 - \frac{(1-Q)}{4N}\frac{Re}{Fr}[2\ln l + (1-l^2)]\}$$

$$+\frac{(1-Q)}{4N}\frac{Re}{Fr}[l^2 - r^2 - 2\ln(\frac{\ell}{r})], \qquad (1)$$

$N = \mu_2/\mu_1$, $l = R_2/R_1$, $Q = \rho_2/\rho_1$,

Re ≡ Reynolds number $= \rho_1 W_o R_1/\mu_1$,

Fr ≡ Froude number $= W_o^2/gR_1$, $R = Re/Fr$,

where the subscript 1 or 2 stands for the liquid or the gas phase respectively, W_o is the magnitude of the jet velocity in the z-axis (c.f. Fig. 1), r is the radial distance normalized with R_1, W(r) is the axial velocity distribution, μ is the dynamic viscosity, ρ is density, and g is the gravitational acceleration in the negative z-direction. Some velocity distributions in a water jet and in the surrounding air flow under one atmosphere are given in figure 1 for various values of Re/Fr. Note the large difference in the slopes of the velocity profiles in the liquid and the gas phases due to the large difference in their viscosity, when R is relatively large.

The stability of the basic state described by (1) with respect to a normal mode axisymmetric disturbance is governed by the well known Orr-Sommerfeld equation (Drazin and Reid, 1985),

$$[\omega - (N'/Re)D^2]D^2\phi_i(r) + ikW_i(r)D^2\phi_i(r) - ikrd[dW_i(r)/r]\phi_i = 0, \quad (i=1,2) \qquad (2)$$

$D^2 = d^2 - r^{-1}d - k^2$, $d = d/dr$, $N' = \nu_i/\nu_1$,

where ν is the kinematic viscosity, the subscript i stands for the liquid phase or the gas phase depending on if i=1 or i=2, ω and k are respectively the dimensionless complex frequency and the wave number of the disturbance, and ϕ_i is the amplitude of the normal mode disturbance related to the Stokes stream function ψ_i by

$$\psi_i(r,z,t) = \phi_i(r)e^{(ikz+\omega t)},$$

where t is time normalized with R_1/W_o. The Stokes stream function is related respectively to the radial and axial components of the disturbance velocity by

$$u_i = \psi_{iz}/r \quad \text{and} \quad w_i = -\psi_{ir}/r,$$

where the subscripts z and r denote partial differentiations.

The boundary conditions at the perturbed liquid-gas interface $r = 1+\eta$ can be linearized by use of the Taylor series expansions of all variables involved about r=1, and retaining only terms of the first order in perturbations. Hence, the interfacial conditions are to be evaluated at r=1 with η as an additional unknown. Since the interface is a material surface, η must satisfy at r=1 the kinematic condition

$$\eta_t + W_i\eta_z = \psi_{iz} .$$

Other interfacial kinematic conditions are the continuity of the radial and tangential components of the velocity across the interface given respectively by

$$[\psi_{iz}]_2^1 \equiv [\psi_{1z} - \psi_{2z}]_{r=1} = 0 , \text{ and}$$

$$[W_{ir}\eta - \psi_{ir}]_2^1 = 0 .$$

The balancing of forces per unit area of the interface in the tangential and normal directions leads respectively to the dynamic conditions at r=1,

$$[N_i\{\eta W_{irr} - (\psi_{ir}/r)_r + \psi_{izz}\}]_2^1 = 0, \text{ and}$$

$$[p_i - (2/Re)N_i(\psi_{iz}/r)_r]_2^1 + (\eta + \eta_{zz})We = 0 ,$$

where p_i is the disturbance pressure,

$$We \equiv \text{Weber number} = S/\rho_1 W_o^2 R_1, \; N_i = \mu_i/\mu_1,$$

in which S is the surface tension. Thus, We signifies the ratio of surface tension force to the inertia force per unit area of the interface. The boundary condition at the pipe wall is the no-slip condition at r=l,

$$\psi_{2z} = 0 , \; \psi_{2r} = 0 .$$

The normal mode axisymmetric pressure disturbance and interfacial displacement are written as

$$[p_i, \eta] = [\zeta_i(r), \xi]e^{(ikz+\omega t)}. \qquad (3)$$

Substituting (3) and the normal mode of ψ_i into the above boundary conditions, we rewrite them in the same order of appearance

$$(\omega + ikW_1)\xi - ik\phi_1 = 0 , \qquad (a)$$

$$[\phi_i]_2^1 = 0 \qquad (b)$$

$$[\xi W_{ir} - \phi_{ir}]_2^1 = 0 , \qquad (c)$$

$$[N_i B\phi_i]_2^1 - (1-Q)R\xi = 0 , \qquad (d)$$

$$B = d^2 - d/r + k^2 ,$$

$$[\zeta_i - (2ik/Re)N_i(\phi_{ir} - \phi_i)]_2^1 + \xi(1-k^2)We = 0, \quad (e)$$

$$\phi_{2r}(l) = 0\,, \quad (f)$$

$$\phi_2(l) = 0\,. \quad (g)$$

The last term in (d) arises from the second derivatives of the basic flows. The pressure amplitude discontinuity in (e) can be obtained from the linearized Navier-Stokes equations, and are found to be

$$[\zeta_i]_2^1 = [Q_i\{(\omega+ikW_i)\phi_{ir} - ikW_{ir}\phi_i\} - N_i(D^2\phi_i)_r/Re]_2^1 \quad (ik)^{-1}, \qquad Q_i = \rho_i/\rho_1\,.$$

Nontrivial solutions of (1) with its boundary conditions (a) to (g) for given flow parameters Re, Fr, We, Q, N, l and k exist only for certain eigenvalues ω. The real part of ω determines the stability of the flow, and the imaginary part of ω determines the characteristic frequency of the disturbance.

SOLUTION. The solution of the problem formulated in the previous section will be expanded in an orthogonal set of functions in each of the flow fields in the liquid and in the gas (lighter imcompressible fluid). The two orthogonal sets are associated with the same differential operator in (2), i.e. D^2, but with different domain boundaries. By use of the change of variable

$$\phi_i = rf_i\,, \quad (i=1,2)\,,$$

we have

$$D^2\phi_i = r(L-k^2)f_i\,,$$

where

$$L = r^{-1}\,d(rd) - r^{-2}\,.$$

The orthogonal functions will be chosen among the solutions of the Bessel equation of the first order with the parameter k_{in}

$$(L^2 + k_{in}^2)F_{in} = 0, \ (n=1,2,...M_i), \quad (4)$$

where F_{in} stands for $F_i(k_{in}r)$, and M_i is an arbitrarilly large integer. The bounded solutions of (4) which forms an orthogonal set of functions in $r\leq 1$ are

$$F_{1n} = J_1(k_{1n}r)\,, \quad (5)$$

where k_{1n} are the roots of

$$k_{1n}J_o(k_{1n}) - J_1(k_{1n}) = 0\,. \quad (6)$$

With these values of k_{1n}, we have

$$\int_o^1 r\,F_{1m}\,F_{1n} = \delta_{mn}^{(1)} \quad (7)$$

where $\delta_{mn}^{(1)} = 0$ if $m \neq n$, if $m = n$ it is given by

$$\delta_{nn}^{(1)} = 0.5 \quad (k_{in}^2 - 1)J_o^2(k_{1m}). \tag{8}$$

The bounded solutions of (4) which form an orthogonal set of functions in the domain $1 \leq r \leq l$ are

$$F_{2n} \equiv F_2(k_{2n}r) = Y_1(k_{2n}l)J_1(k_{2n}r) - J_1(k_{2n}l)Y_1(k_{2n}r)\,, \tag{9}$$

where k_{2n} are the roots of

$$F_2(k_{2n}) - k_{2n}\bar{F}(k_{2n}) = 0\,, \tag{10}$$

$$\bar{F}(k_{2n}r) = J_o(k_{2n}r)Y_1(k_{2n}l) - J_1(k_{2n}l)Y_o(k_{2n}r)\,. \tag{11}$$

With the values of k_{2n} thus determined, we have

$$\int_1^l r\,F_{2n}\,F_{2m}\,dr = \delta_{mn}^{(2)} \tag{12}$$

where $\delta_{mn}^{(2)} = 0$ if $m \equiv n$, otherwise it is given by the integral on the left side of the above equation with F_{2n} given by (9). Note that

$$F_2(k_{2n}l) = 0\,. \tag{13}$$

We now expand the eigen-vector in a truncated series of the above orthogonal functions

$$\phi_i = r\,a_{in}\,F_{in}\,, \quad (i=1,2) \tag{14}$$

where the repeated indecies n denote sumation over n=1 to n=M_i (i=1,2). The number of terms M_i required in the two flow domains may not be the same for the required accuracy. The components of eigenvector will be obtained by use of the Galerkin projection. The following formula which can be derived with integration by parts will be used repeatedly in the reduction of the Galerkin projection,

$$\int_{s_i}^{t_i} rGL(g)dr = \int_{s_i}^{t_i} rgL(G)dr - [rgd(G) - rGd(g)]_{s_i}^{t_i} \tag{15}$$

where g and G are function of r. The Galerkin projection of (2) gives

$$\int_{s_i}^{t_i} rF_{im}[(L-k^2-R_e\omega')(L-k^2)f_i + ik(\nu_1/\nu_i)R_eW_i(L-k^2)f_i]dr = 0\,, \tag{16}$$

where $\omega' = \omega(\nu_1/\nu_i)$. By use of (15), the orthogonality conditions, and the following relations

$$[rL(f_1)d(F_{1m}) - rF_{1m}dL(f_1)]_{r=s_1} = 0\,,$$

$$F_2(k_{2m}t_2) = 0\,,$$

we can reduce (16) to

$$e_{imn}a_{in} - \nu t_1\delta_{1i}[(dF_{1m})_{t_1}\ \alpha - (F_{1m})_{t_1}\ \beta]$$
$$- \nu s_2\delta_{2i}[(t_2/s_2)\,(dF_{2m})_{t_2}\ \gamma + (F_{2m})_{s_2}\ \delta - (d\,F_{2m})_{s_2}\ \varepsilon] = 0$$

$(m, n=1,2,...M_i)$ (17)

where $\nu = (\nu_i/\nu_1)$, the subscripts of parantheses denote the values of r at which the paranthesized functions are to be evaluated, and

$$e_{imn} = \delta_{mn}\,[k^2(\nu k^2+Re\omega) + (2\nu k^2+Re\omega)k_{im}^2 + \nu(k_{in}k_{im})^2]$$
$$+ ikRe(k^2 + k_{in})\int_{s_1}^{t_1} r\, W_i F_{im} F_{in}\, dr - ik\delta_2\ ,\ [(1-Q)RRe/N]\int_{s_2}^{t_2} F_{im}F_{in}\, r^{-1} dr\ ,$$

$$\alpha = [L(f_1)]_{t_1}\quad ,\ \beta = [dL(f_1)]_{t_1}\quad ,$$

$$\gamma = [L(f_2)]_{t_2}\quad ,\ \delta = [dL(f_2)]_{s_2}\quad ,\ \varepsilon = [L(f_2)]_{s_2}\ .$$

It is known that termwise differentiations of truncated series representations of functions do not provide as high an accuracy for the derivatives of functions as for functions themselves. For this reason we treat α to ε in (17), which involve derivatives higher than second, as five additional unknowns. Thus (17) is a system of $M_1 + M_2$ equations in $M_1 + M_2 + 5$ unknowns. The required additional equations are provided by the six, boundary conditions (a) to (f) which contain an additional unknown ξ. Note that boundary condition (g) is already satisfied, because of (13).

Substituting the series solution (14) into (a) to (f) we have

$$ikF_1(k_{1n})a_{1n} - [\omega+ikW_1(1)]\xi = 0\ , \qquad (a)'$$

$$[a_{in}\, F_{in}]_2^1 = 0\ , \qquad (b)'$$

$$[\xi W_{ir} - a_{in}(F_{in} + d\, F_{in})]_2^1 = 0 \qquad (c)'$$

$$[N_i\{(k^2-1)F_{in} + dF_{in} + d^2F_{in}\}a_{in}]_2^1 - (1-Q)\xi = 0 \qquad (d)'$$

$$[\{(\omega+ikW_1)Re\, Q_i(F_{in}+dF_{in})$$
$$- ikRe\, Q_i\, W_{ir}\, F_{in}$$
$$+ N_i[(k^2+k_{in})F_{in} + (3k^2+k_{in})dF_{in}]\}a_{in}]_2^1$$
$$+ ikReWe(1-k^2)\xi = 0\ , \qquad (e)'$$

$$(dF_{2n})_l a_{2n} = 0\ , \qquad (f)'$$

Equation (17) and the above boundary conditions form a system of $(M_1 + M_2 + 6)$ homogeneous linear equations in the same number of unknowns. Making the following identifications

$$a_{1n} = X_n\ (n=1 \text{ to } M_1)$$

$$a_{2n} = X_{n+M_1}\ \ (n=1 \text{ to } M_2)\ ,\ M = M_1 + M_2$$

$$(\alpha, \beta, \gamma, \delta, \varepsilon, \xi) = (X_{M+1}, X_{M+2}, X_{M+3}, X_{M+4}, X_{M+5}, X_{M+6})\ ,$$

this linear homogeneous system can be written in a standard form

$$(A_{mn} + \omega B_{mn})X_n = 0 \, , \, (m,n=1,2,...M+6)$$

where the elements of matrices A_{mn} and B_{mn} can be identified easily from (17) and the above boundary conditions. A nontrivial solution up to an arbitrary multiplicative constant of this system exists only if the determinant of its coefficient matrix vanishes, i.e.,

$$|A_{mn} + \omega B_{mn}| = 0 \, . \qquad (18)$$

To construct the eigenfunctions of the original system, which is not required in this work, we need only the eigenvectors a_{1n} and a_{2n}. The values of α, β, γ, δ, ε and ξ are not required. The explanation of the numerical computation involved in the solution of (18) is in order. All computations are carried out, in double precision with Gould PN 9780 at Clarkson and the supercomputer facility at the Cornell Theory Center. To construct the orthogonal functions F_i, we solve (6) and (10) with (11) respectively for k_{1n} and k_{2n} with the Muller (1956) method. All integrals involved, in (17) except δ_{mn} which has a closed form expression, are evaluated with the Gauss-Kronrod quadrature. For a given set of parameters (Re, We, Fr, Q, N, *l*) the eigenvalue ω is obtained from (18) for various values of k with the method by Kaufman (1974). In this complex eigenvalue solution, M_1 and M_2 are systematically increased until the eigenvalue corresponding to the most amplified or the least damped disturbance converges to the desired significant digits.

RESULTS. Table 1 gives a typical example which demonstrates the convergence of the method of determining the eigenvalues for a given set of flow parameters. It is seen that as M_1 and M_2 are
increased respectively from 6 and 54 to 7 and 63, the eigenvalues remains the same up to the first four significant digits. Note that when $M_1 = 7$ and $M_2 = 63$ there are 76 eigenvalues for the given set of parameters. Only the one corresponding to the most amplified disturbance is given in the table. The same convergence test was carried out for every computation for the most amplified or the least damped eigenvalues for various sets of parameters reported in this work. Preziosi et al. (1989), used Chebyshev polynomials as base functions for their solution of a special case of zero gravity in the present problem. They required 80 terms for satisfactory convergent results. Thus the terms required in the present problem with gravitational effect is slightly less than that required in their problem of zero gravity. A finite element method was used by Hu and Joseph (1989) in their extension of the work of Preziosi et al. The finite element method seemed to be more efficient than the collocation method. Attempts have been made to test the accuracy and convergence by doubling the number of terms in the present problem. It was found that for such a large system, the numerical error with a double precision calculation dominates the reduced truncation error.

Figure 1 shows the velocity distribution in the basic state for various values of R for the given parameters *l=10*, N=0.018, and Q=0.0013. These values of N and Q correspond to a water jet in atmosphere at room temperature. Figure 2 plots the growth rates ω_r against the wave number of the disturbance for various values of We for the set of parameters specified in the figure caption. R = Re/Fr = 0 signifies the absence of gravity. It is clearly seen that as We is decreased from 0.01 to 10^{-5} in steps, the amplification rates decrease for k<1. For k>1 the trend is reversed, although the growth rates are relatively small. The reversal of the trend can be easily understood by looking at the last term in (e). The factor $(1-k^2)\xi$ in this term arises from the curvature of the interface. The ξ-term is associated with the interfacial curvature along a direction perpendicular to the jet axis which gives rise to the necking at $\xi<0$ and expansion at

$\xi>0$. The $-k^2\xi$ term is of opposite sign and is associated with the curvature in the axial direction. This curvature tends to pull the displaced interface back to its basic state position. When $k<1$, the former destabilizing pinching effect dominates the latter stabilizing effect. When $k>1$ the role of surface tension is reversed. For the given parameters the jet instability is clearly due to the Rayleigh mode of capillary pinching, since the maximum growth rate occurs at $k<1$. As We is decreased further from 0.0001, the wave lengths corresponding to the maximum growth rates gradually shift to the region $k>1$. This is examplified by the curve for $We=10^{-5}$ in figure 2. Then the instability judged by the maximum amplification rate is no longer due to capillary pinching, but due to the Taylor mode. This mode will be expounded more clearly later when gravity is taken into account. Contrary to the dramatic effect of surface tension on the Rayleigh mode, the air viscosity has little effect on this mode. The destabilizing effect of the basic state shear rate is significant for shorter waves for which $k>1$. Comparisons between our theoretical results and the experimental results of Goedde and Yuen (1970) and that of Donnelly and Glaberson (1966) are made in figure 3. Unfortunately, the values of We and Re corresponding to the experimental points were not reported. Two amplification curves were obtained from our theory with the parameters corresponding to the lower range of the jet velocity reported by Goedde and Yuen. Rayleigh's amplification curve is also included in this figure for comparison. While the slope of Rayleigh's curve has a discontinuity at $k=1$, and the jet is neutrally stable for $k>1$, our curve is continuous in slope and gives negative ω_r for $k>1.1$. The good agreement between the experiments and our curve for Re = 3000 and We = 0.0013 and Rayleigh's curve, which is independent of We at $Re = \infty$ is probably fortuitous. It has already been shown that the amplification curves depend very sensitively on We, although less so on other parameters. For a better comparison with theories, complete records of all relevant parameters (l, Q, We, Re, N) for each observation of (ω_r, ω_i, k) are needed. Figure 4 further demonstrates the stabilizing effect of the interfacial tension. As We is decreased to values much smaller than Q, both the amplification rates and the wave number of the unstable spectrum are increased dramatically. It is seen that the most unstable disturbances of Taylor's atomization mode are of wave length several orders of magnitude smaller than the jet radius. Moreover, the wave lengths near the maximum growth rates of the amplification curves all scale with the capillary length $a = 2\pi S/\rho_2 W_o^2 R_1$. This can be verified by showing that the following equation is satisfied with the values of the wave number k_m, corresponding to the maximum growth rate, taken from each curve of figure 10,

$$(2\pi R_1/k) \approx a \equiv 2\pi S/\rho_2 W_o^2 R_1 = 2\pi R_1 (We/Q).$$

Recall that in the Rayleigh mode, the most amplified waves scale with R_1 in length. Contrary to the situation in the Rayleigh mode the air viscosity has a more significant effect on the atomization mode, as can be seen in figure 5. When N is increased from 0.0018 to 0.018 the disturbances for which $k<23$ are damped while the disturbances for which $k>23$ are amplified. This seems to reflect the fact that the enhancement of the amplification rate due to the relative increase in gas viscosity more than compensates for the decrease in the damping rate due to the relative decrease in the liquid viscosity for shorter waves such that $k>23$. The reverse is true for longer waves for which $k<23$. This also reveals the crucial roles played by the gas shear stress in the generation of small droplets. Neglecting the gas viscosity, Lin and Kang (1987), and Lin and Creighton (1990) showed that only pressure fluctuation can generate short waves scaling with capillary length. It is clear now that the interfacial shear and pressure fluctuations are equally capable of generating short waves scaling with the capillary length. This view is further substantiated by figure 12 which show qualitatively the same behavior as figure 10, when the ratio of inertia force relative to viscous force is raised respectively by raising the value of Q and Re. Figure 7 shows the destabilizing effect of the basic state shear rate on the Taylor mdoe. A large basic state shear rate at the interface requires a large shear stress fluctuation when the interface

fluctuates from the unperturbed cylindrical surface, in order to satisfy the condition of vanishing shear force at the interface. (c.f. Hinch, 1984; Kelly et al. 1989) This large shear stress fluctuation inevitably brings about a large pressure fluctuation, and causes the growth rate to increase. In contrast to the case of the Rayleigh mode, the radius ratio l has a very significant effect on the Taylor mode (Fig. 8). A decrease in l brings about a larger basic state shear rate which again results in an increase in the growth rate. Unlike the case of Q=1, We=0, and N<1 investigated by Joseph et al., we did not find stability near k=0 when $l \to 1$ for finite values of We and Q<<1. However, when we put We=0, Q=1, N=0.5, R=0 and Re=27.2 we did find that the jet is stable for 6<k and k<0.7. This is consistent with the results of Joseph et al. (1984). Hooper and Boyd (1987) and Renardy (1985) also found a similar stable region for planar Couett flow of two superposed fluids of different viscosities but of the same density.

DISCUSSION. It should be pointed out that the instability waves near k=0 in the present work are not of the type of Yih (1967), since Yih's long shear waves are not apparent when N<1 (c.f. Hooper & Boyd, 1987). The stability analysis of a vsicous liquid jet in an ambient gas reveals that there are two distinct mechanisms of the jet break up. The first is that of Rayleigh mode by capillary pinching, and the second is that of the Taylor mode by interfacial shear and pressure fluctuations. The theory is not yet fully substantiated by experiments. The present theory predicts that the growth rate of disturbances increases significantly with the Weber number as it should, since the instability is due to capillary pinching. Unfortunately, the known experiments in the Rayleigh mode regime failed to record the values of relevant parameters including We for each experimental point. Only the ranges of velocity, temperature, and jet diameter were reported. Thus only the ranges of the parameters encountered in experiments can be estimated. This deprives us of a more complete comparison. Consequently the apparent agreement between experiments and the present theory with We = 0.0013 and Re = 3000, and with the Rayleigh theory remain fortuitous. This value of We and the values of the rest of parameters used in figure 7 are in the lower end of the parameter range estimated from the reported experimental data. It is possible that most of the experimental points were obtained in the lower range of the parameters encountered in experiments. The theoretical results on the Taylor mode are only qualitatively substantiated by the experiments of Reitz and Bracco (1982). The average diameters of their atomized droplets seem to all scale with the capillary length as predicted by our theory. Careful measurements of (ω_r, ω_i, k) for varioius given sets (We, Re, Q, N, l) are required for a better comparison with the theory for both modes. There may exist other modes of instability in the parameter ranges not considered in this work. A possible third mode which may correspond to a dripping jet (c.f. Lin and Lian, 1989) is yet to be explored by considering the convective and absolute instabilities of spatially growing disturbances when We>>Q. The known analysis of absolute and convective instabilities of a jet all ignore the effect of gas viscosity (Leib and Goldstein, 1986; Lin and Lian, 1989). Blennerhassett (1980) showed that Tollmien-Schlichting waves are more stable than the interfacial waves in two superposed viscous fluids flowing over a plane. The same situation appears to happen here. The Tollmien-Schlichting wave will probably not appear until Re is raised to a value much greater than those considered in this work.

While the present analysis also applies to the case of N>1, the computation for this case has not yet been carried out. The extension of the present analysis to the case of non-axisymmetric disturbances is quite straightforward. The nonlinear stability analysis of the linearly unstable disturbances described in this work will be useful for many industrial processes which utilize the mechanisms of the jet breakup either in Taylor's atomization mode or Rayleigh's ink-jet mode.

This work was supported in part by Grant No. DAAL03-89-K-0179 of ARO, Grant No. MSM-8817372 of NSF and a New York State Science Grant. The computation was carried out

with the computer facility at Clarkson University and with the Cornell National Computer facility, which is founded by the NSF, the State of New York, and IBM Corporation.

M_1	M_2	ω_r	ω_i
3	15	0.0238	0.7003
5	35	0.0239	0.7013
5	45	0.0239	0.7017
6	54	0.0239	0.7021
7	63	0.0239	0.7021

Table 1. Convergence to the most amplified mode. k=0.7, We = 0.0025, Re = 400.0, Re/Fr = 0.0, Q = 0.0013, l = 10.0, N = 0.018

REFERENCES

BLENNERHASSETT, P.J., 1980, Phil. *Trans. R. Soc. Long.* A, **298,** 451.
CHANDRASEKHAR, S., 1961, *Hydrodynamic and Hydromagnetic Stability*, p. 537, Oxford U Press.
CHEN, K., BAI, R. & JOSEPH, D.D., 1989, J. Fluid Mech., **214,** 251.
DRAZIN, P.G. & REID, W.H., 1981, *Hydrodynamic Stability*, Cambridge U Press.
DONNELLY, R.J. & GLABERSON, W., 1966, *Proc. Roy. Soc.* A., **290,** 547.
GOEDDE, E.F. & YUEN, M.C., 1970, *J. Fluid Mech.*, **40,** 495.
HINCH, E.J., 1984, *J. Fluid Mech.* **128,** 507.
HOOPER, A.P. & BOYD, W.G.C., 1987 *J. Fluid Mech.*, **179,** 201.
HU, H.H. & JOSEPH, D.D., 1989, *J. Fluid Mech.*, **205,** 359.
JOSEPH, D.D., RENARDY, M. & RENARDY, Y., 1984, *J. Fluid Mech.*, **141,** 309.
KELLER, J.B., RUBINOW, S.I. & TU, Y.O., 1972, *Phys. Fluids*, **16,** 2052.
KELLY, R.E., GOUSSIS, D.A., LIN, S.P. & HSU, F.K., 19898, Phys. Fluids A, **12,** 819.
KAUFMAN, L.C., 1974, *SIAM J. Num. Anal*, **11,** 997.
MULLER, D.C., 1956, *Math. Tables & Aids to Comput.*, **10,** 208.
LEIB, S.J. & GOLDSTEIN, M.E., 1986, *J. Fluid Mech.*, **168,** 479.
LEIB, S.J. & GOLDSTEIN, M.E., 1986, *Phys. Fluids*, **29,** 952.
LIN, S.P. & KANG, D.J., 1987, *Phys. Fluids*, **30,** 2000.
LIN, S.P. & LIAN, Z.W., 1990, *AIAA J.* **28,** 120.
LIN, S.P. & LIAN, Z.W., 1989, *Phys. Fluids A*, **1,** 490.
LIN, S.P. & CREIGHTON, B., 1990, *J. Aero. Sci. and Tech.* (to appear).
PREZIOSI, L., CHEN, K. & JOSEPH, D.D., 1989, *J. Fluid Mech.*, **201,** 323.
REITZ, R.D. & BRACCO, F.V., 1982, *Phys. Fluids*, **25,** 1730.
RAYLEIGH, LORD, 1879, *London Math Soc.*, **10,** 361.
RENARDY, Y., 1985, *Phys. Fluids*, **28,** 3441.
SMITH, M.K., 1989, *Phys. Fluids A*, **1,** 494.
TAYLOR, G.I., 1963, *The Scientific Papers of G.I. Taylor*, **3,** No. 25, Cambridge, Cambridge U.
YIH, C.S., 1967, *J. Fluid Mech.*, **27,** 337.

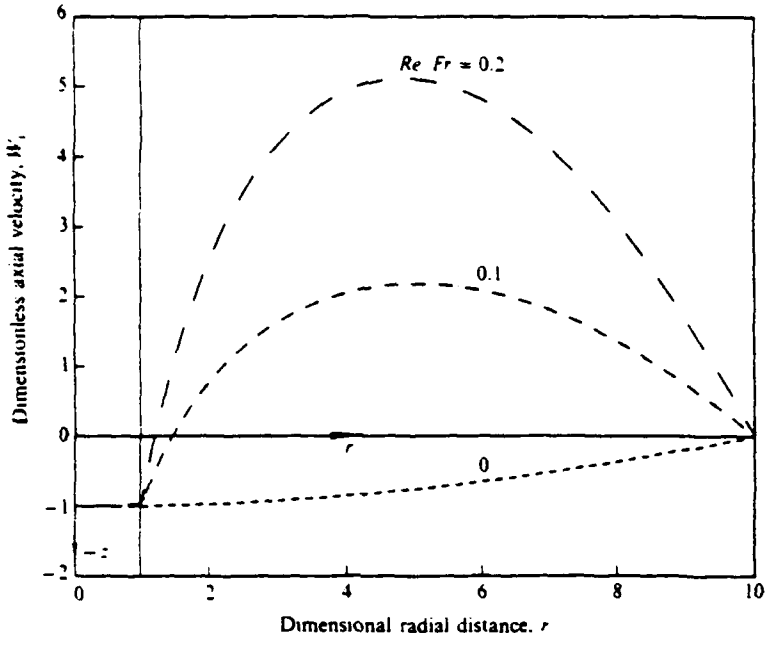


FIGURE 1. Velocity distribution. $Q = 0.0013$. $N = 0.018$. $l = 10$.

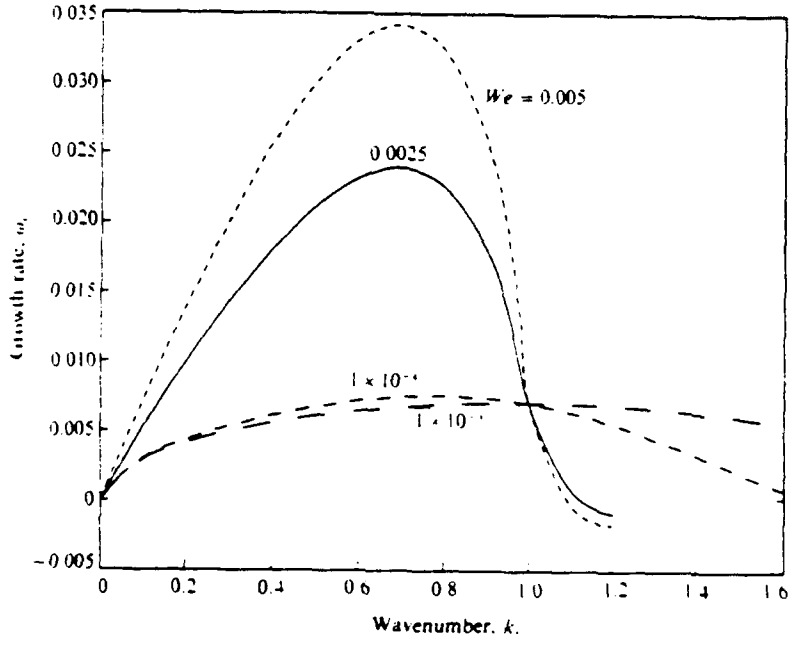


FIGURE 2. Destabilizing effect of surface tension on the Rayleigh mode $Q = 0.0013$. $N = 0.018$ $l = 10$. $R = 0$. $Re = 400$.

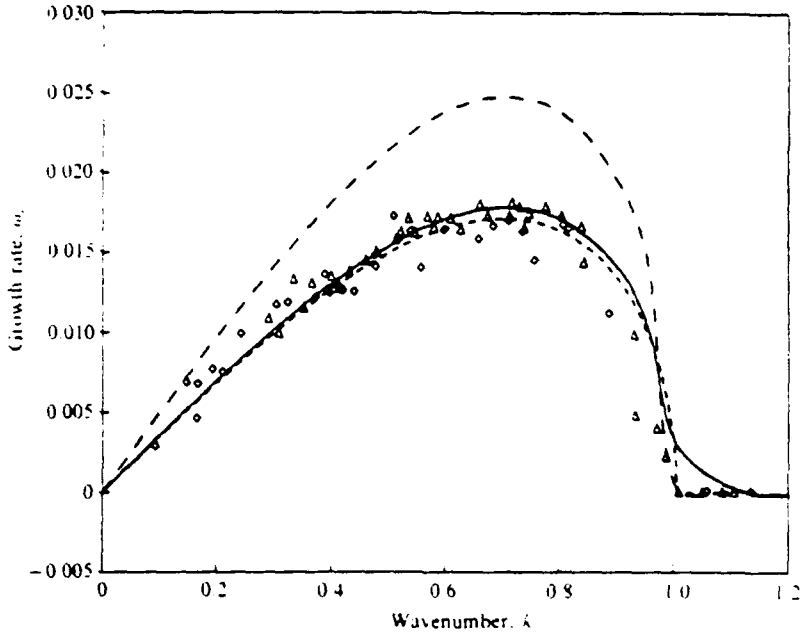


FIGURE 3. Comparisons of theories with experiments —— present work, $We = 0.0013$ - - - present work, $We = 0.0025$ ··· Rayleigh △ Goedde & Yuen ○ Donnelly & Glaberson $Re = 3000$, $Re\ Fr = 0$, $Q = 0.0013$, $N = 0.018$, $l = 10$

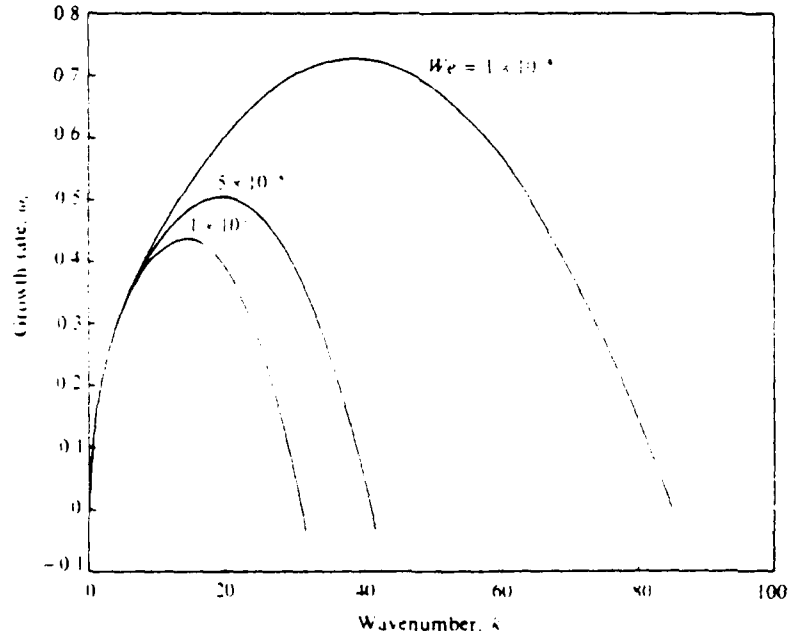


FIGURE 4. Effects of We on the Taylor mode. $l = 10$, $N = 0.018$, $Q = 0.0013$, $R = 0.2$, $Re = 400$

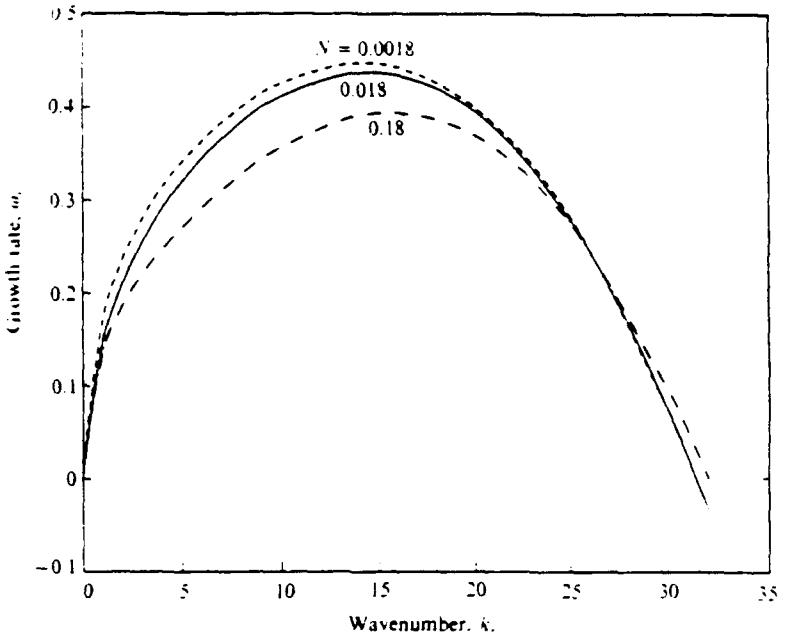


FIGURE 5. Effects of N on the Taylor mode. $l = 10$, $Q = 0.0013$, $We = 10^{-3}$, $R = 0.2$, $Re = 400$

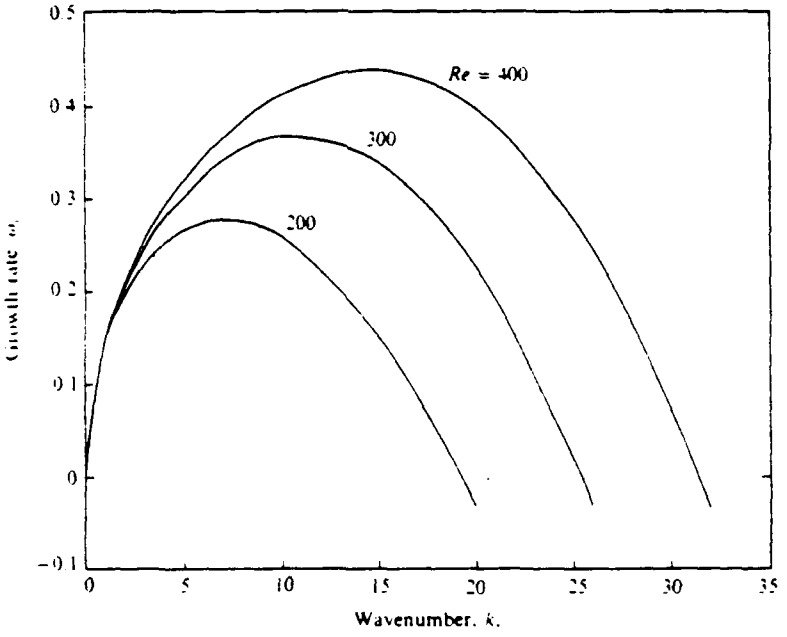


FIGURE 6. Effects of Re on the Taylor mode. $l = 10$, $N = 0.018$, $Q = 0.0013$, $We = 10^{-3}$, $Fr = 2000$

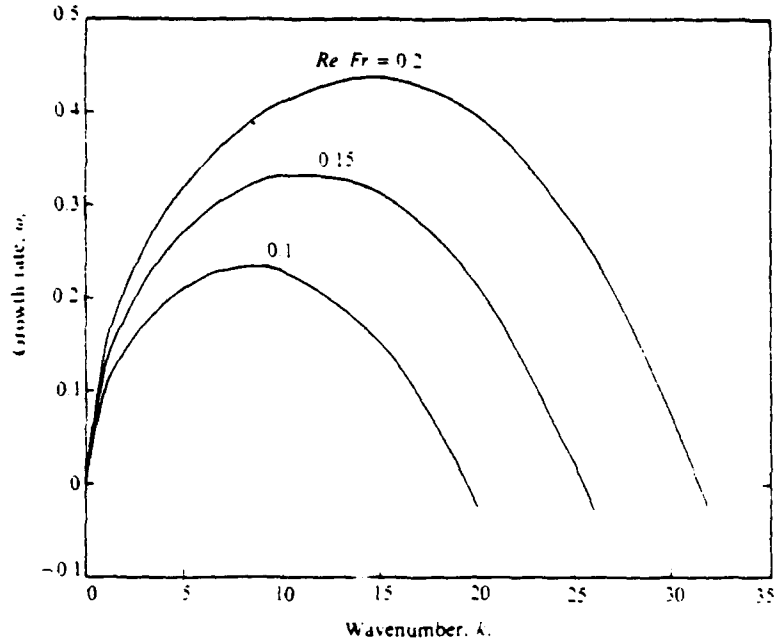


FIGURE 7. Effects of R_e on the Taylor mode. $l = 10$, $N = 0.015$, $Q = 0.0013$, $W_e = 10^{-3}$, $R = 0.2$, $Re = 400$.

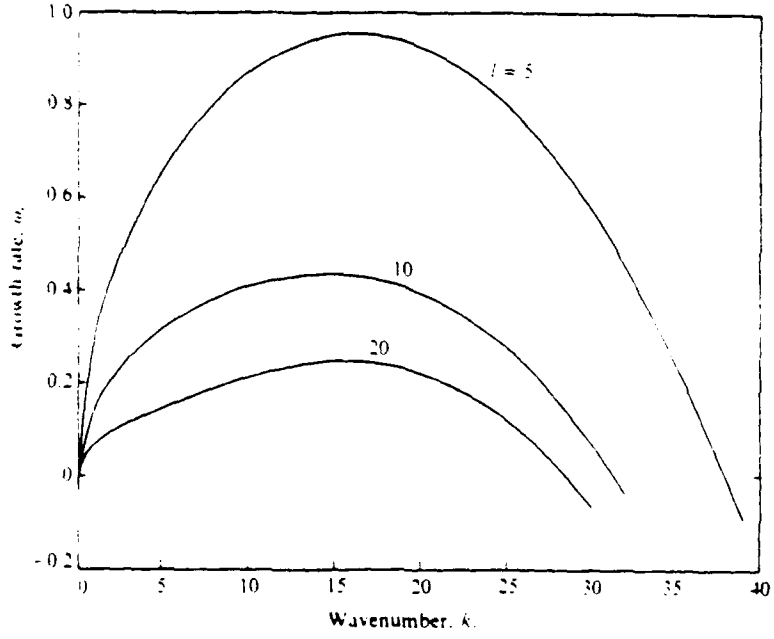


FIGURE 8. Effects of l on the Taylor mode. $N = 0.015$, $Q = 0.0013$, $W_e = 10^{-3}$, $R = 0.2$, $Re = 400$.

Communication and Control in SPMD Parallel Numerical Computations*

Dan C. Marinescu, John R. Rice, and Emmanuel A. Vavalis

Computer Sciences Department
Purdue University
West Lafayette, IN 47907, USA

May 8, 1990

Abstract

This paper investigates communication and control in SPMD parallel computations, and introduces the Events/Threads of control, E/T, model which allows qualitative and quantitative analysis of parallel execution. The principal component of the E/T model is the characteristic function $g(P)$ which relates the number of events to the number P of threads of control (usually a processor). Many properties of a computation follow from the behavior of $g(P)$, such as limits on the potential speedup. The model includes the effects of reads and writes in communication, algorithmic blocking, work intensity, etc. It is most appropriate for SPMD (Single Program Multiple Data) computations that are common in scientific applications. An experiment is described briefly which relates the the detailed behavior of a parallel computation, observed through monitoring, with the high level characterization provided by the E/T model.

1 Overview

The E/T model describes a parallel computation C as a collection of P threads of control and E events. Informally a *thread of control* is an agent capable to perform some work in behalf of C and an *event* is an explicit action perform by a thread of control in order to coordinate its activity with other threads of control. In a wider sense an event is a change of state of a thread of control.

Modeling and analysis of numerical problems which lend themselves to the Same Program Multiple Data, SPMD, paradigm are the focus of our investigation [11], [12]. Communication and control latency can strongly influence the performance of these computations and we use the E/T model to analyze this influence. Informally a *SPMD computation* is performed whenever all processing elements, PE's of a parallel machine execute the same program on different data. SPMD computations lead to a collection of similar threads of control therefore their modeling and analysis seems an easier task than the analysis of non-homogeneous computations with a large number of unrelated threads of control.

The E/T model can used for *qualitative analysis* of a parallel computation C, an analysis based upon the study of the *characteristic function* g, which relates the number of events, E, and the number of threads of control P. $E = g(P)$ of C. An *optimal* parallel computation with P threads of

*Work supported in part by ARO grant DAAG03-86-K-0106.

control is a computation characterized by a linear function $g(P)$. If $g(P)$ cannot be expressed as a polynomial then there is little hope that $\mathcal{C}$ will ever be performed efficiently. Consider two parallel computations $\mathcal{C}_1$ and $\mathcal{C}_2$ with P threads of control which represent two different implementation of an algorithm $\mathcal{A}$ or implementations of two different algorithms $\mathcal{A}_1$ and $\mathcal{A}_2$ which perform the same task. If the characteristic functions of $\mathcal{C}_1$ and $\mathcal{C}_2$ are in the relation $g_1(P) < g_2(P)$ for $P_1 \leq P \leq P_2$ then we have a high degree of confidence that $\mathcal{C}_1$ performs better than $\mathcal{C}_2$ in the range $P_1 \leq P \leq P_2$ for a wide variety of parallel architectures.

Whenever more information about a parallel computation $\mathcal{C}$ is available, for example when the sequence of events occurring in a thread of control can be identified or when the characteristics of the parallel machine executing $\mathcal{C}$ are known then the E/T model is capable of providing more accurate assessments about the expected performance of $\mathcal{C}$. A *quantitative analysis* can only be carried out if the threads of control exhibit some form of invariance to data dependencies, in other words if data dependencies can only alter the timing but not the order of events in a thread of control.

A first type of quantitative analysis is a *static analysis*. This is an analysis of the *mapping* from a directed acyclic graph, $\mathcal{D}$ to a parallel computation $\mathcal{C}$. The E/T model is used to determine the computation and communication workload. The computation workload can be analyzed at different levels, e.g., the amount of computation between two consecutive events, the workload per thread of control, and the total workload of $\mathcal{C}$. Similarly, communication workload can be characterized by the amount of data transferred during a single event, the amount of data transferred per thread of control, and the total amount of data transferred at the computation level. The effects of synchronization and blocking are not captured by the static analysis.

A second type of quantitative analysis is the *dynamic analysis* concerned with *schedules* which associate times with events. At this stage a detailed knowledge of the hardware is necessary in order to determine the time required to perform computations and the time to send and receive data. Alternatively, performance monitors and execution traces for a selection of data may provide sufficient knowledge to carry out a dynamic analysis. This analysis reveals the effects of synchronization and blocking.

Static analysis is often susceptible of an analytical approach but the dynamic models only seldom lead to a tractable analysis. Often dynamic models can be constructed only through monitoring the actual execution of $\mathcal{C}$ on a particular parallel system.

2 Qualitative Analysis Of Parallel Computation in the E/T Model

2.1 Basic assumptions

We propose a model for parallel computing based upon events and threads of control, the *E/T model*. A parallel computation $\mathcal{C}$ with P threads of control and E events is described by its *characteristic function* g defined by $E = g(P)$. The model is based upon two assumptions:

(a) *Conservation of work.* Any work required by a computation $\mathcal{C}(1)$ with one thread of control has to be performed by one of the threads of control of $\mathcal{C}(P)$, the parallel computation with P threads of control.

(b) *$W(P)$, the work required by a parallel computation is an increasing function of the number of threads of control, P.*

The first assumption needs little justification. It is an immediate consequence of the view that a thread of control is an agent performing some work in behalf of $\mathcal{C}$. To carry out a computation

with P threads of control simply means to redistribute in some fashion the work which otherwise would be carried out by only one thread. Call this constant amount of work reflecting the work conservation principle W_{cons}.

The second assumption is supported by the following arguments. An event is associated with every communication and control act. Any thread of control needs to communicate with other threads at least at the instance when it is initiated when some work is assigned to it, and at the termination time, when it has to communicate its results. It follows that $g(P)$ is an increasing function of P. Moreover any event requires a small amount of additional work, say θ, to be carried out by the thread of control when an event occurs. Let $W_{cc}(P)$ denote the additional amount of work required by $C(P)$ for communication and control. The previous arguments show that $W_{cc}(P)$ given by

$$W_{cc}(P) \geq \theta \times E = \theta \times g(P) \tag{2.1}$$

which is an increasing function of P. Thus, while $W_{cc}(P)$ might not increase monotonically, it is plausible to assume that the variations from the trend are small and that $W_{cc}(P)$ is increasing. But $W(P)$, the work carried by $C(P)$ consists of at least wo components the first one, W_{cons}, independent of P and the second one, $W_{cc}(P)$, an increasing function of P

$$W(P) = W_{cons} + W_{cc}(P). \tag{2.2}$$

A parallel computation C with P threads of control is considered to be *optimal* iff $E = \mathcal{O}(P)$, the number of events in C is linear in P.

Some of the algorithms we have encountered exhibit a convex characteristic function $g(P)$. We show that if the characteristic function $g(P)$ is convex then: the speedup has a maximum for some finite $P = P_{smax}$ and it is a concave function for $P > P_{smax}$.

2.2 Threads of control and events

The basic idea of the model is to describe a parallel computation C in terms of *threads of control* and *events*. Important properties of C are its duration T and work intensity $w(t)$. The *work intensity* is the actual measure of work performed as a function of time, e.g., operations per second. The work associated with C is

$$W = \int_0^T w(t)dt. \tag{2.3}$$

In view of the previous discussion the work intensity $w^i(t)$ associated with thread ϕ^i has two components

$$w^i(t) = w^i_{cons}(t) + w^i_{cc}(t) \tag{2.4}$$

where $w^{(i)}_{cons}$ is from the work assigned to the thread by virtue of the work conservation principle, and the second one, $w^i_{cc}(t)$ represents the work intensity for communication and control. Note that $w^i_{cons}(t)$ and $w^i_{cc}(t)$ cannot be non-zero simultaneously.

The duration T of $C(P)$ is expected to depend upon the number P of threads of control of $C(P)$. The work performed by the ith thread, ϕ^i, is

$$W^i = \int_0^T w^i(t)dt = \int_0^T w^i_{cons}(t)dt + \int_0^T w^i_{cc}(t)dt. \tag{2.5}$$

The total work required by $C(P)$ is thus

$$W(P) = \sum_{i=1}^{P} W^i = \sum_{i=1}^{P} \int_0^T w^i(t)dt. \tag{2.6}$$

The thread ϕ^i can be in one of two states at time t: *active* if $w^i_{cons}(t) > 0$, and *suspended* if $w^i_{cons}(t) = 0$. When the thread ϕ^i is suspended then it can be either *communicating* if $w^i_{cc}(t) > 0$, or *blocked* if $w^i_{cc}(t) = 0$, as shown in Figure 1 (which is explained later).

A parallel computation $C(P)$ may have several threads of control ϕ^i active at any given time t. Call $\nu_{act}(t)$ *the number of threads active*, $\nu_{cc}(t)$ *the number of threads communicating* and $\nu(t)$ the *number of threads non-blocked*, either active or communicating at time t. Note that $\nu(t)$ is sometimes called the *profile of the parallelism*, [14]. Clearly

$$\nu(t) = \nu_{act}(t) + \nu_{cc}(t) \tag{2.6a}$$

and

$$1 \leq \nu(t) \leq P \text{ for } 0 \leq t \leq T(P). \tag{2.7}$$

We say that the system changes its state at time t if $\nu_{act}(t - \epsilon) \neq \nu_{act}(t + \epsilon)$ for any positive ϵ. To mark the change of state, we say that an *event* $e(t)$ has occurred at time t. If thread ϕ^i has changed state at time t, we denote the event by $e^i(t)$. Note that we make the following convention: an event is associated only with the transition from active to suspended state. The duration of an event is equal to the time spent by the thread in the suspended state.

For the sake of convenience we consider that all P threads of control are created at time $t = 0$ and exist until time $t = T(P)$. In addition, we assume that there are two intervals of time when only one thread of control is active, $\nu(t) = 1$ for $0 \leq t \leq t_s$ and for $T(P) - t \leq t \leq T(P)$. The times t_s and t_e are called *start parallel* and *end parallel* times, respectively. At t_s the thread of control active initially, ϕ^1, explicitly performs an action to assign a part of work to a thread ϕ^2, which changes its state from suspended to active, ϕ^1 is called a *parent* of ϕ^2. This process has to be repeated at least P times, such that each thread must become active at least once.

In case of a serial computation, only one thread of control is active at any time t. Without loss of generality, we assume that a serial computation, $C(1)$ has only one thread of control active at any time t.

In a parallel computation $C(P)$ changes of state occur due to the need for communication and control. Such communication must take place at least once during the lifetime of ϕ^i, otherwise ϕ^i would not be able to coordinate its work with other threads. Communication between two threads of control, ϕ^i and ϕ^j takes place as the sender, say ϕ^i, performs an explicit action of making available private information, and the receiver, say ϕ^j, performs an explicit action to access this information. The terms *sender* and *receiver* are considered in the sense of information theory and the E/T model is not concerned with the mechanisms used for communication. Sending and receiving may be performed in different ways, such as by message passing or by accessing shared data.

Every time a thread ϕ^i performs an explicit action for communication or control, our model assumes the behavior illustrated in Figure 1. Note that the workload intensities associated with the thread ϕ^i exhibit the following behavior

$$\begin{aligned}
&w^i_{cons}(t) > 0 \quad \text{for} \quad t \leq t_{suspend} \text{ and } t \geq t_{reactivate} \\
&w^i_{cons}(t) = 0 \quad \text{for} \quad t_{suspend} < t < t_{reactivate} \\
&w^i_{cc}(t) > 0 \quad \text{for} \quad t_{suspend} < t < t_{block} \text{ and } t_{resume} < t < t_{reactivate} \\
&w^i_{cc}(t) = 0 \quad \text{for} \quad t_{block} \leq t \leq t_{resume}
\end{aligned} \tag{2.8}$$

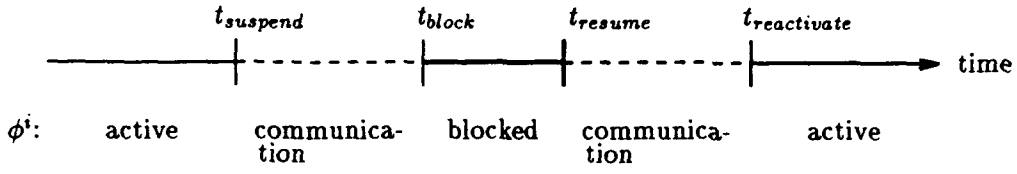


FIGURE 1: The states of the thread of control ϕ^i when an event $e_{t_{suspend}}$ occurs.

The additional work for communication and control, θ in (2.1) reflects the work associated with the periods when $w^i_{cc}(t)$ is non-zero. A blocking period may occur only for some events. For example, in a message passing system, an asynchronous write operation does not experience blocking, while a synchronous read may experience blocking if the data has not been received yet. In a shared memory system, both reading and modifying a shared data element may experience blocking.

It is difficult to predict the duration of a blocking period, therefore, knowing that an algorithm for matrix multiplication requires say, $\mathcal{O}(n^2/p^{2/3})$ communication steps, for two $n \times n$ matrices, using p processors [1], does not translate easily into statements concerning communication time.

2.3 W,T characterization of a parallel computation

Statements about a computation $\mathcal{C}$ can be made when the amount of work, W and the time T required by $\mathcal{C}$ are known. To simplify the discussion, let us assume that the work intensity associated with thread ϕ^i is constant when the thread is not blocked,

$$w^i(t) = \begin{cases} I & \text{if } \phi^i \text{ is active or communicating} \\ 0 & \text{if } \phi^i \text{ is blocked.} \end{cases} \tag{2.9}$$

In this case if $\mathcal{C}$ is performed using a serial execution, i.e., as $\mathcal{C}(1)$, with only one thread as shown in Figure 2a, then there is a clear relationship between $W(1) = W_{cons}$ and $T(1)$, the execution time with one thread only:

$$W(1) = T(1) \cdot I. \tag{2.10}$$

The relationship between W and T is less obvious in case of multiple threads of control, as shown in Figures 2b and 2c, where two alternative computations $\mathcal{C}^A(2)$ and $\mathcal{C}^B(2)$ are used to perform $\mathcal{C}$. We observe that

$$W(1) < W^{(A)}(2) < W^{(B)}(2), \tag{2.11}$$

but

$$T^{(B)}(2) < T^{(A)}(2). \tag{2.12}$$

Even the simple question of which one of the two variations of $\mathcal{C}(2)$, $\mathcal{C}^A(2)$ or $\mathcal{C}^B(2)$, is better cannot be answered unambiguously, as $\mathcal{C}^B(2)$ requires less time, but more work than $\mathcal{C}^A(2)$.

The relationship between $W(P)$ and $T(P)$ is explored next. Consider the case described by equation (2.9). Then the work intensity can be expressed as

$$w(t) = w^i(t) \cdot \nu(t) = I \cdot \nu(t) \tag{2.13}$$

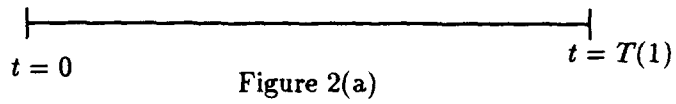


Figure 2(a)

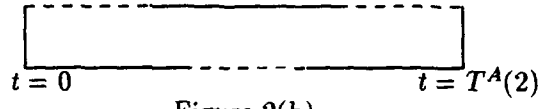


Figure 2(b)

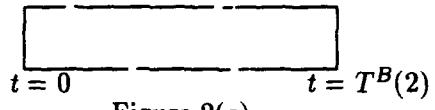


Figure 2(c)

FIGURE 2: (a) Sequential computation $\mathcal{C}(1) = < W(1), T(1) >$. (b) Computation reorganized to be $< \mathcal{C}^A(2) > = < W^A(2), T^A(2) >$ with two threads of control. (c) Computation reorganized in a different way to be $\mathcal{C}^B(2) = < W^B(2), T^B(2) >$ with two threads of control. Solid lines represent work periods and dotted lines blocked periods.

with $\nu(t)$ the number of threads of control non-blocked at time t. The work $W(P)$ associated with $\mathcal{C}(P)$, can be expressed as

$$W(P) = \int_0^{T(P)} w(t)dt = I \int_0^{T(P)} \nu(t)dt. \tag{2.14}$$

Define the *expected number of threads non-blocked* (active or communicating) at time t as

$$\bar{\nu}(P) = \frac{1}{T(P)} \int_0^{T(P)} \nu(t)dt. \tag{2.15}$$

From (2.14) and (2.15) it follows that

$$T(P) = \frac{W(P)}{I} \frac{1}{\bar{\nu}(P)}. \tag{2.16}$$

Similarly

$$w_{cons}(t) = w^i_{cons}(t) \cdot \nu_{act}(t) = I \cdot \nu_{act}(t) \tag{2.17}$$

with $\nu_{act}(t)$ the number of threads of control active at time t.

The work W_{cons} can be expressed as

$$W_{cons} = \int_0^{T(P)} w_{cons}(t)dt = I \int_0^{T(P)} \nu_{act}(t)dt. \tag{2.18}$$

Define the *expected number of threads active at time t* as

$$\bar{\nu}_{act}(P) = \frac{1}{T(P)} \int_0^{T(P)} \nu_{act}(t)dt. \tag{2.19}$$

Then we have

$$W_{cons} = IT(P)\bar{\nu}_{act}(P). \tag{2.20}$$

But $W_{cons} = W(1) = IT(1)$ hence

$$\frac{T(1)}{T(P)} = \bar{\nu}_{act}(P). \tag{2.21}$$

2.4 The expected amount of work per thread of control

To study the asymptotic behavior of a parallel computation C when, P, the number of threads of control increases, we first investigate the behavior of the function

$$\bar{w}(P) = \frac{W(P)}{P}. \tag{2.22}$$

Consider first computations C with $E = \mathcal{O}(P)$ where each thread of control ϕ^i experiences only a few communication events, in addition to the events to initialize and terminate ϕ^i. An example of such a computation is a plotting computation when each thread operates in isolation upon its private data to create its part of the plot and makes the results available at the end. In this case

$$\bar{w}(P) = \frac{W_{cons}}{P} + \mathcal{O}(1). \tag{2.23}$$

For such parallel computations the expected amount of work per thread of control is a monotonically decreasing function of the number of threads of control as shown in Figure 3.

Consider now parallel computations with $E = \mathcal{O}(P^2)$, for example when each thread of control communicates with every other thread of control during its lifetime. In this case the asymptotically expected amount of work per thread of control is

$$\bar{w}(P) = \frac{W(P)}{P} = \frac{W_{cons}}{P} + \mathcal{O}(P). \tag{2.24}$$

The amount of work per thread of control exhibits a minimum for a certain P_{opt} and it is a monotonically increasing function of P when $P > P_{opt}$. Clearly, P_{opt} increases as W_{cons} increases. For a given δ the range of P such that $W(P) - W(P_{opt}) < \delta$ is usually fairly large, $\bar{w}(P)$ is relatively flat around its minimum. This case is illustrated in Figure 4.

If $g(P) = \mathcal{O}(P^n)$ with $n \geq 3$ then $\bar{w}(P)$ increases rapidly with P and massive parallelism is unlikely to be advantageous unless W_{cons} is enormous.

In conclusion $\bar{w}(P)$ provides a useful *signature* of C. This signature indicates that massive parallelism is truly advantageous only when $E = \mathcal{O}(P)$. In this case the $\bar{w}(P)$ is a monotonically decreasing function of P so that if reasonable load balancing is achieved among the threads of control then the processors are used efficiently. When $E = \mathcal{O}(P^2)$ then there exists an optimum number of threads of control which minimize the expected workload per thread, and $\bar{w}(P)$ is relatively flat around that minimum. If the characteristic function $E = g(P)$ is either a polynomial of degree $n \geq 3$ or similar type of behavior, then $\bar{w}(P)$ exhibits a minimum for a lower value of P_{opt} and $w(P_{opt})$ is higher than in the previous case. The efficiency of computations in this class is rather sensitive to the choice of P, the number of threads of control.

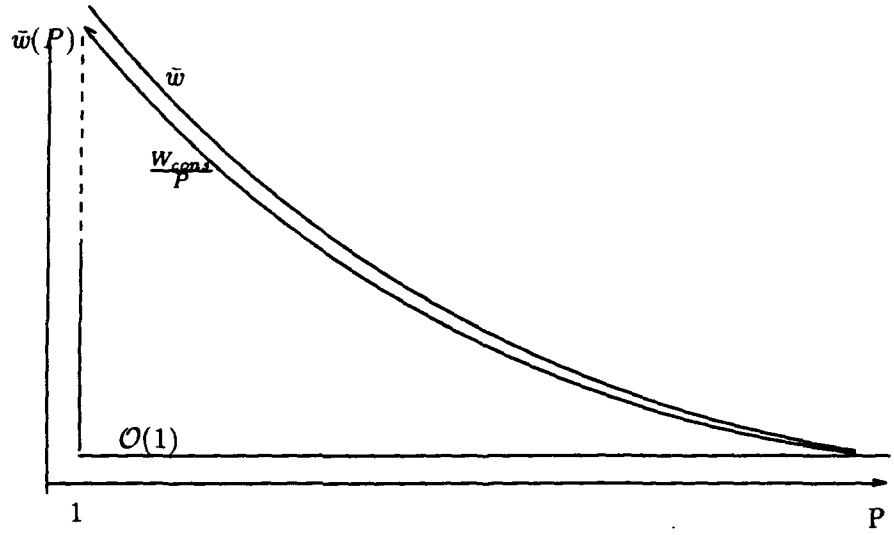


FIGURE 3: The expected work per thread of control $\bar{w}(P)$ function of the number of threads of control, P, for a parallel computation of a fixed problem size with $E = \mathcal{O}(P)$ according to equation (2.23).

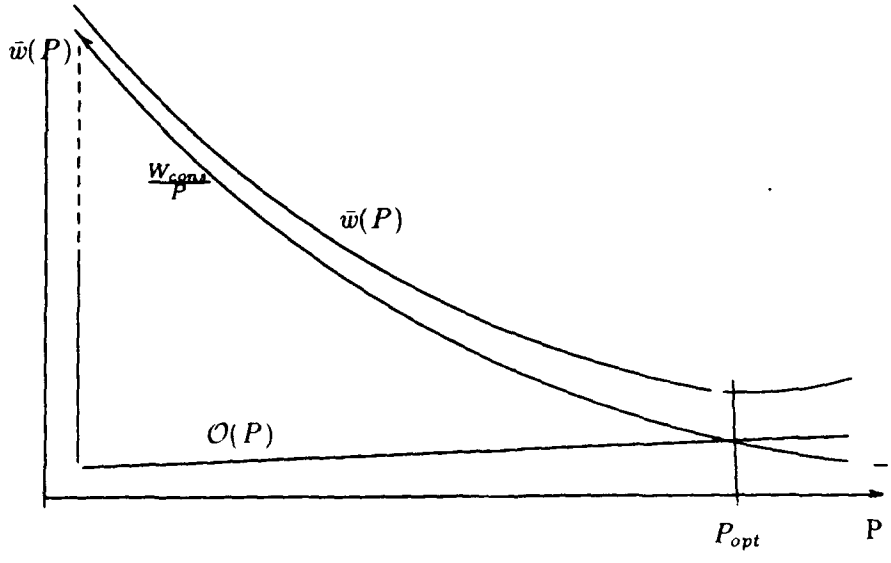


FIGURE 4: The expected work per thread of control $\bar{w}(P)$ function of the number of threads of control, P, for a parallel computation of fixed problem size with $E = \mathcal{O}(P^2)$ according to equation (2.24).

2.5 The speedup

The speedup $S(P)$ is defined as the ratio of the computation time with one thread of control to the computation time with P threads, $P > 1$, that is

$$S(P) = \frac{T(1)}{T(P)} . \tag{2.25}$$

First observe, that according to (2.21),

$$S(P) = \bar{\nu}_{act}. \tag{2.26}$$

Similar results have been reported, see for example [3], but in the framework of the E/T model, *the speedup is equal to the expected number of threads active, performing work assigned by virtue of the conservation law.* The speedup is less than ν, the expected number of threads running (active or communicating). Since $\nu_{act} \leq \nu \leq P$, it follows that

$$S(P) \leq P. \tag{2.27}$$

Consider now the asymptotic behavior of $S(P)$. From (2.16) and (2.20) it follows that

$$S(P) = \frac{W(1)}{W(P)}\bar{\nu}(P). \tag{2.28}$$

We introduce the *efficiency*, $b(P)$ as the ratio between the expected amount of work per thread of control using P threads, $\bar{w}(P) = W(P)/P$, and the work $W(1) = W_{cons}$ using one thread (sequential execution), that is

$$b(P) = \frac{W(P)}{PW(1)} . \tag{2.29}$$

Note that $W(P) \geq W(1)$. Hence

$$b(P) \leq 1/P. \tag{2.30}$$

The expected fraction $a(P)$ of non-blocked threads in $C(P)$ is given by

$$a(P) = \frac{\bar{\nu}(P)}{P} \quad 0 \leq a(P) \leq 1. \tag{2.31}$$

Then we have

$$S(P) = \frac{a(P)}{b(P)} . \tag{2.32}$$

The study of the asymptotic behavior of $S(P)$ when P becomes very large is reduced to the problem of the asymptotic behavior of $a(P)$ and $b(P)$. From the definitions of $a(P)$, $b(P)$ and $W(P)$, the following conclusion can be drawn:

(a) For parallel computations with $g(P) = \mathcal{O}(P)$, we have $b(P) = 1/P + \textit{constant}$ for large P and hence $S(P) < \textit{constant}$ for a large number of threads of control.

(b) For parallel computation with $g(P) = \mathcal{O}(P^n)$ with $n \geq 2$, $b(P)$ is an increasing function of P and hence $S(P)$ tends to zero asymptotically.

Let us now consider the case of *scaled* execution [5] where the computation size increases linearly with the number of processors (threads of control) used, namely

$$W(1) = \mathcal{O}(P)$$
$$T(1) = \mathcal{O}(P). \tag{2.33}$$

Scaled speedup $SS(P)$ is defined for scaled execution by equation (2.25). The quantities $a(P)$ and $b(P)$ are analogously defined and relations (2.27) through (2.32) hold. The asymptotic behaviors of $b(P)$ and $SS(P)$ in this case are as follows:

(sa) For parallel computation with $g(P) = \mathcal{O}(P)$, $SS(P)$ is an increasing function of P.

(sb) For parallel computation with $g(P) = \mathcal{O}(P^2)$, $SS(P) <$ *constant* for large P.

(sc) For parallel computation with $g(P) = \mathcal{O}(P^n)$ and with $n \geq 3$, $SS(P)$ tends to zero for large P.

It seems reasonable to question whether scaled execution and parallel computations with $g(P) = \mathcal{O}(P)$ are compatible with one another. A computation is called *embarrassingly parallel* if

$$W(P) = W(1) + \text{ constant}, \nu(t) = P \text{ for } t_0 \leq t \leq T(P) - t_0$$

and

$$E = \text{ constant } \times P \ .$$

This terminology is especially appropriate if the constants involved are small. For these computations we have $a(P) = 1 - 2t_0/T(P)$ which is asymptotically 1 and $b(P) = (W(1) + \textit{constant}/(P \cdot W(1))$ which is asymptotically $1/P$. Thus for embarrassingly parallel computations, we have

$$SS(P) = P + \mathcal{O}(1). \tag{2.34}$$

Such computations arise when the work can be partitioned into P parts at the beginning and then done completely independently by the processors. Thus we can achieve optimal speedup for such computations.

Divide and conquer algorithms may provide scaled speedup nearly as great. Let $P = 2^k$ and assume conservatively that

1. The work after each division of the problem is the same as $W(1)$.
2. The events take place only at dividing the computation up and recombining the results.

Then we see that $W(P) \leq W(1) \log P$, $E = \mathcal{O}(P)$ and we compute that, asymptotically,

$$\begin{array}{lcl} b(P) & = & \log P/P \\ T(P) & \leq & T(1) \times 2 \log P \\ \bar{\nu}(P) & \leq & \text{constant } \times P \\ SS(P) & = & \mathcal{O}(P/\log P). \end{array} \tag{2.35}$$

In [3] the *average parallelism* was proposed as a high level characterization of software structure. The average parallelism is defined as the speedup, given an unbounded number of processors. The previous discussion shows that there are parallel algorithms, such that $S(P)$ or $SS(P)$ tend asymptotically to zero, hence the average parallelism does not provide a useful characterization of such applications.

2.6 Analysis when E = g(P) is a convex function

The qualitative analysis continues with the case when $g(P)$ is a convex function. Several algorithms we have examined suggest that $g(P)$ is often a convex function of P as well as increasing.

Theorem 2.1 *If $E = g(P)$ is increasing and convex function for $P \geq 1$ then $W(P)$ is also convex. Let P_{smax} be the unique solution of*

$$P = [\alpha + g(P)]/g'(P). \tag{2.36}$$

where $\alpha = W(1)/\theta$. Then $S(P)$ is increasing for $P < P_{smax}$ and decreasing for $P > P_{smax}$.

Proof: We have $W(P) = W_{cons} + \theta \times g(P)$ so $W(P)$ is convex if $g(P)$ is. The speedup $S(P)$ may be expressed by

$$S(P) = T(1)/T(P) = \frac{P}{1 + [\theta/W(1)]g(P)} = \frac{W(1) + \theta \times g(P)}{W(1) + \theta \times g(P)} =$$

$$\frac{W(1)/I}{(W(1) + \theta \times g(P))/(PI)}. \tag{2.37}$$

Combine $\theta/W(1)$ into the constant $1/\alpha$ and differentiate this expression to obtain

$$S'(P) = \frac{\alpha + g(P) - Pg'(P)}{(\alpha + g(P)^2}. \tag{2.38}$$

Set $g(P) = Ph(P)$ with $h'(P) \geq 0$ by the convexity assumption. Then we have

$$S'(P) = \frac{\alpha - P^2h'(P)}{(\alpha + g(P))^2} = [\alpha + Ph(P) - P(h(P) + P^2h'(P))]/(\alpha + Ph(P))^2. \tag{2.39}$$

Since $h'(P)$ is positive, (2.39) is zero exactly once. A manipulation of (2.38) allows one to obtain (2.36) as asserted by the theorem.

We may use this result to provide estimates of maximum speedups and the corresponding number of threads of control (processors) for a few cases as given in Table 1. Note that the speedups given are maximums, other factors (e.g., lack of load balancing) can make them smaller.

2.7 Additional workload due to algorithmic effects

To characterize the work required by a computation C with P threads of control, we have identified two components so far, the intrinsic work, W_{cons} assigned to the threads by virtue of the conservation law and $W_{cc}(P)$ the work for communication and control. However, the transformation from a computation $C(1)$ with one thread of control only to a computation $C(P)$ with P threads of control, often introduces additional work called in the following *algorithmic workload* and denoted by $W_{alg}(P)$, i.e., we have

TABLE 1: Values of maximum speedups and corresponding P_{smax} for $\alpha = W(1)/\theta = 10^6$, 10^4, 10^2 and $g(P) = P^n$ for $n = 1.5$, 2, 2.5, and 3.

$g(P) = P^{1.5}$	$\alpha = 10^6$	$\alpha = 10^4$	$\alpha = 10^2$
Speedup	5291	245	11
P_{smax}	16000	736	34

$g(P) = P^2$	$\alpha = 10^6$	$\alpha = 10^4$	$\alpha = 10^2$
Speedup	500	50	5
P_{smax}	1000	100	10

$g(P) = P^{2.5}$	$\alpha = 10^6$	$\alpha = 10^4$	$\alpha = 10^2$
Speedup	128	20	3.2
P_{smax}	213	34	5

$g(P) = P^3$	$\alpha = 10^6$	$\alpha = 10^4$	$\alpha = 10^2$
Speedup	53	11.4	2.4
P_{smax}	79	17	4

$$W(P) = W_{cons} + W_{cc}(P) + W_{alg}(P). \qquad (10)$$

We study alternative parallel ways to do the same work and exclude from considering complete changes of algorithms. However, we do allow the work to be transformed in various ways using simple equivalences of operations. Specifically,

$$C(1) = 4 * 5 * 9 * 412$$

is equivalent to $C^A(2)$ defined by

Thread 1	*Thread 2*
4 * 5 = 20	9 * 412 = 3707
20 * 3707 = 74160	

but is not equivalent to $C^B(2)$ defined by

Thread 1	*Thread 2*
log 4 + log 5 = 1.301	log 9 + log 412 = 3.569
Blocked	1.301 + 3.569 = 4.871
$10^{4.871} = 74160$	

Similarly

C(1) = sort (list1), sort (list2), sort (list3), concatenate (list1,list2,list3)

is equivalent to $C^A(2)$ defined by

Thread 1	*Thread 2*
sort (list1)	sort (list2)
sort (list3)	concatenate (list1,list2) = list4
concatenate (list4,list3)	

The question of when two computations are equivalent is a subtle one, which we do not attempt to make precise here because most parallelizations of algorithms introduce new work, called $W_{alg}(P)$ above. However, this concept is useful in heuristic discussions of the work of different algorithms for the same task. The range of possible effects of parallelization of an algorithm are very large, but there seem to be two common ones. Later we examine realistic algorithms in some detail, but here we consider a simple algorithm.

(a) *Algorithmic overhead associated with individual events.* An algorithm may have internal information that needs to be updated when new external information is received. For example, an iteration on one domain receives values from an adjacent part of another domain. These values affect the error estimates along the domain boundary, which in turn, affect the estimate of convergence ratio and relaxation factors. All these estimates and factors must be recomputed when new information is received. A very high rate of events could distort the computation until most of the work done is recomputing these estimates and factors, instead of carrying out the iteration. This example of algorithmic overhead behaves like communication and control work (indeed, the update computations are to control the numerical behavior of the iterations).

It is plausible to merge this work into W_{cc} even though it appears to be algorithmic work. There seems to be no advantage in carrying along two independent sources of work proportional to the number of events.

(b) *Algorithmic overhead associated with (long) event free periods.* The computation in one thread might profitably use information from other threads to reduce its algorithmic work. For example, a search party of 10 men covering an area will be forced to have all 10 men search the entire area unless there is communication between them about which areas have already been searched. Many analogs of this simple situation exist in computational search algorithms. Another example occurs when long periods between events force one thread to save data for future communications, buffers or queues can become full, requiring extra work to save data in special ways, or the thread can even become idle (introducing an unnecessary event) waiting to empty space for saving data. A more subtle example is a set of parallel iterations where long periods of no communication means that the iterations are running but not accomplishing anything useful.

The *average event interval* is $T(P) \cdot P/E$ and this type of algorithmic overhead is modeled by

$$W_{alg}(P) = \mathcal{O}\left(\frac{T(P) \cdot P}{E}\right). \tag{2.42}$$

Thus, $W_{alg}(P)$ is proportional with the expected time between successive events. The larger this interval, the more likely it is that a thread will perform unnecessary work, or it will duplicate work done by another thread.

This informal presentation shows that the avoidance of additional work for the algorithm occurs only in embarrassingly parallel computations.

Note that long event-free periods are usually associated with other types of undesirable effects besides work duplication, effects related to practical implementation of partial computations on real machines with finite memory. Whenever the *lifetime of partial results* defined as the interval between the instance partial results are produced by thread ϕ^i and the instance they are consumed by thread ϕ^j, is large, then the memory requirements for $\mathcal{C}$ become substantial as observed in [10].

The algorithmic workload and the lifetime of partial results discussed in this section are difficult to be captured and they will be largely ignored in this study. But it is conceivable to assume that a second conservation law of the type

$$W_{cc}(P) + W_{alg}(P) = f(P) \tag{2.43}$$

is valid and captures the effect that for a given P the sum of the work for communication and control and algorithmic work is constant and any gain obtained by reducing the number of events is compensated by increasing the level of work duplication.

2.8 Extensions of the model to non-SPMD computations

The SPMD parallel computations are homogeneous, all threads of control ϕ^i, $1 \leq i \leq P$, exhibit similar behavior. In the framework of the E/T model, this translates into the fact that the characteristic functions of all threads of control $g^i(P)$ are identical

$$g^i(P) = \frac{1}{Pg(P)} \quad 1 \leq i \leq P. \tag{2.44}$$

In case of non-SPMD computations, the dissimularities among different threads of control is reflected by a partial ordering of threads, based upon the number of events associated with each thread. Call ϕ^ℓ the thread which has the largest number of events. In such a case, the E/T model requires one to identify the thread ϕ^ℓ and to study its characteristic function $g^\ell(P)$. Of course, things can become even more complex when the thread with the largest number of events is data dependent or its behavior changes widely with the data.

3 Quantitative Analysis of SPMD Parallel Computations

In this section we focus upon quantitative analysis of a parallel computation $\mathcal{C}$ and attempt to estimate measures of performance such as speedup, execution time, processor utilization, etc. Such an analysis is possible only if $\mathcal{C}$ exhibits only limited data dependencies. Dynamic computations in which the actual sequence of events in every thread of control are data dependent lead to an intractable analysis. Fortunately, most SPMD computations satisfy this condition, while the timing of different events in a thread of control may change depending upon the data, the actual sequence of computations and events occurring in a thread of control is invariant to input data.

First, we consider a *static analysis* which is an analysis of the *mapping* from a directed acyclic graph, $\mathcal{D}$ to $\mathcal{C}$. The computation and communications workloads can be analyzed at different levels as discussed in the overview. The effects of synchronization and blocking are not captured by

the static analysis. The *dynamic analysis* is concerned with *schedules* which associate times with events. At this stage a detailed knowledge of the hardware is necessary.

The transition from the static to the dynamic stage of the quantitative analysis is difficult. A "superposition" property which would allow the extension of results obtained in stage one to stage two would be desirable but it seems that the occurrence of such a property is an exceptional event. Some of the problems encountered in this area are due to the difficulties to estimate the communication time between two processors, π_i and π_j. This time depends upon factors as

(a) The architecture of the system $\mathcal{H}$ and the number of processors in the system n. For example if we call τ the communication delay for a message of unit length then $\tau(n)$ is of the order of $\sqrt{n}$ for a grid, $\log_2 n$ for a hypercube and n for a ring interconnection network.

(b) The communication software, the communication protocols, the routing strategy, etc.

Effects unrelated to the parallel computation $\mathcal{C}$ but determined by the need to share resources in a multiuser system can only be considered during the dynamic analysis. Such effects are: fragmentation in communication, the need to split a large message into a number of packets, or processor sharing, in case of multiuser systems. These effects are extremely difficult to be captured by any model.

While static analysis is often susceptible to an analytical approach the dynamic models only rarely lead to a tractable analysis. Often dynamic models can be validated only through the process of monitoring $\mathcal{C}$. Static analysis may be useful to make decisions concerning scheduling in a multiuser e vironment. For example if the thread ϕ^i expects a large message from the thread ϕ^j then the erating system of the node where ϕ^i runs may suspend it, run another process and return to it ter the message has been received. Performance measures as *the average degree of parallelism*, [3] are clearly related to stage one in our approach and can be used successfully to make scheduling decisions as pointed out in [14].

3.1 Static analysis – Mapping in the E/T model

Mapping in the context of the E/T model is the process of deciding which computations **and** which problem data are assigned to every thread of control of $\mathcal{C}$. The two aspects of mapping, computation mapping and data mapping are closely related.

Let us for the moment consider the general case when a parallel algorithm $\mathcal{A}$ is given as a directed acyclic graph, DAG, $\mathcal{D} = (V, A)$ whose nodes $d_i \in V$ are computational tasks with given workload requirements and whose arcs, $a_i \in A$, represent both temporal and functional dependencies. In addition, the arcs have associated with them communication load requirements. The mapping of the DAG $\mathcal{D}$ to a parallel computations $\mathcal{C}$ is a process of deciding how many threads of control should $\mathcal{C}$ have and how different nodes of $\mathcal{D}$ will be assigned to the threads of control. The *computation mapping* consists of the following steps

(a) Choose the number P of threads of control ϕ^i, $1 \leq i \leq P$.

(b) Group together a number of nodes of $\mathcal{D}$ and assign them to ϕ^i.

The *data mapping* is the process of assigning problem data to different threads of control. Data mapping follows computation mapping and allows us to

(c) Define the sequence of actions performed by every thread ϕ^i, $1 \leq i \leq P$.

(d) Determine the sequence of events associated with every thread ϕ^i, $1 \leq i \leq P$.

Note that the SPMD execution corresponds to the case when the computation mapping assigns all nodes of $\mathcal{D}$ to every thread of control ϕ^i. The data mapping makes the execution of the P threads of control different.

Formally, the computation mapping is described as follows. Denote by N the number of nodes of $\mathcal{D} = (D, A)$, $N = |D|$. Then the work associated with the DAG, $\mathcal{D}$ is characterized by a $N \times 1$ vector $\mathbf{W}_{\mathcal{D}}$

$$\mathbf{W}_{\mathcal{D}} = [w_j] \tag{3.1}$$

with w_j the work associated with node $d_j \in V$. The mapping from $\mathcal{D}$ to $\mathcal{C}$ is characterized by the $(P \times N)$ ***computation mapping matrix*** $\mathcal{M}_{\mathcal{D}}$

$$\mathcal{M}_{\mathcal{D}} = [m_{ij}] \tag{3.2}$$

with

$$\begin{array}{l} m_{ij} = 1 \quad \text{if } d_j \textit{ is mapped into } \phi^i,\ 1 \leq i \leq P \text{ and } 1 \leq j \leq N \\ m_{ij} = 0 \quad \textit{otherwise.} \end{array} \tag{3.3}$$

The integer $m_i = \sum_{j=1}^{N} m_{ij}$ is called the *grouping factor of* ϕ^i and represents the number of nodes of $\mathcal{D}$ assigned to ϕ^i, with $1 \leq i \leq P$.

The data mapping associated with a domain $\mathcal{B}$ decomposed into K subdomains, b_j, is characterized by the $(P \times K)$ *data mapping matrix* $\mathcal{Q}_{\mathcal{B}}$

$$\mathcal{Q}_{\mathcal{B}} = [q_{ij}] \tag{3.4}$$

with

$$\begin{array}{l} q_{ij} = 1 \quad \text{if } b_j \textit{ is mapped into } \phi^i,\ 1 \leq i \leq P \text{ and } 1 \leq j \leq N \\ q_{ij} = 0 \quad \textit{otherwise.} \end{array} \tag{3.5}$$

The work performed by $\mathcal{C}(P)$ is characterized by a $P \times 1$ vector $\mathbf{W}_{\mathcal{C}}$,

$$\mathbf{W}_{\mathcal{C}} = [w^i] \tag{3.6}$$

with w^i the work assigned to ϕ^i by the mapping $\mathcal{M}$. Clearly,

$$\mathbf{W}_{\mathcal{C}} = \mathcal{M}_{\mathcal{D}} \mathbf{W}_{\mathcal{D}} \tag{3.7}$$

$$w^i = \sum_{j=1}^{N} m_{ij} w^j. \tag{3.8}$$

We consider a nondeterministic model and assume that the work process of $\mathcal{D}$ is stationary and the random variables w_j, $1 \leq j \leq N$ have mean μ_D and variance σ_D. The assumption of a stationarity process is common in the analysis of stochastic processes and it is necessary in order to promote tractability of our model. In general, the w_i are dependent random variables and their dependence is characterized by the *work covariance matrix of* $\mathcal{D}$, $\sigma_{\mathcal{D}}$

$$\sigma_{\mathcal{D}}^2 = [(\sigma_D^2)_{ij}] \tag{3.9}$$

with

$$(\sigma_D^2)_{ij} = \mathrm{Cov}(w_i, w_j). \tag{3.9'}$$

If we assume that the work process associated with the DAG $\mathcal{D}$ is stationary, it follows immediately that the work process associated with the mapping $\mathcal{C}$ is also stationary. The random variables w^i with $1 \leq i \leq P$ have a distribution with mean μ^i and variance σ^i such that

$$\mu^i = m_i \mu_D \tag{3.10}$$

$$(\sigma^i)^2 = m_i \sigma_D^2 + \sum_{j=1}^{m_i} \sum_{k=1}^{m_j} \mathrm{Cov}\ (w^j, w^k). \tag{3.11}$$

The work covariance matrix of $\mathcal{C}$ is $\sigma_{\mathcal{C}}^2$ defined by

$$\sigma_{\mathcal{C}}^2 = [(\sigma_{\mathcal{C}}^2)_{ij}] \tag{3.12}$$

with

$$(\sigma_{\mathcal{C}}^2)_{ij} = \mathrm{Cov}(w^i, w^j). \tag{3.13}$$

An important aspect of mapping is related to load balance. Intuitively, it seems desirable to assign to every thread of control an equal amount of work with the hope that such a load balanced mapping will eventually lead to the shortest possible execution time and to the best possible utilization of resources. An SPMD computation seems an ideal case from the load balance point of view since in this case all threads of control have assigned to them identical workload. But data mapping makes the execution different, the actual instruction execution sequence in each thread of control is different due to data dependencies and a perfect load balance is unlikely to be achieved. For this reason we consider a nondeterministic analysis and regard the workload associated with any thread of control as a random variable.

In addition to such *algorithmic load imbalance effects* there are *non-algorithmic* causes such as hardware failures and retrys, message retransmission, etc. To characterize the load imbalance associated with any given mapping we introduce a load imbalance factor, Δ defined in the following. Denote by Y the workload associated with the most heavily loaded thread of control.

$$Y = \max(w^1, w^2, \ldots, w^i, \ldots, w^P). \tag{3.14}$$

Call $\bar{Y}$ the expected value of the random variable Y and call $\bar{w}$ the mean value of μ^i defined as

$$\bar{w} = \frac{1}{P} \sum_{i=1}^{P} \mu^i. \tag{3.15}$$

Then Δ is defined implicitly as

$$\bar{Y} = \bar{w}(1 + \Delta) \tag{3.16}$$

In this expression $\bar{w}$ is the expected workload per thread of control and $\bar{Y}$ is the expected workload of the most heavily loaded thread of control.

To conclude this section we summarize the parameters which may be used to characterize statically a parallel computation $\mathcal{C}$, in addition to P, E and $W(P)$, which have been discussed previously.

κ^i – The number of events in the thread of control ϕ^i, $1 \leq i \leq P$.

α^i_j – The amount of work performed by the thread ϕ^i between two consecutive events e^i_j and e^i_{j+1}, for $1 \leq i \leq P$, $1 \leq j \leq \kappa^i$.

$\bar{\alpha}^i$ – The expected amount of work associated with an event in thread ϕ^i, computed as the mean value of α^i_j, $1 \leq i \leq P$, $1 \leq j \leq \kappa^i$.

W^i – The total workload associated with the thread ϕ^i, $1 \leq i \leq P$.

μ^i – The expected workload associated with the thread ϕ^i, $1 \leq i \leq P$.

$\bar{w}$ – The expected workload per thread of control.

β^i_j – The amount of data transferred when the event e^i_j in thread ϕ^i occurs, $1 \leq i \leq P$, $1 \leq j \leq \kappa^i$.

$\bar{\beta}^i$ – The expected amount of data transferred per event in thread ϕ^i, computed as the mean value of β^i_j, $1 \leq j \leq \kappa^i$.

β^i – The total communication load associated with thread ϕ^i, $1 \leq i \leq P$.

$\bar{\beta}$ – The expected value of $\bar{\beta}^i$.

Δ – The load imbalance factor.

Finally, we stress again that without the detailed information required to compute a schedule, the performance of a parallel computation can only be estimated. Such estimates may be used to compare different mappings $\mathcal{M}_i$ of a given algorithm $\mathcal{A}$ but no definite statements about the actual execution time, the speedup and/or the efficiency can be made at the stage. Two examples applying these ideas follow.

3.2 Examples

3.2.1 A parallel algorithm for matrix multiplication

Consider a parallel algorithm for matrix multiplication with P threads of control. This algorithm is described and analyzed in [1]. Let A and B be two $(n \times n)$ matrices. The algorithm requires the partitioning of A and B into q^2 disjoint submatrices A_{ij} and B_{jk}. Each submatrix is of size $(m \times m)$. The values of q and m are given by

$$\begin{aligned} q &= P^{1/3} \\ m &= \frac{n}{q} = \frac{n}{P^{1/3}} \,. \end{aligned} \tag{3.17}$$

Without loss of generality assume that $P = 2^a$ and we have $n = mq$ with a and m positive integers. The algorithm proceeds as follows. For every triplet (i, j, k) compute

$$\ell(i, j, k) = (i - 1)q^2 + (k - 1)q + j. \tag{3.18}$$

Clearly $1 \leq \ell(i, j, k) \leq P$ when $(i, j, k) \in [1, q]$. There are two steps:

A. Let the thread $\phi^{\ell(i,j,k)}$ perform the following actions

(A1) - Read the submatrix A_{kj} with m^2 elements.

(A2) - Read the submatrix B_{jk} with m^2 elements.

(A3) - Compute the submatrix $C_{ijk} = A_{im} \times B_{jk}$. This requires $\mathcal{O}(m^3)$ operations.

B. Organize the threads ϕ^ℓ with $1 \leq \ell \leq P$ as q^2 complete binary trees such that $\phi^{\ell(i,1,k)}$ will compute

$$C_{ik} = \sum_{j=1}^{q} C_{ijk} \quad 1 \leq i \leq q,\ 1 \leq k \leq q. \tag{3.19}$$

This addition is done in a pipelined manner. Each group of q threads $\phi^{\ell(i,j,k)}$ with i and k fixed and with $1 \leq j \leq q$ computes the corresponding C_{ik}.

An example with $P = 2^6$ and $n = 12$ is shown in Figure 5. Figure 6 shows the q threads of control used to compute C_{11} for this example.

For this algorithm the total number of events is

$$E = 3P + \frac{P}{q}(2^q - 1) = 3P + P^{\frac{2}{3}}(2^{P^{\frac{1}{3}}} - 1). \tag{3.20}$$

Note the number of events per thread of control ranges from

$$\begin{array}{ll} \kappa^\ell = \mathcal{O}(\log_2 q) & \text{when} \quad \ell = \ell(i, l, k) \quad \text{to} \\ \kappa^\ell = \mathcal{O}(1) & \text{when} \quad \ell = \ell(i, q, k). \end{array} \tag{3.21}$$

The expected active period between two consecutive events is the same for all threads and it is equal to the workload required to multiply (add) two submatrices of size $m \times m$

$$\bar{\alpha}^\ell = \mathcal{O}(m^3). \tag{3.22}$$

The expected duration of an event varies from

$$\begin{array}{ll} \beta^\ell = Q(m^2) & \text{when} \quad \ell = \ell(i, l, k) \quad \text{to} \\ \beta^\ell = Q(m^2 q) & \text{when} \quad \ell = \ell(i, q, k). \end{array} \tag{3.23}$$

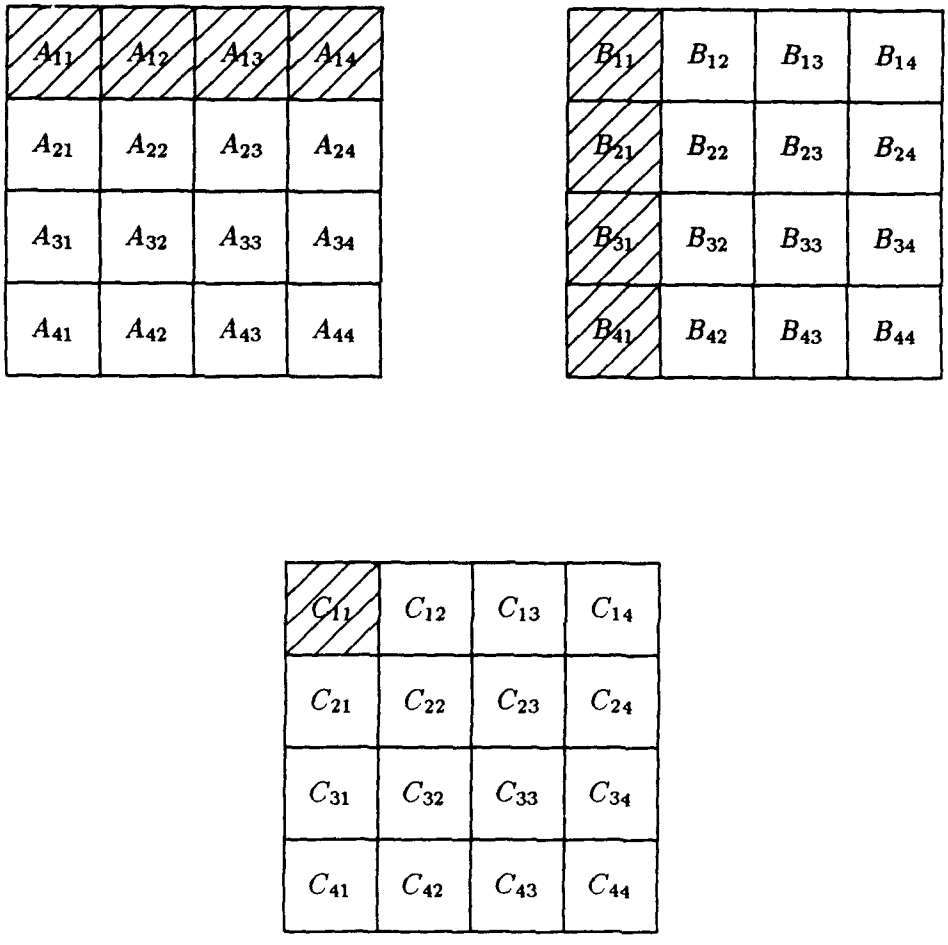


FIGURE 5: A parallel algorithm for matrix multiplication, with P threads of control. ($P = 64$, $n = 12$, $q = P^{1/3} = 4$, $m = \frac{n}{p} = 3$).

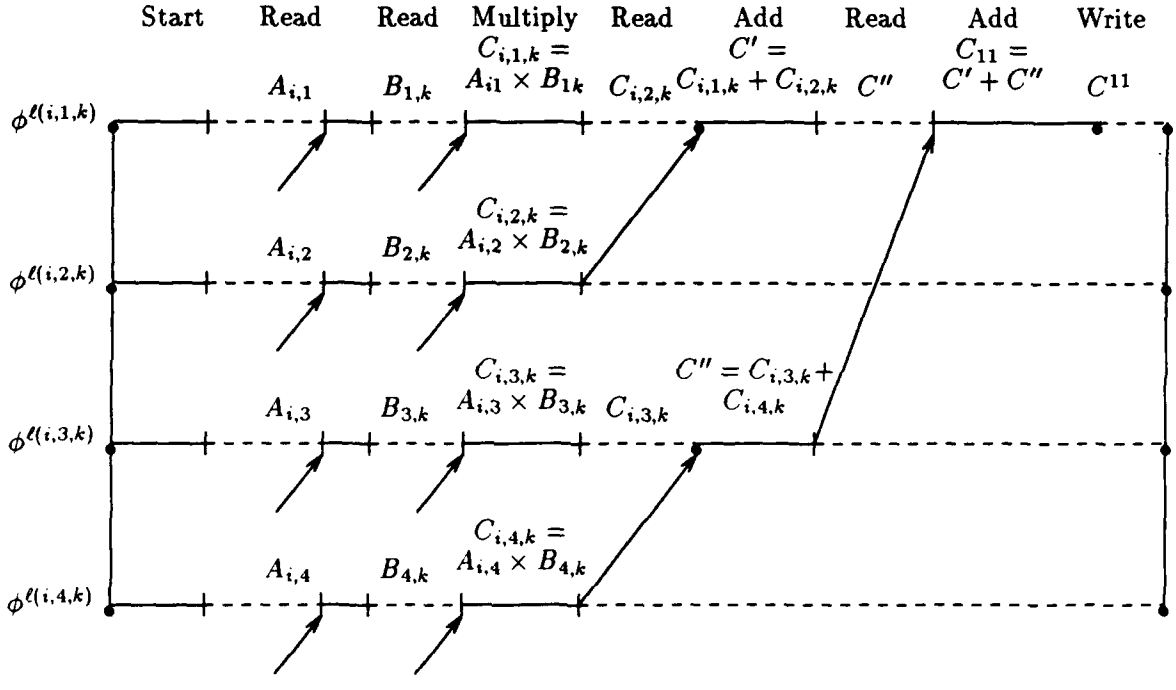


FIGURE 6: The pipelined algorithm for matrix multiplication. The time series of events are shown for $q = 4$ threads of control which compute C_{11}.

3.2.2 A parallel algorithm for banded matrix LU decomposition

The next example is from a concurrent algorithm for banded matrix LU decomposition. A detailed discussion of this algorithm can be found in Chapter 20 of [4] where its implementation on a hypercube is analyzed. The solution of many PDE's can be formulated as the solution of the linear systems of equations

$$AX = B \tag{3.24}$$

where A is a banded $M \times M$ matrix with bandwidth b and where X and B are $M \times n$ matrices of n solutions and free terms, respectively. Only the case when

$$1 << b << M \tag{3.25}$$

is considered here. To solve (3.24) a window of size $m \times m$ ($m = 2b + 1$) is defined and an iterative algorithm is used as follows. At iteration k the window covers the submatrix consisting of rows k to $k + m - 1$ and column k to $k + m - 1$. The process of solving (3.24) consists of three steps.

1. An LU decomposition of A is performed. During the k-th iteration, the following transformations are made:

   Set

   $$L_{k,k} = 1,$$

   and

   $$L_{k+1,k} = \frac{A_{k+i,k}}{A_{k,k}}, \quad 1 \leq i \leq m\ . \tag{3.26}$$

   $$U_{k,k} = \frac{1}{A_{k,k}},$$

   $$U_{k,k+j} = A_{k,k+j}, \quad 1 \leq j \leq m$$

   and

   $$U_{k+i,k+j} = A_{k+i,k+j} - L_{k+i,k} \cdot U_{k,k+j}, \quad 1 \leq i \leq m-1, \quad 1 \leq j \leq m-1.$$

2. A forward reduction of B is performed by

   $$B_{k+i,j} = B_{k+i,j} - B_{k,j} L_{k+i,k} \text{ for } 1 \leq i \leq m,\ 0 \leq j \leq n-1. \tag{3.27}$$

3. A back-substitution from last to first now generates the solution by

   $$X_{k,j} = \frac{B_{k,j}}{U_{k,k}}, \quad 0 \leq j \leq b-1,$$
   $$B_{k-i,j} = B_{k-i,j} - X_{k,j} U_{k-i,j}. \tag{3.28}$$

In this presentation, only the LU decomposition (Step 1) is analyzed. The LU decomposition takes a square matrix A and performs an in-place transformation. The L matrix overlaps the lower triangular part and the U matrix overlaps the upper triangular part of A, including the diagonal. The $m \times m$ window slides along the diagonal such that at step k the k-th row forms the upper side of the window and the k-th column is the left hand side of the window. To simplify the presentation of the algorithm, assume that the current iteration k satisfies the condition

$$k \bmod m = 0 \tag{3.29}$$

and that the matrix element $A_{k+i,k+j}$ is assigned to the thread $\phi^{i_1 q + j_1}$ with

$$i_1 = i \bmod q,\ j_1 = j \bmod q, \tag{3.30}$$

where $q = \sqrt{P}$. This assignment of the matrix elements corresponds to a scatter decomposition as described in [4]. Our analysis corresponds to the case when the size of the window m is a multiple of q and when no pivoting is necessary. In this case $n = m^2/q^2 = m^2/P$ elements of A are handled by every thread of control ϕ^i.

The k-th iteration proceeds as follows:

Step 1. The main diagonal element is updated by ϕ^0 as

$$U_{k,k} = \frac{1}{A_{k,k}}. \tag{3.31}$$

Step 2. *The k-th row of A is updated by*

$$U_{k,k+i} = A_{k,k+i},\ 1 \le i \le m-1. \tag{3.32}$$

This requires no computation and no communication.

Step 3. The k-th row is transmitted. Each ϕ^i with $0 \le i \le q-1$ multicasts $\sqrt{n}$ elements to $(q-1)$ related threads ϕ^{i+jq} with $1 \le j \le q-1$. This stage requires $q(q-1)$ events and no computation.

Step 4. The k-th column of A is updated. The threads ϕ^{iq} with $1 \le i \le q-1$ compute

$$L_{k+j,k} = A_{k+j,k} \cdot U_{k,k} \text{ with } 1 \le j \le m-1. \tag{3.33}$$

Here $(m-2)$ computations are performed. The total amount of work per iteration is $\alpha = 1 + (m-1) + 2(m-1)^2 = 2m^2 - 3m + 3$. The total number of events per iteration is

$$\kappa = 2q(q-1) + q^2 = 3q^2 - 2q = 3P - 2\sqrt{P}. \tag{3.34}$$

The expected amount of work per event is

$$\bar{\alpha}_e = \frac{2m^2 - 3m + 3}{3P - \sqrt{P}}. \tag{3.35}$$

The expected amount of data transferred per event is $\bar{\beta} = \sqrt{\frac{m}{P}}$. Since M iterations are needed and P events are involved in the start-up process, we have

$$E = M\left(3P - \sqrt{P}\right) + P = P(3M+1) - \sqrt{P}M. \tag{3.36}$$

3.3 The dynamic analysis – Schedules in the E/T model

During the previous stage of analysis only static estimates of the performance of parallel computation $\mathcal{C}$ can be obtained since so far the model does not include the concept of time. To extend the model we have to consider a parallel hardware $\mathcal{H}$ with n processors, π_i, $1 \leq i \leq n$ such that $n \geq P$. A *schedule* in the sense of the E/T model is a mapping of every thread of control $\phi^{(i)}$ to a processor π_i.

If detailed information concerning the architecture $\mathcal{H}$ is available then a schedule can be constructed and accurate performance data can be obtained. Following [11] a schedule means to decide for every node of the DAG associated with the computation, $\mathcal{C}$ the processor and the time when the node could execute. In the context of the E/T model, creating a schedule means to determine when every event will occur and how long it will take. The information about the processing speed will allow us to map the amount of work between two consecutive events in a thread of control into the corresponding execution time. The information about communication speed will allow us to map the volume of data to be transferred into a communication time. In this case the time when every event e_j^i occurs is well determined.

Note that at the time a schedule in the E/T sense is constructed, in addition to the *algorithmic events*, the events required by the mapping process, new events may need to be considered. Among the classes of new events we recognize

(a) Events related to the monitoring of $\mathcal{C}$. The next section will discuss in detail this class of events.

(b) Events related to different functions of the operating system. Is is conceivable that in the future more sophisticated operating systems for parallel machines will allow multiprocessing. In this case events related to processor scheduling, memory allocation, etc., will affect the performance of any computation $\mathcal{C}$.

Consider a thread of control ϕ^i assigned to processor π^i and define the following quantities related to ϕ^i, $1 \leq i \leq P$, and to $\mathcal{C}$

α_j^i – The interval of time ϕ^i is active, following the jth event e_j^i in thread ϕ^i.

$\bar{\alpha}^i$ – The expected active time of ϕ^i between two consecutive events.

$\bar{\alpha}$ – The expected active time of $\mathcal{C}$ between two consecutive events.

β_j^i – The duration of the jth event e_j^i in thread ϕ^i.

$\bar{\beta}^i$ – The expected duration of an event in ϕ^i.

$\bar{\beta}$ – The expected duration of an event in $\mathcal{C}$.

γ^i - The expected ratio, $\frac{\bar{\beta}^i}{\bar{\alpha}^i}$, of suspended time to active time in thread ϕ^i.

γ - The expected ratio, $\frac{\bar{\beta}}{\bar{\alpha}}$, of suspended time to active time in $\mathcal{C}$.

If $\bar{\kappa}$ is the expected number of events per thread of control then the average fraction $\eta(P)$ of the lifetime of a thread of $\mathcal{C}$ devoted to computation is given by

$$\eta(P) = \frac{(\bar{\kappa}-1)\bar{\alpha}}{(\bar{\kappa}-1)\bar{\alpha}+\bar{\kappa}\bar{\beta}} \approx \frac{1}{1+\gamma(P)} . \tag{3.37}$$

As expected $\eta(P) \to 1$ when $\gamma(P) \to 0$, i.e., when $\bar{\beta} << \bar{\alpha}$, or suspended time is much less than active time. In the case of a one-to-one mapping from threads of control to processors η represents the average processor utilization for $\mathcal{C}$.

The speedup is given by (2.25). We have $T(1) \leq P(\bar{\kappa}-1)\bar{\alpha}$ with equality when $W_{alg}(P) = 0$. Clearly we have

$$T(P) = (\bar{\kappa}-1)\bar{\alpha} + \bar{\kappa}\bar{\beta} \approx \bar{\kappa}(\bar{\alpha}+\bar{\beta}). \tag{3.38}$$

and hence we have the bound

$$S(P) = T(1)/T(P) \leq P\eta(P). \tag{3.39}$$

The last inequality shows that a low processor utilization leads to low speedup, as expected.

3.4 Monitoring parallel computations

A model of a physical system (process or phenomena) is an abstraction which distinguishes between essential and non-essential aspects of the system being modeled and attempts to predict the behavior of the system using a small subset of essential parameters. To validate a model means to compare predictions of the system performance obtained through the analysis of the model for a particular set of input parameters, with actual data gathered from observations of the real system.

In this section some of the issues pertinent to the validation of the E/T model are discussed and it is argued that the parameters necessary for the validation of the model of a parallel computation $\mathcal{C}$ can be obtained easily through monitoring. A detailed discussion of monitoring as well as a formal model for the monitoring process and an architectural model for software and hardware tools for monitoring parallel and distributed software is presented in [8]. Here we discuss only the relationship between monitoring and validation of the E/T model.

The E/T model characterizes a parallel computation $\mathcal{C}$ at two levels, at the *thread of control level* when information about one thread of control is necessary for the model and at the *global level*, when the entire collection of P threads of control are taken into account. At the first level, the total number κ^i of events in thread ϕ^i, as well as the mean time $\bar{\alpha}^i$ between two consecutive events, and the mean duration $\bar{\beta}^i$ of an event, are the parameters of the model. How κ^i, $\bar{\alpha}^i$ and $\bar{\beta}^i$ can be obtained directly by monitoring the execution of ϕ^i is shown later.

At the global level data concerning all threads of control must be gathered. While the global level characterization of $\mathcal{C}$ is not qualitatively different at the thread of control level, the validation of the model through monitoring does require a much larger volume of data to be collected and analyzed. Hence this type of characterization becomes more difficult to validate for massively parallel computations when a large number of threads of control must be monitored and analyzed.

Note also that in case of SPMD parallel computations, all threads of control exhibit quasi-identical behavior and monitoring a sampling of threads (or even one) can provide enough information to estimate accurately the global level characterization of C. For other types of parallel computations (non-SPMD) it is rather difficult to extrapolate the knowledge acquired through monitoring one thread of control to the global characterization of C.

Monitoring a parallel computation C is the process of recording the events of interest which occur during the lifetime of processes running different threads of control of C. A *monitoring event* is defined [8] as a change of state of a process. An event is "of interest" depending upon the goals of the monitoring process. Monitoring the execution of a parallel computation C is necessary for debugging a particular implementation of C and for performance evaluation. During monitoring each *event of interest* generates a trace record which contains all relevant information concerning the thread, e.g. the type of event, the time of the event, the state of the process, etc.

The definition of a monitoring event is more general than the one discussed so far in the context of the E/T model. For example "$I = 5$" can be defined as an event of interest for monitoring, it is indeed a change of state of the corresponding process since a new value is assigned to the variable I, but it is not an event in the sense of the E/T model since the corresponding thread of control does not change its state as a result of this assignment. However, as soon as a change of state of a process is designated as an event of interest and it is decided to monitor it, then this monitoring event becomes also an event in the sense of the E/T model since monitoring means an interruption of the original flow of control done to record the pertinent trace data. On the other hand, any event in the context of the E/T model corresponds to a change of state of the process which embeds the corresponding thread of control, hence it can be monitored.

A first important conclusion of this discussion is that there is a one to one mapping between monitoring events and the events defined in the context of the E/T model. In other words the E/T model is "observable" through monitoring, all the parameters required by the E/T model can be obtained as part of the trace data gathered through monitoring. A second conclusion is that monitoring a parallel computation may affect the timing of events as well as the total number of events in the original computation. This is an undesirable effect associated with every process of measurement.

3.5 Monitoring the performance of iterative methods on a distributed memory system

An experiment to study the performance of iterative methods on a distributed memory system is described in detail in [9]. The experiment uses the parallel ELLPACK, PELLPACK system developed at Purdue [6], running on a 128 processor NCUBE. The TRIPLEX tool set [7] is used to monitor the execution and to collect trace data.

The purpose of the experiment was to collect detailed information concerning the execution of a particular SPMD application, to study how this data relates to the high level characterization of parallelism in the framework of the E/T model, and to investigate how similar or dissimilar the behavior of the threads of control of an SPMD computation are.

The experiment monitors the execution of the code implementing a Jacobi iterative algorithm for solving a linear system of equations, an important component of a parallel PDE solver. To ensure a load balanced execution, the domain decomposer, part of the PELLPACK environment, attempts to assign to every PE an equal amount of computation. A careful selection of the interface points of the neighboring domains is also necessary in order to achieve a balanced communication. The experiment was conducted by taking a problem of a fixed size and repeating the execution with a number of PEs ranging from 2 to 128.

The detailed behavior of all threads of control was captured by recording all the events, marking changes of state for every thread. For every event the TRIPLEX tool creates a trace record, which contains the pertinent information about the event, type, time stamp, PE, amount of data transferred, etc. All the measurements reported are based upon a clock with resolution of 0.1 msec. To minimize the volume of trace data, only events related to communication and control were recorded. Even so, the trace data collected during a single experiment with 128 PEs amounted to about 25 Mbytes.

The raw data were processed in several stages. First, the events outside the scope of Jacobi iterations were filtered out. Then a preprocessing to gather the data required by the E/T model was performed. The active time between events, the duration of an event (read/write) and the length of a blocking period, were obtained by correlating local events, events occurring in the same thread (on the same PE). The time for communication and control was computed as the difference between the duration of an event and the length of its blocking period. To compute the algorithmic blocking (defined as the interval from the instance a read is issued until the corresponding write takes place) it was necessary to correlate non-local events, events involving more than one thread. Finally, a statistical processing was performed in order to obtain data as described in Section 3.3.

Preliminary results indicate that in spite of all the precautions to achieve a well balanced communication, the behavior of threads of control can be considerably different. The number of events, the active time, the total read time per thread, may be within a factor of two from one thread to another as shown in Figure 7 for the expected active time. Figure 8 represents the characteristic function $g(P)$ of the parallel computation, which indicate a $\mathcal{O}(P^2)$ behavior. This happens, since at the end of every iteration a global communication implemented as broadcast-collapse takes place in order to communicate values between threads of control. There is also a global exchange of information every few iterations to obtain information for convergence control. While this could be done by fan-in, fan-out communication in principle, the NCUBE system forces the use of broadcasting which is another source of $\mathcal{O}(P^2)$ events. One familiar with Jacobi iteration would not expect $g(P) = \mathcal{O}(P^2)$. This behavior arises primarily because the NCUBE system does not provide adequate communication utilities, one must use a broadcast when one actually wants to do a multicast to just a handful of "nearby" processors. Using the system provided "supposed" multicast facility actually increases the communication time, because it is implemented using a broadcast. The allocation of threads of control to actual processors can also affect $g(P)$, while the optimal allocation for this problem is NP-hard in general, there are heuristic algorithms which keep the distances between actual processors to a reasonable level. The expected active time per event decreases linearly (Figure 9), while the write time is essentially constant (Figure 10).

The read time per event experiences a sharp increase (Figure 11) and most of it is due to the algorithmic blocking (Figure 12). The active time fraction of the total non-blocked time decreases, due to the $\mathcal{O}(P^2)$ of the number of events per thread (Figure 13).

These results, though preliminary, seem to indicate that the point of view taken by the E/T model, namely, that the work for communication and control is essential in understanding the behavior of parallel computation is well motivated. The measured speedup of a parallel computation C may be disappointingly low, even when a high PE utilization is observed, simply due to the overhead associated with communication and control. This overhead is difficult to measure, but it can be estimated when $g(P)$ is known.

Further experiments are necessary to establish a sound relation between the algorithmic blocking and $g(P)$. The present data show a strong interdependence between the two.

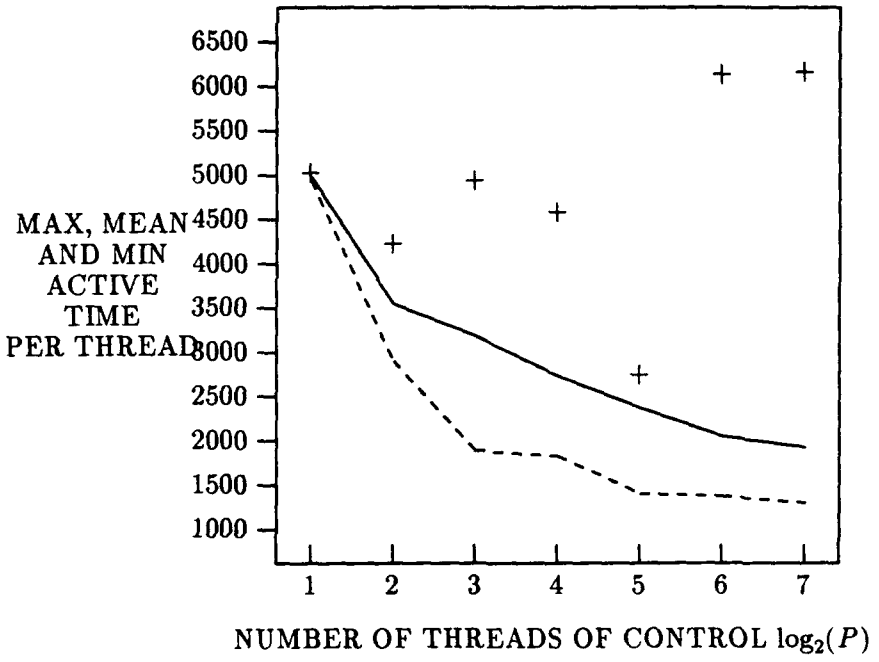


FIGURE 7: The minimum (dashed) the average (solid) and the maximum (plus) *active* time for a thread of control. Irregular domain, 33×33 grid.

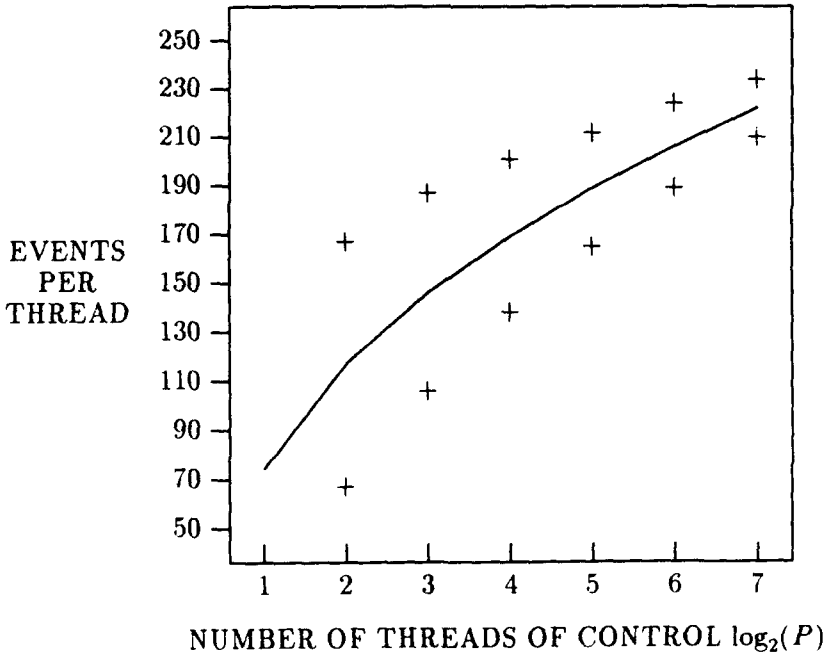


FIGURE 8: The expected number of events per thread of control (solid line) and a 95 percent confidence interval for it.

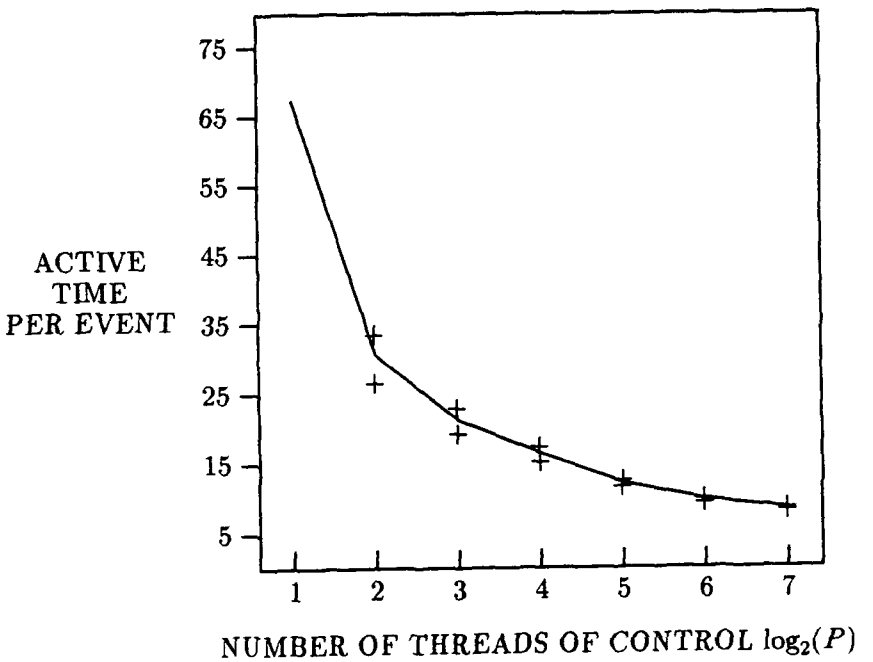


FIGURE 9: The expected time of an *active* period, between two consecutive events (solid line) and a 95 percent confidence interval for it.

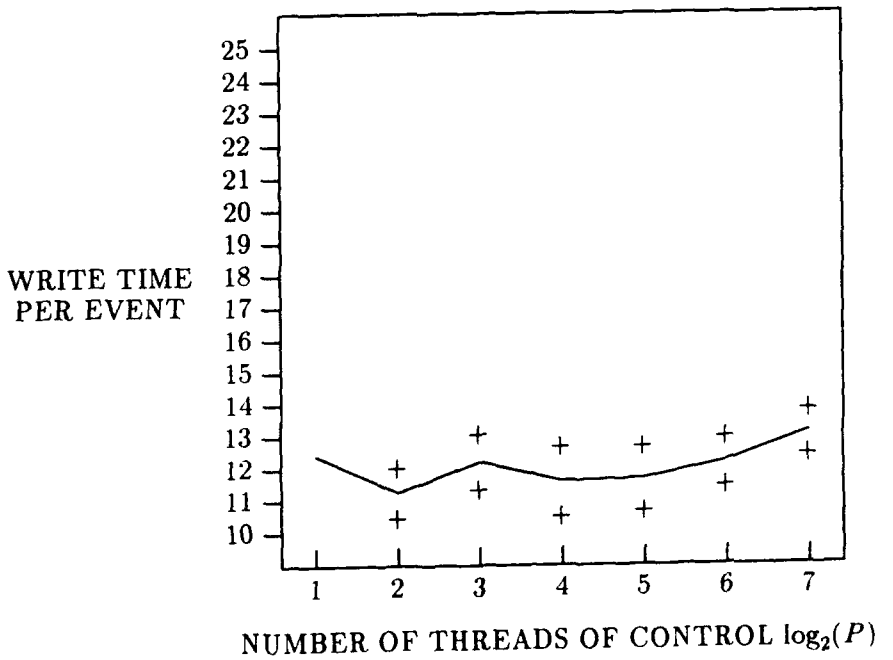


FIGURE 10: The expected time for a single *write* operation (solid line) and a 95 percent confidence interval for it.

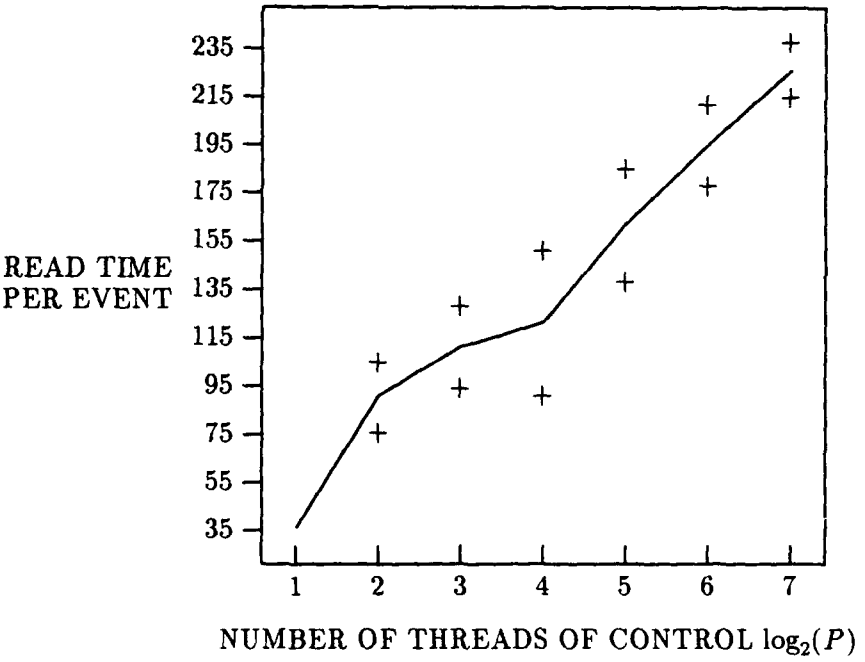


FIGURE 11: The expected time for a single *read* operation (solid line) and a 95 percent confidence interval for it.

4 Conclusions

It is extremely difficult to provide concise characterization of parallel computations, invariant to problem size and especially to the architecture of parallel systems. The E/T model of parallel execution is best suited to the important class of SPMD applications and provides little or no insight for dynamic computations where data dependencies affect the sequence and possibly the number of events in a thread of control.

The characteristic function $E = g(P)$ defined by the E/T model, is less sensitive to the architecture of the parallel system and the problem size than other types of high level characterizations of parallel computations. It allows a uniform treatment for both message passing and shared memory paradigms. The average workload per thread of control, $\bar{w}(P)$ provides a signature of a parallel computation and allows interesting conclusions concerning the asymptotic behavior of different classes of parallel computations.

The E/T model allows quantitative characterizations of the model as well. The static characterization is based upon the computation and data mapping and it is insensitive to timing characteristics (instruction execution rate, communication speed) of the parallel machine, but it does not capture the effects of blocking and synchronization.

A final strength of the E/T model is its closeness to the experiment. Monitoring tools typically provide precisely the data required by the model. The measurements reported in the previous section capture the detailed behavior of all threads of control of a parallel computation in a PDE solver. A preliminary analysis of the measurements shows the effects of the additional work for communication and control and of blocking. For a problem of a given size, the fraction of the time a PE is active, working on W_{cons} to its total running time, including work for communication and

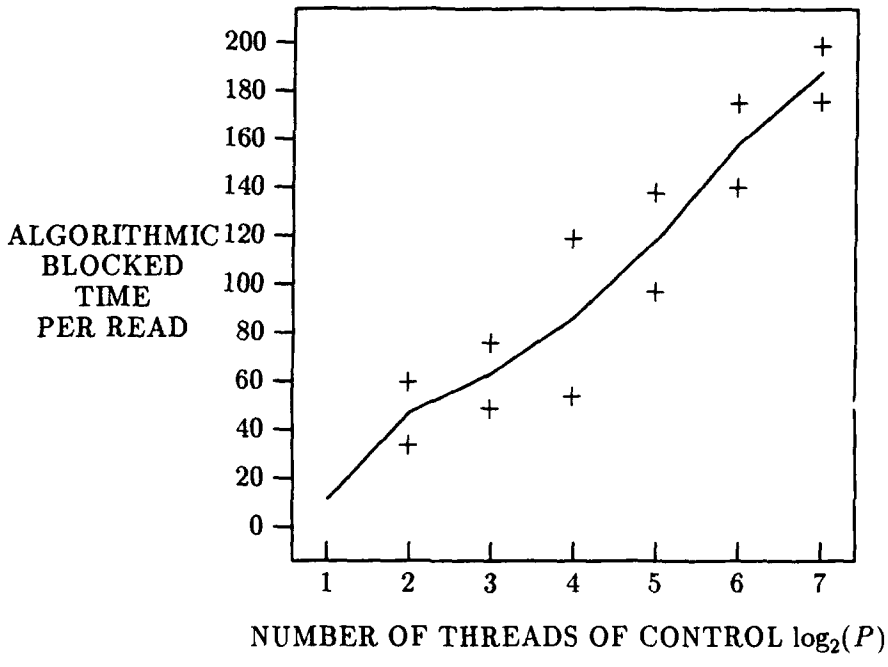


FIGURE 12: The expected *algorithmic blocking time* during a read operation (solid line) and a 95 percent confidence interval for it. The algorithmic blocking is defined as the interval from the instance a *read* is issued until the corresponding *write* is initiated.

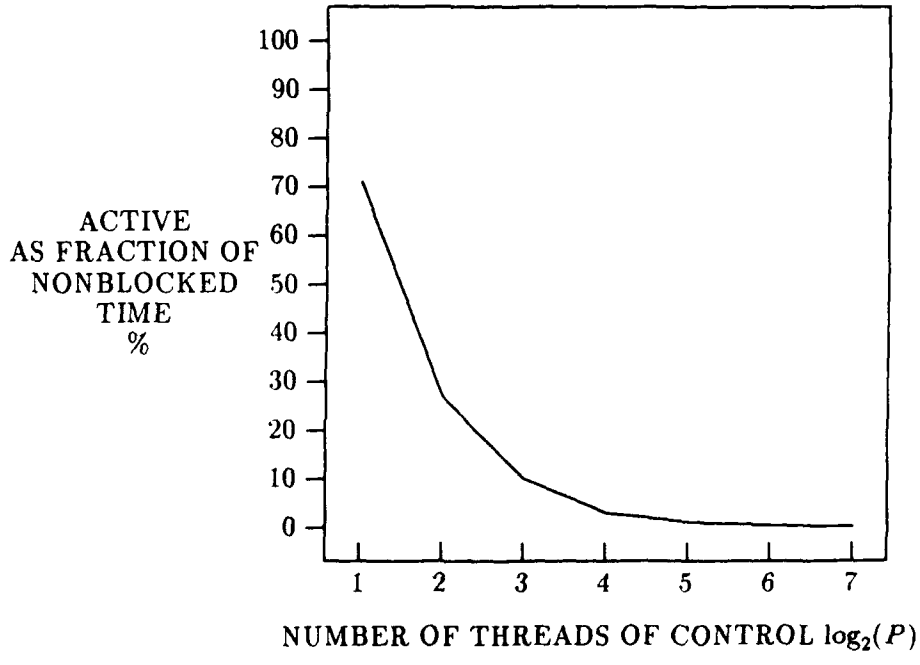


FIGURE 13: The expected *active time* fraction of the non-blocked time per thread.

control, decreases when the number of PEs increases under these conditions. The actual shape of both functions seems closely related to $g(P)$. Further investigations of other parallel computations and comparisons of results obtained from running the same computation on distributed memory and shared memory systems, are necessary to gain insight into the relationship between the characteristic function $g(P)$ and other measures of performance.

Acknowledgements

The authors express their thanks to Mo Mu for his helpful comments.

5 Literature

[1] A. Aggarwal, A.K. Chandra and M. Snir, "Communication complexity of PRAMs", Research Report Rc 14998, IBM Research, February 1989.

[2] A. Aggarwal, A.K. Chandra and M. Snir, "On communication latency in PRAM computations", Research Report RC 14973, IBM Research, September 1989.

[3] D.L. Eager, J. Zahorjan and E.E. Lazowska, "Speedup versus efficiency in parallel systems", IEEE Trans. on Computer Systems, vol. 38, no. 3, pp. 408-423, March 1989.

[4] G. Fox et al., "Solving problems on concurrent processors", Prentice Hall, 1988.

[5] J.L. Gustafson, "Reevaluating Amdahl's law", CACM 31, 5, pp. 532-533, May 1988.

[6] E.N. Houstis and J.R. Rice, "Parallel ELLPACK: An expert system for parallel processing of partial differential equations", Math. Comp. Simulation, vol. 31, pp. 497-507, 1989.

[7] D.W. Krumme, A.L. Couch and B.L. House, "The TRIPLEX tool set for the NCUBE multiprocessors", Technical Report Tufts University, June 1989.

[8] D.C. Marinescu, J. E. Lumpp, T.L. Casavant and H.J. Siegel "Models for monitoring and debugging tools for parallel and distributed software", Journal of Parallel and Distributed Computing, June 1990, (to appear).

[9] D.C. Marinescu, J.R. Rice and E. Vavalis, "Performance of iteration methods for distributed memory processors", CSD-TR 979, Computer Sciences Department, Purdue University, May 1990.

[10] Mo Mu and J.R. Rice, "The structure of parallel sparse matrix algorithms for solving partial differential equations on hypercubes", CSD-TR 976, Computer Sciences Department, Purdue University, May, 1990.

[11] C.H. Papadimitriou and M. Yannakakis, "Towards an architecture-independent analysis of parallel algorithms", Proc. of 20th Annual ACM Symp. on Theory of Computing, pp. 510-513, May 1988.

[12] J.R. Rice and D.C. Marinescu, "Analysis of a two level asynchronous algorithm for PDEs", in *Aspects of Computations on Asynchronous Parallel Processors*, (M. Wright ed), North Holland, pp. 23-33, 1989.

[13] J.R. Rice and D.C. Marinescu, "Multi-Level asynchronous PDEs", in *Iterative Methods*, (D.R. Kincaid and L.J. Hayes eds), Academic Press, pp 193-212, 1990.

[14] K.C. Sevcik, "Characterizations of parallelism in applications and their use in scheduling", Proc. of Sigmetrics and Performance'89, Berkeley, PA, May 1989.

Recursive LLSP in Distributed-Memory Multiprocessors

Jaeyoung Choi, Adam W. Bojanczyk*

School of Electrical Engineering

Cornell University, Ithaca, N.Y. 14853-5401

Abstract

We discuss implementations of block algorithms for recursive linear least squares problems (LLSP) on a linear array of p processors. A recursive least squares problem is a composite task that involves triangularization of the data matrix and solution of the resulting triangular systems of linear equations. These two problems when mapped on to a linear array of processors have different communication requirements that may lead to a high communication overhead for this composite problem. In transforming the data matrix to a triangular form, we consider the sliding rectangular window approach to eliminate the influence of old data on the current solution vector. This approach involves both updating and downdating of the data matrix. The back-substitution process is normally used for solving the triangular system. However, as the back-substitution process is communication bound, we also consider updating and downdating of the inverse of the triangular factor directly and hence replace the back-substitution process by direct matrix-vector multiplication. We discuss block and wrap mappings of the data matrix onto a linear array of processors. The effects of each mapping on the overall execution time are also discussed. We propose ways to decrease the execution time by modifying the algorithms for this problem. We present results of numerical tests on a linear array of transputers.

* This work was supported by the ARO grant number DAA LO3-90-G-0092

1. Introduction

Let $\mathbf{A}$ be a full rank $m \times n$ matrix, and let $\mathbf{b}$ be a real m dimensional vector. In the linear least squares problem (LLSP for short), we want to find a real n-dimensional vector $\mathbf{x}$ such that

$$\|\mathbf{A} \cdot \mathbf{x} - \mathbf{b}\|_2 = \min_{\mathbf{y} \in \mathbb{R}^n} \|\mathbf{A} \cdot \mathbf{y} - \mathbf{b}\|_2. \tag{1.1}$$

The first step in solving LLSP is to reduce $\mathbf{A}$ to the upper triangular form $\mathbf{R}$ by applying an orthogonal transformation $\mathbf{Q}$ to the matrix $\mathbf{A}$. Then the problem (1.1) is equivalent to the problem of finding $\mathbf{x}$ such that

$$\|\mathbf{R} \cdot \mathbf{x} - \mathbf{c}\|_2 = \min_{\mathbf{y} \in \mathbb{R}^n} \|\mathbf{R} \cdot \mathbf{y} - \mathbf{c}\|_2 \tag{1.2}$$

where $\mathbf{c} = \mathbf{Q}^T \cdot \mathbf{b}$. Now (1.2) can be solved by the back-substitution process.

Suppose now that the new data matrix $(\mathbf{X} \quad \mathbf{b}_X)$ is to be added to the matrix $(\mathbf{A} \quad \mathbf{b})$. Let $(\mathbf{A}_{new} \quad \mathbf{b}_{new}) = \begin{pmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{X} & \mathbf{b}_X \end{pmatrix}$. The new problem to be solved is the following:

$$\|\mathbf{A}_{new} \cdot \mathbf{x}_{new} - \mathbf{b}_{new}\|_2 = \min_{\mathbf{y} \in \mathbb{R}^n} \|\mathbf{A}_{new} \cdot \mathbf{y} - \mathbf{b}_{new}\|_2. \tag{1.3}$$

It is easy to see that $\mathbf{x}_{new}$ satisfies the following equivalent relation:

$$\|\mathbf{A}_{new} \cdot \mathbf{x}_{new} - \mathbf{b}_{new}\|_2 = \min_{\mathbf{y} \in \mathbb{R}^n} \left\| \begin{pmatrix} \mathbf{R} \\ \mathbf{X} \end{pmatrix} \cdot \mathbf{y} - \begin{pmatrix} \mathbf{c} \\ \mathbf{b}_X \end{pmatrix} \right\|_2 \tag{1.4}$$

Thus, while solving the new problem (1.4) we can exploit the fact that we already know $\mathbf{R}$. What remains to be done is to transform $\begin{pmatrix} \mathbf{R} \\ \mathbf{X} \end{pmatrix}$ to upper triangular form. This is known as the updating problem.

Similarly, suppose that a subset of rows in the data matrix is no longer representative to the behavior of the system and should be removed from the data set. Formally, let $(\mathbf{A} \quad \mathbf{b}) = \begin{pmatrix} \mathbf{A}_{current} & \mathbf{b}_{current} \\ \mathbf{Y} & \mathbf{b}_Y \end{pmatrix}$. Then we would like to solve the problem (1.3) where $\mathbf{A}_{new}$ and $\mathbf{b}_{new}$ are replaced by $\mathbf{A}_{current}$ and $\mathbf{b}_{current}$ respectively. Again, we would like to exploit the fact that we already know $\mathbf{R}$. We are now looking for an orthogonal transformation $\mathbf{P}$ such th

$$\mathbf{P}^T \cdot \begin{pmatrix} \tilde{\mathbf{R}} & \tilde{\mathbf{b}} \\ \mathbf{Y} & \mathbf{b}_Y \end{pmatrix} = \begin{pmatrix} \mathbf{R} \\ \mathbf{c} \end{pmatrix} \tag{1.5}$$

where $\tilde{\mathbf{R}}$ is upper triangular. This is known as downdating.

In applications, we often want to remove an old data set, add a new data set and possibly repeat this process over and over again. In signal processing, this combined process of updating and downdating is referred to as a rectangular sliding window process. In the rectangular sliding window approach at each recursive step, we want to find an orthogonal matrix $\mathbf{W}$ such that

$$\mathbf{W}^T \cdot \begin{pmatrix} \mathbf{R} & \mathbf{c} \\ \mathbf{X} & \mathbf{b}_X \\ \mathbf{0} & \mathbf{0} \end{pmatrix} = \begin{pmatrix} \tilde{\mathbf{R}} & \tilde{\mathbf{b}} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{Y} & \mathbf{b}_Y \end{pmatrix} \tag{1.6}$$

where $\tilde{\mathbf{R}}$ is upper triangular. In (1.6), $\mathbf{R}$ represents the old window and $\tilde{\mathbf{R}}$ represents the new window in the current step.

Once the new factor $\tilde{\mathbf{R}}$ is known, the corresponding triangular system of linear equations is solved. The back-substitution process is normally used for finding the solution of this system. On parallel architectures, back-substitution is communication bound. Thus, on system where interprocessor communication is expensive relative to computation, back-substitution may increase the overall execution time. One way around this problem is to work directly with the inverse of the triangular factor $\mathbf{R}$. Then back-substitution process can be replaced by a matrix-vector multiplication that is usually very efficient on any parallel architecture. It is known that the transformations which update and downdate the triangular factor $\mathbf{R}$ are closely related to those that update and downdate the inverse $\mathbf{R}^{-1}$. We will consider two approaches to update and downdate $\mathbf{R}^{-1}$. In the first approach, the transformations to update and downdate $\mathbf{R}^{-1}$ are computed based on the factor $\mathbf{R}$, the data to be added $\mathbf{X}$, and the data to be deleted $\mathbf{Y}$. In the second approach, the transformations are computed based on $\mathbf{R}^{-1}$, $\mathbf{X}$ and $\mathbf{Y}$. In this sliding approach, we will compare these two approaches with respect to the execution time on a linear array of processors. The important issue of numerical behavior will be considered separately in a companion study.

We consider two popular mappings, block mapping and wrap mapping. In block mapping the matrix is partitioned into blocks of column with each block assigned to exactly one processor, and consecutive blocks mapped onto consecutive processors in the array. In wrap mapping, consecutive columns are mapped to consecutive processors modulo the number of processors. In general, block mappings lead to smaller communication requirements and wrap mappings exhibit a better load balance. We compare various strategies for solving recursive LLSP on a linear array of 8 transputers.

2. Up-Downdatings of Order k

One of the two major steps in solving recursive linear least squares problems is updating and downdating of the data matrix. We will consider this process where data to be added to the matrix consists of k-rows and the data to be deleted consists of another k-rows ($k \geq 1$). Formally, given a nonsingular $n \times n$ upper triangular matrix $\mathbf{R}$, a $k \times n$ matrix $\mathbf{X}$ and a $k \times n$ matrix $\mathbf{Y}$, where $\text{rank}(\mathbf{X}) = \text{rank}(\mathbf{Y}) = k$, we want to find an orthogonal matrix $\mathbf{W}$ and an $n \times n$ upper triangular matrix $\tilde{\mathbf{R}}$ such that

$$\mathbf{W}^T \cdot \begin{pmatrix} \mathbf{R} \\ \mathbf{X} \\ \mathbf{0} \end{pmatrix} = \begin{pmatrix} \tilde{\mathbf{R}} \\ \mathbf{0} \\ \mathbf{Y} \end{pmatrix} \tag{2.1}$$

Note that (2.1) implies that

$$\tilde{\mathbf{R}}^T\tilde{\mathbf{R}} = \mathbf{R}^T\mathbf{R} + \mathbf{X}^T\mathbf{X} - \mathbf{Y}^T\mathbf{Y} \tag{2.2}$$

Thus as long as the right hand side of (2.2) is positive definite, the problem (2.1) is well-defined. The problem (2.2) is that of finding the Cholesky factor after two rank-k modifications of $\mathbf{R}^T\mathbf{R}$ (one rank-k addition of $\mathbf{X}^T\mathbf{X}$ and one rank-k deletion of $\mathbf{Y}^T\mathbf{Y}$). We will call the joint process of rank-k updating and rank-k downdating as ***up-downdating of order k.***

We can treat the problem (2.2) as a sequence of $2k$ rank-1 modification problems, k rank-1 updating problems and k rank-1 downdating problems. Each rank-1 modification could be realized by a sequence of n two-dimensional rotations [6]. Such an approach would require $2n^2 + 3n + O(1)$ flops per rank-1 modification for a total of $2kn(2n + 3)$ flops. It is well-known that Householder transformations require fewer operations than the corresponding sequence of rotations. If we treat (2.2) as a two step process consisting of a rank-k updating followed by a rank-k downdating, then each step can be realized by a sequence of Householder transformations in $(k + 1)n(n + 1)$ flops for a total of $2(k + 1)n(n + 1)$ flops, a two-fold saving over the rotation based approach.

Recently hyperbolic Householder transformations have been proposed to realize the joint process (2.2) of updating and downdating. Let $\Phi = \mathrm{diag}(\mathrm{I}_n, \mathrm{I}_k, -\mathrm{I}_k)$, then a transformation $\mathbf{H}$ is called hyperbolic with respect to Φ if and only if

$$\Phi = \mathbf{H}^T \cdot \Phi \cdot \mathbf{H}. \tag{2.3}$$

Note that if $\mathbf{H}$ is hyperbolic with respect to Φ such that

$$\mathbf{H} \cdot \begin{pmatrix} \mathbf{R} \\ \mathbf{X} \\ \mathbf{Y} \end{pmatrix} = \begin{pmatrix} \tilde{\mathbf{R}} \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix} \tag{2.4}$$

where $\tilde{\mathbf{R}}$ is upper triangular, then (2.2) holds. We will consider three algorithms for recursive LLSP based on hyperbolic transformations.

3. Algorithms for Recursive LLSP via Hyperbolic Transformations

3.1 Algorithm A

The recursive algorithm for up-downdating of order k based on the hyperbolic Householder transformation is simple. The hyperbolic transformation $\mathbf{H}$ is constructed as a product of hyperbolic Householder transformations $\mathbf{H}_i\ (1 \le i \le n)$, where $\mathbf{H}_i$ operates on $\mathbf{X}$, $\mathbf{Y}$ and the i-th row of $\mathbf{R}$, i.e.,

$$\mathbf{H}_n\mathbf{H}_{n-1}\cdots\mathbf{H}_1 \cdot \begin{pmatrix} \mathbf{R} & \mathbf{d} \\ \mathbf{X} & \mathbf{b}_X \\ \mathbf{Y} & \mathbf{b}_Y \end{pmatrix} = \begin{pmatrix} \tilde{\mathbf{R}} & \tilde{\mathbf{d}} \\ \mathbf{0} & \tilde{\mathbf{b}}_X \\ \mathbf{0} & \tilde{\mathbf{b}}_Y \end{pmatrix} \tag{3.1.1}$$

and $\tilde{\mathbf{R}}$ is an $n \times n$ upper triangular matrix. The definition of a hyperbolic Householder transformation and the way it can be constructed and applied to a matrix are discussed in section 7.

Algorithm A requires $(2k + 1)n(n + 3)$ multiplications.

3.2 Algorithm B

Let us start with a procedure for the updating the inverse of a triangular matrix. The algorithm for rank-1 updating of the inverse of the triangular factor $\mathbf{R}$ [6] can be extended to rank-k updating of the inverse. Let $\mathbf{R}$ be an upper triangular $n \times n$ matrix, $\mathbf{X}$ be a $k \times n$ matrix, and $\mathbf{H} = \mathbf{H}_n\mathbf{H}_{n-1}\cdots\mathbf{H}_1$ be a

product of Householder transformations which transform the matrix $\begin{pmatrix} \mathbf{R} \\ \mathbf{X} \end{pmatrix}$ to upper triangular form. The transformation $\mathbf{H}$ when applied to $\begin{pmatrix} \mathbf{R}^{-T} \\ \mathbf{0} \end{pmatrix}$ gives $\tilde{\mathbf{R}}^{-T}$. To see this, let us consider the following identity:

$$\mathbf{I} = \underbrace{(\mathbf{R}^{-1} \quad \mathbf{O}^T) \cdot \mathbf{H}^T} \cdot \mathbf{H} \cdot \begin{pmatrix} \mathbf{R} \\ \mathbf{X} \end{pmatrix}$$

$$= (\hat{\mathbf{R}}^{-1} \quad \mathbf{Z}^T) \cdot \begin{pmatrix} \hat{\mathbf{R}} \\ \mathbf{O} \end{pmatrix}$$

Note that because of the form of $\mathbf{H}$, matrix $\mathbf{R}^{-1}$ is upper triangular, hence it must be the inverse of $\mathbf{R}$. It can be shown that $\mathbf{Z}$ is such that $\mathbf{Z}^T\mathbf{Z} = \mathbf{R}^{-1}\mathbf{V}^T \left(\mathbf{I} + \mathbf{V}\mathbf{V}^T\right)^{-1} \mathbf{V}\mathbf{R}^{-T}$ and $\mathbf{V} = \mathbf{X}\mathbf{R}^{-1}$.

The same idea can be extended to up-downdate $\mathbf{R}$ and $\mathbf{R}^{-T}$ simultaneously. Let $\mathbf{F} = \mathbf{F}_n\mathbf{F}_{n-1}\cdots\mathbf{F}_1$ be a product of hyperbolic matrices to up-downdate $\mathbf{R}$ with $\mathbf{X}$ and $\mathbf{Y}$, respectively. The same $\mathbf{F}$ can be applied to up-downdate $\mathbf{R}^{-T}$. Let

$$\mathbf{F} \cdot \begin{pmatrix} \mathbf{R} & \mathbf{d} \\ \mathbf{X} & \mathbf{b}_X \\ \mathbf{Y} & \mathbf{b}_Y \end{pmatrix} = \begin{pmatrix} \tilde{\mathbf{R}} & \tilde{\mathbf{d}} \\ \mathbf{0} & \tilde{\mathbf{b}}_X \\ \mathbf{0} & \tilde{\mathbf{b}}_Y \end{pmatrix}.$$

Then

$$\mathbf{I} = \underbrace{(\mathbf{R}^{-T} \quad \mathbf{O}^T \quad \mathbf{O}^T) \cdot \mathbf{F}^T} \cdot \mathbf{F} \cdot \begin{pmatrix} \tilde{\mathbf{R}} \\ \mathbf{X} \\ \mathbf{Y} \end{pmatrix}$$

$$= (\tilde{\mathbf{R}}^{-T} \quad \mathbf{Z}^T \quad \mathbf{W}^T) \cdot \begin{pmatrix} \tilde{\mathbf{R}} \\ \mathbf{O} \\ \mathbf{O} \end{pmatrix}$$

i.e.

$$\mathbf{F} \cdot \begin{pmatrix} \mathbf{R}^{-T} \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix} = \begin{pmatrix} \tilde{\mathbf{R}}^{-T} \\ \mathbf{Z} \\ \mathbf{W} \end{pmatrix}$$

where $\mathbf{W}^T\mathbf{W} = \mathbf{R}^{-1}\mathbf{N}^T \left(\mathbf{I} - \mathbf{N}\mathbf{N}^T\right)^{-1} \mathbf{N}\mathbf{R}^{-T}$ and $\mathbf{N} = \mathbf{Y}\mathbf{R}^{-1}$.

This algorithm requires $2(2k+1)n(n+2)$ multiplications. The number of multiplications is almost twice as large as that of Algorithm A, hence Algorithm B is not recommended for a sequential processor where the number of operations determines the cost. However as the inverse $\tilde{\mathbf{R}}^{-T}$ is known, back-substitution can now be replaced by matrix-vector multiplication. Matrix-vector multiplication, in general, requires less communication among processors than back-substitution, thus potentially leading to a faster execution time of the recursive sliding window approach.

3.3 Algorithm C

Note that Algorithm B requires storing both matrix $\mathbf{R}$ and its inverse $\mathbf{R}^{-1}$. It is possible to operate on the inverse $\mathbf{R}^{-1}$ without keeping the triangular matrix $\mathbf{R}$. Algorithm C is based on the algorithm for rank-1 downdating of the inverse in [6].

Let us rewrite the rank-k updating problem as:

$$\mathbf{R}^T\mathbf{R} + \mathbf{X}^T\mathbf{X} = \hat{\mathbf{R}}^T\hat{\mathbf{R}} \tag{3.3.1}$$

Assume that $\mathbf{X}$ is an $1 \times n$ data matrix, then (3.3.1) is a rank-1 updating problem. Let us find a sequence of plane rotations $\mathbf{G}_{n,n+1}, \mathbf{G}_{n-1,n+1}, \cdots, \mathbf{G}_{1,n+1}$, such that for $\mathbf{G} = \mathbf{G}_{n,n+1}\mathbf{G}_{n-1,n+1}\cdots\mathbf{G}_{1,n+1}$,

$$\mathbf{G} \cdot \begin{pmatrix} -\mathbf{R}^{-T}\mathbf{X}^T & \mathbf{R}^{-T} \\ 1 & \mathbf{0}_{1\times n} \end{pmatrix} = \begin{pmatrix} \mathbf{0}_{n\times 1} & \tilde{\mathbf{R}}^{-T} \\ u & \mathbf{Z} \end{pmatrix}. \tag{3.3.2}$$

$\mathbf{G}_{i,n+1}$ $(1 \le i \le n)$ is the Givens rotation operating in plane $(i, n+1)$ which forces the i-th component of the first column of the matrix on the left in (3.3.2) to zero. Then $\tilde{\mathbf{R}}^{-T}$ is the inverse of the desired factor in (3.3.1). It can be shown that $\mathbf{Z}^T\mathbf{Z} = \mathbf{R}^{-1}\mathbf{V}^T\left(\mathbf{I} + \mathbf{V}\mathbf{V}^T\right)^{-1}\mathbf{V}\mathbf{R}^{-T}$ and $\mathbf{V} = \mathbf{X}\mathbf{R}^{-1}$.

We can generalize the process (3.3.2) of rank-1 updating to rank-k updating by treating the rank-k updating as a sequence of k rank-1 updatings. Assume that $\mathbf{X}$ is a $k \times n$ data matrix, and consider the matrix

$$\mathbf{B} = \begin{pmatrix} -\mathbf{R}^{-T}\mathbf{X}^T & \mathbf{R}^{-T} \\ \mathbf{I}_k & \mathbf{0}_{k\times n} \end{pmatrix}.$$

Let $\mathbf{G}_j^u$ be the product of Givens rotations as defined in (3.3.2) operating on the top n rows and the $(n+j)$-th row of $\mathbf{B}$ such that the first n components of the j-th column of $\mathbf{B}$ are zero. Then for $\mathbf{G}^u = \mathbf{G}_k^u\mathbf{G}_{k-1}^u\cdots\mathbf{G}_1^u$,

$$\mathbf{G}^u \cdot \begin{pmatrix} -\mathbf{R}^{-T}\mathbf{X}^T & \mathbf{R}^{-T} \\ \mathbf{I}_k & \mathbf{0}_{k\times n} \end{pmatrix} = \begin{pmatrix} \mathbf{0}_{n\times k} & \tilde{\mathbf{R}}^{-T} \\ \mathbf{U}_u & \mathbf{Z} \end{pmatrix} \tag{3.3.3}$$

where $\tilde{\mathbf{R}}^{-1}$ is the desired updated factor.

Similarly, we expand the same idea to rank-k downdatings. Let $\mathbf{Y}$ be a $k \times n$ data matrix and consider the following equation:

$$\mathbf{R}^T\mathbf{R} - \mathbf{Y}^T\mathbf{Y} = \tilde{\mathbf{R}}^T\tilde{\mathbf{R}}, \tag{3.3.4}$$

and the matrix

$$\mathbf{C} = \begin{pmatrix} \mathbf{R}^{-T}\mathbf{Y}^T & \mathbf{R}^{-T} \\ \mathbf{I}_k & \mathbf{0}_{k\times n} \end{pmatrix}.$$

Let $\mathbf{G}_j^d$ be the product of hyperbolic Givens rotations operating on the top n rows and the $(n+j)$-th row of $\mathbf{C}$ such that the first n components of the j-th column of $\mathbf{C}$ are zero. Then for $\mathbf{G}^d = \mathbf{G}_k^d\mathbf{G}_{k-1}^d\cdots\mathbf{G}_1^d$,

$$\mathbf{G}^d \cdot \begin{pmatrix} \mathbf{R}^{-T}\mathbf{Y}^T & \mathbf{R}^{-T} \\ \mathbf{I}_k & \mathbf{0}_{k\times n} \end{pmatrix} = \begin{pmatrix} \mathbf{0}_{n\times k} & \tilde{\mathbf{R}}^{-T} \\ \mathbf{U}_d & \mathbf{W} \end{pmatrix} \tag{3.3.5}$$

where $\tilde{\mathbf{R}}^{-T}$ is the desired lower triangular factor. It can be also shown that $\mathbf{W}^T\mathbf{W} = \mathbf{R}^{-1}\mathbf{N}^T\left(\mathbf{I} - \mathbf{N}\mathbf{N}^T\right)^{-1}\mathbf{N}\mathbf{R}^{-T}$ and $\mathbf{N} = \mathbf{Y}\mathbf{R}^{-1}$.

The combined process of (3.3.3) and (3.3.5) is an algorithm for up-downdating of order k of the triangular factor. Let

$$\mathbf{R}^T\mathbf{R} + \mathbf{X}^T\mathbf{X} - \mathbf{Y}^T\mathbf{Y} = \tilde{\mathbf{R}}^T\tilde{\mathbf{R}},$$

and consider the matrix

$$\mathbf{D} = \begin{pmatrix} -\mathbf{R}^{-T}\mathbf{X}^T & \mathbf{R}^{-T}\mathbf{Y}^T & \mathbf{R}^{-T} & \mathbf{d} \\ \mathbf{I}_k & \mathbf{0}_k & \mathbf{0}_{k\times n} & \mathbf{b}_X \\ \mathbf{0}_k & \mathbf{I}_k & \mathbf{0}_{k\times n} & \mathbf{b}_Y \end{pmatrix}.$$

Let $\mathbf{S}_j$, $1 \le j \le k$, be the product of Givens rotations operating on the top n rows and the $(n+j)$-th row of $\mathbf{D}$ such that the first n components of the j-th column of $\mathbf{P}$ are zero. And let $\mathbf{S}_j$, $k+1 \le j \le 2k$,

be the product of hyperbolic Givens rotations operating on the top n rows and the $(n+j)$-th row of D such that the first n components of the $(j-k)$-th column of D are zero. Then for $S = S_{2k}S_{2k-1}\cdots S_1$,

$$S \cdot \begin{pmatrix} -R^{-T}X^T & R^{-T}Y^T & R^{-T} & d \\ I_k & 0_k & 0_{k\times n} & b_X \\ 0_k & I_k & 0_{k\times n} & b_Y \end{pmatrix} = \begin{pmatrix} 0_{n\times k} & 0_{n\times k} & \tilde{R}^{-T} & \tilde{d} \\ U_s & P & Z & \tilde{b}_X \\ 0_k & U_d & W & \tilde{b}_Y \end{pmatrix} \tag{3.3.6}$$

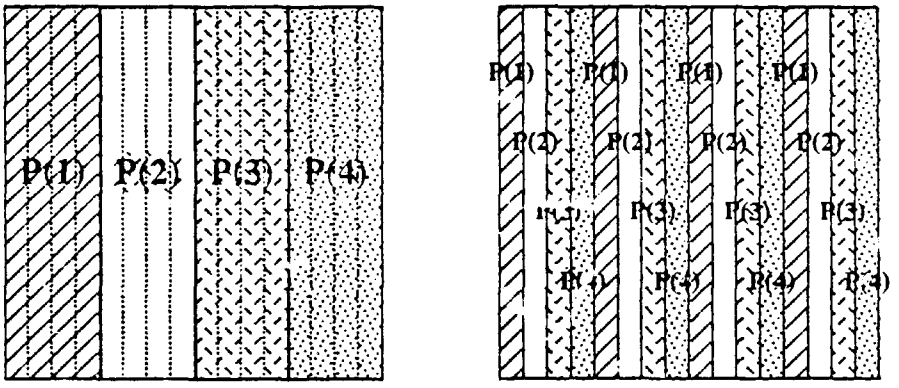
where P is a square matrix and $\tilde{R}^{-T}$ is the desired factor.

This algorithm is based on Givens/hyperbolic Givens transformations, so it requires $nk(8k+5n+17)$ multiplications

4. Mapping

We assume that we are given a ring of processors. The processors are numbered from 1 to p with the leftmost processor labeled P_1 and the rightmost processor labeled P_p. The data matrix is distributed among processors. The mapping of the matrix onto the processors affects the communication requirements, the degree of concurrency and the load balance among the processors. The objectives to minimize communication, maximize concurrency and uniformly balance the load tend to conflict [7].

There are two widely used mappings; 'block mapping' and 'wrap mapping'. In the block mapping, contiguous blocks of n/p columns are assigned onto each processor, where n is the size of the matrix and p is the number of processors. The first processor P_1 has the first n/p columns (column 1 to column n/p), the second processor P_2 has the next n/p columns (column $n/p+1$ to column $2n/p$), and so on. In the wrap mapping, consecutive columns are assigned to consecutive processors from the first processor P_1 to the last processor P_p, then wrapping back to P_1 and continuing with further columns. In the wrap mapping, the first processor P_1 contains all columns $j \cdot p + 1$ $(0 \le j < n/p)$, the second processor P_2 has all columns $j \cdot p + 2$ $(0 \le j < n/p)$, and so on. The two mappings are illustrated in Fig. 4.1.



(a) Block Mapping (b) Wrap Mapping

Figure 4.1. Two Mappings for $n = 16$ and $p = 4$

We will compare the two different mappings in implementing Algorithms A and B and their effects on the overall execution time of up-downdating followed by solution of a triangular system of linear equations. At the i-th step of up-downdating ($1 \leq i \leq n$), the processor which is assigned the i-th column of the matrix computes the corresponding transformation and sends the parameters of the transformation to the processors that are assigned the columns whose indices are greater than i. In the wrap mapping, the last processor P_p receives the transformation from P_{p-1} and sends it to P_1 again if necessary. In the block mapping, the last processor in the ring P_p only receives and never sends transformations.

After applying transformations to its assigned columns, each processor becomes idle for the remainder of the current step. The block mapping causes processors containing the earlier blocks to be idle much of the time, whereas the wrap mapping keeps all processors busy as long as possible. Thus the wrap mapping is likely to yield higher concurrency and processor utilization than the block mapping. On the other hand, the block mapping has potentially smaller communication requirements, since transformations generated by a processor need to be sent only to higher numbered processors, rather than all processors as in the wrap mapping. Once the data matrix is triangularized, the resulting triangular system of linear equations has to be solved. This is discussed in the next section.

5. Parallel Triangular Solver and Inverse of the Triangular Matrix

5.1 Parallel Triangular Solver

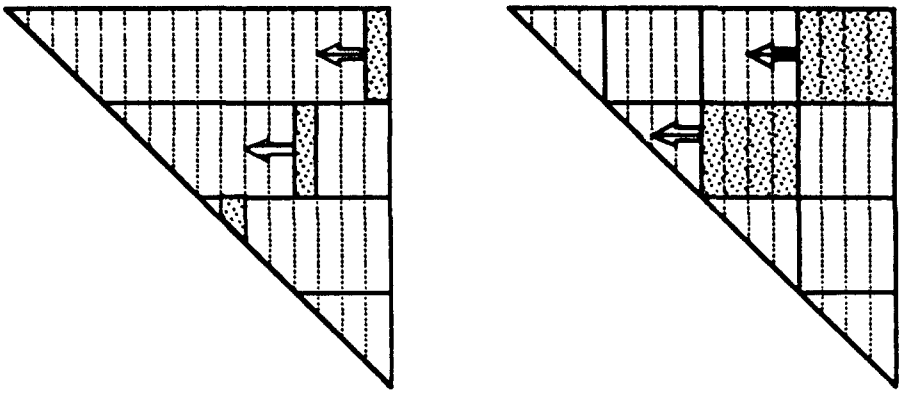
Back-substitution is used to solve the triangular system in Algorithm A. The serial algorithm to compute the solution of $\mathbf{R} \cdot \mathbf{x} = \mathbf{b}$ is as follows:

For $i = n:1$
  $\mathbf{x}_i = \mathbf{b}_i / \mathbf{R}_{i,i}$
  $\mathbf{b}_{1:i-1} = \mathbf{b}_{1:i-1} - \mathbf{R}_{1:i-1,i} \cdot \mathbf{x}_i$
end

where we use MATLAB notation: $\mathbf{b}_{1:i-1} = [\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_{i-1}]^T$, $\mathbf{R}_{1:i-1,i} = [\mathbf{R}_{1,i}, \mathbf{R}_{2,i}, \cdots, \mathbf{R}_{i-1,i}]^T$, $\mathbf{R}_{:,i}$ is the i-th column of $\mathbf{R}$ and $\mathbf{R}_{j,:}$ is the j-th row of $\mathbf{R}$.

Li & Coleman [4] and Heath & Romine [5] implemented the algorithms for solving triangular systems on a hypercube multiprocessor. We adopt the Li & Coleman algorithm for solving the triangular system in the wrap mapping of Algorithm A. In a naive implementation of back-substitution algorithm using the wrap mapping at the i-th step, $1 < i < n$, the processor P_j, $1 < j < p$, receives the vector $\mathbf{b}_{1:i}$ from the right processor P_{j+1}, computes one component $\mathbf{x}_i$ of the solution, updates the vector $\mathbf{b}_{1:i-1}$ and sends the updated vector $\mathbf{b}_{1:i-1}$ to the next processor P_{j-1}. The computations of the algorithm are distributed over processors, but are not executed simultaneously. There are no concurrent computations. Li & Coleman have modified the algorithm so that operations can be overlapped. For details, see [4].

Heath & Romine developed a wavefront algorithm for the wrap mapping. In their algorithm, each processor computes one component $\mathbf{x}_i$ ($1 < i < n$) of the solution, then proceeds to compute the vector $\mathbf{zsum}_{1:i-1} = \mathbf{R}_{1:i-1,i} \cdot x_i$. After computing the first σ components (for some integer σ satisfying $1 \leq \sigma \leq n$),



(a) Wavefront Algotiyhm in Wrap Mapping

(b) Modified Wavefront Algorithm in Block Mapping

Figure 5.1. Snapshots of the Wavefront Algorithm

the processor P_j $(1 < j < p)$ sends them to the left neighbor P_{j-1} so that the latter can compute the next component. Meanwhile, the processor P_j resumes the computation of the next σ components of **zsum**. We modified the algorithm for the block mapping. Each processor, when ready, computes n/p components of **x**. After the processor receives n/p components of **zsum** from the left neighbor, it updates them by multiplying the components by the corresponding $n/p \times n/p$ submatrix of the triangular matrix **R** and sends them to the right neighbor, and so on. A snapshot of the wavefront algorithm for the wrap mapping and a snapshot of the modified wavefront algorithm for the block mapping are illustrated in Figure 5.1.

5.2 Parallel Algorithm for Computing the Transposed Inverse of a Triangular Matrix

In order to avoid the back-substitution in Algorithms B and C, the triangular matrix is inverted during initialization of the recursive process and then the inverse is maintained. The cost of computing the inverse is high but affects initialization only. Here we propose a parallel algorithm for computing the transposed inverse of a triangular matrix.

Assume that $\mathbf{R}$ is a nonsingular and upper triangular $n \times n$ matrix. $\mathbf{R}^{-T}$ is required in the initial step of the Algorithms B and C. Let $\mathbf{R} \cdot \mathbf{T} = \mathbf{B}$. Then $\mathbf{B}_{ij} = \sum_{k=1}^{n} \mathbf{R}_{ik}\mathbf{T}_{kj}$. Assume that $\mathbf{T} = \mathbf{R}^{-1}$, $\mathbf{B} = \mathbf{I}$, so $\mathbf{R} \cdot \mathbf{T} = \mathbf{T} \cdot \mathbf{R} = \mathbf{I}$. $\mathbf{T}$ is also a nonsingular and upper triangular matrix. From $\mathbf{R} \cdot \mathbf{T} = \mathbf{I}$,

$$\begin{cases} \mathbf{R}_{ii}\mathbf{T}_{ii} = 1, & \text{if } i = j \\ \sum\limits_{i \le k \le j} \mathbf{R}_{ik}\mathbf{T}_{kj} = 0, & \text{if } i < j \\ \sum\limits_{j \le k \le i} \mathbf{R}_{ik}\mathbf{T}_{kj} = 0, & \text{if } i > j \end{cases}$$

The elements of $\mathbf{T}$ are

$$\mathbf{T}_{ij} = \begin{cases} 1/\mathbf{R}_{ii}, & \text{if } i = j \\ (\sum\limits_{i < k \le j} \mathbf{R}_{ik}\mathbf{T}_{kj})/\mathbf{R}_{ii}, & \text{if } i < j \\ 0, & \text{if } i > j \end{cases} \tag{5.1}$$

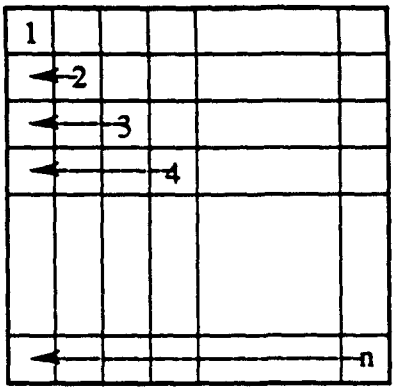


Figure 5.2. Order of Computing $\mathbf{L}(=\mathbf{R}^{-T})$

Let $\mathbf{L} = \mathbf{T}^T = \mathbf{R}^{-T}$. The elements of $\mathbf{L}$ are generated from top row to the bottom row. The order of computation is shown in Figure 5.2.

A sequential code to compute (5.1) could look as follows:

For $i = 1 : n$
  $\mathbf{L}_{i,i} = 1 / \mathbf{R}_{i,i}$
  if $i > 1$
    $temp_{1:i-1} = \mathbf{L}_{i,i} \cdot \mathbf{R}_{1:i-1,i}$
    For $k = i - 1 : -1 : 1$
      $\mathbf{L}_{i,k} = -temp_k \cdot \mathbf{L}_{k,k}$
      $temp_{1:k-1} = temp_{1:k-1} + \mathbf{L}_{i,k} \cdot \mathbf{R}_{1:k-1,k}$ (if $k > 1$)
    end
  end
end

Assume that we are given n linearly connected processors P_i $(1 \leq i \leq n)$ arranged from left to right. Let the i-th processor P_i store initially the i-th column of $\mathbf{R}$ $(\mathbf{R}_{1:i,i})$; at the end of the computations, it will have the i-th column of $\mathbf{L}$ $(\mathbf{L}_{i:n,i})$. The parallel version of the algorithm might look as follows:

$\mathbf{L}_{i,i} = 1 / \mathbf{R}_{i,i}$
$temp_{1:i-1} = \mathbf{L}_{i,i} \cdot \mathbf{R}_{1:i-1,i}$
send $temp_{1:i-1}$ to P_{i-1} (if $i > 1$)
For $k = i + 1 : n$
  receive $temp_{1:i}$ from P_{i+1} (if $i < n$)
  $\mathbf{L}_{k,i} = -temp_i \cdot \mathbf{L}_{i,i}$
  $temp_{1:i-1} = temp_{1:i-1} + \mathbf{L}_{k,i} \cdot \mathbf{R}_{1:i-1,i}$
  send $temp_{1:i-1}$ to P_{i-1} (if $i > 1$)
end

Processor P_i computes $\mathbf{L}_{i,i}$, calculates $i - 1$ components of the intermediate vector $temp$, $temp_{1:i-1} = \mathbf{L}_{i,i} \cdot \mathbf{R}_{1:i-1,i}$, and sends them to the left neighbor P_{i-1}. Next P_i receives $temp_{1:i}$ from the right neighbor P_{i+1}, computes $\mathbf{L}_{i+1,i}$ and updates $temp_{1:i-1}$, and so on. The order of computing $\mathbf{L}_{i:n,i}$ in P_i is shown in Figure 5.3.

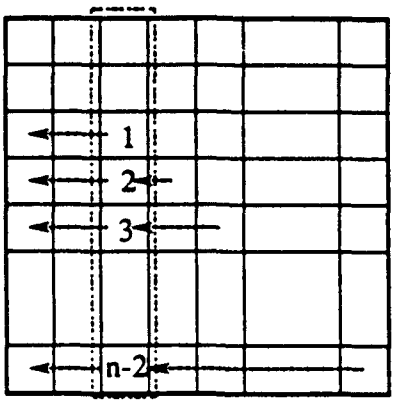


Figure 5.3. Order of Computing $\mathbf{L}(=\mathbf{R}^{-T})$ in Parallel

The transposed inverse **L** is lower triangular and can be conveniently stored together with the upper triangular matrix **R**, forming a square matrix. The parallel algorithm for computing the transposed inverse of a triangular matrix can be easily implemented in both the block mapping and the wrap mapping.

6. Transputer and Occam

The INMOS transputer is a high-speed parallel processor which combines processing, memory and interconnect in a single VLSI chip. It contains 4 inter-processor links which provide high-speed communication. INMOS T800/20 is a 32-bit transputer with floating point processor which is able to perform floating point operations concurrently with the processor, sustaining a rate of 1.5 Mflops [2].

A link between two transputers is implemented by connecting link interfaces on the transputers by two one-directional signal wires, along which data is transmitted serially. Messages are transmitted as a sequence of bytes, each of which must be acknowledged before the next is transmitted.

The test system has 8 transputers on two boards in a Mac IIx. Each board contains up to four transputers and has one INMOS C004 programmable link switch designed to provide a full crossbar switch between 32 input links and 32 output links. A message has to pass through the C004 switch once for on-board communication, and twice for off-board communication. We configured the transputers as a ring and measured the communication time for this configuration. We measured the time for sending a 64-bit message around the ring of 8 transputers 1000 times. Then the average time for the toral of 8000 communications was recorded as T_{comm}. We also measured the execution time for the loop,

```
For i = 1:8000
    y = xy + z
end
```

where all operations are double-precision (64-bit) floating point operations. In order to decrease the overhead caused by the loop, the loop was modified by repeating 10 multiplications and additions within the loop. The average computation time was recorded as T_{comp}. The ratio of the communication time to the computation time of the system is,

$$\frac{T_{comm}}{T_{comp}} \simeq \frac{8.944usec}{2.120usec} \simeq 4.219.$$

The communication time is about 4.219 times more expensive than the computation time in the transputer system. The communication speed plays an important role in the execution time of parallel algorithms. By reducing the communication requirements of the algorithms, even at the expense of increasing computational work load, we may save the total execution time of parallel algorithms.

The main programming language supported by transputers is OCCAM. All programs were written in OCCAM. Occam is designed to support explicit hardware concurrency and it reflects the concurrency found in the transputer [9]. Although transputers provide implementation of most modern programming languages, concurrency in transputer systems is most effectively supported by Occam.

7. Implementation of Hyperbolic Householder Transformation

In this section, we present a possible implementation of a Householder transformation. The variant we consider is related to that suggested in [10].

We start with a definition of a hyperbolic Householder transformation. An $n \times n$ hyperbolic Householder matrix has the form

$$\mathbf{H_Y} = \Phi - 2\mathbf{v}\mathbf{v}^T / \mathbf{v}^T\Phi\mathbf{v} \tag{7.1}$$

where $\Phi = \mathrm{diag}(\pm 1)$ and $\mathbf{v}$ is any vector for which $\mathbf{v}\Phi\mathbf{v} \neq 0$.

In our recursive up-downdating Φ is of the form,

$$\Phi = \mathrm{diag}\,(\mathrm{I}, \mathrm{I}_p, -\mathrm{I}_q)$$

where the middle I_p corresponds to rank-p updating and the last $-\mathrm{I}_q$ corresponds to rank-q downdating. Let $\mathbf{c}$ be a vector such that $\|\mathbf{c}\|_\Phi{}^2 = \mathbf{c}^T\Phi\mathbf{c}$ is positive. Then the choice $\mathbf{v} = \mathbf{c} \mp \|\mathbf{c}\|_\Phi \mathbf{e}_1$ guarantees $\mathrm{H}_\mathbf{c} = \pm\|\mathbf{c}\|_\Phi \mathbf{e}_1$, i.e., H can compress the hyperbolic norm of the vector $\mathbf{c}$ into a selected component of $\mathbf{c}$.

Let us consider the l-th column of Eq.(3.1.1) in the k-th modification step. From (7.1), we have that

$$\begin{pmatrix} \tilde{\mathrm{R}}_{k,:} \\ \tilde{\mathrm{X}}_{:,:} \\ \tilde{\mathrm{Y}}_{:,:} \end{pmatrix} = \mathrm{H}_{\mathbf{Y}k} \cdot \begin{pmatrix} \mathrm{R}_{k,:} \\ \mathrm{X}_{:,:} \\ \mathrm{Y}_{:,:} \end{pmatrix} = \begin{pmatrix} \mathrm{R}_{k,:} - 2\,\mathbf{v}_1\ M_{k:}\ / \mathbf{v}\Phi_k\mathbf{v} \\ \mathrm{X}_{:,:} - 2\,\mathbf{v}_{2:p+1}\,M_{k:} / \mathbf{v}\Phi_k\mathbf{v} \\ -\mathrm{Y}_{:,:} + 2\mathbf{v}_{p+2:p+q+1}\,M_{k:}/\mathbf{v}\Phi_k\mathbf{v} \end{pmatrix} \tag{7.2}$$

where $M_{k:} = \mathbf{v}^T \cdot (\,\mathrm{R}_{k,:}^T, \mathrm{X}_{:,:}^T, \mathrm{Y}_{:,:}^T\,)^T$.

Let

$$S_{kk} = \mathrm{R}_{k,k}^2 + \|\mathrm{X}_{:,k}\|^2 - \|\mathrm{Y}_{:,k}\|^2 = \tilde{\mathrm{R}}_{k,k}^2,$$

$$s_k = \mathrm{sqrt}(S_{kk}),$$

and

$$S_{k:} = \mathrm{R}_{k,k}\mathrm{R}_{k,:} + \sum_{i=1}^{p} \mathrm{X}_{i,k}\mathrm{X}_{i,:} - \sum_{i=1}^{q} \mathrm{Y}_{i,k}\mathrm{Y}_{i,:}.$$

According to the definition of $\mathbf{v}$, $\mathbf{v}_1 = \mathbf{c}_1 \pm \|\mathbf{c}\|_\Phi = \mathrm{R}_{k,k} \pm s_k$. $\tilde{\mathrm{R}}_{k,:}$ can be represented as follows:

$$\tilde{\mathbf{R}}_{k,:} = \mathbf{R}_{k,:} - 2\,\mathbf{v}_1 \frac{M_{k:}}{\mathbf{v}\Phi_k\mathbf{v}} = \mathbf{R}_{k,:} - 2(\mathbf{R}_{k,k} \pm s_k)\frac{S_{k:} \pm s_k \mathbf{R}_{k,:}}{2\,s_k(s_k \pm \mathbf{R}_{k,k})} = \mp\frac{S_{k:}}{s_k}. \tag{7.3}$$

Then, using the above result,

$$\tilde{\mathbf{X}}_{:,:} = \mathbf{X}_{:,:} - 2\,\mathbf{v}_{2:p+1}\frac{M_{k:}}{\mathbf{v}\Phi_k\mathbf{v}} = \mathbf{X}_{:,:} - \frac{\mathbf{v}_{2:p+1}}{\mathbf{v}_1}(\mathbf{R}_{k,:} - \tilde{\mathbf{R}}_{k,:})$$

and

$$\tilde{\mathbf{Y}}_{:,:} = \mathbf{Y}_{:,:} - 2\,\mathbf{v}_{p+2:p+q+1}\frac{M_{k:}}{\mathbf{v}\Phi_k\mathbf{v}} = \mathbf{Y}_{:,:} - \frac{\mathbf{v}_{p+2:p+q+1}}{\mathbf{v}_1}(\mathbf{R}_{k,:} - \tilde{\mathbf{R}}_{k,:})$$

By defining $\mathbf{v}_{2:p+q+1} = \mathbf{v}_{2:p+q+1}/\mathbf{v}_1$, (7.2) can be represented as follows:

$$\begin{pmatrix} \tilde{\mathbf{R}}_{k,:} \\ \tilde{\mathbf{X}}_{:,:} \\ \tilde{\mathbf{Y}}_{:,:} \end{pmatrix} = \begin{pmatrix} \mp S_{k:}/s_k \\ \mathbf{X}_{:,:} - \mathbf{v}_{2:p+1}\,(\mathbf{R}_{k,:} - \tilde{\mathbf{R}}_{k,:}) \\ -\mathbf{Y}_{:,:} + \mathbf{v}_{p+2:p+q+1}(\mathbf{R}_{k,:} - \tilde{\mathbf{R}}_{k,:}) \end{pmatrix}. \tag{7.4}$$

$\tilde{\mathbf{R}}_{k,:}$ is computed directly from (7.3). By forming the component $\tilde{\mathbf{R}}_{k,:}$ first, the other components, $\tilde{\mathbf{X}}_{:,:}$ and $\tilde{\mathbf{Y}}_{:,:}$, can be formed in a straitforward way.

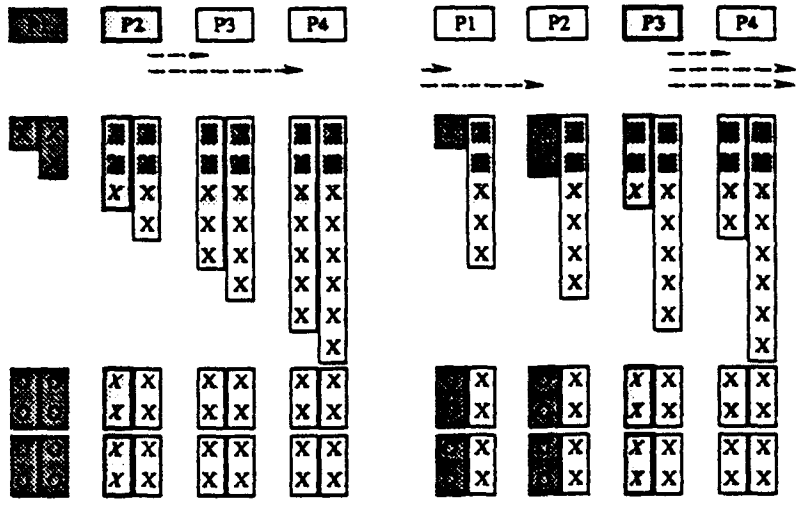
8. Implementations

We implemented the three algorithms for up-downdating on the transputer array. Figures 8.1, 8.2 and 8.3 show how the data is distributed in the transputer array for Algorithms A, B and C, respectively. The figures show snapshots for the cases of a 16 x 16 data matrix, 4 processors and up-downdating of order 2. The darkly shaded area represents data for which corresponding computations are finished and no more changes will occur. The data in the lightly shaded area represents active data which is operated on. The white area represents data that will be operated on.

Figure 8.1a shows the typical block mapping of the algorithm. The computations start from the left-most node and propagate to the right. The other processors receive transformations, send them to the right and update their own data according to (7.4). The rightmost processor only receives the transformations from the left. The block mapping of Algorithm A has the simplest communication scheme. However the load balance of the block mapping among the nodes is not as good as that of the wrap mapping. The wrap mapping of Algorithm A is shown Figure 8.2b.

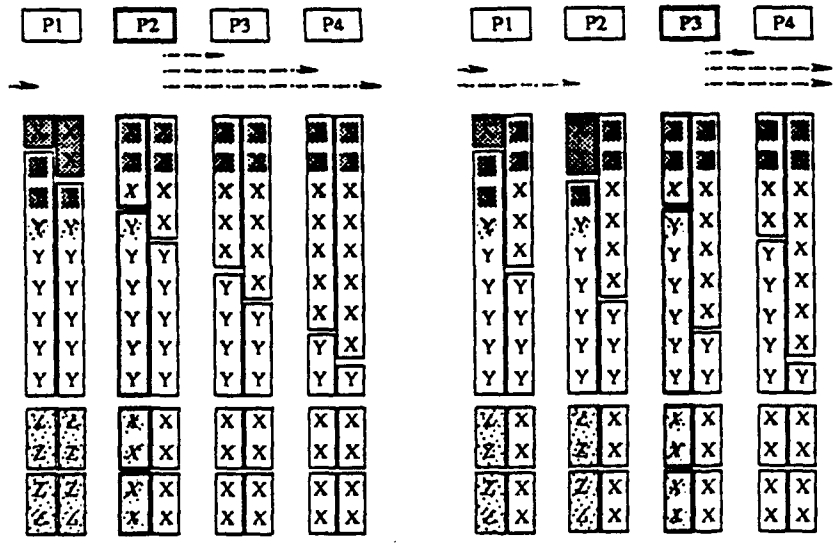
Algorithm B operates on the triangular matrix and its inverse. The transposed inverse matrix is placed at the bottom of the triangular matrix. The computation load is perfectly balanced among processors. The block mapping of Algorithm B has simpler communication strategy than the wrap mapping.

Each processor in Algorithm C computes its own partial products of $-\mathbf{R}^{-T}\mathbf{X}^T$ and $\mathbf{R}^{-T}\mathbf{Y}^T$ of (3.3.6) based on its own partial data $\mathbf{R}^{-T}$, $\mathbf{X}$ and $\mathbf{Y}$. All partial pruducts in all processors have to be added to construct the elements of $-\mathbf{R}^{-T}\mathbf{X}^T$ and $\mathbf{R}^{-T}\mathbf{Y}^T$. The nodes send their own partial products to the right neighbors, receive the others' from the left, and then update their own partial products with the received partial products. Then the nodes send the received partial products to the right neighbor, receive new partial products from the left and update their partial products again. After repeating $p-1$ times, all of



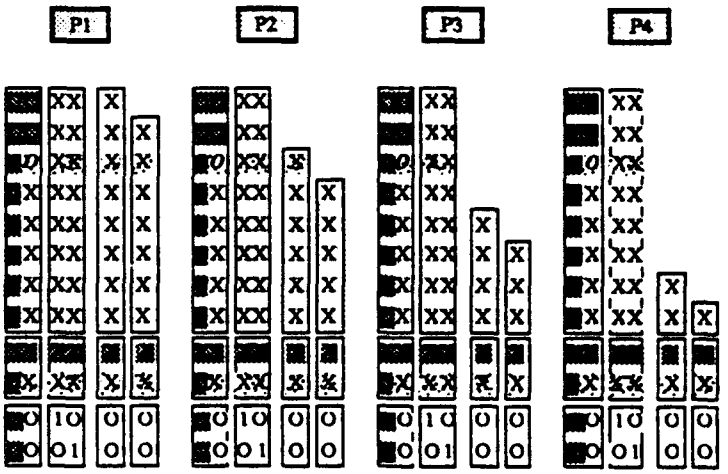
a) Block Mapping b) Wrap Mapping

Figure 8.1. Snapshots of Algorithm A ($n = 8$, $p = 4$ and $k = 2$)

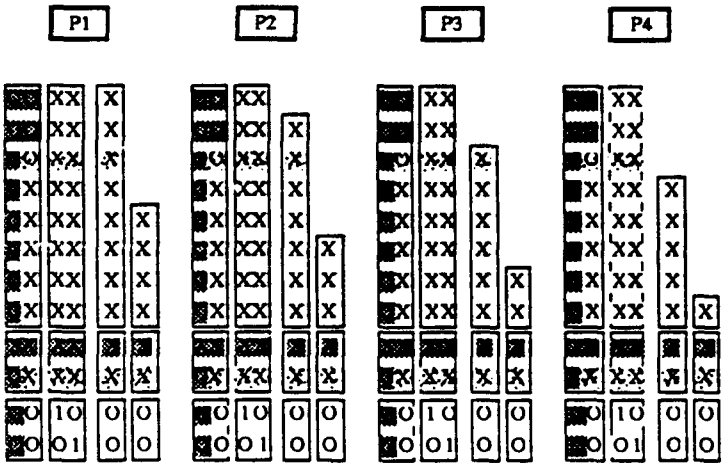


a) Block Mapping b) Wrap Mapping

Figure 8.2. Snapshots of Algorithm B ($n = 8$, $p = 4$ and $k = 2$)



a) Block Mapping



b) Wrap Mapping

Figure 8.3. Snapshots of Algorithm C ($n = 8$, $p = 4$ and $k = 2$)

Table 9.1. Timing Results of Algorithm A (msecs)

a) Block Mapping

Size of Data		1 Proc	2 Proc	4 Proc	8 Proc
16 x 16	1 col up&dn	5.888	4.928	3.584	3.008
	2 col up&dn	8.960	7.168	5.184	4.096
	4 col up&dn	15.040	11.712	8.256	6.400
32 x 32	1 col up&dn	23.104	19.264	12.352	8.320
	2 col up&dn	35.008	28.864	17.792	11.712
	4 col up&dn	58.816	47.552	28.800	18.560
48 x 48	1 col up&dn	50.880	42.432	25.728	16.128
	2 col up&dn	77.312	63.360	37.568	23.104
	4 col up&dn	130.176	105.088	61.248	37.120
64 x 64	1 col up&dn	90.816	75.392	47.488	27.008
	2 col up&dn	138.112	112.256	69.568	38.720
	4 col up&dn	232.768	185.728	114.560	62.656
80 x 80	1 col up&dn	140.800	115.968	71.360	40.384
	2 col up&dn	214.336	173.888	106.240	58.560
	4 col up&dn	361.728	288.192	176.000	95.040

b) Wrap Mapping

Size of Data		1 Proc	2 Proc	4 Proc	8 Proc
16 x 16	1 col up&dn	5.888	6.784	4.288	3.392
	2 col up&dn	8.960	10.240	6.208	4.736
	4 col up&dn	15.040	16.960	10.112	7.424
32 x 32	1 col up&dn	23.104	26.368	14.592	9.152
	2 col up&dn	35.008	40.064	21.696	13.184
	4 col up&dn	58.816	67.392	35.776	21.312
48 x 48	1 col up&dn	50.880	56.704	30.528	17.984
	2 col up&dn	77.312	86.528	45.696	26.368
	4 col up&dn	130.176	146.240	75.904	43.008
64 x 64	1 col up&dn	90.816	101.312	54.592	30.144
	2 col up&dn	138.112	155.648	82.432	44.608
	4 col up&dn	232.768	256.920	137.984	73.472
80 x 80	1 col up&dn	140.800	156.160	81.920	44.928
	2 col up&dn	214.336	240.832	124.160	66.944
	4 col up&dn	361.728	410.176	208.512	111.040

the nodes collect products of $-\mathbf{R}^{-T}\mathbf{X}^T$ and $\mathbf{R}^{-T}\mathbf{Y}^T$. Once the elements of $-\mathbf{R}^{-T}\mathbf{X}^T$ and $\mathbf{R}^{-T}\mathbf{Y}^T$ are computed, no more communication is required. Now, each processor computes Givens rotations based on $-\mathbf{R}^{-T}\mathbf{X}^T$ and hyperbolic Givens rotations based on $\mathbf{R}^{-T}\mathbf{Y}^T$, and applies the rotations to $\mathbf{R}$ as well as to $-\mathbf{R}^{-T}\mathbf{X}^T$ and $\mathbf{R}^{-T}\mathbf{Y}^T$. This part of the computation is purely sequential. The snapshots are illustrated in Figure 8-3, where the leftmost box represents $-\mathbf{R}^{-T}\mathbf{X}^T$, and the next shows $\mathbf{R}^{-T}\mathbf{Y}^T$. The wrap mapping of Algorithm C exhibits better load balancing than block mapping.

9. Results and Discussions

We implemented the three algorithms and measured the execution time on an 8 transputer array. Tables 9.1, 9.2 and 9.3 show the time results of Algorithms A, B and C, respectively. We tested the algorithms for several sets of data with one, two and four column up-downdatings.

The time results of the algorithms are compared in Figures 9.1, 9.2 and 9.3 for 32×32 and 64×64 data matrices. The results show that Algorithm A is usually the fastest since it requires the minimum number of operations. The block mapping of Algorithm A is better for small data matrices, small numbers

Table 9.2. Timing Results of Algorithm B (msecs)

a) Block Mapping

Size of Data		1 Proc	2 Proc	4 Proc	8 Proc
16 x 16	1 col up&dn	10.176	6.784	4.288	3.392
	2 col up&dn	15.616	10.240	6.208	4.736
	4 col up&dn	26.624	16.960	10.112	7.424
32 x 32	1 col up&dn	41.152	26.368	14.592	9.152
	2 col up&dn	63.808	40.064	21.696	13.184
	4 col up&dn	108.992	67.392	35.776	21.312
48 x 48	1 col up&dn	92.096	56.704	30.528	17.984
	2 col up&dn	143.232	86.528	45.696	26.368
	4 col up&dn	245.504	146.240	75.904	43.008
64 x 64	1 col up&dn	165.568	101.312	54.592	30.144
	2 col up&dn	257.920	155.648	82.432	44.608
	4 col up&dn	442.560	256.920	137.984	73.472
80 x 80	1 col up&dn	257.472	156.160	81.920	44.928
	2 col up&dn	402.048	240.832	124.160	66.944
	4 col up&dn	691.136	410.176	208.512	111.040

b) Wrap Mapping

Size of Data		1 Proc	2 Proc	4 Proc	8 Proc
16 x 16	1 col up&dn	10.176	7.104	5.056	4.160
	2 col up&dn	15.616	10.688	7.360	5.696
	4 col up&dn	26.624	17.856	11.904	8.768
32 x 32	1 col up&dn	41.152	26.688	16.064	11.456
	2 col up&dn	63.808	40.896	23.808	16.064
	4 col up&dn	108.992	69.184	39.104	25.472
48 x 48	1 col up&dn	92.096	57.280	32.576	21.568
	2 col up&dn	143.232	88.192	48.832	30.784
	4 col up&dn	245.504	149.888	81.408	49.472
64 x 64	1 col up&dn	165.568	99.712	56.512	34.560
	2 col up&dn	257.920	154.240	85.760	50.432
	4 col up&dn	442.560	263.232	144.192	82.304
80 x 80	1 col up&dn	257.472	153.024	84.992	50.112
	2 col up&dn	402.048	237.568	129.728	73.664
	4 col up&dn	691.136	406.848	219.264	121.152

of up-downdating columns and large numbers of processors. On the other hand, wrap mapping is better for large data matrices, large numbers of up-downdating columns and small numbers of processors. For example, if each node has a light computation load, block mapping is preferable. In the case of a heavy load, however, wrap mapping is preferable.

Algorithm B operates on the triangular matrix and its transposed inverse. It uses direct matrix-vector multiplications to obtain the solution of the triangular system. The number of multiplications in Algorithm B is twice that of Algorithm A, so it is usually slower than Algorithm A. The block mapping of Algorithm B always has better performance than wrap mapping, since the computation load is perfectly balanced among processors for both mappings, and the former has a simpler communications scheme.

The construction of $-\mathbf{R}^{-T}\mathbf{X}^T$ and $\mathbf{R}^{-T}\mathbf{Y}^T$ in Algorithm C impose a large communication overhead since each processor needs its own copy of $-\mathbf{R}^{-T}\mathbf{X}^T$ and $\mathbf{R}^{-T}\mathbf{Y}^T$, and this requires additional transfer of data among processors. Each needs $p-1$ communication and addition steps to acquire their own copy of the products. In addition, the algorithm uses Givens and hyperbolic Givens rotations instead of hyperbolic Householder reflections. Therefore, it requires more multiplications than the other two

Table 9.3. Timing Results of Algorithm C (msecs)

a) Block Mapping

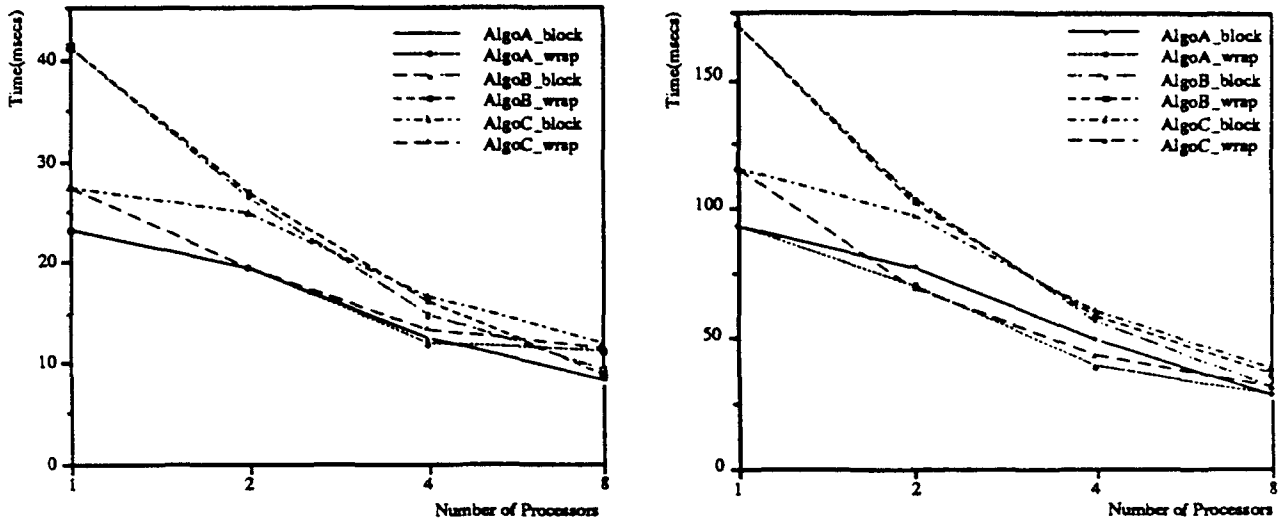
Size of Data		1 Proc	2 Proc	4 Proc	8 Proc
16 x 16	1 col up&dn	7.872	6.784	5.056	4.160
	2 col up&dn	15.616	13.696	10.368	8.768
	4 col up&dn	33.600	29.504	23.680	20.544
32 x 32	1 col up&dn	27.200	24.704	16.384	11.776
	2 col up&dn	56.576	48.704	32.704	24.128
	4 col up&dn	116.480	101.696	70.784	54.208
48 x 48	1 col up&dn	63.488	53.760	33.600	22.528
	2 col up&dn	122.112	103.744	67.008	45.312
	4 col up&dn	245.504	213.248	139.648	98.944
64 x 64	1 col up&dn	111.872	94.080	58.624	37.184
	2 col up&dn	214.080	178.688	113.856	73.408
	4 col up&dn	428.800	361.280	237.504	156.416
80 x 80	1 col up&dn	172.992	145.536	89.536	54.272
	2 col up&dn	329.984	274.048	173.376	107.072
	4 col up&dn	657.152	555.072	354.944	226.240

b) Wrap Mapping

Size of Data		1 Proc	2 Proc	4 Proc	8 Proc
16 x 16	1 col up&dn	7.872	5.760	4.672	4.544
	2 col up&dn	15.616	11.584	9.664	9.280
	4 col up&dn	33.600	25.728	22.272	21.504
32 x 32	1 col up&dn	27.200	19.392	13.184	11.264
	2 col up&dn	56.576	38.208	26.560	22.912
	4 col up&dn	116.480	80.832	59.008	51.776
48 x 48	1 col up&dn	63.488	39.936	25.024	19.712
	2 col up&dn	122.112	77.568	49.600	39.680
	4 col up&dn	245.504	158.208	107.456	87.936
64 x 64	1 col up&dn	111.872	66.944	41.408	30.272
	2 col up&dn	214.080	129.024	81.152	60.096
	4 col up&dn	428.800	268.416	174.336	130.944
80 x 80	1 col up&dn	172.992	101.440	60.544	42.304
	2 col up&dn	329.984	194.624	117.952	83.648
	4 col up&dn	657.152	400.896	247.168	180.672

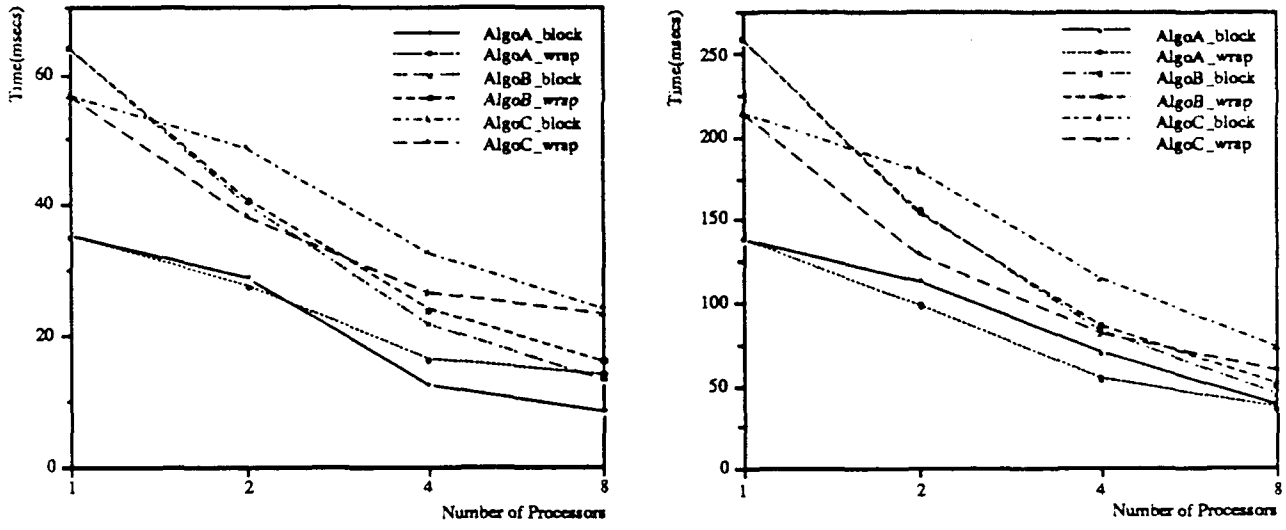
algorithms. Householder transformations have almost the same number of multiplications as additions, and the number of additions in Givens rotations is half the number of multiplications. Algorithm C needs slightly more multiplications than Algorithm B. But the number of additions required by Algorithm C is almost half of the number of multiplications, and much less than the number of additions required by Algorithm B. Thus, Algorithm C is faster than Algorithm B for small numbers of up-downdating columns though more multiplications are required. The wrap mapping of the algorithm is even faster than that of the Algorithm A for 1 column up-do' ndatings with 2 processors. The algorithm performs poorly for the large number of up-downdating columns since each node has to modify $-\mathbf{R}^{-T}\mathbf{X}^T$ and $\mathbf{R}^{-T}\mathbf{Y}^T$ as well as portions of the matrix inverse.

A buffer between two processors is necessary to smooth communication among nodes. The transputer system does not support the hardware buffer. However, the buffer can be simulated with software. A software buffer is different from a hardware buffer, in that communication between a software buffer and a process can be accomplished through a software channel, which is a memory-read or memory-write operation. One software buffer is inserted between every two processors. Figure 9.4 shows performance of the algorithms without using buffers. Note that processors in Algorithm C do not need buffers since they



a) $n = 32$ b) $n = 64$

Figure 9.1. Comparison of Algorithms for 1 column up-downdatings

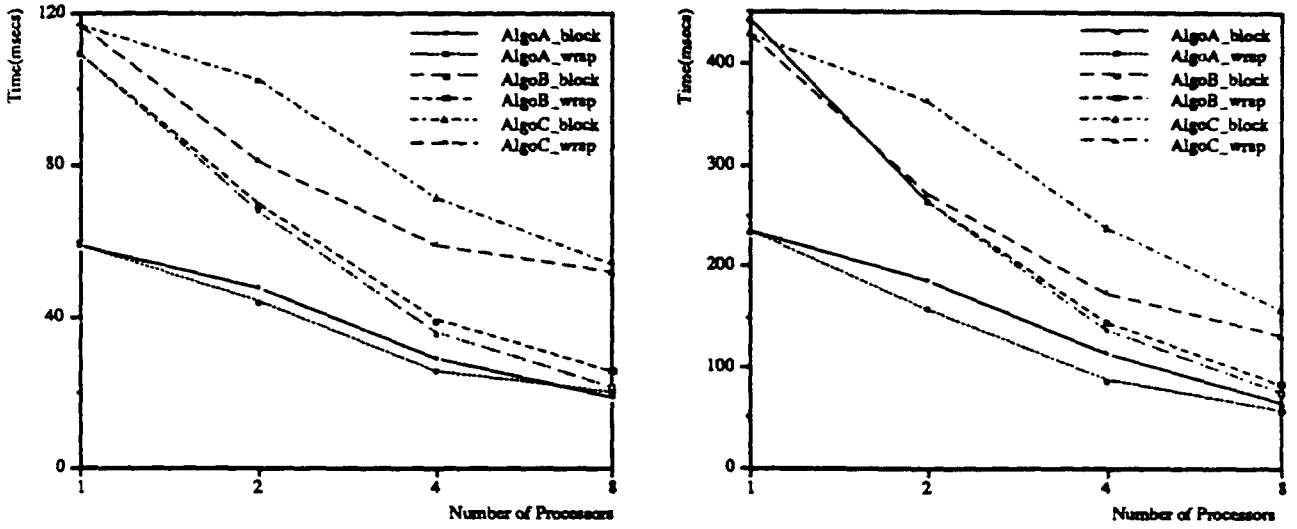


a) $n = 32$ b) $n = 64$

Figure 9.2. Comparison of Algorithms for 2 column up-downdatings

do not have to communicate to modify their own data. Clearly, the buffered implementations have better performance than the unbuffered, especially in the wrap mapping of Algorithm A.

The communication speed can be altered in the transputer array. The algorithms were initially run with 20 Mbits/sec transfer rate on each channel. In addition, they were run with 10 Mbits/sec transfer rate. The ratio of the communication time to the computation time is changed to the following ratio when



a) $n = 32$ b) $n = 64$

Figure 9.3. Comparison of Algorithms for 4 column up-downdatings

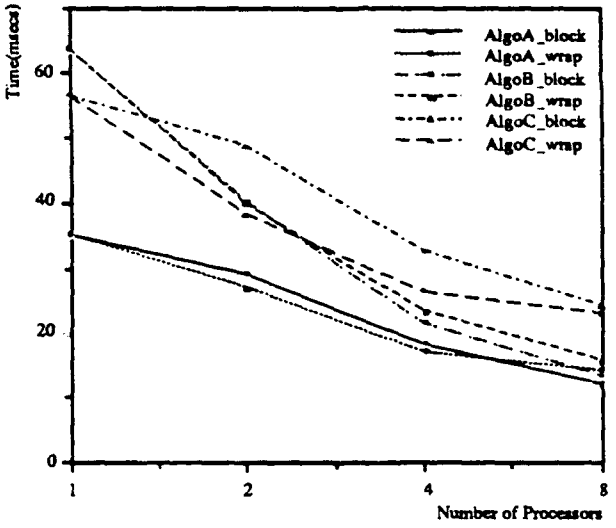


Figure 9.4. 2 column up-downdatings without buffer for $n = 32$

the link speed is 10 Mbits/sec:

$$\frac{T'_{comm}}{T_{comp}} \simeq \frac{14.380usec}{2.120usec} \simeq 6.783.$$

Comparisons of elapsed times for the algorithms when the link speed is 10 Mbits/sec are presented in Figure 9.5, which is compared with Figure 9.3b. The time difference between the wrap mappings of the Algorithms A and C using 2 processors becomes slightly larger than that of Figure 9.3b. The block mapping of Algorithm A has better performance than the wrap mapping using 8 processors, since the wrap mapping requires more communication.

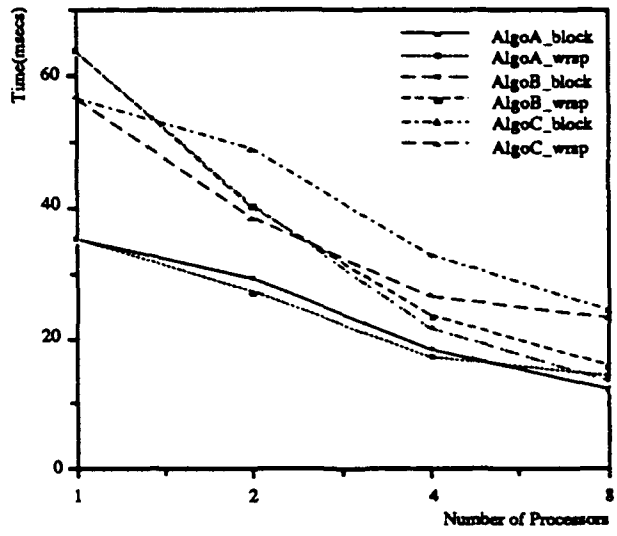


Figure 9.5. 1 column up-downdatings with 10 Mbits/sec for $n = 64$

10. Future Work

We implemented the three algorithms, measured their elapsed time and compared the performance results on the transputer array. The future work for the time analyses of the algorithms will be presented at a later date. The propagation and truncation errors resulting from the different algorithms will be monitored and analyzed. Finally, the algorithms will be implemented on a two-dimensional transputer array, and the comparison of the time results conducted by the different algorithms will be presented.

REFERENCES

[1] G.H.Golub, C.F.Van Loan, "Matrix Computations," The Johns Hopkins University Press, 1989.

[2] Inmos, "The Transputer Databook," Inmos, 1989.

[3] Inmos, "Communication Process Architecture," Prectice Hall, 1988.

[4] G.Li, T.F.Coleman, "A Parallel Triangular Solver for a Distributed-Memory Multiprocessor," SIAM J. of Sci. Stat. Computing, Vol. 9, No 3, pp. 485-502.

[5] M.T.Heath, C.H.Romine, "Parallel Solution of Triangular Systems on Distributed-Memory Multiprocessors," SIAM J. of Sci. Stat. Computing, Vol. 9, No 3, pp. 558-588.

[6] A.W.Bojanczyk, A.O.Steinhardt, "Matrix Downdating Techniques for Signal Processing," SPIE Vol. 975, 1988.

[7] G.A Geist, M.T.Heath, "Matrix Factorization on a Hypercube Multiprocessor," Hypercube Multiprocessors 1986, Society for Industrial abd Applied Mathmatics, Philadelpia, 1986, pp161-180.

[8] D.Heller, "A Survey of Parallel Algorithms in Numerical Linear Algebra," SIAM Review, 20(1978), pp. 740-777.

[9] R.W.Hockney, C.R.Jesshope "Parallel Computers 2," Adam Hilger Press, 1988.

[10] N-K.Tsao "A Note on implementing the Householder Transformation," SIAM J. of Numerical Analysis, Vol.12, No.1, 1975, pp53-58.

The Hyperbolic Singular Value Decomposition and Applications[1]

Ruth Onn, Allan O. Steinhardt, and Adam Bojanczyk

Department of Electrical Engineering
Phillips Hall, Cornell University, Ithaca, NY 14853-5401

Abstract

A new generalization of the singular value decomposition (SVD), the *hyperbolic SVD*, is advanced, and its existence established under mild restrictions. The hyperbolic SVD accurately and efficiently finds the eigenstructure of any matrix that is expressed as the difference of two matrix outer products. Signal processing applications where this task arises include the covariance differencing algorithm for bearing estimation in sensor arrays, sliding rectangular windowing, and array calibration. Two algorithms for effecting this decomposition are detailed. One is sequential and follows a similar pattern to the sequential bidiagonal based SVD algorithm. The other is for parallel implementation and mimics Hestenes' SVD algorithm. Numerical examples demonstrate that, like its conventional counterpart, the hyperbolic SVD exhibits superior numerical behavior relative to explicit formation and solution of the normal equations. Furthermore the hyperbolic SVD applies in problems where the conventional SVD cannot be employed.

1. INTRODUCTION

The singular value decomposition (SVD) is a tool of both practical and theoretical importance in digital signal processing. The SVD of an $n \times m$ complex valued matrix A is given by [8]:

$$A = USV^{\dagger},$$

where S is an $n \times m$ diagonal matrix with non-negative diagonal (with the entries ordered from largest to smallest), U is an $n \times n$ unitary matrix[2], V is an $m \times m$ unitary matrix and † denotes Hermitian conjugation in case of a complex valued matrix and simple

[1] A more comprehensive description of the algorithmic aspects of this decomposition can be found in "The Hyperbolic Singular Value Decomposition and Applications" to be published in the IEEE ASSP.

[2] A square matrix U is unitary if it satisfies $UU^{\dagger} = U^{\dagger}U = I$ the identity matrix.

transposition in case of a real valued matrix.

The SVD provides a solution to the following problem:

> **P1** Given a matrix A, find the eigenvalues and eigenvectors of $AA^\dagger$.

The SVD can thus be interpreted as follows: the eigenvalues of $AA^\dagger$ are the squares of the entries of S, (note that all the eigenvalues of $AA^\dagger$ are real and non-negative), and the columns of U are the corresponding eigenvectors. We get the eigenvectors of $A^\dagger A$ "for free" in the columns of V. For numerical reasons it proves more accurate to not explicitly form the product $AA^\dagger$, but rather to perform an algorithm on the data matrix A directly[3] [8]. The outer product $AA^\dagger$ arises in the normal equations encountered (among other places) in various estimation and adaptive filtering problems. Further details on the normal equations and the role of the SVD in its analysis and solution are found in [11], and in [8].

Consider now the following (related) problem:

> **P2** Given two $n \times m$ matrices A_1, A_2 find the eigenvectors and eigenvalues of $A_1\, A_1^\dagger - A_2\, A_2^\dagger$.

We would like to solve **P2** *without* forming the outer products and subtracting them explicitly. To this end we introduce a new generalization of the ordinary SVD which we call the *hyperbolic* singular value decomposition or HSVD. Just as the ordinary SVD was initially "designed" to allow efficient solution of **P1**, our HSVD is designed to efficiently solve **P2**. Again the primary motivation is numerical accuracy. In addition, as we shall see, the HSVD can be implemented on a parallel machine. The HSVD actually solves the following slightly more general version of **P2**:

> **P3** Given a matrix A and a matrix Φ that is diagonal with ± 1 on the diagonal, find the eigenvalues and the eigenvectors of $A\Phi A^\dagger$.

P2 is indeed a special case of **P3**, which is easily made evident by setting:

$$A = \begin{bmatrix} A_1 & A_2 \end{bmatrix}$$

$$\Phi = \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix}.$$

---

[3] It is not advisable to form outer products of matrices explicitly. Suppose we are working in fixed point, with an accuracy of three digits after the decimal point, and

$$A = \begin{bmatrix} 5.015 & -4.995 \\ -4.995 & 5.015 \end{bmatrix}.$$

Then the eigenvalues of A are: {10.01 and .02} and A is not singular. Square A to get:

$$AA^\dagger = \begin{bmatrix} 50.1002 & -50.0998 \\ -50.0998 & 50.1002 \end{bmatrix}.$$

But with the limited (three digit) accuracy:

$$AA^\dagger \approx \begin{bmatrix} 50.100 & -50.100 \\ -50.100 & 50.100 \end{bmatrix}.$$

Thus the computed $AA^\dagger$ is singular, while the unlimited accuracy one isn't. The singular values, computed as the square roots of the eigenvalues of $AA^\dagger$ will of course be, not {10.01 and .02}, but {10.01 and 0}.

Note also that **P1** is likewise a special case of **P3** associated with $\Phi = I$. The problem **P2** comes up in at least three distinct physical scenarios. One is the downdating problem, another (the one that initially caught our interest) is the so-called covariance differencing problem, and a third is array calibration. Note that the data in these problems can be complex [11], and therefore that the solution must work for both real and complex data.

In downdating, an estimation or filtering problem (the solution of which involves the SVD) is solved for a string of data. Anticipating subtle changes in the data and in the solution, the older data is expected to be outdated, and should be discarded[4]. The columns of A_2 in **P2** are then a subset of the columns of A_1, and the resulting difference is actually non-negative definite. This simplifies the problem somewhat, and indeed there are two efficient algorithms for solving the problem by serial computation. One approach is to form the Cholesky factor of the covariance matrix using the hyperbolic Householder scheme [20] and then proceed using the conventional SVD. The second approach is the one described in [13] which is tailored to single column downdating. In contrast, in the event that a parallel computer is used the HSVD algorithm is the only one we know of for this problem.

The covariance differencing problem arises in high resolution bearing estimation in unknown noise fields [1], [14], [16], [18], [19], [24], [25], [26], [27]. Bearing estimation in unknown coherent noise is a topic which has received a great deal of attention recently, as evidenced by the large number of references cited above. The failure of standard eigenstructure schemes (such a MUSIC, ESPRIT etc.) in the presence of coherent noise fields was demonstrated by Bienvenue in 1979 [1]. A solution, based on covariance differencing, was presented by Paulraj and Kailath [16] in 1986, and independently by Tuteur and Rockah [24]. Since then, variants have been proposed [14], [18], and order determination and performance analysis issues addressed [25], [26], [27]. A general overview of the bearing estimation problem is available in [15].

The key idea behind this bearing estimation scheme is as follows. Suppose that the noise remains the same (in statistical average) between A_1, and A_2, but that the signals of interest change. Then the covariance difference will contain only a residual amount of noise, but the signals will still be present. Details as to how the two snapshot matrices A_1, A_2 are formed, and how parameter estimates are found from the eigenstructure of the difference covariance, are found in the above references.

Despite the interest in covariance differencing for coherent noise bearing estimation, no (prior) published work has addressed computational issues.

A related problem is array calibration. Suppose we have an array which (due to component errors, or other physical considerations) has an unknown array response. We are hampered in our efforts to measure the response by ambient noise. An (offline) solution, proposed in [23], is as follows. First form a covariance matrix from the array with no sources present. Then turn on a calibration source at bearing angle θ and frequency ω, and form the new array covariance. Next find the principle eigenvector of the covariance difference. This eigenvector should be an accurate estimate of the array response at the specified ω, θ co-ordinates.

[4] In the adaptive filtering community this procedure is often called sliding rectangular windowing [11].

The structure of this paper is as follows. The hyperbolic decomposition is introduced in §2.. We also establish that the HSVD is canonic in the sense that it exists subject only to mild restrictions on the attendant A and Φ matrices. In sections §3. and §4. we describe both sequential and parallel algorithms for implementing the HSVD. A simple numerical example of the sequential algorithm is detailed in §5., where we also explore the numerical behavior of the HSVD using numerical experiments. We use simulations from a realistic covariance differencing based bearing estimation application as well as a more artificial example selected for its analytic tractability. Both cases demonstrate the anticipated superior numerical behavior of both sequential and parallel HSVD as contrasted to direct outer product formation.

2. THE HYPERBOLIC SVD

We now state and prove the fundamental existence theorem for the hyperbolic SVD.

> Theorem: Let Φ be an $m \times m$ diagonal matrix, with entries ± 1 and let A be an $n \times m$ matrix, such that $A\Phi A^\dagger$ is non degenerate in the sense that $rank(A\Phi A^\dagger) = min\{n,m\}$[5]. Then there exists an $n \times n$ unitary matrix U, and an $m \times m$ matrix V with
>
> $$V^\dagger \Phi V = \hat{\Phi} \tag{1}$$
>
> where $\hat{\Phi}$ is a diagonal matrix with entries ± 1 (possibly different from Φ), and an $n \times m$ diagonal matrix D with positive real diagonal entries, such that
>
> $$A = UDV^\dagger. \tag{2}$$

Following the nomenclature in [20] we call a matrix W satisfying $W^\dagger \Phi W = \Phi$ a *hypernormal* matrix. A matrix satisfying the more general condition in (1) will be called a *hyperexchange* matrix. Note that a hyperexchange matrix and a hypernormal matrix are related by the equation $WP = V$ where P is a permutation matrix such that $P\Phi P^\dagger = \hat{\Phi}$. Note also that "hyperexchangivity" is partially preserved under multiplication, in particular, if $V^{(1)}$ is hyperexchange with respect to $\Phi^{(1)}$, such that $V^{(1)\dagger}\Phi^{(1)}V^{(1)} = \Phi^{(2)}$, and if $V^{(2)}$ is hyperexchange with respect to $\Phi^{(2)}$, such that $V^{(2)\dagger}\Phi^{(2)}V^{(2)} = \Phi^{(3)}$, then $V^{(2)\dagger}V^{(1)\dagger}\Phi^{(1)}V^{(1)}V^{(2)} = \Phi^{(3)}$, and $V^{(1)}V^{(2)}$ is obviously hyperexchange with respect to $\Phi^{(1)}$.

As mentioned above we call (2) the *hyperbolic singular value decomposition* of A. As a special case when $\Phi = I$ we obtain the ordinary SVD. For real data the proof remains valid, and the Hermitian operator † reverts to the ordinary transpose operator.

The columns of the matrix U of the hyperbolic SVD of A are the eigenvectors we seek, and the diagonal entries of $D\hat{\Phi}D^\dagger$ are the eigenvalues we seek.

[5]This result can be extended for rank deficient matrices. However, though this extension is not trivial, it is mostly of theoretical value, as "real world" matrices are not rank deficient (due to noise at least). For details see [22].

It would be interesting at this point to note some of the properties of the hyperexchange matrices.

- The hyperexchange matrix always has an inverse. This inverse bears an interesting relation to the matrix's Hermitian transpose. To see this consider the hyperexchange matrix V in (1), and multiply both sides of the equation by the matrix $\hat{\Phi}$, to obtain,

  $$\hat{\Phi}V^{\dagger}\Phi V = I.$$

  It is directly apparent that the inverse of V (which is unique) is given by,

  $$V^{-1} = \hat{\Phi}V^{\dagger}\Phi, \tag{3}$$

  and that it always exists.

- The singular values of a hyperexchange matrix have an interesting structure. The singular values come in pairs of reciprocals, i.e., if σ is a singular value of a hyperexchange matrix V, then so is $1/\sigma$. This is a direct consequence of (3) because while the singular values of reciprocal matrices are reciprocals of each other, those of transpose matrices are identical (to within the number of zeros, which is irrelevant here), and by (3) it can be seen that V^{-1} and $V^{\dagger}$ have the same singular values.

- The eigenvalues of a hyperexchange matrix don't in general have any structure, but if it is a hypernormal matrix (recall that means that $\hat{\Phi}$ and Φ in (1) are equal), then *the eigen-structure is similar to the singular-structure*. More precisely, if λ is an eigenvalue of V, then so is $1/\lambda^{*}$. This property was already discussed in [20], where it was also proven, but it can also be easily seen from (3), for if $\hat{\Phi}$ and Φ are equal, then $V^{\dagger}$ and V^{-1} are orthogonally similar, and have the same eigenvalues.

An existence proof of the hyperbolic singular value decomposition follows. It is not a straight forward extension of the existence proof of the SVD. Rather it has its own interesting features.

Proof of the Hyperbolic SVD Existence Theorem: For (2) to hold U must satisfy:

$$A\Phi A^{\dagger} = UD\hat{\Phi}D^{\dagger}U \tag{4}$$

We can find such a U by diagonalizing the left hand side of (4), i.e. U is a unitary matrix such that:

$$U^{\dagger}(A\Phi A^{\dagger})U = \Lambda \tag{5}$$

where Λ is a diagonal matrix whose entries are eigenvalues of the Hermitian matrix $A\Phi A^\dagger$. U always exists since Hermitian[6] matrices are always diagonalizable by unitary transforms [8]. Let the elements of Λ be ordered,

$$|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_n|$$

If $m \leq n$ $|\lambda_{m+1}| = \cdots = |\lambda_n| = 0$. Let

$$D = \begin{bmatrix} |\lambda_1|^{1/2} & & & \\ & |\lambda_2|^{1/2} & & \\ & & \ddots & \\ & & & |\lambda_m|^{1/2} \end{bmatrix},$$

and

$$\hat{\Phi} = \begin{bmatrix} sgn(\lambda_1) & & & \\ & sgn(\lambda_2) & & \\ & & \ddots & \\ & & & sgn(\lambda_m) \end{bmatrix}, \text{ where } sgn(x) \equiv \begin{cases} \frac{x}{|x|} & x \neq 0 \\ 0 & x = 0 \end{cases}.$$

Then if $V = A^\dagger U D^{-1}$, $A = UDV^\dagger$ and

$$V^\dagger \Phi V = D^{-1} U^\dagger A \Phi A^\dagger U D^{-1} = \hat{\Phi}.$$

This completes the proof for the case $m \leq n$.

If $m > n$ let

$$\tilde{D} = \begin{bmatrix} |\lambda_1|^{1/2} & & & \\ & |\lambda_2|^{1/2} & & \\ & & \ddots & \\ & & & |\lambda_n|^{1/2} \end{bmatrix},$$

and

$$\tilde{\Phi} = \begin{bmatrix} sgn(\lambda_1) & & & \\ & sgn(\lambda_2) & & \\ & & \ddots & \\ & & & sgn(\lambda_n) \end{bmatrix},$$

so $\Lambda = \tilde{D}\tilde{\Phi}\tilde{D}$. Let

$$D = \begin{bmatrix} \tilde{D} & 0 \end{bmatrix}. \tag{6}$$

[6] A matrix M is Hermitian if it satisfies $M^\dagger = M$, for a real matrix that is equivalent to M being symmetric.

To establish (2) it suffices to show that for some hyperexchange matrix V,

$$A^\dagger A = VD^\dagger U^\dagger UDV^\dagger = V \begin{bmatrix} \tilde{D}^2 & 0 \\ 0 & 0 \end{bmatrix} V^\dagger. \tag{7}$$

Let us partition $V = \begin{bmatrix} \tilde{V} & \bar{V} \end{bmatrix}$, and set

$$\tilde{V} \equiv A^\dagger U \tilde{D} \tag{8}$$

then (7) is satisfied.

Eq (1) can be restated in the following manner:

$$\begin{array}{ll} (a) & \tilde{V}^\dagger \Phi \tilde{V} = \tilde{\Phi} \quad , \\ (b) & \bar{V}^\dagger \Phi \tilde{V} = 0 \quad , \\ (c) & \bar{V}^\dagger \Phi \bar{V} = \bar{\Phi} \quad , \end{array} \tag{9}$$

where $\tilde{\Phi}$, $\bar{\Phi}$ and $\hat{\Phi}$ are related by:

$$\hat{\Phi} = \begin{bmatrix} \tilde{\Phi} & 0 \\ 0 & \bar{\Phi} \end{bmatrix}.$$

(9)(a) follows directly from (4) (6) and (8).

Note that since by assumption $A\Phi A^\dagger$ is full rank (when $m > n$,) so is A. By the way in which $\tilde{V}$ is defined in (8) both $\tilde{V}$ and $\Phi\tilde{V}$ are full rank. If $\bar{V}$ is chosen to lie in the exactly $m-n$ dimensional subspace orthogonal to $\Phi\tilde{V}$, i.e. $\bar{V} \in (\Phi\tilde{V})^\perp$ then (9)(b) is satisfied.

Let Q be an orthogonal matrix that spans $(\Phi\tilde{V})^\perp$, then $\bar{V}$ can be decomposed as $\bar{V} = QX$ with X some $m \times m$ matrix. To assure satisfaction of (9)(c), all we must show is that a matrix X exists, so that $X^\dagger Q^\dagger \Phi Q X = \bar{\Phi}$. This rests on the fact that $Q^\dagger \Phi Q$ is non-singular, which we prove by contradiction[7].

Suppose that $Q^\dagger \Phi Q$ were singular, then a vector $y \neq 0$ would exist such that $Q^\dagger \Phi Q y = 0$. This implies that $\Phi Q y$ lies in the space orthogonal to Q, i.e., in the space spanned by $\Phi\tilde{V}$. Thus there exists a vector $z \neq 0$ so that

$$\Phi Q y = \Phi \tilde{V} z.$$

Multiply both sides on the left by $\tilde{V}^\dagger$ to get,

$$\tilde{V}^\dagger \Phi Q y = \tilde{V}^\dagger \Phi \tilde{V} z. \tag{10}$$

The left hand side of (10) is clearly zero, which implies that $z = 0$ as well, which is a contradiction.

[7] An alternate proof can be found in [7] where Φ norms are studied as a particular case of indefinite scalar products.

Having established that $Q^\dagger \Phi Q$ is nonsingular, we can diagonalize it by an orthogonal matrix Y:

$$Y^\dagger Q^\dagger \Phi Q Y = \Delta.$$

Δ of course has no zero diagonal entries, and we can therefore let

$$\bar{D} = \begin{bmatrix} |\delta_1|^{1/2} & & & \\ & |\delta_2|^{1/2} & & \\ & & \ddots & \\ & & & |\delta_m|^{1/2} \end{bmatrix},$$

and

$$\bar{\Phi} = \begin{bmatrix} sgn(\delta_1) & & & \\ & sgn(\delta_2) & & \\ & & \ddots & \\ & & & sgn(\delta_m) \end{bmatrix},$$

and $\Delta = \bar{D}\bar{\Phi}\bar{D}$. We choose $X = YD^{-1}$, and $\bar{V} = QX$. Then,

$$\bar{V}^\dagger \Phi \bar{V} = D^{-1} Y^\dagger Q^\dagger \Phi Q Y D^{-1} = \bar{\Phi},$$

which completes the proof of the HSVD existence theorem. ■

3. A SEQUENTIAL ALGORITHM FOR HYPERBOLIC SVD

The serial hyperbolic SVD algorithm, outlined below, mimics the classic SVD procedure [8]. The main difference is in the use of the hyperbolic Householder transform described in [20] and the hyperbolic Givens rotation, described in [3]. Both are here further perfected for the matter at hand. The main difference between the transforms and rotations described previously and the ones used (and described) here, is in the concept of the exchange. Thus if (for reasons that become apparent later) a transform or a rotation cannot be completed as expected, the elements of the matrices involved are exchanged in a way that preserves pertinent properties, but enables the algorithm to proceed.

3.1. The Hyperbolic Householder Transform and the Hyperbolic Givens Rotation

The original Householder transformation [10] of a (column) vector v involves finding an orthonormal matrix M so that

$$Mv = \pm \| v \| e_1, \tag{11}$$

where $\| v \| = \sqrt{v^\dagger v}$ and e_1 is the unit vector with the first element one and all the rest zeros. This can be viewed as *compressing* all the vector's energy into the first entry. It

is easily verified that a matrix M given by: $M = I - 2bb^\dagger/b^\dagger b$ where $b = v \mp \| v \| e_1$ satisfies (11). The hyperbolic Householder transform will take on a similar form, and a signature matrix Φ has to be specified as well as the vector v. We will then expect H to be hypernormal (see definition after equation (2)), and $Hv = \pm \| v \|_\Phi e_1$ where,

$$\| v \|_\Phi \equiv sgn(v^\dagger \Phi v) \sqrt{|v^\dagger \Phi v|}. \tag{12}$$

Note that despite the notation $\| v \|_\Phi$ is not a norm, because norms are always non-negative.

The natural thing to try would be to let $b = v \mp \| v \|_\Phi e_1$ and $H = \Phi - 2bb^\dagger/b^\dagger \Phi b$. This was done in [20] and H was defined only for v's and Φ's for which $v^\dagger \Phi v \geq 0$ and thus $\| v \|_\Phi$ is well defined. But in order to be useful for HSVD a matrix H should be obtainable for *any* pair Φ and v.

In order to achieve this generality, look more carefully into what actually happens when a vector is transformed. If v denotes the original vector and $\hat{v}$ denotes the transformed vector, then we expect two things:

- $\hat{v}^\dagger \Phi \hat{v} = v^\dagger \Phi v$, and
- $\hat{v}$ has the form $\pm \| v \|_\Phi e_1$, this can be viewed as compressing all its ***hyperbolic "energy"*** into its first entry.

It turns out that the two conditions cannot generally be met simultaneously, because the sign of
$(\| v \|_\Phi e_1)^\dagger \Phi \| v \|_\Phi e_1 = \| v \|_\Phi^2 \Phi(1,1)$ is determined by $sgn(\Phi(1,1))$ and is independent of the sign of $\| v \|_\Phi$. Suppose for the moment that $\| v \|_\Phi \neq 0$, then a solution springs to mind: why insist on e_1? Why not choose another unit vector e_k, or in other words, why not try to compress the hyperbolic "energy" into the k^{th} entry, where $\Phi(k,k)$ is the "right" sign, i.e., it equals -1 if $\| v \|_\Phi < 0$ and $+1$ otherwise. Alternately, permute Φ and v, exchanging the k^{th} and the 1^{st} entries, in the manner described in the pseudo computer code below:

if $sgn(\| v \|_\Phi) \neq \Phi(1,1)$
  then
    find a k so that $sgn(\| v \|_\Phi) = \Phi(k,k)$
    permute entries 1 and k in v, and entries $(1,1)$ and (k,k) in Φ
end if
$b := v \mp \sqrt{|\| v \|_\Phi|} e_1$
$H = \Phi - 2\frac{bb^\dagger}{b^\dagger \Phi b}$

The resulting matrix H is a hyperexchange matrix with respect to Φ. This hyperbolic Householder scheme is used mainly in the introductory part of bidiagonalizing the data matrix[8], see subsection 3.2..

For use in the main part of the serial HSVD scheme another hyperbolic extension of a classic algorithm will be introduced, it is the hyperbolic Givens Rotation. The original

[8]This extension of the hyperbolic Householder transform for nonpositive normed vectors was also developed (independently) by Cybenko [5] in a different context.

Givens rotation, [8] is a two by two matrix that is similar to the Householder transformation for a length two vector. Let v be a length two real column vector $v^\dagger = [v_1\ v_2]$, and let

$$G = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}, \tag{13}$$

where, $s = v_2/\sqrt{v_1^2 + v_2^2}$, and $c = v_1/\sqrt{v_1^2 + v_2^2}$. Then $Gv = \| v \| [1\ 0]^\dagger$, and $GG^\dagger = I$, or otherwise phrased $(Gv)^\dagger(Gv) = v^\dagger v$ and Gv is proportional to e_1. Now suppose we are interested in finding a matrix H so that Hv is proportional to e_1, and $(Hv)^\dagger\hat{\Phi}(Hv) = v^\dagger\Phi v$, where $\hat{\Phi}$ and Φ equal $\pm\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$. Again, as in the hyperbolic Householder transform, we have to allow $\hat{\Phi} \neq \Phi$. In order to achieve that we note that s and c in (13) are the cosine and sine of $tan^{-1}(v_2/v_1)$. The hyperbolic Givens rotation will naturally use the hyperbolic trigonometric functions *sinh* and *cosh*. Thus, given $v^\dagger = [v_1\ v_2]$ and $\Phi = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$, and supposing $\| v \|_\Phi \neq 0$, the required matrix will be given by:

$$H = \begin{bmatrix} c_h & \sigma s_h \\ -\sigma s_h & c_h \end{bmatrix} \tag{14}$$

with

$s_h = v_2/\sqrt{v_1^2 - v_2^2}$, $c_h = v_1/\sqrt{v_1^2 - v_2^2}$, and $\sigma = sgn(\| v \|_\Phi\ \Phi(1,1))$.

Note that s_h and c_h are indeed the hyperbolic sine and cosine of $\tanh^{-1}(v_2/v_1)$. Note also that although all the references above were to column vectors, and that the matrices were employed from the left, the above is trivially translated to row vectors with matrices employed on the right.

At this point a problem that is common both to the hyperbolic Householder and to the hyperbolic Givens algorithms should be addressed: what happens when $\| v \|_\Phi = 0$? The answer is that both procedures per se fail (see [5] for some implications of this problem). But when put in the context of the whole HSVD algorithm a mechanism for recovery exists, and will be described subsequently.

What we rely upon in recovering from a situation of $\| v \|_\Phi = 0$ is that the hyperbolic Householder and the hyperbolic Givens are applied to whole matrices, not merely to isolated column or row vectors, and that while hyperbolic transformations are applied to the columns orthogonal transformations may be performed on the rows. To demonstrate the recovery suppose we attempt to transform A into a bidiagonal matrix using an orthogonal matrix on the left and hyperbolic Householder transformations on the right (the overall HSVD procedure is composed of just such tasks). We start by attempting to "squeeze" all of $a_1^\dagger$ into its first element. We employ the hyperbolic Householder procedure, but to our chagrin discover that $\| a_1 \|_\Phi = 0$. A solution can be found by using $a_2^\dagger$ in the following way:

If we choose an angle θ and let $c = cos\theta$ and $s = sin\theta$ then the block diagonal matrix $Q = \begin{bmatrix} c & -s & \\ s & c & \\ & & I \end{bmatrix}$ is orthogonal. Its affect on A is as follows:

$$\tilde{A} = \begin{bmatrix} \tilde{a}_1^\dagger \\ \tilde{a}_2^\dagger \\ \tilde{a}_3^\dagger \\ \vdots \\ \tilde{a}_n^\dagger \end{bmatrix} \equiv Q^\dagger A = \begin{bmatrix} ca_1^\dagger + sa_2^\dagger \\ ca_2^\dagger - sa_1^\dagger \\ a_3^\dagger \\ \vdots \\ a_n^\dagger \end{bmatrix}.$$

Now calculate the Φ-norm squared of $\tilde{a}_1$:

$$| \| \tilde{a}_1 \|_\Phi |^2 = |\tilde{a}_1^\dagger \Phi \tilde{a}_1| = |s^2 a_2^\dagger \Phi a_2 + 2cs Re(a_1^\dagger \Phi a_2)|.$$

Since $a_2^\dagger \Phi a_2$ and $Re(a_1^\dagger \Phi a_2)$ are just real numbers, we can find θ such that $\| \tilde{a}_1 \|_\Phi \equiv sgn(\tilde{a}_1^\dagger \Phi \tilde{a}_1)\sqrt{|\tilde{a}_1^\dagger \Phi \tilde{a}_1|} \neq 0$ thereby recovering from the initial "trap" of the vanishing norm. The choice of θ is rather unlimited (so long as the new norm isn't zero as well), but we might for instance choose to maximize the resulting norm.

3.2. The Serial HSVD Algorithm

The best available serial algorithm for the full SVD is the bidiagonalization based scheme due to Golub and Kahan [8]. Our serial HSVD algorithm borrows heavily from Golub and Kahan's original algorithm. We assume that the reader is familiar with this algorithm, (to which we will refer as the "classical" algorithm,) and will just note where our differs.

The first step is to bidiagonalize the matrix A, i.e., find unitary matrices U and V so the $UAV^\dagger$ is bidiagonal. This is classically achieved through the use of Householder matrices. Our algorithm requires a unitary matrix on the left, but a hyperexchange matrix on the right, we use Householder and hyperbolic Householder matrices respectively to build these matrices. We will call the bidiagonal matrix "B". If the matrix A was not square, B will have columns of zeros. If only the eigenvalues and eigenvectors of $A\Phi A^\dagger$ are of interest, not the HSVD of A per se[9], $A\Phi A^\dagger$ can be summarized as $\bar{B}\bar{\Phi}\bar{B}^\dagger$ where $\bar{B}$ is the matrix composed of the first n rows of B, and $\bar{\Phi}$ is the $n \times n$ top left hand corner of Φ.

We note that the matrix B (or $\bar{B}$) can now be made into a real matrix with diagonal unitary (and thus also hypernormal) matrices on the right and on the left. Once this is achieved we proceed with a hyperbolic version of the QR algorithm, which we call the hyperbolic QR algorithm. It differs again from the classical version by using unitary matrices on the left and hyperexchange matrices on the right.

An important point is that the proofs of the theorems used, the implicit Q theorem and the convergence of the QR algorithm, do not depend on the positive definiteness of the composite matrix ($AA^\dagger$), nor on how it was formed. Thus they are applicable for $A\Phi A^\dagger$ as well.

[9] As is true for all applications considered by us.

4. A SYSTOLIC ALGORITHM FOR HYPERBOLIC SVD

Our parallel algorithm for the hyperbolic SVD is an adaptation of a well known biorthogonalization technique developed by Hestenes [9] for computing the singular value decomposition of an $n \times m$ matrix A. The method is known as a one-sided Jacobi method.

The one-sided (Hestenes) method can be modified to carry out the singular value decomposition of a matrix, $A = U\Sigma V^\dagger$. The technique finds a unitary matrix U such that $U^\dagger A = \Sigma V^\dagger$ has orthogonal columns, i.e., $U^\dagger A$ has orthogonal rows.

If we insist that $U^\dagger A$ be hypernormal with respect to the matrix Φ, then $(U^\dagger A)\Phi(U^\dagger A)^\dagger$ will give the eigenvalues of $A\Phi A^\dagger$, the precise quantities we were interested in in the first place. The process of finding V is iterative and proceeds by constructing a sequence of matrices $A^{(k)}, k = 0, 1, \ldots$, $A^{(k+1)} = J^{(k)}A^{(k)}$, here $A^{(0)} = A$ and $J^{(k)}$, $k = 0, 1, \ldots$, are plane rotations operating on pairs of rows of A. Angles of rotations are chosen in such a way that, for a single rotation, the resulting rows become hypernormal. By applying rotations to all different pairs of rows in a sweep, and iterating the sweeps, the limit matrix $\lim_{k\to\infty} A^{(k)}$ becomes hypernormal.

Parallel implementations of the algorithm are based on earlier works on one-sided Jacobi methods [2]. The key observation is that as long as rotations operate in different planes, they are independent and can be executed by different processing units all at the same time; such a simultaneous transformation is called a parallel rotation. The success of this approach heavily depends on the ordering of rotations in a sweep. There are various strategies for choosing the order of rotations, or pivot rows, in the sweep. The choice is influenced by the communication geometry directly supported by the target computer architecture. In several studies it has been shown that there exist strategies which are well suited for linear array, ring and hypercube topologies. It turns out that one can essentially mimic the Hestenes sweep selection process for hyperbolic rotations.

5. SIMULATION RESULTS

We coded both the sequential and the linear systolic array versions of the hyperbolic SVD scheme and applied them to the square root covariance differencing task. The primary purpose of simulations is to explore the numerical accuracy of our new algorithms, as compared to direct, explicit, formation of covariances matrices followed by differencing, and finally eigenanalysis. We shall refer to this latter approach as the "power domain" scheme, since it demands the explicit formation, storage, and processing of squared-data, that is power-like, quantities.

We conducted two numerical experiments. The first of these involved the example furnished in the original paper on covariance differencing by Paulraj and Kailath [16]. There is little point in detailing this example here, the interested reader can consult [16] for specifics. We used 8 sensors, 3 signals at $-30°$, $-12°$, and $-15°$, and $50°$ of rotation. We retained the same noise field as in [16]. However, to better differentiate numerical errors from statistical fluctuations, we increased the number of snapshots from 300 to one thousand giving matrices of size 8×1000. (Spot checks for 300 snapshots assured us that

this had no effect on the relative stability of either method.) When the signal power levels used were as per [16] the problem is well conditioned, and both the systolic and the power domain schemes work well with a 64-bit floating point number system[10]. To produce ill conditioning for the rather extended precision of our computer language we increased the dynamic range of the signals to 80dB, by increasing the first signal's amplitude, and leaving the rest alone. This led to a breakdown in the power domain method, but not in either of the two new methods based on the hyperbolic SVD. More specifically, all three methods accurately identified the first signal. However the next two signals were identified correctly using the HSVD method (with an error of 1° and 4° respectively) while the power domain gave estimates off by 5° and 41°. An error of a few degrees ($\approx 1-4$) is within the range of statistical fluctuations for this estimation task, while an error of 41° intimates numerical collapse.

In the Paulraj/Kailath example exact assessment of numerical errors is impossible since the "exact" answer, furnished by mapping the computations onto an infinite precision computer, is unknown. To avoid this shortcoming the second experiment we conducted involved a covariance difference whose eigenvalues we knew explicitly. This allowed us to determine *exactly* how much numerical error each algorithm produced. The drawback here, in contrast to example one, is that the covariance difference no longer has a nice physical interpretation, and does not (as far as we know) correspond to any known practical application.

To begin we form the n by m matrix $\Psi \equiv [\mathrm{diag}(\lambda_1, ..., \lambda_n)|0]$. If we define the signature matrix Φ via $\Phi \equiv \mathrm{diag}(\{(-1)^i,\ i = 0, ..., n-1\})$ then the eigenvalues of $\Psi\Phi\Psi^\dagger$ are quite clearly $\lambda_1^2, -\lambda_2^2, ..., \lambda_n^2$. Now suppose we multiply Ψ on the left by any $n \times n$ unitary matrix U and on the right by any $m \times m$ matrix V hypernormal with respect to Φ to form the matrix A. Then $A\Phi A^\dagger = U\Psi V\Phi V^\dagger\Psi^\dagger U^\dagger = U\Psi\Phi\Psi^\dagger U^\dagger$ has the same eigenvalues as $\Psi\Phi\Psi^\dagger$, but is now a full matrix. We can readily generate random choices for U and V as products of respectively Householder and hyperbolic Householder transformations. By selecting λ_i we can influence the condition number of the eigenanalysis problem. As in the first example both hyperbolic SVD schemes lead to better numerical behavior for a given condition number and a given wordlength. Some examples are given below.

The preliminary simulations were conducted using MATLAB^{TM} for which relative precision ϵ is 2^{-48}. For a given data matrix $A \equiv U\Psi V_k$ we constructed the corresponding "covariance" matrix $A\Phi A^\dagger$. We chose $\Psi \equiv \mathrm{diag}\ (10^8, 10^4, 1)$, and generated the hypernormal matrix V_k as a product of k, $k = 1, 2, 3, 4, 6$, random hyperbolic Householder matrices. Note that the condition number of $A\Phi A^\dagger$ is 10^{16} which is comparable to the reciprocal of the relative precision used in the computations. We computed the eigenvalues of $A\Phi A^\dagger$ via hyperbolic Hestenes method, on the original data matrix A. Next we repeated the computation via the two-sided Jacobi method which operated on $A\Phi A^\dagger$.

Let us denote the exact eigenvalues of $A\Phi A^\dagger$ as λ_i^E, the computed eigenvalues by Hestenes method as λ_i^H, and by Jacobi method as λ_i^J.

[10]The simulations were conducted on an IBM-PC^{TM} using MATLAB^{TM}. (All data types are 64bits in MATLAB^{TM}.) The power domain scheme used tridiagonalization followed by the QR method [8] to excise the eigenstructure of the covariance difference.

Let $\gamma_i^J \equiv \frac{\lambda_i^E - \lambda_i^J}{\lambda_i^E}$ and $\gamma_i^H \equiv \frac{\lambda_i^E - \lambda_i^H}{\lambda_i^E}$.

We observed, see Table 1 below, that the hyperbolic Hestenes method always gave better approximation of the eigenvalues than the Jacobi method. However, the accuracy of the hyperbolic Hestenes was influenced by the number of terms in the product V_k and varied from simulation to simulation. This can be explained by the loss of accuracy while computing the hyperbolic norm.

Table 1

k	λ	γ^J	γ^H
1	λ_1	10^{-4}	10^{-9}
2		10^{-4}	10^{-8}
3		10^{-3}	10^{-8}
4		10^{-3}	10^{-8}
6		10^{0}	10^{-4}
1	λ_3	10^{-14}	10^{-13}
6		10^{-12}	10^{-12}

We also ran the QR eigenanalysis [8] method on $A\Phi A^\dagger$ and the serial hyperbolic SVD on A. The results did not differ markedly from the ones indicated above, so we decided not to include them in the table above.

6. CONCLUSIONS

We presented an extension of the SVD, which we called the hyperbolic SVD. Its existence was established under mild restrictions. We derived two algorithms for effecting this decomposition, one sequential and the other parallel.

The HSVD is indicated whenever one seeks to evaluate the eigenstructure of a covariance (outer product) matrix provided:

(i) The covariance involves a difference of outer products,

(ii) all (or most) of the eigenvectors and eigenvalues are required,

(iii) numerical ill-conditioning is anticipated (using the available wordlength for a given computer), and

(iv) either a sign indefinite matrix arises or a parallel implementation is required.

We mentioned three signal processing applications of this new canonic decomposition, which satisfy the above conditions, mainly bearing estimation, sliding rectangular windowing, and array calibration.

Numerical experiments demonstrated the enhanced numerical accuracy available using the HSVD in contrast to competing schemes. Theoretical backing for this improved accuracy remains a topic for future investigation.

We feel confident that there are many more applications within and beyond digital signal processing where the HSVD will be useful for its numerical stability, fast computational characteristics, and as a theoretical structure.

Acknowledgements

We would like to thank Paul Van-Dooren for calling reference [7] to our attention.

This work was supported by the NSF under contract MIP-8710835 the Air Force under contract AFOSR-89-0267 and in part by the SDIO, through ARO, under contract number DAA L03-90-G-0092.

References

[1] G. Bienvenu, "Influence of the spatial coherence of the background noise on high resolution passive methods", *Proceedings of the Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP) 1979*, Washington, D.C.

[2] R.P. Brent and F.T. Luk, "The solution of singular value and symmetric eigenvalue problems on multiprocessor arrays", SIAM J. Sci. Stat. Comput., vol 6, p. 69-84, 1985.

[3] J. Chun, T. Kailath, and H. Lev-Ari, "Fast Parallel Algorithms for QR and Triangular Factorizations", *SIAM J. SSC*, Vol. 8 No. 6 Nov. 1987.

[4] T. F. Chan, "An Improved Algorithm for Computing the Singular Value Decomposition", ACM Transactions on Mathematical Software 8, pp 72-93, 1982.

[5] G. Cybenko and M. Berry, "Hyperbolic Householder Algorithms for Factoring Structured Matrices," University of Illinois CSRD Report No.877, Submitted to SIAM J. on Matrix Analysis & Applications, May 1989.

[6] T. Coleman, Class notes, CS621, Cornell University, Fall 1988.

[7] I. Gohberg, P. Lancaster, and L. Rodman, *Matrices and Indefinite Scalar Products*, Birkhauser Verlag, Basel, Switzerland, 1983.

[8] G. Golub and C. Van Loan, *Matrix Computations*, John Hopkins Press, Baltimore, MD, 1983.

[9] M.R. Hestenes, "Inversion of matrices by biorthogonalization and related results", J. Soc. Indust. Appl. Math., vol 6, p. 51-90, 1958.

[10] Alston S. Householder, "Unitary Triangularization of a Non-symmetric Matrix," *J. Ass. Comp. Mach.*, Vol. 5, 1958. *Also found in* A. Householder, *The Theory of Matrices in Numerical Analysis,* Dover, London, 1964.

[11] S. Haykin, *Adaptive Filter Theory*, Prentice-Hall, Englewood Cliffs, New Jersey, 1986.

[12] C. Lawson and R. Hansen, *Solving Least Squares Problems.* Englewood Cliffs, NJ: Prentice Hall, 1974.

[13] Gang Li and Kai-Bor Yu "Adaptive rank-2 Update Algorithm for Eigen Value Decomposition", *ICASSP 88* April 1988, New York, New York pp.1510-1513.

[14] A. Moghaddamjoo, "Transform Based Covariance Differencing Approach to the Estimation of Multiple Signals in a Multi-sensor Environment", 4-th ASSP Workshop on Spectral Estimation and Modeling, U. of Minn., Aug. 1988, p.327-332.

[15] J. Munier and G. Deliste, "Spatial Analysis in Passive Listening Using Adaptive Techniques", Proc. IEEE, Nov. 1987, pp.1458-1471.

[16] A. Paulraj and T. Kailath, "Eigenstructure methods for direction of arrival estimation in the presence of unknown noise fields", *IEEE ASSP*, Feb. 1986.

[17] B. Parlett, "The Symetric Eigenvalue Problem", Prentice-Hall, Englewood Cliffs, New Jersey, 1980.

[18] S. Prasad, R. Williams, A.Mahalanabis, and L.Sibul, "A transform based covariance differencing approach to bearing estimation", *ICASSP 1987*, Dallas, TX.

[19] S. Prasad, R. Williams, A.Mahalanabis, and L.Sibul, "The Covariance Differencing Approach for Parametric Estimation Problems", *IEEE TASSP*, May 1988.

[20] C. Rader and A. Steinhardt, "Hyperbolic Householder Transformations", *IEEE Trans. Acoust., Speech, Signal Proc.*, Dec. 1986.

[21] A. Steinhardt, "Householder Transformations in Signal Processing", invited paper, *IEEE ASSP Magazine*, July, 1988.

[22] R. Onn, A. Steinhardt, and A. Bojanczyk, "Hyperbolic Singular Value Decomposition- A New Canonic Form" (to be submitted to *LAA Special Issue on Canonic Matrix Decompositions*).

[23] J.Pierce, M.Jacobsen, and M. Kaveh, "A Laboratory Testbed for Sensor Array Processing", 32nd Midwest Symp. on Circuits and Systems, Champaign, Ill., Aug. 1989, Conf. Proceedings, to appear.

[24] F.Tuteur and Y.Rockah, "A new method for signal detection and estimation using the eigenstructure of the covariance difference", *ICASSP 1986*, Tokyo, Japan.

[25] F. Tuteur and Y. Rockah, "The Covariance Differencing Method in Signal Detection", *3RD ASSP Workshop on Spectral Estimation*, Boston, MA, 1986.

[26] L. Zhao, P. Krishnaiah, and Z. Bai, "A new Method to Estimate the DOA of Signals", *4-th ASSP Workshop on Spectral Estimation and Modeling*, U. of Minn., Aug. 1988, p.108-110.

[27] L. Zhao, P. Krishnaiah, and Z. Bai, "On Detection of the Number of Signals when the Noise Covariance is Arbitrary", *J. Multivariate Anal.*, Vol. 20, 1988.

Taylor series and the overall properties of composites [1]

Oscar P. Bruno
School of Mathematics, University of Minnesota
206 Church St. SE, Minneapolis, MN, 55455.

Abstract. We deal with the effective electrical conductivity and the effective elastic moduli L^* of multiphase, ordered or disordered composite materials. The problem of estimation of L^* can be approached through perturbation expansions around the uniform case. Interestingly, knowledge of truncated expansions for L^* leads to rigorous bounds on this quantity. We describe a general construction which permits one to find a hierarchy of bounds from truncated expansions for L^*. Our approach is motivated by previous work of Beran and others. These bounds give good estimations for weakly heterogeneous materials, i.e., composites in which the departure from homogeneity is not too strong. For strongly heterogeneous materials, such as conductor–insulator mixtures, these bounds lead to broad intervals of uncertainty. We thus address this problem, and show that tight bounds can be found for the effective conductivity of two–phase ordered or disordered composites, in which material of very large or very low conductivity is arranged in the form of particles inside a conducting matrix. These bounds depend on two parameters A and B which are related to the particle shapes and the interparticle distances. For instance, if the particles are assumed to be spherical, the bounds depend only on a parameter q which is related to the minimum interparticle distance, and they give excellent estimations for a wide range of values of the parameter q. In many cases, they improve substantially over previous estimations for the effective conductivity of this type of composites.

Introduction. We deal with a physical property, elasticity or conductivity, whose constitutive linear relation is

$$f = Le. \tag{1}$$

In the case of conductivity, L is a symmetric tensor of second order which represents the local conductivity, and f and e are the current and electric field respectively. In the case of elasticity, L is the Hookean fourth order tensor, and f and e are the tensors of stress and strain respectively. In this case, products are understood as follows:

$$(Le)_{ij} = L_{ijkl}e_{kl},$$

$$eLe = e_{ij}L_{ijkl}e_{kl}$$

where the usual summation convention has been used.

An r-phase composite material is usually modeled by a spatially varying tensor L, which is locally constant and assumes r different values. Similarly, in a polycrystal,

[1] Supported by the U.S. Army Research Office under grant DAAL-03-88-K-0110

the tensor L is locally constant and its values vary among all rotations of a given fixed tensor L_0. The quantity of interest is then the effective tensor L^* defined by (see e.g. [2,13])

$$\langle f \rangle = L^* \langle e \rangle \tag{2}$$

where $\langle h \rangle$ stands for the average of the quantity h, and f and e are fields of strain and stress, or current and electric field that are compatible with the uniform applied field $\langle e \rangle = e_0$. The averages above should be understood to be ensemble averages if one deals with random materials, or averages over the unit cell if one deals with periodic materials. Of course, all the quantities involved are assumed to be statistically stationary, so that ensemble averages do not depend on the point in space at which they are performed.

The minimum energy principle states (see [2,3]) that the true field e in the composite that is compatible with an uniform applied field e_0 is the one that minimizes the energy

$$\langle e^t L e^t \rangle \tag{3}$$

among all curl free trial fields verifying $\langle e^t \rangle = e_0$. Furthermore, the true energy in the material due to the applied field e_0 is given by

$$\frac{1}{2}\langle eLe \rangle = \frac{1}{2} e_0 L^* e_0. \tag{4}$$

The complementary principle of minimum energy states (see [2,3]) that the true field f in the composite that is compatible with an uniform applied field f_0 is the one that minimizes the expression

$$\langle f^t L^{-1} f^t \rangle \tag{5}$$

among all divergence free trial fields f^t such that $\langle f^t \rangle = f_0$. The true energy in the body due to the applied field is

$$\frac{1}{2}\langle f(L)^{-1} f \rangle = \frac{1}{2} f_0 (L^*)^{-1} f_0. \tag{6}$$

Series expansions. Series expansions are central in the theory of bounds for effective moduli. Typically, the fields e and f and/or the effective modulus L^* are expanded in series of the form

$$e = \sum_{i=0}^{\infty} e_i \tag{7}$$

$$f = \sum_{i=0}^{\infty} f_i \tag{8}$$

$$L^* = \sum_{i=0}^{\infty} L_i^*. \tag{9}$$

The order of smallness of the quantities e_i, f_i and L_i^* is, roughly speaking, the i-th power of the order of smallness of the heterogeneity. One simple way to obtain such expansions is to consider Taylor series. Other kinds of expansions have been considered, which lead to consideration of the n-point correlation function of the microstructure. We will assume, however, that the expansions above are simply the Taylor series of the various quantities. For instance, in the case of conductivity of a mixture of two components of conductivities z and w, we take w as fixed, and consider expansions in powers of $(z-w)$. In the case of elastic moduli of a mixture of two isotropic components of bulk moduli κ_1 and κ_2 and shear moduli μ_1 and μ_2, we take κ_1 and μ_1 as fixed, and consider expansions in powers of $(\mu_2 - \mu_1)$ and $(\kappa_2 - \kappa_1)$.

We note that for all i, e_i is a curl free field and f_i is a divergence free field, and that, for $i \geq 1$ the mean values of e_i and f_i are zero. It follows (see also [8]) that the true fields in a composite verify the equations

$$\langle e_m L e \rangle = 0 \tag{10}$$

and

$$\langle f_m (L)^{-1} f \rangle = 0 \tag{11}$$

for all $m \geq 1$.

Bounds for the effective moduli. To obtain upper bounds on a given effective modulus, we follow Beran (see [2]) in using a truncated expansion together with variational parameters. We choose to consider a Taylor series expansion (7). We substitute the trial field

$$e^t = e_0 + \sum_{i=1}^{k} \alpha_i e_i \tag{12}$$

in the energy expression (3). The α_i's are real constants to be determined. Since (12) is an admissible trial field, we have the upper bound

$$e_0 L^* e_0 \leq \sum_{i,j=0}^{k} \alpha_i \alpha_j \langle e_i L e_j \rangle. \tag{13}$$

where we have put $\alpha_0 = 1$. This is an upper bound on L^* for any choice of the parameters α_i, $i \geq 1$. Finally, minimizing this expression on the parameters α_i, $(i \geq 1)$, yields the Beran type upper bound for L^*.

Of course, this Beran type bound can only be made explicit if the quantities $\langle e_i L e_j \rangle$ are known. Quantities of this type can be computed as integrals of certain statistical

functions: the n-point correlation functions (see, for example, [2] in the case of conductivity of mixtures, [3] in the case of elastic properties of mixtures, [24] in the case of conductivity of polycrystals). These functions are, in general, extremely difficult to determine. For low orders, however, the quantities needed to evaluate the bounds have been computed for cell like mixtures as introduced by Miller [20] (see, for instance [20,11,19]).

The point we wish to establish here is that, because of equations (9), (10) and (11), the quantities $\langle e_i L e_j \rangle$ $(1 \leq i, j \leq k)$ can be simply read off from the first $2k+1$ terms in the Taylor expansion of L^*.

To illustrate the procedure, we derive the first upper bound in the hierarchy. We substitute a trial field of the form

$$e^t = e_0 + \alpha_1 e_1 \tag{14}$$

in the variational principle (3), and obtain

$$e_0 L^* e_0 \leq e_0 \langle L \rangle e_0 + 2\alpha_1 \langle e_0 L' e_1 \rangle + \alpha_1^2 (\langle e_1 L' e_1 \rangle + \langle e_1 L_0 e_1 \rangle), \tag{15}$$

where L_0 is a constant isotropic tensor, and $L' = L - L_0$ represents the oscillation of L about L_0 (Note that $\langle e_0 L_0 e_1 \rangle = 0$ since $\langle e_1 \rangle = 0$). The choice of α_1 that minimizes the right hand side is

$$\alpha_1 = -\frac{\langle e_0 L' e_1 \rangle}{\langle e_1 L' e_1 \rangle + \langle e_1 L_0 e_1 \rangle} \tag{16}$$

Because of equations (9), (10) and (11), one can show that

$$\begin{aligned} \langle e_1 L' e_1 \rangle &= e_0 L_3^* e_0 \\ \langle e_1 L_0 e_1 \rangle &= -e_0 L_2^* e_0, \end{aligned} \tag{17}$$

and

$$\langle e_0 L' e_1 \rangle = -\langle e_1 L_0 e_1 \rangle = e_0 L_2^* e_0. \tag{18}$$

Therefore, the upper bound reads

$$e_0 L^* e_0 \leq e_0 \langle L \rangle e_0 - \frac{(e_0 L_2^* e_0)^2}{e_0 L_3^* e_0 - e_0 L_2^* e_0}. \tag{19}$$

where L_i^* denotes the term of order i in the Taylor expansion of L^*.

An explicit upper bound for different kinds of materials and physical properties can now be obtained simply by substitution into equation (19) of the results for the truncated expansions given in (or easily obtainable from) [1,7,15,24,11]. Lower bounds, and bounds of higher order can be obtained in an entirely analogous manner.

Strongly heterogeneous composites. Here we discuss the problem of determining the effective electrical conductivity of two–phase, isotropic, strongly heterogeneous mixtures. Because the constitutive relation (Ohm's law) is linear, we can assume that the conductivity of one of the components equals one. The conductivity of the second component will be denoted by z, and the effective conductivity will now be denoted by $m = m(z)$.

It is a well known fact that the function m can be extended as an analytic function to complex values of z outside the negative real axis [4,13]. There are examples of materials for which the function m is singular in the whole negative real axis (see [12]).

For ordered or disordered materials which consist in a matrix containing separated inclusions, the zeroes and singularities of the function m cannot accumulate neither at $z = 0$ nor at $z = \infty$. More than that, an explicit interval in the negative real axis can be found depending on the distribution and shapes of the particles, outside which the function m is regular and nonzero [9]. This allows us to find, by means of the complex variable method of Bergman [4,21,13], new bounds on the effective conductivity of mixtures with separated inclusions. (The problem of describing the singularities of the function m is also connected to the properties of absorptance of solar energy exhibited by some ceramic–metal mixtures [10]. Numerical calculations of the singularities for periodic arrays have been conducted [5,6,17,18]. Our results are in agreement with these numerical calculations.)

Consider a microgeometry which consists of a matrix containing a number of randomly placed, separated inclusions. Associated with any such material, there are two constants A and B which determine a region in the negative real axis where the singularities and zeroes of m can occur [9]. The constants A and B depend on the shapes of the inclusions and on the interparticle distances. Assuming, for simplicity, equally sized randomly placed spherical particles, the constants A and B depend on a single parameter q, $0 \leq q \leq 1$, equal to the ratio of the particle diameter to the minimum distance between two centers. Values of A and B for arrays of spheres are given in table 1, for the whole range of values of the parameter q.

The bounds on m depend on the constants A and B. Explicitly, calling p_1 the volume fraction of material of conductivity z, $p_2 = 1 - p_1$, and defining

$$s_m = \frac{A}{1 + A}, \tag{20}$$

$$s_M = \frac{1}{1 + B}, \tag{21}$$

$$\delta = s_M - s_m, \tag{22}$$

$$V(z) = 1 - \frac{p_1}{\frac{1}{1-z} - \frac{p_2}{d}} \tag{23}$$

and

$$W(z) = \frac{1-(1-z)s_M}{1-(1-z)s_m}(1 + \frac{(1-\frac{p_1}{\delta})^2}{(1-\frac{p_1}{\delta})(\frac{1-(1-z)s_M}{(1-z)\delta}) + \frac{p_1}{\delta^2}(\frac{p_2}{d} - s_m)}), \tag{24}$$

our bounds read (see [9])

$$W(z) \leq m(z) \leq V(z) \text{ for } 0 \leq z \leq 1 \tag{25}$$

and

$$V(z) \leq m(z) \leq W(z) \text{ for } z \geq 1. \tag{26}$$

The upper bound in the case $0 \leq z \leq 1$ and the lower bound in the case $z \geq 1$ coincide with the corresponding bounds given by Hashin and Shtrikman [14]. The two remaining bounds are, in many cases, much tighter than the corresponding bounds of Hashin and Shtrikman, reflecting our assumption that the particles cannot touch.

To illustrate numerically our results, we consider a microgeometry formed by spheres of conductivity z coated by a corona of material of conductivity 1 immersed, in a random manner, in a matrix of conductivity 1. This is a random array of closely packed coated spheres. The coated spheres fill about 60 percent of the volume [22]. In figures 1.a–3.b we plot our bounds for this microgeometry. In table 2.a–2.b we give the values of the bounds V and W for the complete range of values of the parameter q. For comparison, we include the values B_l and B_u of the lower and upper bounds obtained, for the same microgeometry, by the method of security spheres (see [16,23]). We thank S. Torquato and J. Rubinstein for communicating their preliminary results to us.

References

[1] Avellaneda, M. and Bruno, O., *Effective conductivity and average polarizability of random polycrystals,* J. Math. Phys. **31** (1990), 2047–2056

[2] Beran, M., *Use of the variational approach to determine bounds for the effective permittivity in random media,* Il Nuovo Cimento **38** (1965), 771–782

[3] Beran, M. and Molyneux, J. *Use of classical variational principles to determine bounds for the effective bulk modulus in heterogeneous media,* Quart. Appl. Math. **24** (1966), 107–118

[4] Bergman, David J. *The dielectric constant of a composite material–A problem in classical physics,* Physics Reports **43** (1978), 378–407

[5] Bergman, D. J., *Dielectric constant of a two-component granular composite: A practical scheme for calculating the pole spectrum,* Physical Review B **19** (1979), 2359–2368

[6] Bergman, D. J., *The dielectric constant of a simple cubic array of identical spheres,* J. Phys. C **12** (1979), 4947–4960

[7] Bruno, O. P., *The effective conductivity of an infinitely interchangeable mixture,* Communications in Pure and Applied Mathematics **43** (1990), 769–807

[8] Bruno, O. P., *Taylor series and bounds for the effective conductivity and the effective elastic moduli of multicomponent composites and polycrystals,* To appear in Asymptotic Analysis

[9] Bruno, O. P., *The effective conductivity of strongly heterogeneous composites,* In preparation

[10] Cohen, R. W., Cody, G. D., Coutts, M. D. and Abeles, B., *Optical properties of granular silver and gold films,* Physical Review B **8** (1973), 3689–3701

[11] Dederichs, P. H. and Zeller, *Variational treatment of the elastic constants of disordered materials,* Z. Physik **259**, (1973), 103–116

[12] Dykhne, A. M., *Conductivity of a two dimensional two phase system,* Soviet Physics JETP **32** (1971), 63–65

[13] Golden, K. and Papanicolaou, G., *Bounds for effective parameters of heterogeneous media by analytic continuation,* Commun. Math. Phys. **90** (1983), 473–491

[14] Hashin, Z. and Shtrikman, S., *A variational approach to the theory of effective magnetic permeability of multiphase materials,* Journal of Applied Physics **33** (1962), 3125–3131

[15] Hori, M., *Statistical theory of effective electrical, thermal, and magnetic properties of random heterogeneous materials.1. Perturbation expansions for the effective permittivity of cell materials,* J. Math. Phys. **14** (1973), 514–523

[16] Keller, J. B., Rubenfeld, L. and Molyneux, J., *Extremum principles for slow viscous flows with applications to suspensions,* J. Fluid Mech. **30** (1962), 97–125

[17] McPhedran, R. C. and McKenzie, D. R., *The conductivity of lattices of spheres I. The simple cubic lattice,* Proc. R. Soc. Lond. A **359** (1978), 45–63

[18] McPhedran, R. C. and McKenzie, D. R., *Electrostatic and optical resonances of arrays of cylinders,* Appl. Phys. **23** (1980), 223–235

[19] McPhedran, R. C. and Milton, G. W., *Bounds and exact theories for the transport properties of inhomogeneous media,* Appl. Phys. **A 26** (1981), 207–220

[20] Miller, M. N., *Bounds for effective electrical, thermal, and magnetic properties of heterogeneous materials,* J. Math. Phys. **10** (1969), 1988–2004

[21] Milton, G. W., *Bounds on the complex permittivity of a two-component composite material,* J. Appl. Phys. **52** (1981), 5286

[22] Sangani, A. S. and Acrivos, A., *The effective conductivity of a periodic array of spheres,* Proc. R. Soc. Lond. A **386** (1983), 263–275

[23] Torquato, S. and Rubinstein, J., In preparation

[24] Willemse, M.W.M. and Caspers W.J., *Electrical conductivity of polycrystals,* J. Math. Phys. **20** (1979), 1824–1831

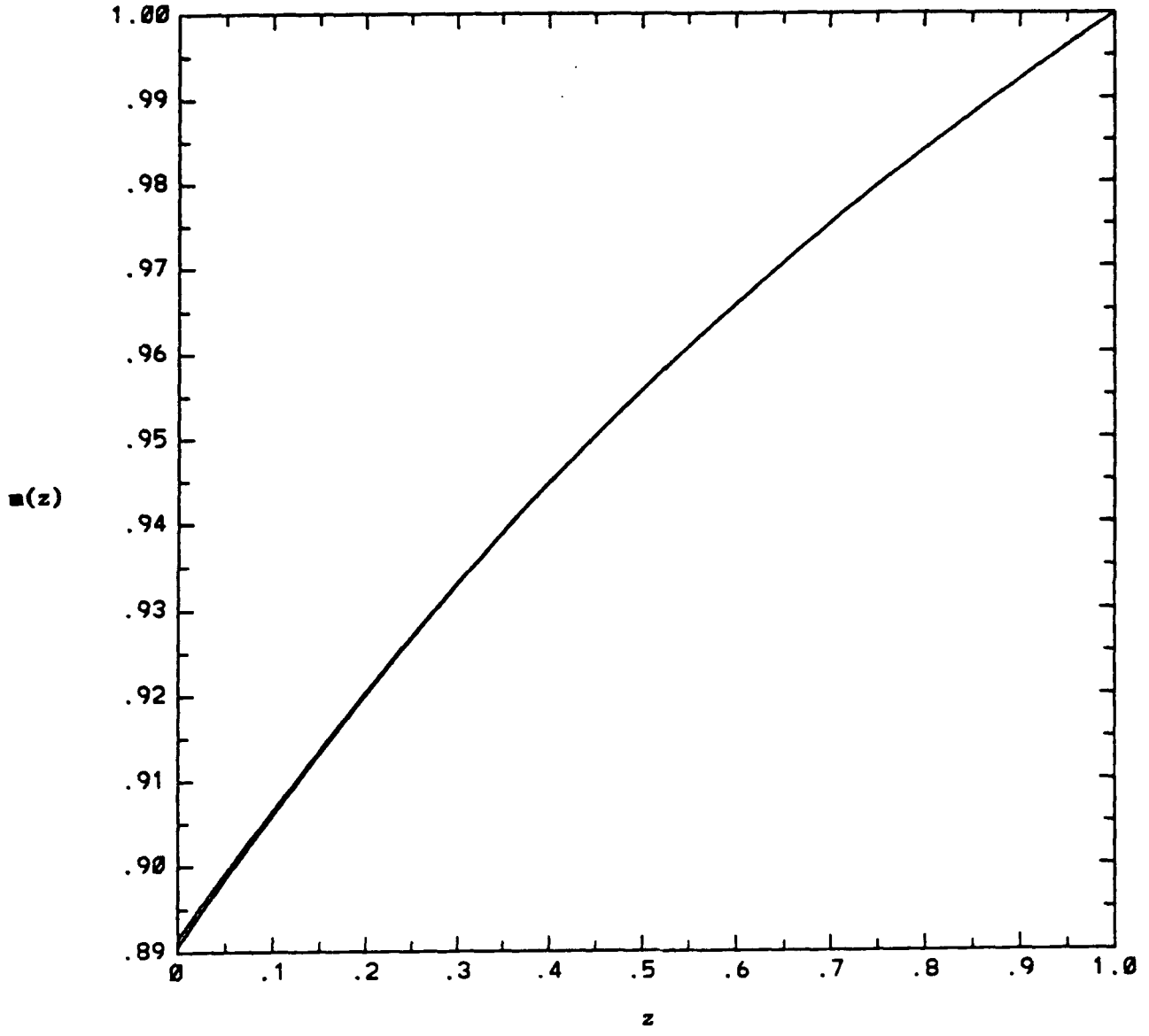
Table 1

q	A	$z_m = -B$	z_M	s_m	s_M
.00	1.00000	-2.00000	-1.00000	0.33333	0.50000
.10	1.00000	-2.00300	-1.00000	0.33300	0.50000
.20	1.00000	-2.02419	-1.00000	0.33067	0.50000
.30	1.00000	-2.08325	-1.00000	0.32433	0.50000
.40	1.00000	-2.20513	-1.00000	0.31200	0.50000
.50	1.00000	-2.42857	-1.00000	0.29167	0.50000
.60	1.00000	-2.82653	-1.00000	0.26133	0.50000
.70	1.28311	-3.56621	-0.77936	0.21900	0.56200
.80	2.07377	-5.14754	-0.48221	0.16267	0.67467
.90	4.53506	-10.07011	-0.22050	0.09033	0.81933
.91	5.08694	-11.17389	-0.19658	0.08214	0.83571
.92	5.77776	-12.55552	-0.17308	0.07377	0.85246
.93	6.66702	-14.33404	-0.14999	0.06521	0.86957
.94	7.85395	-16.70789	-0.12732	0.05647	0.88706
.95	9.51709	-20.03418	-0.10507	0.04754	0.90492
.96	12.01360	-25.02719	-0.08324	0.03842	0.92316
.97	16.17681	-33.35361	-0.06182	0.02911	0.94178
.98	24.50669	-50.01339	-0.04081	0.01960	0.96079
.99	49.50305	-100.00631	-0.02020	0.00990	0.98020
1	∞	$-\infty$	0	0	1

Table 1: values of the different constants related to the bounds (25) and (26) for an arbitrary array of spheres, as functions of the parameter q. The interval $z_m \leq z \leq z_M$ contains all the zeroes and singularities of m in the z plane.

FIGURE 1.a

Case q = .5



Figures 1.a–3.b: the bounds (25) and (26) in the case of a random closely packed array of coated spheres. The coated spheres are assumed to fill 60 percent of the volume of the sample. Figures 1.a–1.b: case $q = .5$. Figures 2.a–2.b: case q=.8. Figures 3.a–3.b: case q=.99. Other quantities of interest can be found in tables 2.a and 2.b.

FIGURE 1.b

Case q = .5

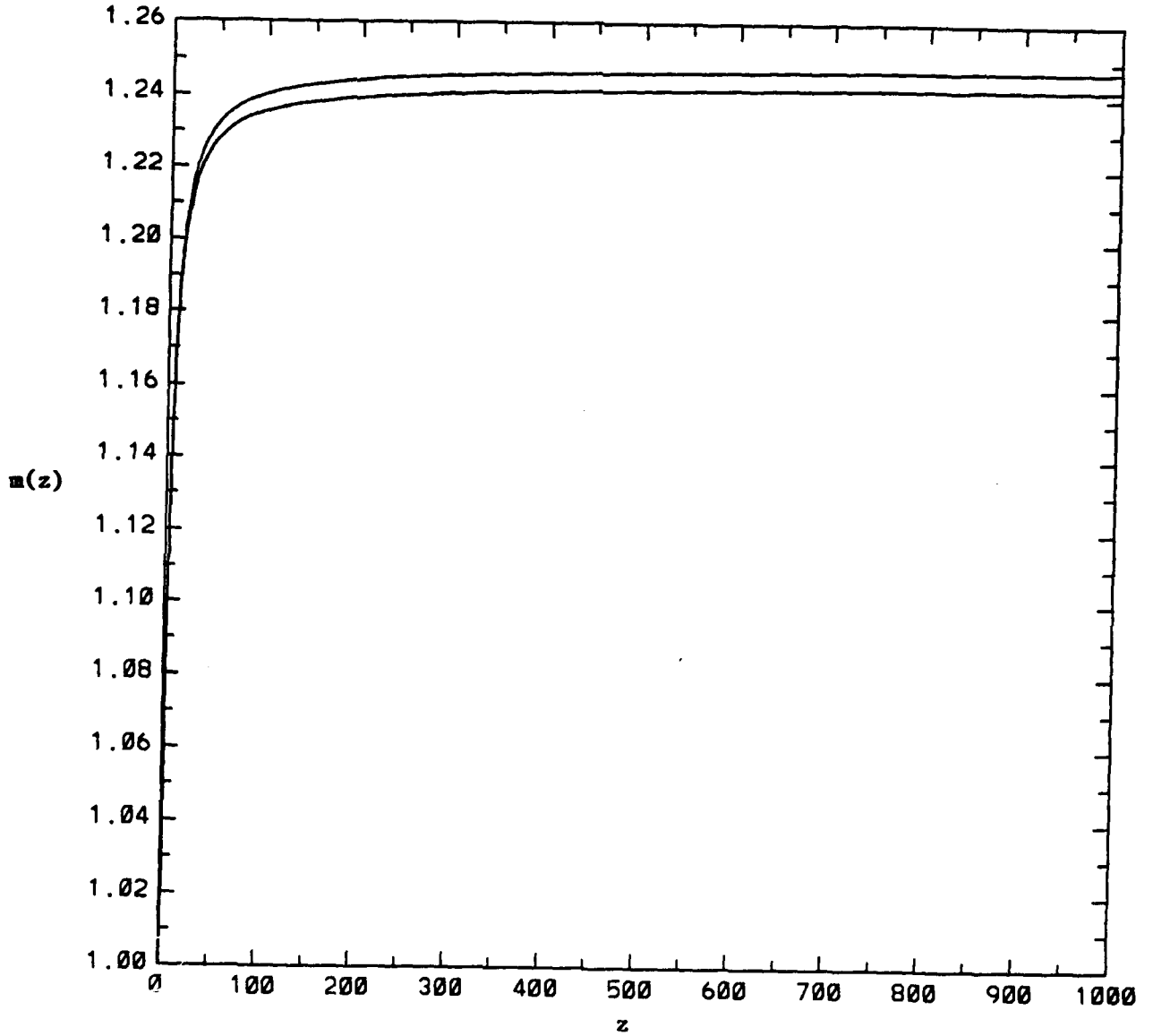


FIGURE 2.a
Case q = .8

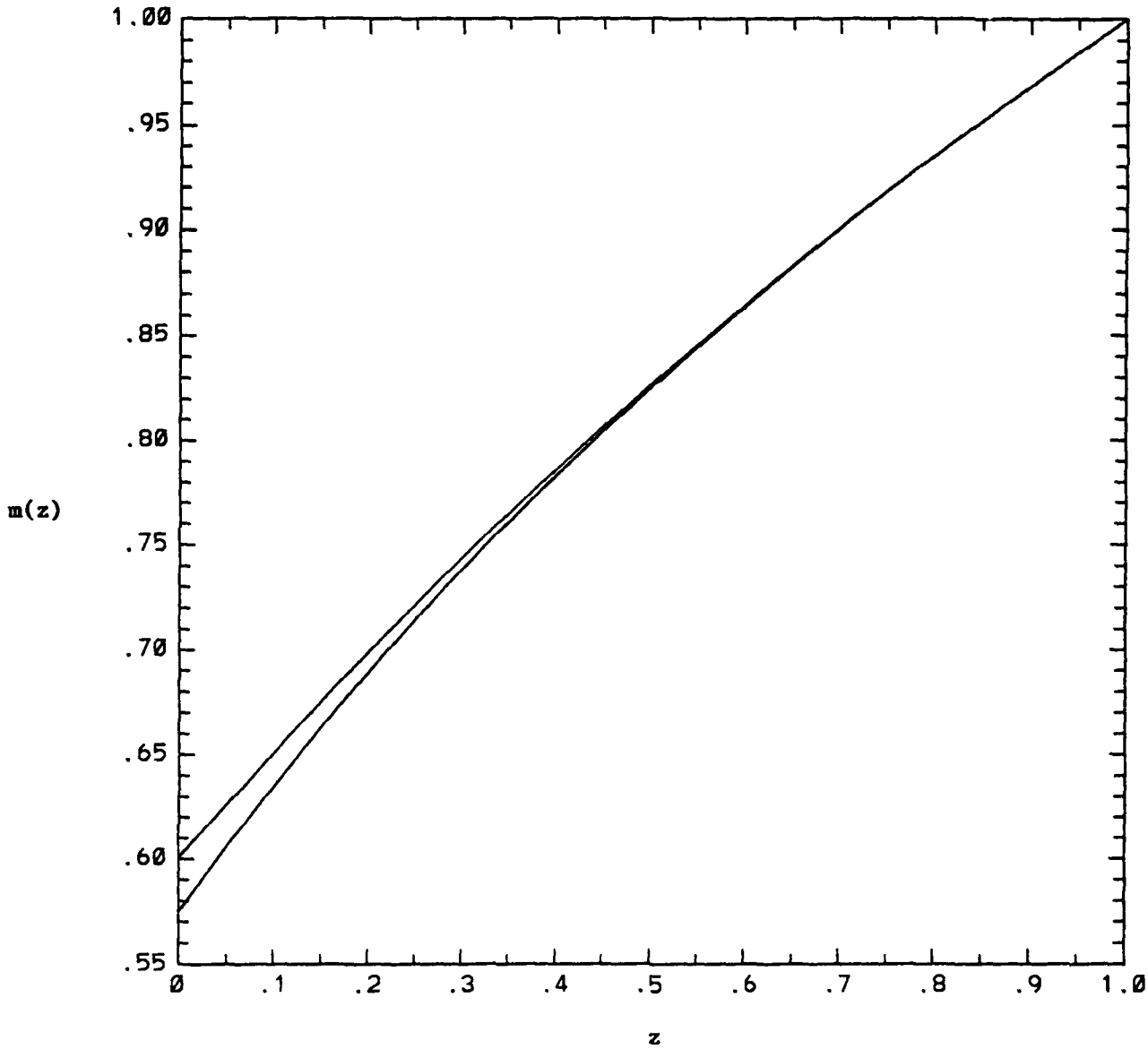


FIGURE 2.b

Case q = .8

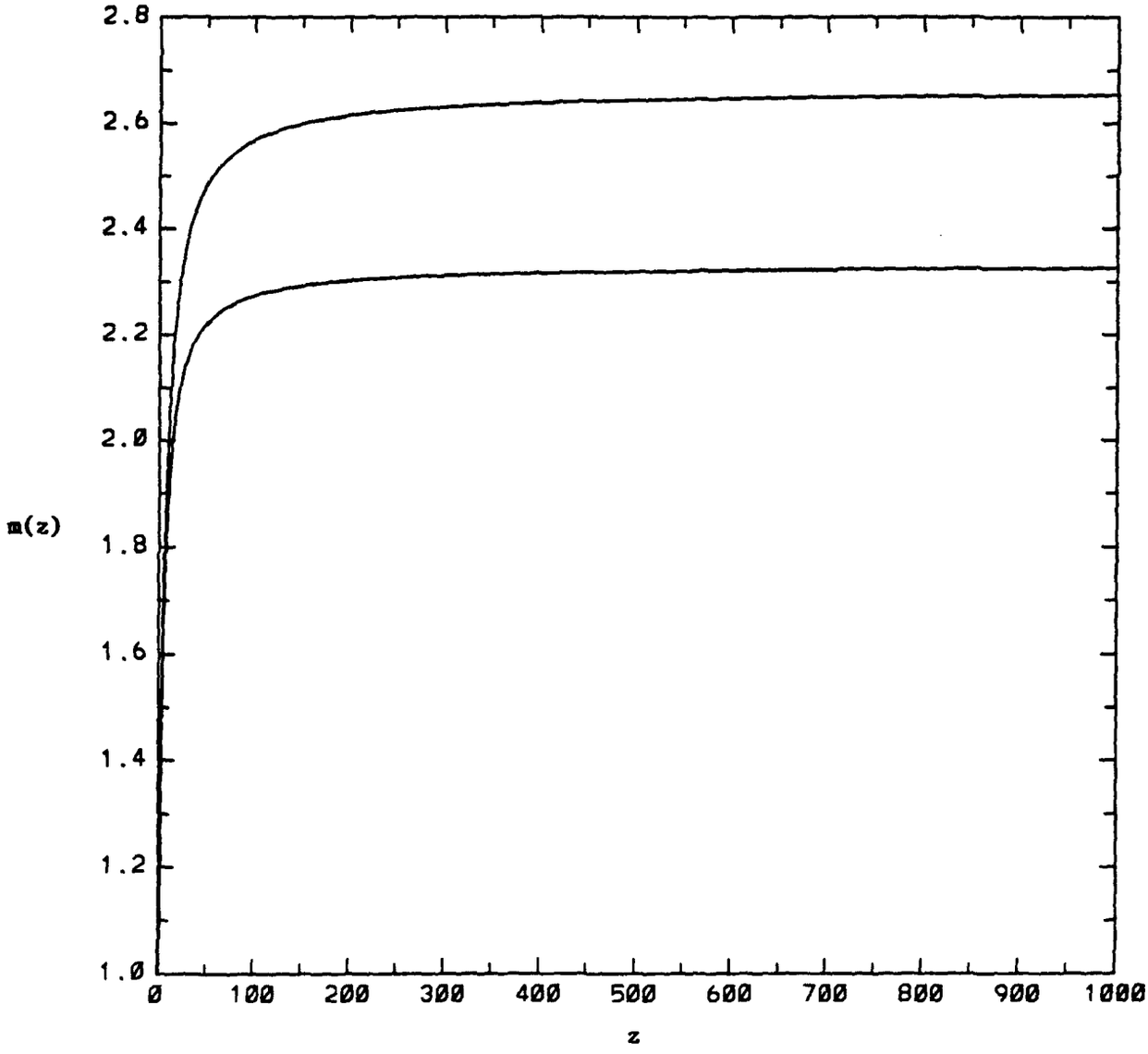


FIGURE 3.a

Case q = .99

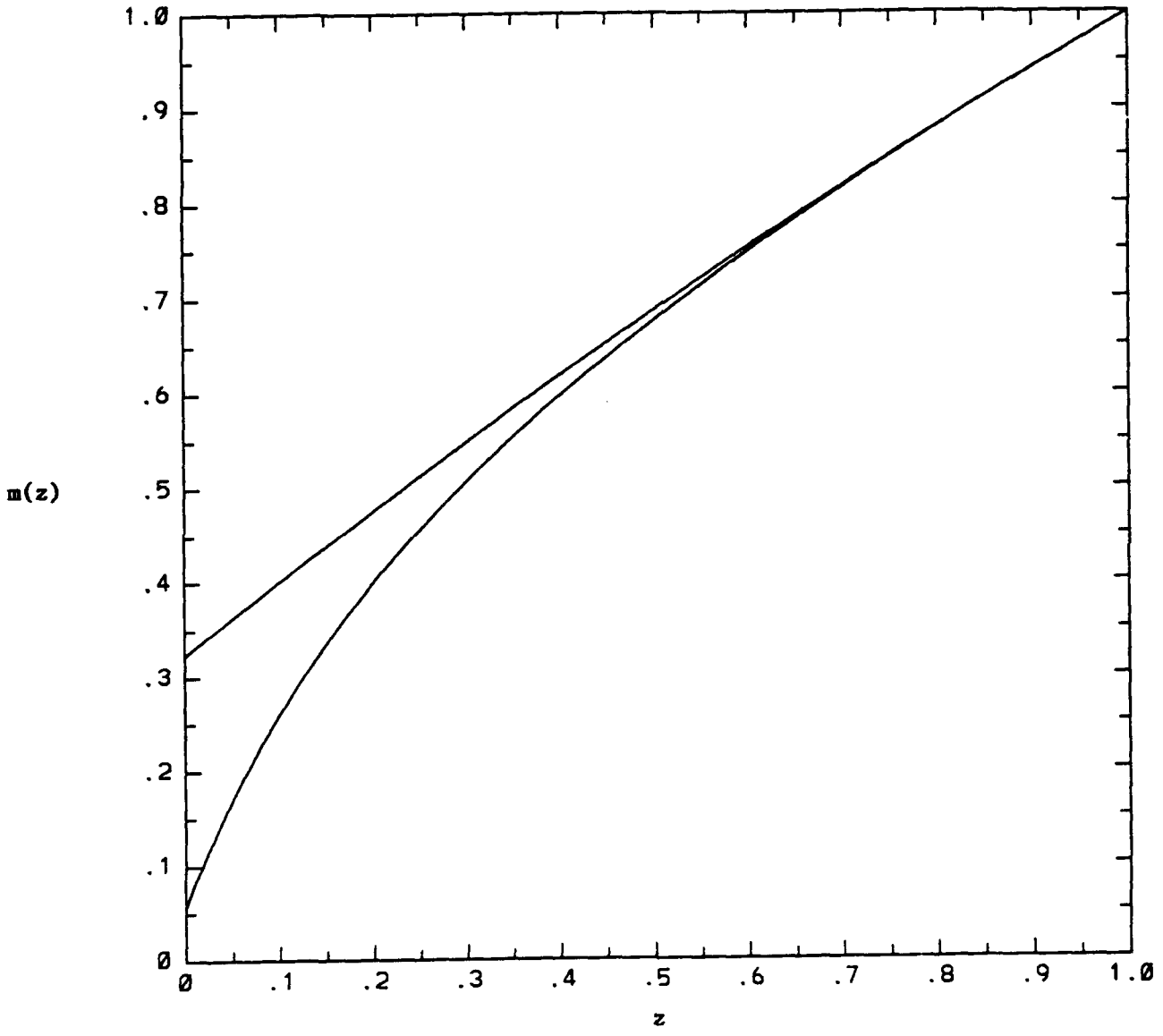


FIGURE 3.b

Case q = .99

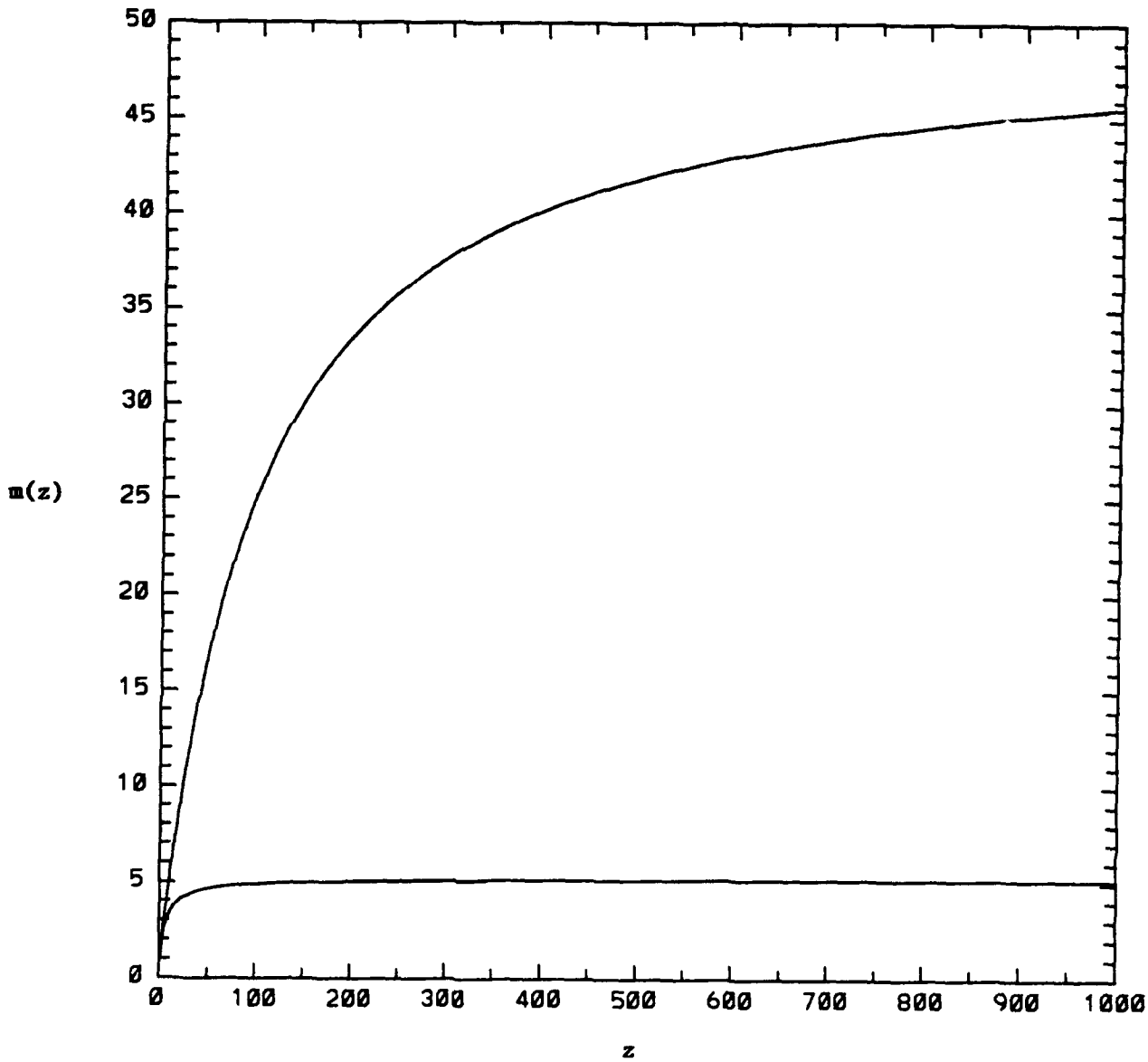


Table 2.a

Case z = 0

q	p_1	V	W	B_l	B_u
0.10	0.000600	0.999100	0.999100	0.999100	0.999100
0.20	0.004800	0.992813	0.992817	0.992794	0.992829
0.30	0.016200	0.975851	0.975895	0.975634	0.976024
0.40	0.038400	0.943238	0.943485	0.942029	0.944186
0.50	0.075000	0.890653	0.891566	0.886076	0.894118
0.60	0.129600	0.815090	0.817430	0.801308	0.824549
0.70	0.205800	0.712703	0.720102	0.680336	0.736492
0.80	0.307200	0.574567	0.600555	0.514334	0.633121
0.90	0.437400	0.376004	0.461639	0.292309	0.519165
0.93	0.482614	0.293637	0.416807	0.212756	0.483717
0.96	0.530842	0.191347	0.370753	0.126452	0.447948
0.99	0.582179	0.056292	0.323619	0.032892	0.411999

Tables 2.a, 2.b: the bounds (25), (26) and the corresponding security sphere bounds B_l and B_u. Table 2.a: a closely packed array of coated spheres, $z = 0$. Table 2.b: a closely packed array of coated spheres, $z = \infty$.

Table 2.b

Case $z = \infty$

q	p_1	W	V	B_l	B_u
0.10	0.000600	1.00180	1.00180	1.00180	1.00180
0.20	0.004800	1.01447	1.01449	1.01438	1.01452
0.30	0.016200	1.04940	1.04959	1.04834	1.04995
0.40	0.038400	1.11980	1.12097	1.11374	1.12308
0.50	0.075000	1.24324	1.24841	1.21951	1.25714
0.60	0.129600	1.44669	1.46525	1.37270	1.49592
0.70	0.205800	1.77739	1.85262	1.57777	1.93973
0.80	0.307200	2.33025	2.66324	1.83599	2.88852
0.90	0.437400	3.33239	5.14329	2.14523	5.84207
0.93	0.482614	3.79838	7.28052	2.24720	8.40043
0.96	0.530842	4.39443	12.6322	2.35310	14.8163
0.99	0.582179	5.18012	50.1269	2.46269	59.8044

DYNAMIC SHEAR BAND DEVELOPMENT IN PLANE STRAIN COMPRESSION OF A BIMETALLIC BODY*

R. C. Batra and Z. G. Zhu
Department of Mechanical and Aerospace Engineering
and Engineering Mechanics
University of Missouri-Rolla
Rolla, MO 65401-0249, USA

ABSTRACT. We study plane strain thermomechanical deformations of a prismatic viscoplastic body of square cross-section and deformed at a nominal strain-rate of 5000 sec^{-1}. The body has two thin layers placed symmetrically about the horizontal centroidal axis. The layer material differs from that of the body in only the value of the yield stress in a quasistatic simple compression test. The yield stress for the layer material is taken to be either one-fifth or five times that of the matrix material. Three cases, namely, when there is an elliptical void with its major axis aligned along the horizontal centroidal axis or the vertical centroidal axis, but the void center coincides with the center of the cross-section, and when there are two elliptical voids with major axes aligned along the vertical centroidal axis and a void tip abuts the layer/matrix interface are studied. The deformations are assumed to be symmetrical about the vertical and horizontal centroidal axes.

It is found that in each case shear bands initiate from points on the traction free edges where the matrix/layer interfaces intersect them and propagate into the softer material. For the soft layer these bands initially merge into one and propagate horizontally. Subsequently, each of these bands bifurcates into two which propagate into the matrix material along the direction of the maximum shear stress. There is minimal interaction between these bands and those initiating from points near the void tips.

INTRODUCTION. Adiabatic shear bands are narrow regions of intense plastic deformation that form during high strain-rate processes, such as shock loading, ballistic penetration, metal forming, and machining. As these bands generally precede material fracture, a knowledge of factors that inhibit or enhance their growth is essential to the production of durable materials and more efficient manufacturing processes. These bands form in both ferrous and nonferrous alloys.

Johnson (1987) has recently pointed out that Tresca (1878) and Massey (1921) observed hot lines, now referred to as adiabatic shear bands, during the forging of platinum. Both Tresca and Massey stated that these were the lines of greatest sliding, and also therefore the zones of greatest development of heat. Wulf (1978) has reported experimental observations of adiabatic shear bands in high strain rate (2000 to 25000 sec^{-1}) compression of 7039 aluminum armour. He found that the cross-section of the cylindrical specimens changed from circular to elliptical after the compression test, and adiabatic shear bands formed in the specimens which subsequently failed by crack propagation along the dominant band. Further references to the analytical, numerical and experimental work on shear banding may be found in two recent books (Dodd and Bai (1987), Semiatin and Jonas (1984)).

Recently, LeMonds and Needleman (1986), Needleman (1989), Anand et al. (1988), Zbib and Aifantis (1988), Batra and Liu (1989,1990), Zhu and Batra (1990), Batra and Zhu (1990), and Batra and Zhang (1990) have studied the phenomenon of shear banding in plane strain deformations of a viscoplastic solid. Whereas Needleman studied a purely mechanical problem, other works have treated a coupled thermomechanical problem. LeMonds and Needleman, Zbib and Aifantis, and Anand et al. neglected the effect of inertia forces on the ensuing deformations of the body. These investigations have employed

* Supported by the U. S. Army Research Office Contract DAAL03-88-K-0184 to the University of Missouri - Rolla.

different constitutive relations, different techniques to integrate the stiff set of governing partial differential equations, and have generally assumed that the entire body or the portion of the body whose deformations were analyzed had only one defect in it. The prismatic body whose plane strain thermomechanical deformations are studied herein is of a square cross-section and has two thin layers made of a viscoplastic material different from that of the body and placed symmetrically about and parallel to the horizontal centroidal axis. These horizontal layers may be thought of as representing planes of chemical inhomogeneity. Also, as stated above in the abstract, there is either an elliptical void at the center of the cross-section or two ellipsoidal voids with major axes along the vertical centroidal axis and tips touching the layer/matrix interfaces. The voids can form during manufacturing. However, the symmetrical situation considered herein is to simplify the problem. The vertices of the ellipsoidal void on its major axis and the points on the free edges where the thin layer and the matrix materials meet act as nuclei for the initiation of shear bands. It thus becomes an interesting exercise to study the initiation and growth of various bands and the interaction, if any, amongst them. We account for the effect of inertia forces, strain-rate sensitivity of the materials, thermal softening effects, heat conduction, and the heat generated due to plastic working.

FORMULATION OF THE PROBLEM. Figure 1 depicts the cross-section of the prismatic body, and the relative dimensions of the ellipsoidal void and the two thin layers. For this case, the centers of the void and the cross-section coincide and the major axis of the void coincides with the vertical centroidal axis of the cross-section. It is assumed that the body is loaded along the vertical axis, plane strain state of deformation prevails, and that the deformations are symmetrical about the two centroidal axes. Thus the deformations of the material in the first quadrant are analyzed. We use a fixed set of rectangular cartesian coordinate axes and the referential description of motion to describe the thermomechanical deformations of the body. The governing equations are:

balance of mass: $(\rho J)^{\cdot} = 0,$ (2.1)

balance of linear momentum: $\rho_0 \dot{v}_i = T_{i\alpha,\alpha},$ (2.2)

balance of moment of momentum: $T_{i,\alpha}\, x_{j,\alpha} = T_{j,\alpha}\, x_{i,\alpha},$ (2.3)

balance of internal energy: $\rho_0 \dot{e} = -Q_{\alpha,\alpha} + T_{i,\alpha}\, v_{i,\alpha},$ (2.4)

where

$$T_{i\alpha} = (\rho_0/\rho)\, \sigma_{ij}\, X_{\alpha,j},$$
$$\sigma_{ij} = -B\,(\rho/\rho_0 - 1)\,\delta_{ij} + 2\mu\, D_{ij}, \quad (2.5)$$

$$2\mu = [\sigma_0/\sqrt{3I}]\,(1 + bI)^m\,(1 - \alpha\theta), \quad (2.6)$$

$$I^2 = (1/2)\,\bar{D}_{ij}\,\bar{D}_{ij}, \quad (2.7)$$

$$\bar{D}_{ij} = D_{ij} - (1/3)\,D_{kk}\,\delta_{ij}, \quad (2.8)$$

$$Q_\alpha = (\rho_0/\rho)\, q_i\, X_{\alpha,i},\ q_i = -k\,\theta_{,i}, \quad (2.9)$$

$$\dot{e} = c\,\dot{\theta} + B(\rho/\rho_0 - 1)\,\dot{\rho}/\rho^2\ . \quad (2.10)$$

In these equations x_i gives the position at time t of the material particle X_α, $v_i = \dot{x}_i$ is its velocity in the x_i-direction, ρ is its present mass density, ρ_0 its mass density in the reference configuration, $J = \det [x_{i,\alpha}]$, $x_{i,\alpha} = \partial x_i/\partial X_\alpha$, $T_{i\alpha}$ is the first Piola-Kirchoff stress tensor, σ_{ij} is the Cauchy stress tensor, e is the specific internal energy, Q_α is the heat flux

measured per unit area in the reference configuration, $\underset{\sim}{D}$ is the strain-rate tensor and $\underset{\sim}{\bar{D}}$ is its deviatoric part, a superimposed dot indicates material time derivative, a comma followed by index α (j) implies partial differentiation with respect to X_α (x_j), and a repeated index implies summation over the range (1,2) of the index. In the constitutive relations (2.5), (2.9) and (2.10), the material parameter B represents the bulk modulus, σ_o is the yield stress in a quasistatic simple compression test, parameters b and m characterize the strain-rate sensitivity of the material, α describes its thermal softening, θ equals the temperature change of a material particle from that in the reference configuration, k is the constant thermal conductivity and c is the constant specific heat. Here we have not considered the stresses caused by the thermal expansion.

The foregoing equations hold in regions occupied by the matrix and the layers. The values of material parameters for the matrix and the layer materials are the same except that either

$$\sigma_o \text{ layer} = 5\, \sigma_o \text{ matrix}, \tag{2.11a}$$

or

$$\sigma_o \text{ layer} = (1/5)\, \sigma_o \text{ matrix}. \tag{2.11b}$$

In terms of the deviatoric stress $\underset{\sim}{s}$ defined by

$$\underset{\sim}{s} = \underset{\sim}{\sigma} + [B(\rho/\rho_o - 1) - (2\mu/3)\ \text{tr}\ \underset{\sim}{D}]\underset{\sim}{1}, \tag{2.12a}$$

$$= 2\mu\ \underset{\sim}{\bar{D}}, \tag{2.12b}$$

equations (2.12), (2.5) and (2.6) give

$$(1/2\ \text{tr}\ \underset{\sim}{s}^2)^{1/2} = (\sigma_o/\sqrt{3})(1 - \alpha\theta)\ (1 + bI)^m. \tag{2.13}$$

We assume that the body is initially at rest at a uniform temperature, has a constant mass density and is initially stress free. That is

$$\rho(\underset{\sim}{x},o) = 1,\ \underset{\sim}{v}(\underset{\sim}{x},o) = \underset{\sim}{o},\ \theta(\underset{\sim}{x},o) = 0\ . \tag{2.14}$$

For the material in the first quadrant, we impose the following boundary conditions.

$$v_2 = -h(t),\ T_{12} = 0 \text{ and } Q_2 = 0, \quad \text{on the top surface AB}, \tag{2.15}$$

$$T_{11} = 0,\ T_{21} = 0 \text{ and } Q_1 = 0, \quad \text{on the right surface BC}, \tag{2.16}$$

$$v_2 = 0,\ T_{12} = 0 \text{ and } Q_2 = 0, \quad \text{on the bottom surface CD}, \tag{2.17}$$

$$T_{i\alpha}N_\alpha = 0,\ Q_\alpha N_\alpha = 0, \quad \text{on the void surface DE}, \tag{2.18}$$

$$v_1 = 0,\ T_{21} = 0 \text{ and } Q_1 = 0, \quad \text{on the left surface EA}. \tag{2.19}$$

That is the top surface is moving downward with a speed h(t), contact between it and the loading device is smooth, the right surface is traction free, and the entire boundary is ther-

mally insulated. If at any time during the deformations of the body, a point on the void surface touches the vertical axis, the boundary condition on it is changed to (2.19). The boundary conditions (2.17) and (2.18) reflect the presumed symmetry of deformations about the x_1 and x_2 axes. For the loading function h(t) we take

$$h(t) = v_o\, t/t_r, \quad 0 \le t \le t_r, \qquad = v_o \quad t > t_r. \tag{2.20}$$

The steady speed v_0 of the top surface of the block is reached in time t_r.

The matrix and the layer are assumed to be perfectly bonded. Thus at the common interfaces between them, the velocity field, surface tractions, the temperature and the normal component of the heat flux are assumed to be continuous.

For other configurations of the voids, the boundary conditions are appropriately modified.

RESULTS. The finite element code developed by Batra and Liu (1989, 1990) was modified to analyze the present problem. In order to compute results, we used the following values of various material and geometric parameters.

$$b = 10000 \text{ sec}, \ \sigma_o = 333 \text{ MPa}, \ k = 49.22 \text{ W m}^{-1} \text{ °C}^{-1},$$
$$m = 0.025, \ c = 473 \text{ J Kg}^{-1} \text{ °C}^{-1}, \ \rho_o = 7860 \text{ Kg m}^{-3},$$
$$B = 128 \text{ GPa}, \ H = 5 \text{ mm}, \ v_o = 25 \text{ m sec}^{-1}, \tag{3.1}$$
$$\alpha = 0.0025 \text{ °C}^{-1}.$$

Thus the average applied strain-rate $\dot{\gamma}_{avg}$ equals 5000 sec^{-1}, the reference temperature $\theta_o = \sigma_o/(\rho_o c) = 89.6°$ C, and $\nu = \rho_o v_o^2/\sigma_o = 0.015$. The nondimensional number ν signifies the effect of inertia forces relative to the flow stress of the material. For the simple shearing problem, Batra (1988) noted that the inertia forces play a noticeable role when $\nu = 0.004$. Thus for the present problem, the inertia forces will very likely play a significant role.

LAYER MATERIAL SOFTER THAN THE MATRIX MATERIAL. Figure 2 depicts the contours of the maximum principal logarithmic strain ϵ at an average strain of 0.079 when the center of the ellipsoidal void coincides with the center of the cross-section and the major axis of the void is aligned along the vertical centroidal axis. These contours and other results reported by Zhu and Batra (1990b) reveal that shear bands initiate from points on the right traction free edge where the matrix /layer interfaces intersect it. Because the layer material is softer and its thickness quite small, these bands merge into one band that initially propagates horizontally into the layer. When the matrix material has softened somewhat due to the rise in its temperature, the horizontally propagating band bifurcates into two bands that propagate into the matrix along ±45° directions. The band propagating into the matrix material above the layer has more severe deformations associated with it than the one propagating into the matrix material below the layer.

The band initiating from a point near the void tip on the vertical centroidal axis propagates along a line that makes an angle of 45° with the horizontal. This band seems to pass through the soft layer rather easily.

Figure 3 shows the contours of the maximum principal logarithmic strain ϵ at an average strain of 0.0333 when the ellipsoidal void is at the center of the cross-section and the major axis of the void is aligned along the horizontal centroidal axis. These plots look quite similar to that for the case when the major axis of the void coincides with the vertical centroidal axis. For a further discussion and details of results in this case, see Batra and Zhu (1991).

When the void tip touches the matrix/layer interface and the major axis of the void coincides with the vertical centroidal axis, the contours of ϵ plotted in Figure 4 at an average strain rate of 0.0175 look quite different from the previous two cases. Results given by

Batra and Zhu (1990) and these contours of ϵ suggest that a shear band initiates within the matrix surrounding the void tip near the matrix/layer interface and propagates into the matrix material below the common interface, the direction of propagation being nearly 45° to the vertical axis. The shear bands initiating at points of intersection of the matrix/layer interfaces with the right traction free surface propagate into the soft layer and then bifurcate into the matrix material along lines making an angle of approximately 45° with the vertical. The band in the layer near the upper matrix/layer interface bifurcates into the matrix prior to that near the lower interface. Also the band in the layer near the upper matrix/layer interface continues to propagate horizontally into the layer too, while that near the lower surface does not.

LAYER MATERIAL STRONGER THAN THE MATRIX MATERIAL. We first study the case when the center of the ellipsoidal void coincides with that of the cross-section. In Figure 5, we have plotted contours of ϵ at an average strain of 0.122 when the major axis of the void is vertical. Now the bands initiating from points on the right traction free edge where the matrix/layer interfaces intersect it propagate into the matrix material along lines making an angle of approximately ±45° with the horizontal. Recall that the matrix material is softer than the layer material. The quarter of the square cross-section studied is divided into five subregions, each of which is deforming essentially rigidly, and there is a shear band at the four common boundaries (e.g., see Zhu and Batra (1990b)). However, when the major axis of the void is horizontal, contours of ϵ depicted in Figure 6 at an average strain of 0.031 suggest a picture different from the one when the major axis of the void was vertical. We should add that the average strains in the two cases are quite different. Hence, a direct comparison is not very meaningful.

Figure 7 shows contours of ϵ at an average strain of 0.057 when a void tip is at the matrix/layer interface. These contours and other results given by Batra and Zhu (1990) reveal that a shear band initiating from the void tip abutting the matrix/layer interface propagates initially along the interface and then into the matrix material along a line making an angle of nearly 45° with the vertical. The shear band initiating from the lower void tip also propagates into the matrix material along a line making an angle of approximately 45° with the vertical. Two shear bands also initiate from points on the right traction free edge where matrix/layer interfaces intersect it, and these bands propagate into the matrix material along lines making an angle of 45° with the vertical. Even though it seems that near the vertical centroidal axis a shear band has propagated into the layer, there is no localization of deformation occurring in the layer material. This is evidenced by the plots of ϵ versus the average strain at several points in the layer that are given in Fig. 8c of Batra and Zhu's (1990) paper. The contours of the temperature rise, not included herein, support the picture laid out above for the development of four bands, two from void tips and two from points on the right traction free surface where the layer and the matrix materials meet.

CONCLUSIONS. We have analyzed the problem of the initiation and growth of shear bands in a prismatic viscoplastic body containing an ellipsoidal void, and two thin layers made of a different viscoplastic material placed symmetrically about the horizontal centroidal axis. The body is deformed in plane strain compression along the vertical axis at an average strain-rate of 5000 sec^{-1}, and its deformations are assumed to be symmetrical about the two centroidal axes.

Two shear bands initiate from points on the right traction free edge where the matrix/layer interfaces meet it. These bands propagate into the softer material. When the matrix material is softer, these bands propagate along lines that make an angle of approximately ±45° with the horizontal. However, when the layer material is softer, these bands essentially merge into one and initially propagate horizontally into the layer. Subsequently, this band bifurcates into two bands that propagate into the matrix material along ±45° directions.

Shear bands also initiate from points near the void tips and propagate in the ±45° directions. When the layer material is stronger than the matrix material, these bands do not pass through the layer and are deflected back upon arriving at the matrix/layer interface.

However, they pass through easily through a softer layer.

REFERENCES

Anand, L., Lush, A. M., and Kim, K. H., 1988, "Thermal Aspects of Shear Localization in Viscoplastic Solids", Thermal Aspects in Manufacturing, M. H. Attia and L. Kops., eds., ASME - PED Vol. 30, pp. 89-103.

Batra, R. C., and Kim, C. H., 1990, "Effect of Viscoplastic Flow Rules on the Initiation and Growth of Shear Bands at High Strain Rates", J. Mechs. Phys. Solids (in press).

Batra, R. C., and Liu, D. S., 1990, "Adiabatic Shear Banding in Dynamic Plane Strain Compression of a Viscoplastic Material", Int. J. Plasticity, Vol. 6, pp. 231-246.

Batra, R. C., and Liu, D. S., 1989, "Adiabatic Shear Banding in Plane Strain Problems", ASME J. Appl. Mechs., Vol. 56, pp. 527-534.

Batra, R. C., and Zhang, X. T., 1990, "Shear Band Development in Dynamic Loading of a Viscoplastic Cylinder Containing Two Voids", Acta Mechanica (in press).

Batra, R. C., and Zhu, Z. G., 1990, "Dynamic Shear Band Development in a Thermally Softening Bimetallic Body Containing Two Voids", Acta Mechanica (in press).

Batra, R. C. and Zhu, Z. G., 1991, "Dynamic Adiabatic Shear Band Development in a Bimetallic Body Containing a Void", Int. J. Solids Structures (in press).

Johnson, W., 1987, "Henri Tresca as the Originator of Adiabatic Heat Lines", Int. J. Mech. Sci., Vol. 29, pp. 301-310.

Le Monds, J., and Needleman, A., 1986, "An Analysis of Shear Band Development Incorporating Heat Conduction", Mech. Mat., Vol. 5, pp. 363-373.

LeMonds, J., and Needleman, A., 1986, "Finite Element Analyses of Shear Localization in Rate and Temperature Dependent Solids", Mech. Mat., Vol. 5, pp. 339-361.

Massey, H. F., 1921, "The Flow of Metal During Forging", Proc. Manchester Assoc. Engineers, pp. 21-26.

Needleman, A., 1989, "Dynamic Shear Band Development in Plane Strain", ASME J. Appl. Mechs., Vol. 59, pp. 1-9.

Semiatin, S. L., and Jonas, J. J., 1984, Formability and Workability of Metals: Plastic Instability and Flow Localization, ASM, Metals Park.

Tresca, H., 1878, "On further Application of the Flow of Solids", Proc. Inst. Mech. Engr., Vol. 30, pp. 301-345.

Wulf, G. L., 1978, "The High Strain Rate Compression of 7039 Aluminum", Int. J. Mech., Vol. 20, pp. 609-615.

Zhu, Z. G., and Batra, R. C., 1990a, "Dynamic Shear Band Development in Plane Strain Compression of a Viscoplastic Body Containing a Rigid Inclusion", Acta Mechanica (in press).

Zhu, Z. G., and Batra, R. C., 1990b, "Analysis of Shear Banding in Plane Strain Compression of a Bimetallic Thermally Softening Viscoplastic Body Containing an Elliptical Void", in Thermal Problems in Elasticity and Plasticity (V. Birman and D. Hui, eds.), ASME Press (in press).

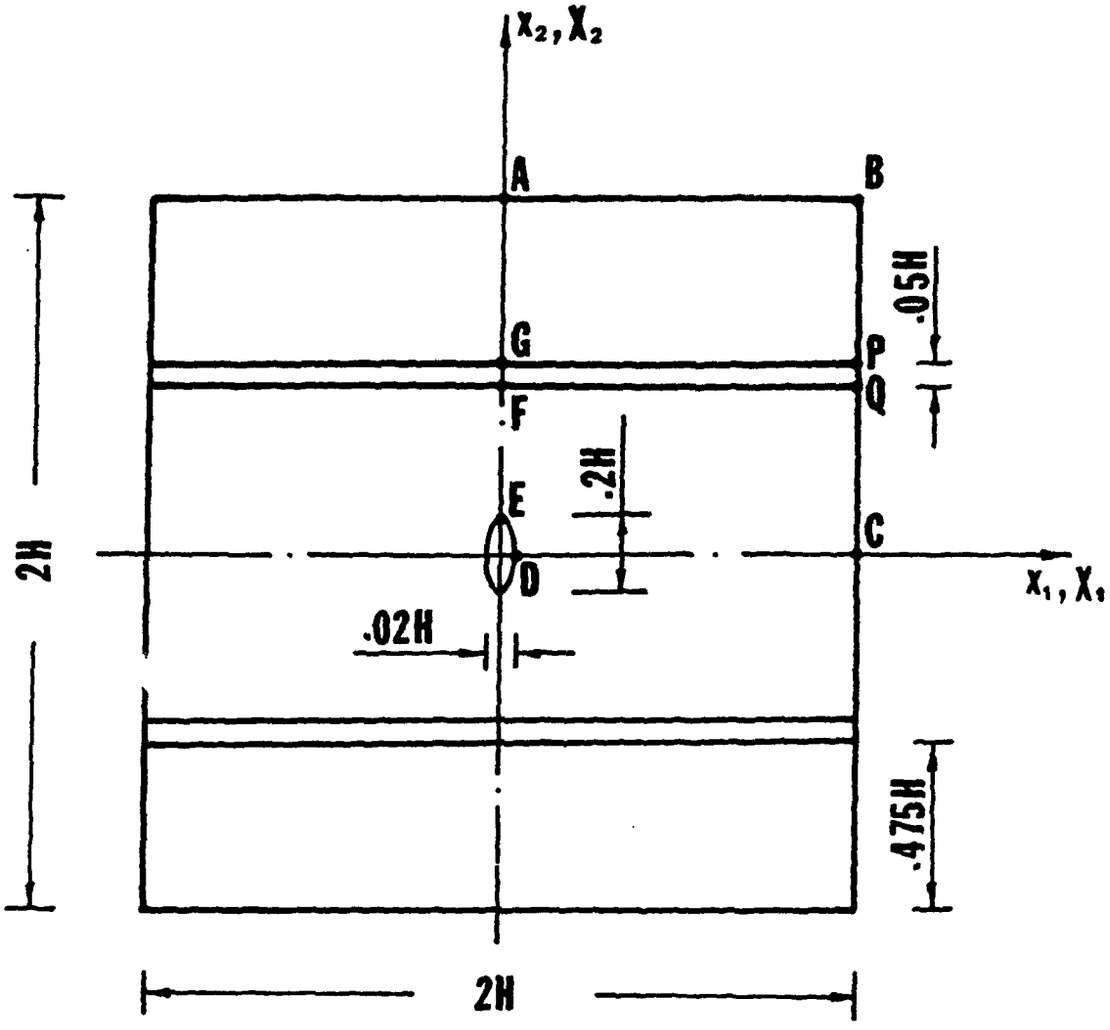


Fig. 1. Cross-section of the prismatic body studied

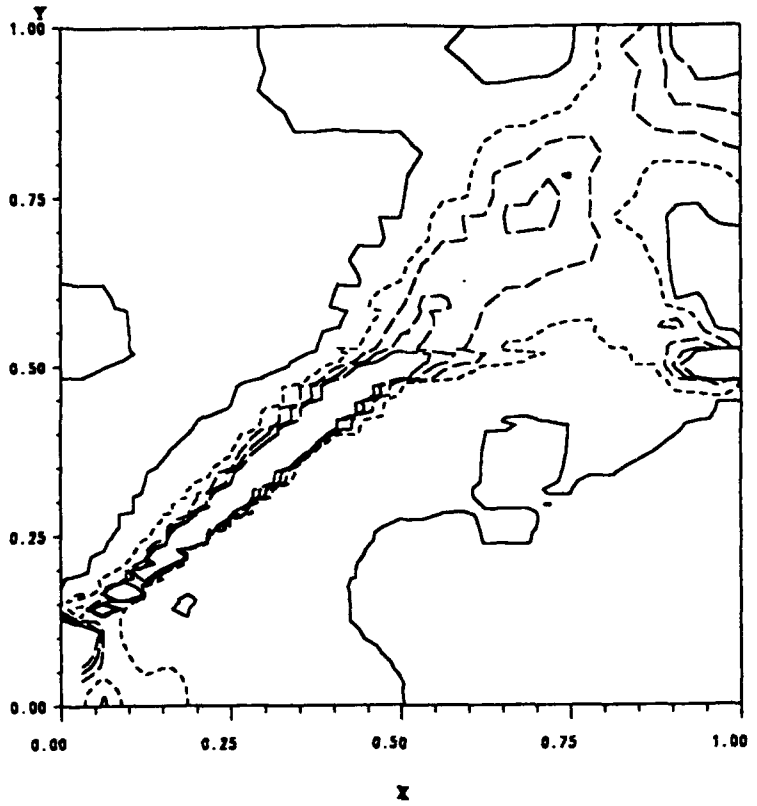


Fig. 2. Contours of the maximum principal logarithmic strain at $\gamma_{avg} = 0.079$. $\sigma_0^{layer} = (1/5)\,\sigma_0^{matrix}$. The major axis of the ellipsoidal void coincides with the vertical centroidal axis.
——— 0.05, • • • • • • 0.10, ---- 0.15, — — — — 0.20, — — — — 0.25

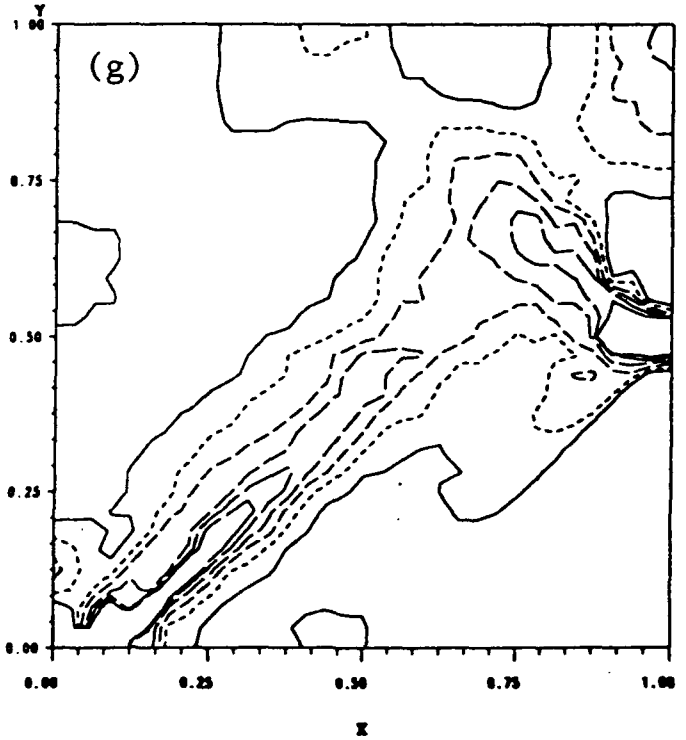


Fig. 3. Contours of the maximum prir ipal logarithmic strain at $\gamma_{avg} = 0.0333$. $\sigma_0^{layer} = (1/5)\,\sigma_0^{matrix}$, the major axis of the ellipsoidal void coincides with the horizontal centroidal axis.
——— 0.025, • • • • • 0.035, ---- 0.045, — — — — 0.055, — — — — 0.065.

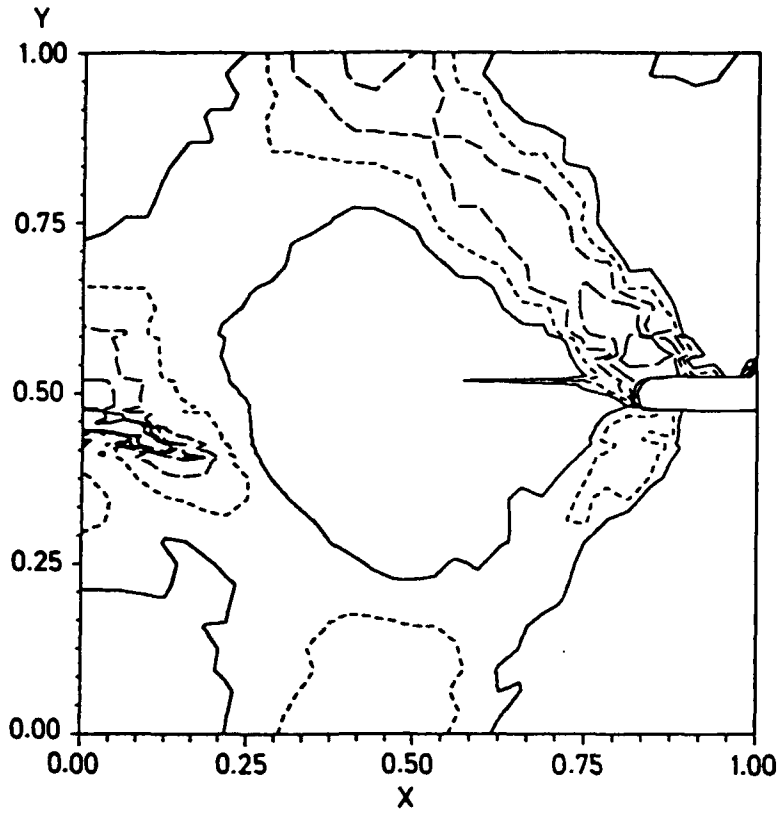


Fig. 4. Contours of the maximum principal logarithmic strain at $\gamma_{avg} = 0.0175$. $\sigma_0^{layer} = (1/5)\,\sigma_0^{matrix}$. A void tip is at the matrix/layer interface.
—— 0.015, • • • • • 0.20, — 0.025, — — — — 0.030, — — — — 0.035.

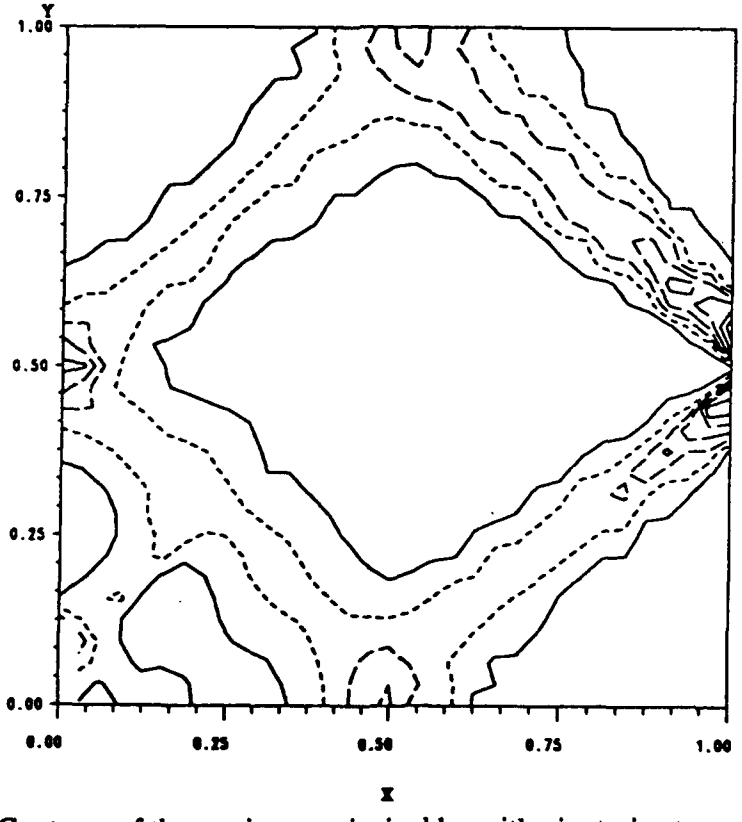


Fig. 5. Contours of the maximum principal logarithmic strain at $\gamma_{avg} = 0.122$. $\sigma_0^{layer} = 5\,\sigma_0^{matrix}$. The major axis of the ellipsoidal void coincides with the vertical centroidal axis.
—— 0.1, • • • • • 0.2, — 0.3, — — — — 0.4, — — — — 0.5.

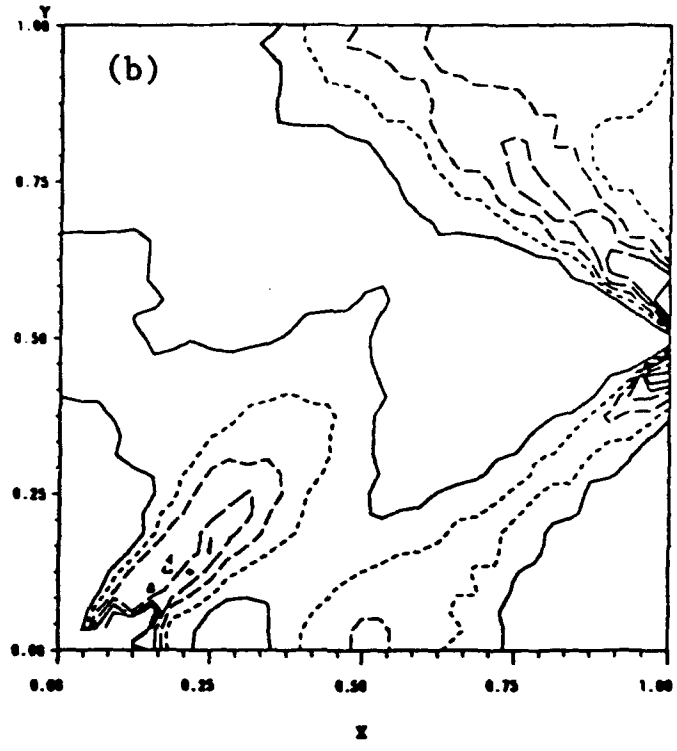


Fig. 6. Contours of the maximum principal logarithmic strain at $\gamma_{avg} = 0.0308$. $\sigma_0^{layer} = 5\sigma_0^{matrix}$, the major axis of the ellipsoidal void coincides with the horizontal centroidal axis.
—— 0.025, • • • • • 0.035, --- 0.045, — — — — — 0.055, —— - —— - 0.065.

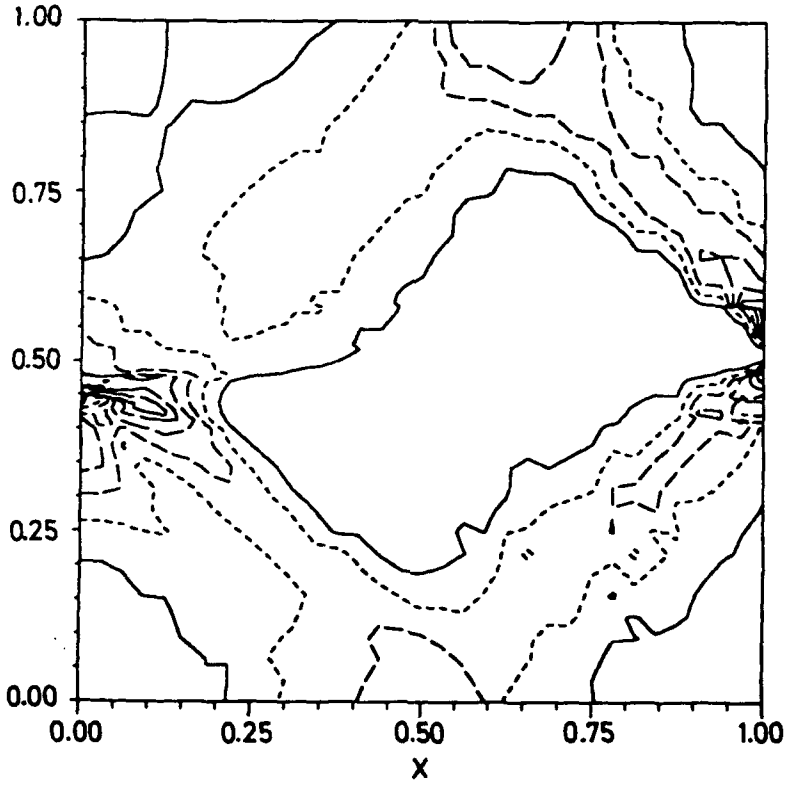


Fig. 7. Contours of the maximum principal logarithmic strain at $\gamma_{avg} = 0.057$. $\sigma_0^{layer} = 5\sigma_0^{matrix}$. A void tip is at the matrix/layer interface.
—— 0.05, • • • • • 0.06, --- 0.07, — — — — — 0.08, —— - —— - 0.09.

An Analysis of the Torsion Specimen Used in Constitutive Modeling

Charles S. White

Materials Dynamics Branch
Materials Reliability Division
U.S. Army Materials Technology Laboratory
Watertown, MA 02172-0001

ABSTRACT

In the constitutive modeling of metals there has long been the need for a simple, test methodology to achieve large deformation with a uniform stress and strain state. The torsion test has been recognized for its potential to fill this need. The geometry of the test specimen does not change significantly even for very large deformation. In recent years, the thin-walled tubular specimen having a short gauge length has been proposed to meet this need. In this paper, a nonlinear finite element analysis of a typical specimen geometry has been conducted using the ABAQUS finite element code. Several different three dimensional models have been used.

The analysis shows the suitability and limitations of applying a simple shear approximation to the deformation of the gauge section. Quantitative assessment is made of the macroscopically derived stress and strain states with the numerically predicted variations within the gauge region. A correction factor is found to be required when converting the applied twist at the grips to average shear strain across the gauge section. Plastic deformation extends from the gauge section into the shoulder region whether there is an abrupt or gradual transition from the large shoulder region to the gauge section. This also causes the development of axial strain in the gauge section even when the axial motion of the grips is constrained to be zero.

The usefulness of the thin-walled, short gauge length specimen is discussed in light of the detailed analysis. The need for accurate modeling of experimental procedures is highlighted.

Introduction

Determining the behavior of materials, particularly metals, to large deformation conditions has presented substantial experimental difficulties from the time of Tresca's early metalworking experiments [Tresca, 1864] to the present. Primary difficulties have been in determining the stress and strain accurately in an experimental test specimen that undergoes gross deformation. A homogeneous test region is required which is free from large stress or strain gradients and is large enough to measure displacements from which strain can be inferred. Tension, compression, rolling, extrusion, torsion and drawing are all types of procedures that have been applied to metals to achieve large deformation.

Torsion of solid rods and thin-walled tubular specimens have been particularly attractive because they do not undergo a large change in dimensions during a test. The biggest drawback of the solid rod torsion specimen is the large radial gradient in stress and strain. The distribution of shear strain in a twisted rod is linear varying from zero at the center to a maximum on the surface. The variation of shear stress depends upon the constitutive behavior. A solid specimen can have a very three dimensional stress distribution even if the ends of the specimen are free to expand or contract. Symmetry conditions require that the cross sectional planes of material must remain plane during deformation [Crandall, Dahl and Lardner, 1972]. An unmeasurable axial stress distribution is required to produce that deformation for all but the simplest constitutive models. There are two very attractive features of this type of test: the geometry is very simple and easily machined, and the solid rod is very stable against buckling. These two features must be measured against the uncertainty in the results due to the nonhomogeneous stress and strain states.

The thin wall torsion specimen has received considerable attention in the literature in recent years because it offers the possibility of a simple deformation field and homogeneous stress and strain states. A specimen which has a wall which is only a small fraction of the radius of the section will have a nearly uniform strain distribution through the wall thickness. The stress state is given by dividing the required torque or axial (thrust) load by the cross sectional area and mean radius. The tubular test specimen has been historically used to probe biaxial stress states and infinitesimal theory flow rules [Taylor and Quinney, 1931; Phillips and Lu, 1984]. The specimen is known to be unstable at large strains when it is proportioned by conventional means having long uniform gauge sections with gradual transition to the gripping region. Torsional buckling is a primary mode of failure for such a specimen. The problem of torsional buckling can be avoided by shortening the gauge length so that it is only a fraction of the diameter of the gauge section. It is much harder to buckle a short, squat cylinder than a long, thin one. A specimen of this type was first proposed by Hodierne in 1962 for use in hot working studies [Hodierne, 1962] and popularized by Lindholm et al. [1980]. Figure 1 shows a generic sketch of this type of specimen. Notice the short, thin wall gauge section which quickly transitions to the thick walled shoulder region where the specimen is gripped.

Torsion test results have been originally used to compare normalized flow stress behavior with tension/compression results [Hecker, 1982]. A landmark paper by Nagtegaal and deJong [1982] showed that simple shear deformation can yield startling stress predictions which are very sensitive to material model. These results show the importance of simple shear deformation for discriminating material behavior, especially kinematic hardening. Nagtegaal and deJong showed that classical Prager type kinematic hardening predicts very

large normal stresses in simple shear. The stress components are seen to oscillate with plastic strain for very large shearing due to the objective stress rate (Jaumann) which was used. In the past decade there has been intensive work aimed at describing material behavior in large simple shear and even formulating constitutive models with the specific goal of describing simple shear in a more intuitively acceptable manner. In light of these results the torsion test has a new importance insofar as it provides an approximation to simple shear deformation. Just how close an approximation this is will be examined below.

The kinematics of the finite strain tension torsion of a thin-walled tube have been examined in detail by McMeeking [1982]. For a uniformly deforming tube, the stretching tensor (symmetric part of the velocity gradient) can be written in curvilinear coordinates as:

$$D = \begin{bmatrix} \dot{t}/t & 0 & 0 \\ 0 & \dot{r}/r & \dot{\gamma}/2 \\ 0 & \dot{\gamma}/2 & \dot{\varepsilon} \end{bmatrix},$$

where the 1, 2, and 3 directions are in the radial, hoop and axial directions, respectively. The torsion test provides an approximation to simple shear only as well as it restrains the changes in wall thickness, t, mean radius, r, and axial strain , ε, to be zero. The geometry shown in Figure 1 attempts to enforce the gauge section to be free from normal stretches by the presence of the large shoulders near the gauge region. The massive shoulders provide the restraint against changes in the radius. The shoulders also provide the restraint against axial deformation. The grips can be restrained against axial motion to within the stiffness of the testing machine frame. The thick walls of the shoulder region transmit this axial stiffness to the gauge region. The effectiveness of this geometry to restrain the radial, hoop and axial normal straining is one of the primary things to be determined in evaluating the torsion test.

The thin walled torsion specimen has been used largely without investigation, either experimentally or numerically as to its effectiveness in simulating simple shear. The single exception is Lipkin, Chiesa and Bammann [1987] who conducted a numerical analysis of their test specimen using the DYNA3D finite element hydrocode. This analysis will be referred to below in comparison with the present results but a few comments are in order.

A number of features of this type of specimen were first observed. The gauge length shows a slight lengthening during torsion even for perfectly fixed specimen ends. Also, the shear strain at an element in the gauge section can differ from the average shear strain calculated from the twist of the grips.

This analysis was conducted for the express purpose of comparing experimental results of the authors with a particular constitutive model. As such, no comparison was made with better understood, more classical material models.

The finite element mesh that was used was very coarse containing only 3 elements through the thickness. No mesh convergence comparisons were reported.

The results of Lipkin et al. have shed new light on the behavior of the thin-walled torsion specimen but have not provided a detailed, quantitative assessment as to the limitations and usefulness of the specimen. The current analysis addresses these questions.

Finite Element Model

The specimen geometry that was simulated corresponds to the 316 stainless steel specimen reported in White, Anand and Bronkhorst [1990]. The gauge length was 5.9 mm. The inside diameter was 19.05 mm. The outside diameter of the gauge region was 20.52 mm. The outside diameter of the shoulder region was 38.1 mm.

The finite element analysis was conducted with the ABAQUS finite element program [ABAQUS, 1989]. All of the elements were eight node, linear displacement bricks having a full eight material integration points (C3D8 elements in ABAQUS). Two different types of mesh designs were employed. These are shown in Figure 2.

The first type of mesh is one that represented the full three-dimensional geometry of the specimen, Fig. 2a. Only one half of the specimen was discretized due to symmetry about the midplane. The mesh had 5 elements through the thickness of the specimen wall. In the one-half gauge region 8 elements were used in the axial discretization but were not evenly spaced, they were smaller near the shoulder than at the specimen midsection. The mesh was discretized into 12 rows of elements around the circumference. Thus each element covered 30 degrees of arc. The applied boundary conditions were very simple. The nodes on the specimen midplane were restrained against movement in the axial, and local hoop directions but were allowed free movement in the radial direction. The outermost ring of nodes at the top of the shoulder region farthest from the gauge section was constrained to move in a circle at the original radius. This was to simulate the twist applied to the specimen by the hydraulic, collet grips of the testing machine. All nodes on the plane furthest from the midplane were restrained against axial motion. All other free motion of the specimen boundaries was allowed without constraint.

The second type of mesh, Fig. 2b, discretized the specimen into just one circumferential slice of either one or five degree extent. A fine mesh within that slice was used. Each plane had 9 elements through the wall thickness and 15 along the axial length of the one-half gauge length. Again, the nodes on the midplane of the gauge region were restrained against axial or circumferential motion but radial motion was allowed. The nodes along the top face of the shoulder region were restrained against axial motion. The outer node on the top face was constrained to move in a circular arc simulating the applied twist. The compatibility enforced on this strip to make it simulate an entire circumference was by requiring the initially straight radial lines of nodes to remain on straight lines and by requiring the corresponding nodes on the two faces of the strip to retain the prescribed circumferential angle (either one or five degrees) between them.

The material models that were used in the simulations were classical, large strain plasticity laws: isotropic hardening, kinematic hardening, and perfect plasticity. The material was assumed to behave elastically with Young's modulus of 200. GPa and Poisson's ratio of 0.33 up to a yield stress of 250. MPa. Above this stress the elastic plastic behavior had a constant plastic hardening modulus of 1500. MPa. The objective stress rate employed was the Jaumann derivative. These material models satisfactorily bound the behavior both of experimental results and of most of the constitutive laws in the literature.

Convergence studies were conducted of the step size and nodal force tolerance in the acceptance criterion for the finite element studies. The automatic load stepping was used in full large deformation analysis. Approximately 200 displacement increments were used to twist the specimen to a nominal engineering shear strain of unity.

Results

The majority of the comparisons of the numerical results are shown in terms of macroscopic variables that would be determined in an experiment. The nominal shear stress in the gauge section was determined by summing the reaction forces in the circumferential direction on the top face of the specimen at the shoulder (the applied torque) and dividing by the cross sectional area and average radius of the gauge section. The axial, normal stress was determined in similar manner by summing the axial reaction force on the top face of the specimen and dividing by the gauge cross sectional area. The average strains in the gauge section were determined assuming an extensometer could measure the circumferential and axial displacements on the outside of the gauge section at the intersection with the transition region to the shoulder. The nominal axial strain was determined by dividing the axial displacement by the original gauge length and the nominal average shear strain (engineering) by dividing the arc displacement by the gauge length. These numerical results then can be compared with the variables measured in an experimental test.

In results not shown here, the different models described above were compared for calculations using both isotropic and kinematic hardening. The macroscopic stress-strain results and the stress and strain contours within the specimen were virtually identical. The models: full circumferential, one degree slice and five degree slice all yielded results that were almost indistinguishable. This was taken as verification that the mesh was sufficiently fine and the boundary conditions for the single slice meshes were appropriate. Most of the succeeding results that will be discussed were obtained with the single slice model having five degrees circumferentially.

In order to evaluate how well the thin walled torsion specimen approximates simple shear a comparison is made between the shear and normal stress response inferred from the finite element simulation with that for the same constitutive models integrated directly assuming only simple shear deformation. The results of this comparison are shown in Figure 3 for both isotropic and kinematic hardening. For isotropic hardening we see an almost exact correlation. No normal stress develops and the shear stress linearly increases with strain. For kinematic hardening (using Jaumann derivatives) a substantial axial normal stress is predicted both by the finite element model of the specimen and the assumed simple shear deformation. The finite element results an axial normal stress approximately 20% larger than predicted for simple shear. This provides a good way to estimate how well the specimen approximates simple shear. This kind of result is okay but not terrific. Notice that the shear stress is quite well correlated by the specimen and simple shear.

Stress contours are provided in Figure 4, for kinematic hardening, to demonstrate what kind of variability exists in the gauge sections. Notice that the shear stress is quite uniform except right at the shoulder transition. The axial normal stress and the hoop normal stress show a much larger variation across the wall of the gauge region. For a perfect simple shear specimen these would be constant and of equal magnitude and opposite sign with the axial stress being compressive and the hoop stress tensile. In the specimen the tensile hoop stress tries to pull the radius of the gauge in to the center and creates a slight bending in the wall which creates the stress nonuniformity. For the geometry shown we do notice that there is a cross section through the wall that has a fairly constant stress both for the axial and hoop stresses. This cross section is about one wall thickness away from the shoulder transition. At this location simple shear appears to be maintained quite well. Thus the

macroscopic variables do give pretty good results.

The reason for the increased axial normal stress when compared to simple shear (Fig. 3) can be traced to the fact that the plastic deformation is not completely contained in the gauge section. Figure 5 shows the contours of equivalent plastic strain at a shear strain of approximately 78% . Notice that the contour of 1% plastic strain extends into the transition region for about one-half gauge length. Thus the plastic deformation is not completely contained in the gauge section. One effect of this is that the axial restraint is not perfect. Figure 6 shows the development of average axial strain across the gauge length with shear strain. For the three plasticity laws considered, an extension of the gauge length is observed during testing. The magnitude of this strain is small but it is enough to increase the axial stress, especially for kinematic hardening.

Another effect of the plastic strain extending into the transition shoulder region is upon the shear strain. In the results presented above, the shear strain was calculated by taking the twist measured across the gauge section to convert into strain. In experimental practice, the twist applied by the testing machine has been assumed to be entirely transmitted into the gauge section and hence it has been used to calculate the shear strain. From the finite element results we compare the twist measured across the gauge section with the twist applied across the entire specimen. In Figure 7 the ratio of these twists is plotted against rotation. For the geometry considered here, the ratio is independent of hardening model. It is independent of applied strain after an initial transient. The rotation ratio does depend slightly whether it is measured along the inner surface or the outer surface of the specimen. For this specimen, about 78% of the twist that is applied at the grips actually goes into the deformation in the gauge section. It is very fortuitous that this ratio is independent of hardening model and deformation level. A simple correction factor can be applied to the experimentally measured twist in converting to shear strain (just multiply by 0.78). This correction factor is dependent upon the particular specimen geometry but can be easily evaluated from finite element modeling. Of course, experimentally one would like to have a rotational extensometer to measure the twist in the gauge section. This correction factor is the next suitable approach.

Conclusions

A finite element model has been used to evaluate the thin-walled torsion specimen. The specimen is shown to provide a reasonable approximation to simple shear. In evaluating the specimen response to twisting the following observations are made:

The shear strains are almost constant in the gauge section.

The axial strain across the gauge section increases with deformation but remains small for fixed grips.

The shear stress is quite uniform in the gauge section.

The axial and hoop normal stresses have a large variation across the wall thickness except at one location where they are uniform.

The plastic deformation extends into the shoulder transition region and has the main effect of reducing the twist that the gauge section experiences to a fraction of the twist applied by the grips. This factor is well characterized and can be used to correct experimental results.

Future work is needed to optimize specimen geometry. The effect of gauge length needs to be explored as well as the onset of torsional buckling. Experimentally, the largest need is to be able to directly measure the strain in the gauge region.

References

ABAQUS Users Manual, Hibbitt Karlsson and Sorenson, Inc., Providence, RI, 1989.

CRANDALL, S.H., DAHL, N.C. AND T.J. LARDNER, **An Introduction to the Mechanics of Solids, McGraw-Hill, Second Edition**, 1972.

HECKER, S.S., STOUT, M.G. AND D.T. EASH, *Experiments on Plastic Deformation at Finite Strains*, **Plasticity of Metals at Finite Strain: Theory, Experiment and Computation**, E.H. Lee and R.L. Mallett, eds., Proceedings of Research Workshop held at Stanford University, June 29 - July 1, 1981, pp. 162-205, 1982.

LINDHOLM, U.S., NAGY, A., JOHNSON, G.R., AND J.M. HOEGFELDT, *Large Strain, High Strain Rate Testing of Copper,* ASME Journal of Engineering Materials and Technology, **102**, pp. 376-381,1980.

LIPKIN, J., CHIESA, M.L. AND D.J. BAMMANN, *Thermal Softeningof 304L Stainless Steel: Experimental Results and Numerical Simulations*, Proceedings of IMPACT '87, Bremen, FRG, May, 1987.

MCMEEKING, R.M., *The Finite Strain Tension Torsion Test of a Thin-Walled Tube of Elastic-Plastic Material*, International Journal of Solids and Structures, **18**, pp. 199-204, 1982.

NAGTEGAAL, J.C. AND J.E. DEJONG, *Some Aspects of Non-Isotropic Workhardening in Finite Strain Plasticity*, **Plasticity of Metals at Finite Strain: Theory, Experiment and Computation**, E.H. Lee and R.L. Mallett, eds., Proceedings of Research Workshop held at Stanford University, June 29 - July 1, 1981, pp. 55, 1982.

PHILLIPS, A. AND W.-Y. LU, *An Experimental Investagation of Yield Surfaces of Pure Aluminum with Stress-Controlled and Strain-Controlled Paths of Loading*, ASME Journal of Engineering Materials and Technology, **106**, p. 349, 1984.

TAYLOR, G.I. AND H. QUINNEY, Phil. Trans. Roy. Soc. A, **230**, pp. 323, 1931.

TRESCA, H., *Memoire sur L'ecoulement des Corps Solides soumis a des Fortes Pressions*, C. Rend. Paris, **59**, pp. 754, 1864.

WHITE, C.S, ANAND, L. AND C. BRONKHORST, *An Improved Isotropic-Kinematic Hardening Model for Moderate Deformation Metal Plasticity*, Mechanics of Materials, to appear.

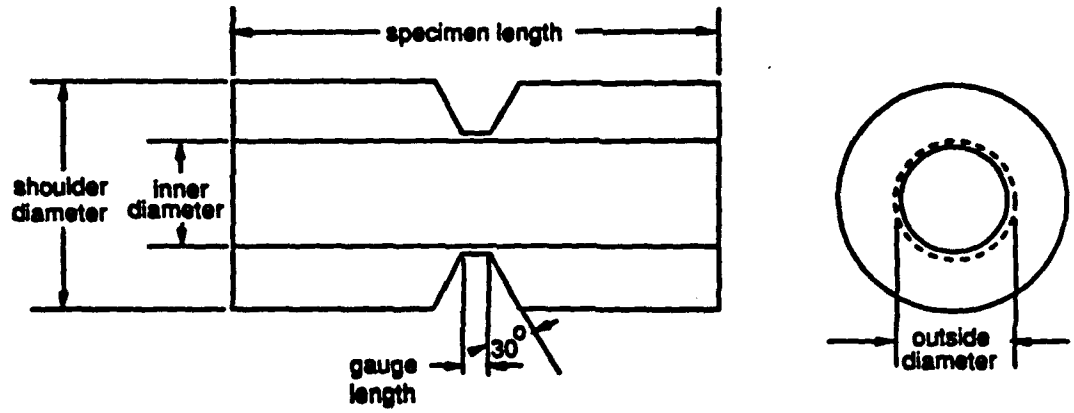
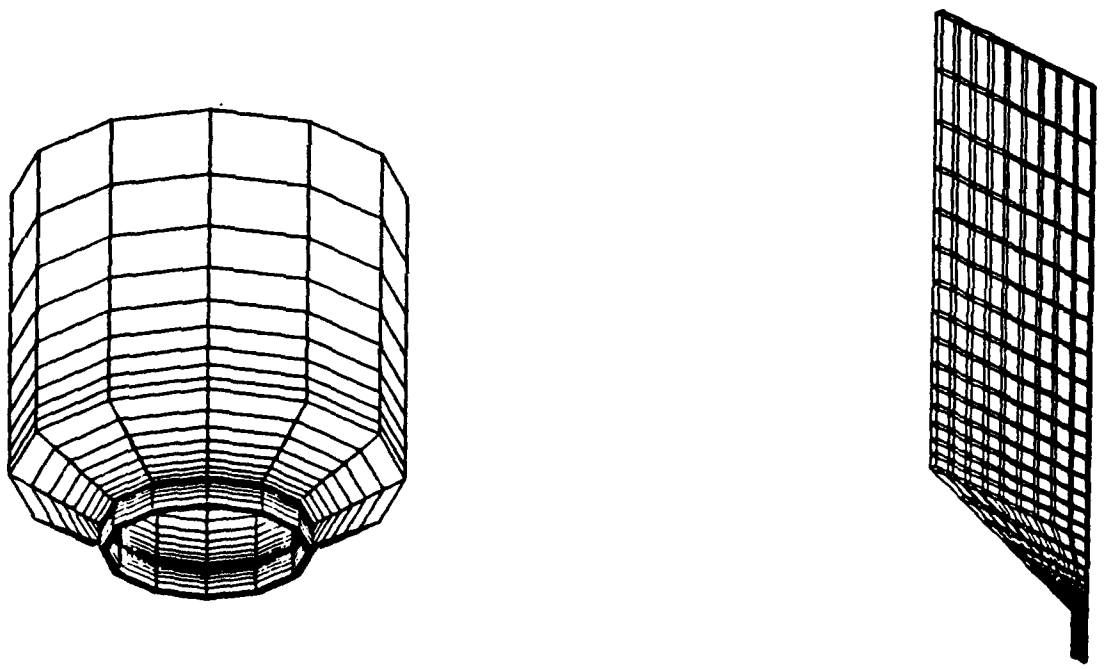


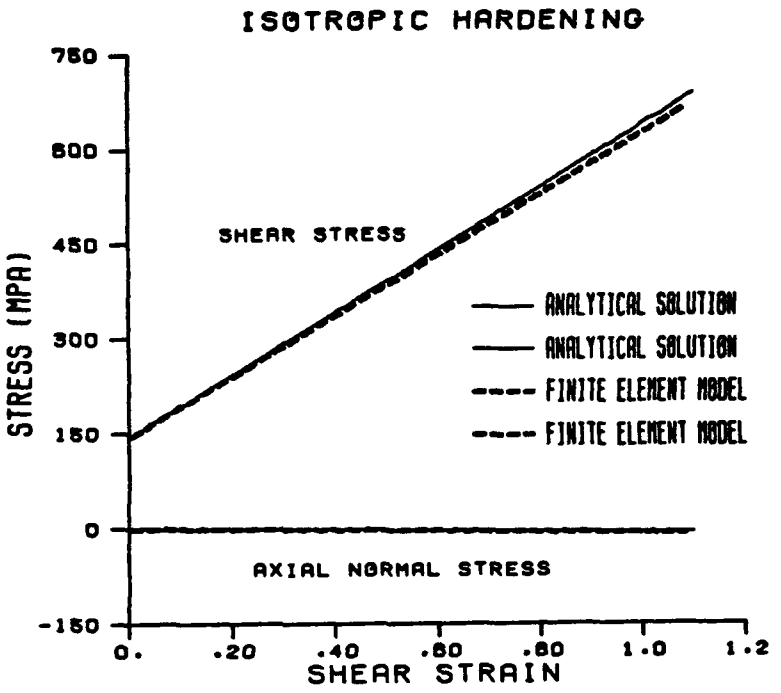
Figure 1, Geometry of the Thin-Walled Torsion Specimen.



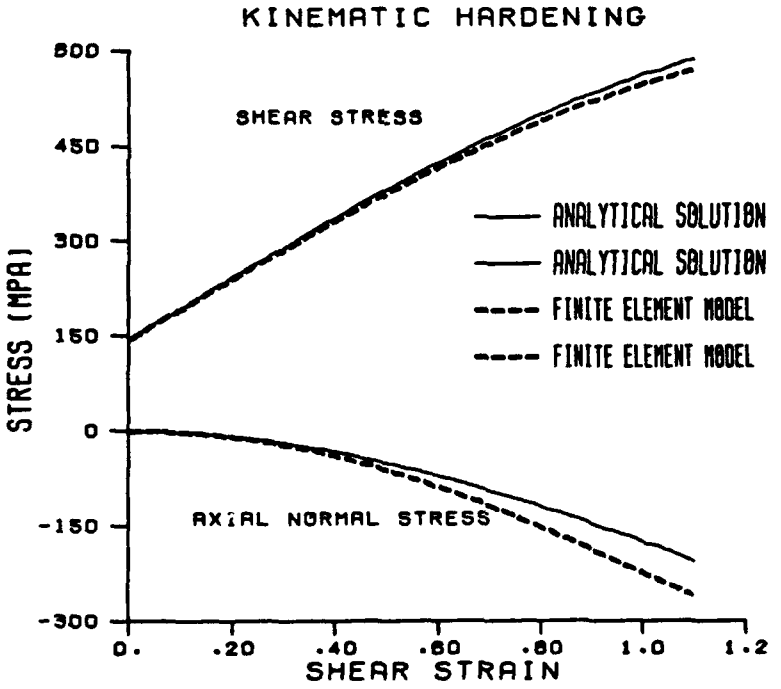
a) Full Circumferential Mesh.

b) Mesh of Single Slice of Specimen.

Figure 2, Finite Element Meshes Used to Simulate the Torsion Specimen.
a) Full Circumferential Mesh.
b) Mesh of Single Slice of Specimen.

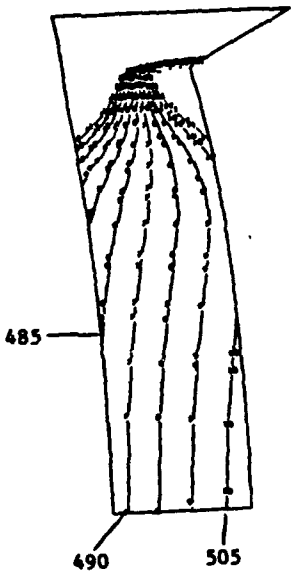


a) Isotropic Hardening

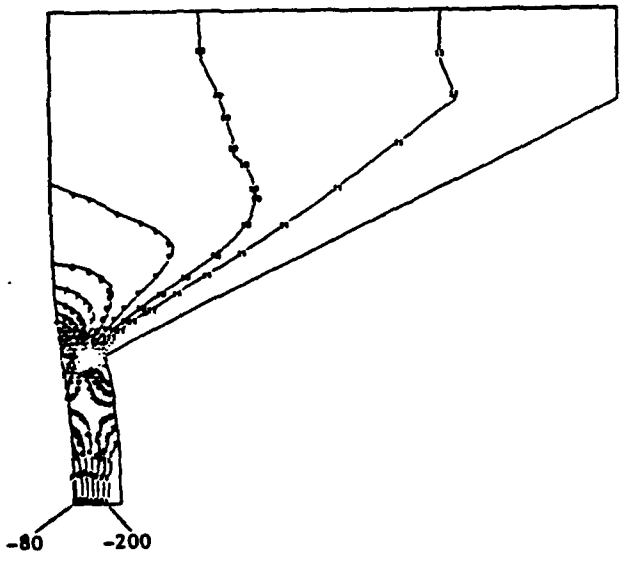


b) Kinematic Hardening

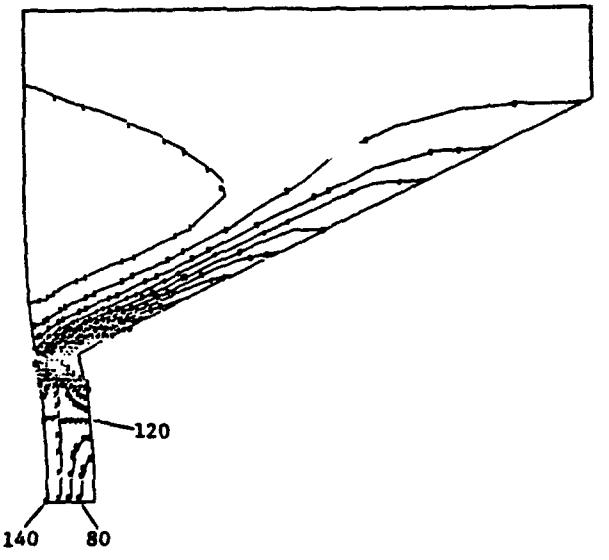
Figure 3, Comparison of Finite Element Simulation with Simple Shear for:
a) Isotropic Hardening, and
b) Kinematic Hardening.



Shear Stress Distribution



Axial Normal Stress Distribution



Hoop Normal Stress Distribution

Figure 4, Stress Contours (MPa) in Gauge Region for Kinematic Hardening.

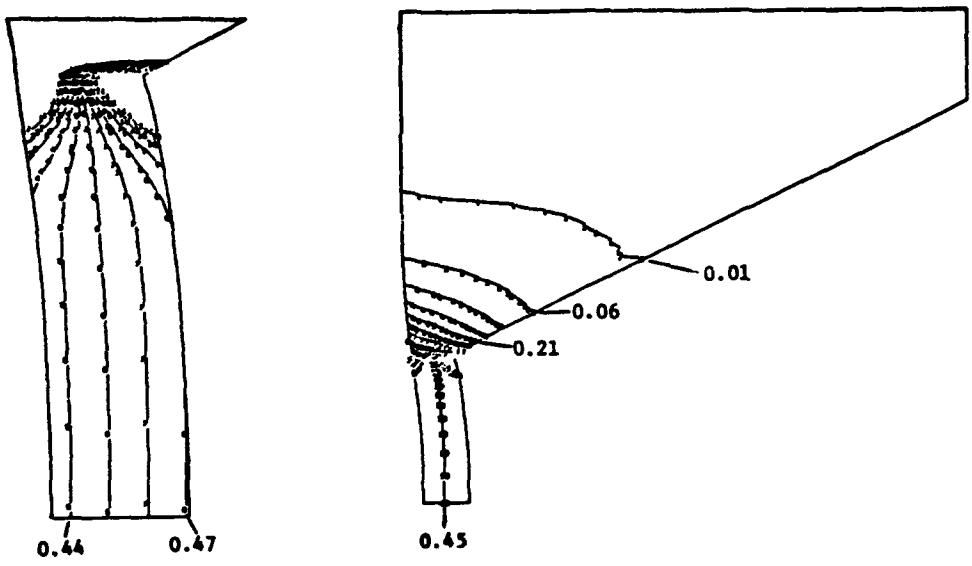


Figure 5, Contours of Equivalent Plastic Strain.

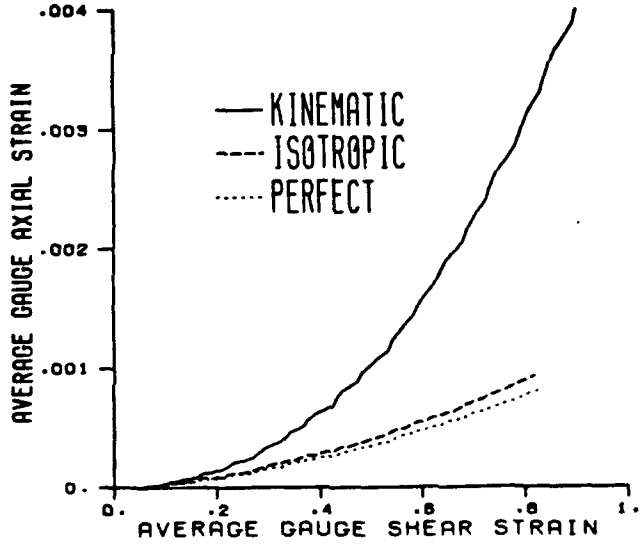


Figure 6, Development of Axial Strain in the Gauge Section with Deformation.

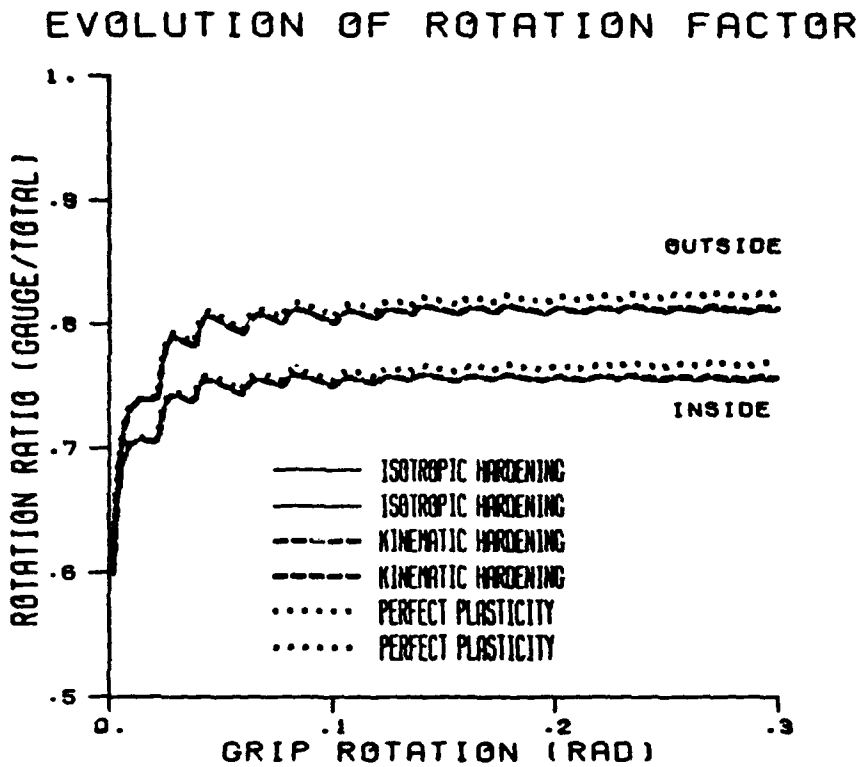


Figure 7, Ratio of Twist in the Gauge Section to Total Twist Applied by Grips as a Function of Applied Twist for Various Hardening Rules.

WRAPPABILITY OF CURVES ON SURFACES

Royce W. Soanes
U.S. Army Armament Research, Development, and Engineering Center
Close Combat Armaments Center
Benet Laboratories
Watervliet, NY 12189-4050

ABSTRACT. In this paper, conditions are derived under which a path on a general smooth surface is wrappable or capable of receiving an essentially one-dimensional flexible filament under tension that clings to the surface throughout its length and does not slip. Wrappability considerations are of practical importance in the fabrication of filament-wound composite pressure vessels, for instance. The general wrappability conditions derived are applied to two special cases: general cylinders and general surfaces of revolution.

INTRODUCTION. Imagine a rotating spindle accepting string from some delivery point which moves parallel to the axis of the spindle. This is the essence of the filament winding process where the "string" is replaced by a band of epoxy impregnated fiber glass, for instance; layer upon layer of this filament is evenly laid down; and the whole thing is ultimately cured or baked into a filament-wound composite structure - often a pressure vessel. The spindle or mandrel is designed so that it may be broken down into parts and removed subsequent to curing, leaving only the wrappings embedded in the matrix material. This paper considers the question of how the winding or wrapping process is limited by the differential geometric nature of the mandrel's surface.

PRELIMINARIES. Begin with point P in three space:

$$P = iX + jY + kZ$$

Restrict P by parameterizing with respect to x and θ, defining a surface S embedded in three space:

$$X = x$$

$$Y = r(x,\theta)\sin\theta$$

$$Z = r(x,\theta)\cos\theta$$

where $r(x,\theta)$ is the radius of the surface. We require r to be sufficiently smooth, positive, and 2π periodic in θ, making S a closed surface with an inside and an outside. If P is further restricted by defining θ in terms of x, P will lie on a curve or path embedded in the surface S.

Let $(\ )'$ denote $\frac{d}{dx}(\ )$ and let s denote distance along curve c. The tangent vector t to curve c is

$$t = \frac{dP}{ds} = P'(s')^{-1}$$

and the curvature vector κ of any curve c is

$$\kappa = \frac{dt}{ds} = t'(s')^{-1}$$

$$= (P''-ts'')(s')^{-2}$$

A family of vectors tangent to surface S at point P is

$$dP = P_x dx + P_\theta d\theta = \frac{\partial P}{\partial x} dx + \frac{\partial P}{\partial \theta} d\theta$$

Two independent vectors spanning the tangent space at P are therefore P_x and P_θ. Now form the vector cross product of P_x and P_θ to obtain ν, a vector normal to the surface and pointing away from the outside of the surface.

$$\nu = P_x \times P_\theta$$

WRAPPABILITY CONDITION I - NO LIFTOFF. Now, in order for c to be a wrappable curve on surface S, it is necessary for a length of flexible filament under tension to cling to c and S. In order for this clinging to take place, it is necessary for c to see the outside of S as being convex. This will be the case only if the curvature vector of c points away from the inside of the surface. It is therefore necessary that the inner or dot product of ν and κ be negative for clinging to take place. Now,

$$\nu \cdot \kappa = \nu \cdot (P''-ts'')(s')^{-2}$$

but

$$\nu \cdot t = 0$$

therefore,

$$\nu \cdot \kappa = \nu \cdot P''(s')^{-2}$$

and since only the sign of $\nu \cdot \kappa$ matters here, the function λ is defined as

$$\lambda = \nu \cdot P''$$

When λ is positive, a filament under tension tends to lift off the surface and form a bridge between two distant points; when λ is negative, the filament tends to cling to the surface.

Now, the evaluation of λ in terms of x, θ, and r is outlined.

First,

$$P' = P_x + P_\theta \theta'$$

and

$$P'' = P_{xx} + 2P_{x\theta}\theta' + P_{\theta\theta}\theta'^2 + P_\theta \theta''$$

but P_θ is in the tangent space, so

$$\nu \cdot P_\theta = 0$$

Hence,

$$\lambda = \nu \cdot P'' = \nu \cdot P_{xx} + 2\nu \cdot P_{x\theta}\theta' + \nu \cdot P_{\theta\theta}\theta'^2$$

Note that all the inner products defining λ are determined at a point on the surface independently of the curve c and that the only thing that changes λ at a point is the direction of c determined by θ'. Therefore, all curves with the same direction through a given point on the surface have the same value of λ at that point.

Continuing the evaluation of $\nu \cdot P''$ we have

$$P_x = i + jY_x + kZ_x$$

$$P_\theta = jY_\theta + kZ_\theta$$

$$\nu = P_x \times P_\theta = i(Y_x Z_\theta - Y_\theta Z_x) - jZ_\theta + kY_\theta$$

$$P_{xx} = jY_{xx} + kZ_{xx}$$

$$P_{x\theta} = jY_{x\theta} + kZ_{x\theta}$$

$$P_{\theta\theta} = jY_{\theta\theta} + kZ_{\theta\theta}$$

The dot products in λ are therefore

$$\nu \cdot P_{xx} = Y_\theta Z_{xx} - Z_\theta Y_{xx}$$

$$\nu \cdot P_{x\theta} = Y_\theta Z_{x\theta} - Z_\theta Y_{x\theta}$$

$$\nu \cdot P_{\theta\theta} = Y_\theta Z_{\theta\theta} - Z_\theta Y_{\theta\theta}$$

Obtaining the partials in these dot products

$$Y = r \sin \theta$$

$$Y_x = Yr_x/r$$

$$Y_\theta = Yr_\theta/r + Z$$

$$Y_{xx} = Yr_{xx}/r$$

$$Y_{x\theta} = (Yr_{x\theta}+Zr_x)/r$$

$$Y_{\theta\theta} = \{Y(r_{\theta\theta}-r) + 2Zr_\theta\}/r$$

$$Z = r \cos \theta$$

$$Z_x = Zr_x/r$$

$$Z_\theta = Zr_\theta/r - Y$$

$$Z_{xx} = Zr_{xx}/r$$

$$Z_{x\theta} = (Zr_{x\theta}-Yr_x)/r$$

$$Z_{\theta\theta} = \{Z(r_{\theta\theta}-r) - 2Yr_\theta\}/r$$

The dot products then become

$$\nu \cdot P_{xx} = rr_{xx}$$

$$\nu \cdot P_{x\theta} = rr_{x\theta} - r_x r_\theta$$

$$\nu \cdot P_{\theta\theta} = rr_{\theta\theta} - 2r_\theta^2 - r^2$$

Hence,

$$\lambda = rr_{xx} + 2(rr_{x\theta}-r_x r_\theta)\theta' + (rr_{\theta\theta}-2r_\theta^2-r^2)\theta'^2$$

$$= a\theta'^2 + 2b\theta' + c$$

$$= a(\theta' + \frac{b}{a})^2 + c - \frac{b^2}{a}$$

It is clear that the sign of λ is completely independent of θ' at points for which $ac > b^2$. Hence, any curve is wrappable where $ac > b^2$ and $a < 0$, but the surface is unwrappable if $ac > b^2$ and $a > 0$ anywhere. If $ac < b^2$, some curves will be wrappable and others won't. In any case, given the radius function r, its partial derivatives at a point, and the direction of a curve through that point, one can immediately compute whether or not a taut filament following the curve will tend to lift from the surface.

Consider two special cases. For a surface of revolution, $r_\theta = 0$. Therefore, $\lambda = rr'' - (r\theta')^2$ [1]. If $r'' < 0$ everywhere, all curves on the

surface of revolution are wrappable. Points for which r" > 0 are also wrappable for curves with θ' sufficiently large. No surface of revolution is unwrappable in the sense of liftoff.

For a cylinder, $r_x = 0$, and

$$\lambda = a\theta'^2 = (rr_{\theta\theta}-2r_\theta^2-r^2)\theta'^2$$

Hence, every curve on a cylinder is wrappable if a < 0 everywhere, but if a > 0 anywhere, the cylinder is unwrappable.

In this section the phrase "is wrappable" has meant "does not experience filament liftoff or bridging during winding." In the next section, the definition of "wrappable" is augmented by considering friction between filament and surface.

WRAPPABILITY CONDITION II - NO SLIPPAGE. If there were no friction between filament and surface (or between filament and filament since the surface is filament after the first layer is laid down), there would be only one type of path along which one might wind filament without the filament slipping - a path with no transverse (tangent to the surface and perpendicular to the filament) forces acting on the filament - a path which curves neither left nor right in the surface - a path with zero geodesic curvature - a geodesic path. If there were no friction available, and only geodesic paths could be wound, there would be no filament winding industry. In fact, for numerous reasons, it is seldom if ever possible to wind along geodesic paths in practice [1]. The geodesic path remains as an ideal, however, and in this section the degree of closeness to this ideal is quantified.

Let

ϕ = force per unit length that filament exerts on surface = $\tau\kappa$

where τ is the scalar tension in the filament and κ is the vector curvature of the filament.

Note that the curvature vector can be resolved into a component tangent to the surface and a component normal to the surface $\kappa = \kappa_g + \kappa_n$ [2] where the tangent component of κ is called the geodesic curvature vector and the normal component is called the normal curvature vector.

The force that a small length of filament exerts on the surface is

$$\phi ds = \tau\kappa ds = (\kappa_g+\kappa_n)\tau ds = \kappa_g\tau ds + \kappa_n\tau ds$$

Now, in order to avoid slippage of this small section of filament, the ratio of the magnitude of the tangent force to the magnitude of the normal force should be less than μ, the coefficient of friction

$$\sigma = \frac{\|\kappa_g\|\tau ds}{\|\kappa_n\|\tau ds} = \left|\frac{k_g}{k_n}\right| < \mu$$

(or more precisely, $0 \leq \sigma = \frac{-|k_g|}{k_n} < \mu$, since we want $k_n < 0$). We call this ratio of geodesic to normal curvature the slippage function σ. This function measures how close a given path comes to the ideal geodesic path ($\sigma \equiv 0$). The evaluation of σ is now detailed. First, if P is on the surface,

$$dP = P_x dx + P_\theta d\theta$$

and

$$ds^2 = dP \cdot dP = P_x \cdot P_x dx^2 + 2P_x \cdot P_\theta dx d\theta + P_\theta \cdot P_\theta d\theta^2$$

$$= E dx^2 + 2F dx d\theta + G d\theta^2$$

Hence,

$$(s')^2 = E + 2F\theta' + G\theta'^2$$

for a point on a path on the surface.

We now define a Cartan frame [2-4] relative to the surface, i.e., an orthonormal basis having two vectors tangent to the surface and a third normal to it. Let

$$e_1 = P_x/\|P_x\|$$

$$= P_x/(P_x \cdot P_x)^{1/2}$$

$$= P_x/E^{1/2}$$

and

$$e_3 = P_x \times P_\theta/\|P_x \times P_\theta\| = \nu/\|\nu\|$$

but from vector algebra,

$$(A \times B) \cdot (C \times D) = (A \cdot C)(B \cdot D) - (A \cdot D)(B \cdot C)$$

Hence,

$$(P_x \times P_\theta) \cdot (P_x \times P_\theta) = \|P_x \times P_\theta\|^2 = \|\nu\|^2$$

$$= (P_x \cdot P_x)(P_\theta \cdot P_\theta) - (P_x \cdot P_\theta)^2$$

$$= EG - F^2$$

Therefore,

$$e_3 = P_x \times P_\theta / (EG-F^2)^{\frac{1}{2}}$$

Now

$$e_2 = e_3 \times e_1$$

$$= (P_x \times P_\theta) \times P_x / (E^{\frac{1}{2}}(EG-F^2)^{\frac{1}{2}})$$

but again from vector algebra,

$$(A \times B) \times C = B(A \cdot C) - A(B \cdot C)$$

Hence,

$$(P_x \times P_\theta) \times P_x = P_\theta(P_x \cdot P_x) - P_x(P_x \cdot P_\theta)$$

$$= EP_\theta - FP_x$$

and

$$e_2 = (EP_\theta - FP_x)/(E^{\frac{1}{2}}(EG-F^2)^{\frac{1}{2}})$$

Now, e_1 and e_2 span the tangent space (plane) at any point; therefore, the path tangent vector t can be written

$$t = Ae_1 + Be_2$$

If t makes an angle ω with e_1,

$$e_1 \cdot t = A = \cos \omega$$

and

$$e_2 \cdot t = B = \sin \omega$$

Hence

$$t = e_1 \cos \omega + e_2 \sin \omega$$

where ω is the angle between the path and a meridian (θ = constant). Now

$$\kappa = \frac{dt}{ds} = \frac{de_1}{ds} \cos \omega + \frac{de_2}{ds} \sin \omega + (e_2 \cos \omega - e_1 \sin \omega) \frac{d\omega}{ds}$$

Let

$$T = e_3 \times t$$

$$= e_3 \times (e_1 \cos \omega + e_2 \sin \omega)$$

$$= e_2 \cos \omega - e_1 \sin \omega$$

Therefore,

$$\kappa = \frac{d\underline{e}_1}{ds} \cos \omega + \frac{d\underline{e}_2}{ds} \sin \omega + T \frac{d\omega}{ds}$$

and

$$T \cdot \kappa = T \cdot \frac{d\underline{e}_1}{ds} \cos \omega + T \cdot \frac{d\underline{e}_2}{ds} \sin \omega + \frac{d\omega}{ds}$$

but since

$$e_1 \cdot e_1 = e_2 \cdot e_2 = 1$$

one has

$$e_1 \cdot \frac{d\underline{e}_1}{ds} = e_2 \cdot \frac{d\underline{e}_2}{ds} = 0$$

and since

$$e_1 \cdot e_2 = 0$$

one has

$$\frac{d\underline{e}_1}{ds} \cdot e_2 = - \frac{d\underline{e}_2}{ds} \cdot e_1$$

We therefore have

$$T \cdot \frac{d\underline{e}_1}{ds} = e_2 \cdot \frac{d\underline{e}_1}{ds} \cos \omega$$

and

$$T \cdot \frac{d\underline{e}_2}{ds} = -e_1 \cdot \frac{d\underline{e}_2}{ds} \sin \omega = e_2 \cdot \frac{d\underline{e}_1}{ds} \sin \omega$$

Therefore

$$T \cdot \kappa = e_2 \cdot \frac{d\underline{e}_1}{ds} \cos^2 \omega + e_2 \cdot \frac{d\underline{e}_1}{ds} \sin^2 \omega + \frac{d\omega}{ds} = e_2 \cdot \frac{d\underline{e}_1}{ds} + \frac{d\omega}{ds}$$

but

$$\kappa = \kappa_g + \kappa_n \quad \text{and} \quad T \cdot \kappa_n = 0$$

hence,

$$T \cdot \kappa = T \cdot \kappa_g = \|\kappa_g\| = k_g$$

and the geodesic curvature is

$$k_g = e_2 \cdot \frac{d\underline{e}_1}{ds} + \frac{d\omega}{ds} = (e_2 \cdot e_1' + \omega')/s'$$

Now

$$\sigma = \left| \frac{k_g}{k_n} \right|$$

so k_n must be computed, but most of the work has already been done to find k_n.

$$k_n = e_3 \cdot \kappa = \frac{\nu}{\|\nu\|} \cdot \kappa = \frac{\nu \cdot P''}{\|\nu\|(\dot{s}')^2} = \frac{\lambda}{\|\nu\|(\dot{s}')^2}$$

Therefore,

$$\sigma = \left| \frac{e_2 \cdot e_1' + \omega'}{s'} \cdot \frac{\|\nu\|(\dot{s}')^2}{\lambda} \right|$$

$$= \left| \frac{s' \|\nu\|}{\lambda} (e_2 \cdot e_1' + \omega') \right|$$

At this point, a few dot products must be computed. We have the identity

$$(P_u \cdot P_v)_w = P_v \cdot P_{uw} + P_u \cdot P_{vw}$$

Letting v = u,

$$P_u \cdot P_{uw} = \tfrac{1}{2}(P_u \cdot P_u)_w$$

Therefore

$$P_x \cdot P_{x\theta} = \tfrac{1}{2}E_\theta$$

$$P_\theta \cdot P_{\theta x} = \tfrac{1}{2}G_x$$

$$P_x \cdot P_{xx} = \tfrac{1}{2}E_x$$

and

$$P_\theta \cdot P_{\theta\theta} = \tfrac{1}{2}G_\theta$$

Letting w = u,

$$(P_u \cdot P_v)_u = P_v \cdot P_{uu} + P_u \cdot P_{uv}$$

$$= P_v \cdot P_{uu} + \tfrac{1}{2}(P_u \cdot P_u)_v$$

or

$$P_v \cdot P_{uu} = (P_u \cdot P_v)_u - \tfrac{1}{2}(P_u \cdot P_u)_v$$

Hence

$$P_x \cdot P_{\theta\theta} = F_\theta - \tfrac{1}{2}G_x$$

and

$$P_\theta \cdot P_{xx} = F_x - \tfrac{1}{2}E_\theta$$

Recalling

$$e_1 = P_x E^{-\frac{1}{2}}$$

and applying d/dx,

$$e_1' = P_x' E^{-\frac{1}{2}} + P_x (E^{-\frac{1}{2}})'$$

$$= (P_{xx} + P_{x\theta}\theta') E^{-\frac{1}{2}} + P_x (E^{-\frac{1}{2}})'$$

but

$$e_2 \cdot P_x = 0$$

Therefore

$$e_2 \cdot e_1' = (e_2 \cdot P_{xx} + e_2 \cdot P_{x\theta}\theta') / E^{\frac{1}{2}}$$

Now,

$$e_2 \cdot P_{xx} = (EP_\theta - FP_x) \cdot P_{xx} / (E^{\frac{1}{2}} \| \nu \|)$$

$$= (E(F_x - \tfrac{1}{2}E_\theta) - F(\tfrac{1}{2}E_x)) / (E^{\frac{1}{2}} \| \nu \|)$$

$$e_2 \cdot P_{x\theta} = (EP_\theta - FP_x) \cdot P_{x\theta} / (E^{\frac{1}{2}} \| \nu \|)$$

$$= (E(\tfrac{1}{2}G_x) - F(\tfrac{1}{2}E_\theta)) / (E^{\frac{1}{2}} \| \nu \|)$$

and therefore

$$e_2 \cdot e_1' = \{E(2F_x - E_\theta) - FE_x + (EG_x - FE_\theta)\theta'\} / (2E \| \nu \|)$$

Now, consider the meridian angle ω

$$e_1 \cdot t = \cos \omega$$

$$= P_x \cdot t / E^{\frac{1}{2}}$$

$$= P_x \cdot \frac{dP}{ds} / E^{\frac{1}{2}}$$

$$= P_x \cdot P' / (s' E^{\frac{1}{2}})$$

$$= P_X \cdot (P_X + P_\theta \theta') / (s' E^{1/2})$$

$$= (E + F\theta') / (s' E^{1/2})$$

Therefore,

$$\text{Tan } \omega = \frac{(E(s')^2 - (E+F\theta')^2)^{1/2}}{E + F\theta'}$$

$$= \frac{(E(E+2F\theta'+G\theta'^2) - (E^2+2EF\theta'+F^2\theta'^2))^{1/2}}{E + F\theta'}$$

$$= \frac{(EG-F^2)^{1/2}\theta'}{E + F\theta'} = \frac{\|\nu\|\theta'}{E + F\theta'}$$

Solving for θ', we have

$$\theta' = \frac{E \text{ Tan } \omega}{(EG-F^2)^{1/2} - F \text{ Tan } \omega}$$

Therefore, σ can be computed in terms of x, θ, ω, and ω'.

The basic metric coefficients in terms of our parameterization are now computed:

$$P = iX + jY + kZ$$

but

$$P_X = i + jY_X + kZ_X$$

and

$$Y_X = Yr_X/r$$

$$Z_X = Zr_X/r$$

hence,

$$E = P_X \cdot P_X = 1 + Y_X^2 + Z_X^2 = 1 + \frac{r_X^2}{r^2}(Y^2+Z^2) = 1 + r_X^2$$

Now

$$P_\theta = jY_\theta + kZ_\theta$$

and

$$Y_\theta = Yr_\theta/r + Z$$

$$Z_\theta = Zr_\theta/r - Y$$

Hence

$$F = P_X \cdot P_\theta = Y_X Y_\theta + Z_X Z_\theta$$

$$= \frac{Yr_X}{r}\left(\frac{Yr_\theta}{r} + Z\right) + \frac{Zr_X}{r}\left(\frac{Zr_\theta}{r} - Y\right)$$

$$= \frac{r_X r_\theta}{r^2}(Y^2+Z^2)$$

$$= r_X r_\theta$$

Also

$$G = P_\theta \cdot P_\theta = Y_\theta^2 + Z_\theta^2$$

$$= \frac{Y^2 r_\theta^2}{r^2} + \frac{2YZr_\theta}{r} + Z^2$$

$$+ \frac{Z^2 r_\theta^2}{r^2} - \frac{2YZr_\theta}{r} + Y^2$$

$$= r^2 + r_\theta^2$$

Now some of the more important relations can be summarized:

$$\lambda = rr_{XX} + 2(rr_{X\theta} - r_X r_\theta)\theta' + (rr_{\theta\theta} - 2r_\theta^2 - r^2)(\theta')^2$$

$$(s')^2 = E + 2F\theta' + G(\theta')^2$$

$$\|\nu\|^2 = EG - F^2$$

$$\sigma = \left|\frac{s'\|\nu\|}{\lambda}(e_2 \cdot e_1' + \omega')\right|$$

$$e_2 \cdot e_1' = \{E(2F_X - E_\theta) - FE_X$$

$$+ (EG_X - FE_\theta)\theta'\}/(2E\|\nu\|)$$

$$\theta' = \frac{E \operatorname{Tan} \omega}{\|\nu\| - F \operatorname{Tan} \omega}$$

$$E = 1 + r_X^2$$

$$F = r_X r_\theta$$

$$G = r^2 + r_\theta^2$$

Now consider the two special cases addressed before. First, the general cylinder ($r_x = 0$):

$$E = 1 \ , \ F = 0 \ , \ G = r^2 + r_\theta^2 \ , \ G_x = 0 = E_x = E_\theta$$

$$\|v\| = G^{1/2}$$

$$e_2 \cdot e_1' = 0$$

$$\theta' = \frac{\text{Tan } \omega}{G^{1/2}}$$

$$s' = (1+\text{Tan}^2\ \omega)^{1/2} = \text{Sec } \omega$$

$$\lambda = (rr_{\theta\theta}-2r_\theta^2-r^2)\ \frac{\text{Tan}^2\ \omega}{G}$$

$$\sigma = \left| \frac{\text{Sec } \omega\ G^{1/2}\ \omega'}{(rr_{\theta\theta}-2r_\theta^2-r^2)\ \frac{\text{Tan}^2\ \omega}{G}} \right|$$

$$= \left| \frac{(r^2+r_\theta^2)^{3/2}\text{Csc } \omega \text{ Cot } \omega\ \omega'}{(rr_{\theta\theta}-2r_\theta^2-r^2)} \right| = \left| \frac{(r^2+r_\theta^2)^{3/2}}{rr_{\theta\theta} - 2r_\theta^2 - r^2} \cdot \frac{d}{dx}\ (-\text{Csc } \omega) \right|$$

hence,

$$\sigma = \frac{(r^2+r_\theta^2)^{3/2}|\ u'\ |}{r^2 + 2r_\theta^2 - rr_{\theta\theta}}$$

where $u = \text{Csc } \omega$. Note that $u = 1$ at turning points and $u > 1$ between turning points. (A turning point is defined as a point at which $\theta' = \infty$ or $i \cdot t = 0$.) Also note that at points for which $u' = 0$, the path is geodesic. In addition, $\sigma \to 0$ if $r_{\theta\theta} \to -\infty$, while r_θ and u' are bounded. This can be called a "knife edge" condition where σ is zero due to infinite normal curvature instead of zero geodesic curvature. Now, consider the general surface of revolution ($r_\theta = 0$):

$$E = 1 + r'^2 \ , \ F = 0 \ , \ G = r^2 \ , \ E_\theta = 0 = G_\theta$$

$$\|v\| = (EG)^{1/2} = r\sqrt{1 + r'^2}$$

$$\theta' = \frac{E\ \text{Tan } \omega}{\|v\|} = \left(\frac{E}{G}\right)^{1/2} \text{Tan } \omega$$

$$e_2 \cdot e_1' = EG'\theta'/(2E\|v\|)$$

$$= \frac{G'}{2(EG)^{1/2}} \cdot \left(\frac{E}{G}\right)^{1/2} \text{Tan } \omega$$

$$= \frac{G'}{2G} \text{Tan } \omega$$

$$(s')^2 = E + G \cdot \frac{E}{G} \operatorname{Tan}^2 \omega$$

$$s' = E^{\frac{1}{2}} \operatorname{Sec} \omega = (1+r'^2)^{\frac{1}{2}} \operatorname{Sec} \omega$$

$$\lambda = rr'' - r^2(\theta')^2 = rr'' - r^2 \cdot \frac{E}{G} \operatorname{Tan}^2 \omega = rr'' - (1+r'^2) \operatorname{Tan}^2 \omega$$

$$\sigma = \left| \frac{E^{\frac{1}{2}} \operatorname{Sec} \omega \, (EG)^{\frac{1}{2}}}{rr'' - E \operatorname{Tan}^2 \omega} \left(\frac{G'}{2G} \operatorname{Tan} \omega + \omega' \right) \right|$$

$$= \left| \frac{r(1+r'^2) \operatorname{Sec} \omega \operatorname{Tan} \omega}{rr'' - (1+r'^2) \operatorname{Tan}^2 \omega} \left(\frac{r'}{r} + \operatorname{Cot} \omega \, \omega' \right) \right|$$

Now,

$$\frac{r'}{r} + \operatorname{Cot} \omega \, \omega' = \frac{d}{dx} \ln r + \frac{d}{dx} \ln \operatorname{Sin} \omega = \frac{d}{dx} \ln(r \operatorname{Sin} \omega)$$

It is clear that if r Sin ω = constant, σ will be zero and the path will be geodesic (Clairaut).

One can define a quasi-geodesic path on a surface of revolution by replacing Clairaut's relation [2,3,5] with r Sin $\omega = r_o$ [1] where the function r_o has the following properties:

- $r_o(x) = r(x)$ at exactly two values of x (turning points), and
- $r_o(x) < r(x)$ at all points between the turning points.

The function r_o is called the polar radius function, because it is the radius of the surface at the boundaries of the uncovered polar regions [1]. Now, ω will be eliminated in favor of r_o. Since

$$\operatorname{Sin} \omega = \frac{r_o}{r}$$

one has that

$$\operatorname{Sec} \omega = \frac{r}{(r^2 - r_o^2)^{\frac{1}{2}}}$$

$$\operatorname{Tan} \omega = \frac{r_o}{(r^2 - r_o^2)^{\frac{1}{2}}}$$

$$\operatorname{Sec} \omega \operatorname{Tan} \omega = \frac{rr_o}{r^2 - r_o^2}$$

and

$$\frac{d}{dx}\ln r_o = \frac{r_o'}{r_o}$$

Hence, one finds after simplification that

$$\sigma = \frac{r^2|r_o'|}{r_o^2 - rr''\left(\frac{r^2-r_o^2}{1+r'^2}\right)}$$

Note the following: $\sigma = |r_o'|$ at turning points; $r_o' = 0$ at geodesic points; $\sigma = \left(\frac{r}{r_o}\right)^2|r_o'|$ if r is linear; $\sigma \to 0$ if $r'' \to -\infty$; while r' and r_o' are bounded (knife edge); and positivity of the denominator in σ implies that

$$r_o^2 > \frac{r^3r''}{1+r'^2+rr''}$$

It has been shown how the slippage function σ can be computed for a general closed surface ($r_x \neq 0 \neq r_\theta$) and what simplifications take place in the $F \equiv 0$ cases. It should be emphasized, however, that σ is more than just a number to be compared with the coefficient of friction μ to determine whether or not slippage occurs. The slippage function σ measures pointwise path quality and should ultimately be usable to synthesize or define quality *wrappable* paths on general closed surfaces.

REFERENCES

1. Royce W. Soanes, "Mathematical Aspects of the Off-Line Programming of Filament Winding Machines for General Surfaces of Revolution," ARDEC Technical Report ARCCB-TR-88036, Benet Laboratories, Watervliet, NY, September 1988.

2. D.J. Struik, Lectures on Classical Differential Geometry, Second Edition, Addison-Wesley, Reading, MA, 1961.

3. H.W. Guggenheimer, Differential Geometry, McGraw Hill, New York, 1963.

4. Barrett O'Neill, Elementary Differential Geometry, Academic Press, New York, 1966.

5. Manfredo P. do Carmo, Differential Geometry of Curves and Surfaces, Prentice Hall, Englewood Cliffs, NJ, 1976.

Optimal Control of Distributed Parameter Systems under Resonant and Unstable Loading

by

Iradj G. Tadjbakhsh
and Yuan–an Su

Rensselaer Polytechnic Institute
Troy, New York 12181

Abstract. Vibrations of linear, conservative distributed–parameter systems can be suppressed by a set of discrete actuators and an optimal control algorithm which minimizes the current value of a positive–definite, time–dependent objective function. The control procedure depends on the initial data and is independent of the ultimate outcome.

The control algorithm that is developed has a limited time interval of applicability about the arbitrary initial point. Outside this interval, negative damping and reduced stiffness may arise. However, the arbitrariness of the initial point allow re–initiation of the end–point and thus provides a means of continuation of control.

Using variational methods and the general Duhamel integral representation of the solution, explicit representation for infinite dimensional gain matrices is obtained. Examples of successful control under conditions of resonant loading and moving loads are carried out. The control algorithm can also be modified to control a beam–column subjected to transverse loads on the span before the beam is buckled. Due to loss of modal stiffness of the buckled beam, the controller can only provide modal damping to the structure. The controller cannot suppress the vibrations of a buckled beam. The question of the effect of control force spillover into higher uncontrolled modes of the system is considered.

Introduction. Active control of continuous flexible structures by means of the classical variational theories has been considered by many authors, among them Barnes (1971), Balas (1978), Soong (1982), Leipholz and Abdel–Rohman (1986), and Abdel–Rohman and Nayfeh (1987). For systems governed by self–adjoint partial differential equation the control can be naturally formulated in terms of the spatial modal functions and modal amplitudes.

In a series of papers Meirovitch et al. (1980, 1982, 1983, 1985) have developed the method of independent modal space control (IMSC). In this method each mode of the system is considered to be controllable independently. The distributed response of the controlled system and the forces of the actuators are obtained from the controlled mode shapes and the modal control forces using the orthogonal properties of the eigenfunctions.

Assuming that a finite number of sensors and actuators are placed on or about the structure that monitor the response and apply the control forces, one would

expect that the control algorithm should be causal i.e. it should depend only on the information available up to the time at which control is being exercised. Yang et al. (1985, 1987) have considered "instantaneous control" of discrete dynamic systems. They consider the discretized state equations of motion and represent the solution in terms of the unknown control forces. Then optimization of the Hamiltonian of the system yields as necessary conditions the optimality algorithm and the response state vector. Komkov (1972) also speaks of "instantly optimal control" algorithms that are obtained by a limiting process from the time optimal law reducing the energy of the system at the maximum possible rate.

The problem of active control of a distributed–parameter system with discrete sets of sensors and actuators is of the category of coupled modal control (Meirovitch, 1987). Here an optimal, coupled–modal control algorithm is developed for continuous self–adjoint structures by minimizing a time–dependent weighted sum of the kinetic energy, potential energy, and the control effort. This leads to a causal optimal algorithm whereby control forces are determined solely on the basis of information available up to the time at which control is being implemented.

The behavior of the controlled structure will be governed by a system which is generally non–self–adjoint and with modified damping and stiffness matrices of infinite order. For the system considered the elements of the matrices consist of harmonics with modal periods. It turns out that the control algorithm that is developed has a limited time–interval of applicability about the arbitrary initial point. Outside this interval negative damping and reduced stiffness may arise. However, the arbitrariness of the initial point allows definition of the end–point of the interval of integration as a new initial point and thus provides a means of continuation of the application of the control.

The method is used to suppress the vibration of an undamped beam subjected to a periodic load resonant with the fundamental frequency of the beam. The control of the beam under a moving load is also considered. Both cases show that the control interval affects the behavior of the controlled system significantly. The control algorithm is applied to a beam–column subjected to an impulse load at midspan. The vibrations are suppressed when the axial load is below the first buckling load of the column.

The Control Problem. Consider a flexible elastic medium such as a plate, a rod, a column, a membrane or a cable. Let $z(\underset{\sim}{r},t)$ denote the deflection of the structure with $\underset{\sim}{r}$ and t respectively representing the position of a material point in a domain D and the time. Assume n actuators are at locations $\underset{\sim}{r}_i$, i = 1,2,3,...,n with control forces $u_i(t)$. Motion of the controlled structure will be described by

$$m(\underset{\sim}{r})\,\ddot{z} + Lz = f(\underset{\sim}{r},t) + \sum_{i=1}^{n} u_i(t)\,\delta(\underset{\sim}{r} - \underset{\sim}{r}_i),\ \underset{\sim}{r} \epsilon D,\ t > t_0 \qquad (1)$$

where $m(\underset{\sim}{r})$ is the mass distribution, $f(\underset{\sim}{r},t)$ is the external excitation, δ is the Dirac delta function and L is a self–adjoint spatial operator with self–adjoint boundary conditions. The initial time t_0 is any arbitrary point on the time scale at which the initial conditions

$$z_o(\underset{\sim}{r}) = z(\underset{\sim}{r},t_o) \quad , \quad \dot{z}_o(\underset{\sim}{r}) = \dot{z}(\underset{\sim}{r},t_o) \tag{2}$$

are known. In (1) and (2) a dot denotes time derivative.

Associated with the above system is the positive definite potential energy U(t) whose variation δ U due to a small variation $\delta z(\underset{\sim}{r},t)$ is given by

$$\delta U = \int_D Lz\ \delta z\ dD \tag{3}$$

A positive performance index J is defined for the current value of time and consists of the kinetic energy, potential energy and the control effort

$$J(t) = Q_1 \left(\int_D \frac{1}{2} m \dot{z}^2\ dD\right) + Q_2 U + \frac{1}{2} \sum_{i=1}^{n} R_i u_i^2 \tag{4}$$

The positive functions Q_1 (t), Q_2 (t) and R_i (t) determine the level of the effectiveness of the control criteria and can be assigned arbitrary values by the designer. The control forces u_i will be determined such that J(t) is a minimum subject to the constraint of the equations of motion (1) – (2). This minimization implies that the system will acquire a state as near its state of rest $z = \dot{z} = 0$ as it is possible with limited control effort.

Using the eigenfunction expansion the displacement z can be represented by an infinite sum of the products of orthonormalized modes ϕ_k $(\underset{\sim}{r})$ and the corresponding modal amplitudes a_k (t), k = 1,2,3,... . When this representation is used in (1) and (2), the following system is obtained

$$\ddot{\underset{\sim}{a}} + \underset{\sim}{\Omega}^2 \underset{\sim}{a} = \underset{\sim}{f} + \underset{\sim}{B}\ \underset{\sim}{u}\ , \quad t > t_o \tag{5}$$

$$\underset{\sim}{a}\ (t_o) = \underset{\sim}{a}_o\ , \quad \dot{\underset{\sim}{a}}(t_o) = \dot{\underset{\sim}{a}}_o \tag{6}$$

where $\underset{\sim}{a}$, $\underset{\sim}{f}$, $\underset{\sim}{a}_o$, $\dot{\underset{\sim}{a}}_o$ are infinite dimensional vectors with their elements given respectively by

$$(a_j,\ a_{oj},\ \dot{a}_{oj}) = \int_D m[z,\ z_o,\ \dot{z}_o]\ \phi_j\ (\underset{\sim}{r})\ dD,$$

$$f_j = \int_D f\phi_j\ dD,\ j = 1,2,3,..., \tag{7}$$

and $\underset{\sim}{\Omega} = \text{diag}(\omega_j)$, $j = 1,2,3,...$, where ω_j are the modal frequencies as determined from (see Meirovitch 1967)

$$\omega_j^2 = \int_D \phi_j \, L\phi_j \, dD \quad , \quad j = 1,2,3,... \tag{8}$$

Also $\underset{\sim}{u} = (u_1, u_2, u_3, ..., u_n)^T$ is the vector of actuator forces and $\underset{\sim}{B}$ is a matrix of infinite rows and n columns with the elements of the jth row given by

$$B_{ji} = \phi_j (\underset{\sim}{r}_i) \quad , \quad i = 1,2,3,...n \tag{9}$$

Solution of (5) – (6) with the aid of the convolution integral can be represented by

$$\underset{\sim}{a} - \underset{\sim}{C}(t - t_o)\, \underset{\sim}{a}_o - \underset{\sim}{S}(t - t_o)\, \dot{\underset{\sim}{a}}_o = \underset{\sim}{S} * (\underset{\sim}{f} + \underset{\sim}{B}\, \underset{\sim}{u})$$

$$\triangleq \int_{t_o}^{t} \underset{\sim}{S}(t - \tau)\, [\underset{\sim}{f}(\tau) + \underset{\sim}{B}\, \underset{\sim}{u}(\tau)]\, d\tau \tag{10}$$

where $\underset{\sim}{C}$ and $\underset{\sim}{S}$ are infinite dimensional diagonal matrices of the modal harmonics given by

$$\underset{\sim}{S} = \text{diag}[\omega_j^{-1} \sin \omega_j t], \; \underset{\sim}{C} = \dot{\underset{\sim}{S}} = \text{diag}[\cos \omega_j t], j = 1,2,3,. \tag{11}$$

Control Algorithm. Introducing the arbitrary variations $\delta\underset{\sim}{u}$ in the interval $[t_o,t]$ we have for the variation δA from (10)

$$\delta\underset{\sim}{a} = \underset{\sim}{S} * \underset{\sim}{B}\, \delta\underset{\sim}{u} \tag{12}$$

By differentiating (12) one obtains

$$\delta\dot{\underset{\sim}{a}} = \underset{\sim}{C} * \underset{\sim}{B}\, \delta\underset{\sim}{u} \tag{13}$$

It is understood that certain modes may be uncontrollable due to coincidence of the actuator locations with nodal points. Also, it may be advantageous to limit the number of modes that participate in the control in order to achieve better controller design. Therefore variations of these uncontrolled modes will be zero either automatically, $\phi_j\,(\underset{\sim}{r}_i) = 0$, or by choice. In all cases the variations δz and $\delta \dot{z}$ can be written from the eigenfunction representation of z as

$$\delta z = \sum_{j=1}^{\infty} \phi_j\, \delta a_j \quad , \quad \delta \dot{z} = \sum_{j=1}^{\infty} \phi_j\, \delta \dot{a}_j \tag{14}$$

where it is understood that certain terms in the sums may vanish. Then from (3) – (4),

$$\delta J = \int_D (Q_1\, m\, \dot{z}\, \delta \dot{z} + Q_2\, Lz\, \delta z)\, dD + \sum_{i=1}^{n} R_i\, u_i\, \delta u_i \tag{15}$$

These together with the relations defining the eigenfunctions, the frequencies and the orthogonality conditions

$$L\phi_j = m\, \omega_j^2\, \phi_j,\ \int_D m\, \phi_j\, \phi_k\, dD = \delta_{jk} \ , \ j,k = 1,2,3,... \tag{16}$$

yield the modal equivalent of (15)

$$\delta J = Q_1\, \underset{\sim}{a}\ \delta \underset{\sim}{\dot{a}}^T + Q_2\, \underset{\sim}{\dot{a}}^T\, \underset{\sim}{\Omega}^2\, \delta \underset{\sim}{a} + \underset{\sim}{u}^T\, \underset{\sim}{R}\, \delta \underset{\sim}{u} \tag{17}$$

where $\underset{\sim}{R} = \mathrm{diag}[R_1, R_2, R_3,... R_n]$.

Substitution of (11) – (13) into (17) yields

$$\delta J = Q_1\, (t)\underset{\sim}{\dot{a}}^T(t) \int_{t_o}^{t} \underset{\sim}{C}(t-\tau)\, \underset{\sim}{B}\delta\, \underset{\sim}{u}(\tau)d\tau + Q_2(t)\underset{\sim}{a}^T(t)\, \underset{\sim}{\Omega}^2 \int_{t_o}^{t} \underset{\sim}{S}(t-\tau)\, \underset{\sim}{B}\delta\underset{\sim}{u}(\tau)d\tau$$

$$+ \underset{\sim}{u}^T\,(t)\, \underset{\sim}{R}(t)\, \delta\underset{\sim}{u}(t) \tag{18}$$

By introducing the Heaviside unit step function H(t) and the delta function $\delta\,(t)$ one can represent (18) as a single integral over the extended interval $t_o \le \tau < \infty$. Thus

$$\delta J = \int_{t_0}^{\infty} \{H(t-\tau)[Q_1(t)\dot{\underset{\sim}{a}}^T(t)\underset{\sim}{C}(t-\tau)+Q_2(t)\underset{\sim}{a}^T(t)\underset{\sim}{\Omega}^2\underset{\sim}{S}(t-\tau)]\underset{\sim}{B}+$$

$$\delta(\tau-t)\underset{\sim}{u}^T(\tau)\underset{\sim}{R}(\tau)\}\delta\underset{\sim}{u}(\tau)d\tau \qquad (19)$$

Vanishing of δJ for arbitrary $\delta\underset{\sim}{u}$ implies the vanishing of the coefficient of $\delta\,\underset{\sim}{u}(\tau)$ in the integrand in (19). This condition is

$$H(t-\tau)[Q_1(t)\dot{\underset{\sim}{a}}^T(t)\underset{\sim}{C}(t-\tau)+Q_2(t)\underset{\sim}{a}^T(t)\underset{\sim}{\Omega}^2\underset{\sim}{S}(\tau-t)]\underset{\sim}{B}+\delta(t-\tau)\underset{\sim}{u}^T(\tau)\underset{\sim}{R}(\tau) = 0,$$

$$\text{for } t > t_0,\ t > t_0 \qquad (20)$$

Integrating this equation with respect to τ, from t_0 to t, one obtains

$$\{Q_1\,\dot{\underset{\sim}{a}}^T\underset{\sim}{S}(t-t_0) + Q_2\,\underset{\sim}{a}^T[\underset{\sim}{I}-\underset{\sim}{C}(t-t_0)]\}\,\underset{\sim}{B} + \underset{\sim}{u}^T\,\underset{\sim}{R} = 0,\ t > t_0 \qquad (21)$$

which determines the control force in the time domain.

Assuming that a certain ordered number Nc of modes are to participate in the control algorithm we denote them by $\underset{\sim}{a}_c = (a_1, a_2, \ldots a_{N_c})^T$. The remaining modes will be denoted by $\underset{\sim}{a}_r$. In a corresponding manner vectors and matrices will be partitioned

$$\underset{\sim}{a} = \begin{bmatrix} \underset{\sim}{a}_c \\ \\ \underset{\sim}{a}_r \end{bmatrix},\ \underset{\sim}{f} = \begin{bmatrix} \underset{\sim}{f}_c \\ \\ \underset{\sim}{f}_r \end{bmatrix},\ \underset{\sim}{B} = \begin{bmatrix} \underset{\sim}{B}_c \\ \\ \underset{\sim}{B}_r \end{bmatrix},\ \underset{\sim}{S} = \begin{bmatrix} \underset{\sim}{S}_c & \underset{\sim}{0} \\ & \\ \underset{\sim}{0} & \underset{\sim}{S}_r \end{bmatrix}$$

$$\underset{\sim}{C} = \begin{bmatrix} \underset{\sim}{C}_c & 0 \\ & \\ \underset{\sim}{0} & \underset{\sim}{C}_r \end{bmatrix},\ \underset{\sim}{I} = \begin{bmatrix} \underset{\sim}{I}_c & \underset{\sim}{0} \\ & \\ \underset{\sim}{0} & \underset{\sim}{I}_r \end{bmatrix},\ \underset{\sim}{\Omega}^2 = \begin{bmatrix} \underset{\sim}{\Omega}_c^2 & \underset{\sim}{0} \\ & \\ \underset{\sim}{0} & \underset{\sim}{\Omega}_r^2 \end{bmatrix} \qquad (22)$$

Then (21) can be solved for the control force

$$\underset{\sim}{u} = -Q_1 \underset{\sim}{D}\, \dot{\underset{\sim}{a}}_c - Q_2 \underset{\sim}{E}\, \underset{\sim}{a}_c \tag{23}$$

where

$$\underset{\sim}{D} = \underset{\sim}{R}^{-1} \underset{\sim}{B}_c^T \underset{\sim}{S}_c (t - t_o), \quad \underset{\sim}{E} = \underset{\sim}{R}^{-1} \underset{\sim}{B}_c^T [\underset{\sim}{I}_c - \underset{\sim}{C}_c (t - t_o)] \tag{24}$$

Meirovitch and Baruh (1985) developed spatial modal filters for reconstruction of the modal states $\underset{\sim}{a}_c$ and $\dot{\underset{\sim}{a}}_c$. Using orthogonality of eigenfunctions they showed that the problem of observer spillover that was discussed by Balas (1978) can be circumvented. Assuming that the effect of time delay due to computation on the system is small enough to be negligible, the behavior of the controlled modes are determined from (5) and (23), to be

$$\ddot{\underset{\sim}{a}}_c + Q_1 \underset{\sim}{B}_c \underset{\sim}{D}\, \dot{\underset{\sim}{a}}_c + (\underset{\sim}{\Omega}_c^2 + Q_2 \underset{\sim}{B}_c \underset{\sim}{E})\, \underset{\sim}{a}_c = \underset{\sim}{f}_c \tag{25}$$

The residual modes are subject to control spillover and are determined from

$$\ddot{\underset{\sim}{a}}_r + \underset{\sim}{\Omega}_r^2\, \underset{\sim}{a}_r = \underset{\sim}{f}_r - Q_1 \underset{\sim}{B}_r \underset{\sim}{D}\, \dot{\underset{\sim}{a}}_c - Q_2 \underset{\sim}{B}_r \underset{\sim}{E}\, \underset{\sim}{a}_c \tag{26}$$

It is evident that the terms corresponding to Q_1, Q_2 in the performance index J respectively modify the damping and the stiffness of the system. The gain matrices $\underset{\sim}{E}$, $\underset{\sim}{D}$ are non–symmetric non–diagonal infinite dimensional matrices that couple all the modal amplitudes. The governing equations for the controlled system are non self–adjoint raising the question that in the event of instability the transition may be through flutter.

The elements of gain matrices consist of modal harmonics $\cos\omega_j(t-t_o)$ and $\sin\omega_j(t-t_o)$ which initially, i.e., near $t = t_o$. are non–negative. As t increases, the higher harmonics begin to change sign thus creating negative damping and reducing the stiffness of system. It thus becomes clear that the optimal algorithm derived here is valid for an interval of time during which occurrence of negative damping or reduction of stiffness have not caused amplification of the response of those higher modes that participate significantly in the dynamic of the system. Therefore the control interval, i.e., $t - t_o$ should not exceed one–half of the period of the highest controlled mode, i.e.,

$$(t - t_o) \leq \frac{\pi}{\omega_{N_c}} \tag{27}$$

Control of A Simply Supported Beam. Consider a uniform beam hinged at both ends and controlled by four actuators, Fig. 1. The corresponding location of the actuators, x_i, i = 1, 2, 3, 4, are 13, 25, 35, and 47m from the left end of the beam respectively. We assume that the length of the beam is 60m, its bending stiffness EI is 5 x 10^5 N–m^2 and its mass per unit length m is 1.0 Kg(m). Then the eigenvalues ω_i^2 and normalized eigenfunctions are given by

$$\omega_i^2 = 0.039(i\pi)^4, \qquad \phi_i(x) = 0.183 \sin(i\pi/60),\ i = 1, 2\ldots$$

The control parameters are assigned fixed values $Q_1 = Q_2 = 5000$ and $R_i = 1$, i = 1, 2, 3, 4. The lowest ten modes of the beam constitute the controlled and the uncontrolled modes.

In the first example, the beam is subjected to a unit cyclic load resonant with the first mode of the beam acting at the midspan. The external excitation in Eq. (1) is expressed as $f(x,t) = \sin(1.95t) \cdot \delta(x - 30)$. The control intervals are 0.01, 0.08 and 0.80 seconds and only the first mode is subjected to control. Control of additional modes is not called for as they are not being excited. The results that are presented refer to the deflection at midspan of the beam as shown in Fig. 2. It can be seen that, unlike the uncontrolled beam the response remains bounded. From Table 1 the maximum control gain for the first mode is approximately 0.81 seconds. Thus the beam is almost completely controlled, without any oscillations, by setting the control interval, i.e., $t–t_0$, equal to 0.80 seconds.

In the second example the beam is subjected to a 20N load that travels along the beam with a speed of 37 m/sec. Now the lowest 5 modes are controlled and the next 5 modes are uncontrolled ($N_c = N_r = 5$). The results corresponding to control intervals of 0.01, 0.03 and 0.06 seconds are shown in Fig. 3. It is clear that the controllers suppress the vibration within 10 seconds.

Control of Simply Supported Beam–Columns. Beam columns are members that are subjected to both bending and axial compression. They will be unstable when the axial compression reaches certain critical values. For a given beam–column with certain specified boundary conditions, theoretically an infinite number of critical loads and associated buckling mode–shapes exist. In this section, we attempt to follow the same procedure derived previously to control the beam–columns at and above their buckling load.

Let P denote the axial compressive load applied to a beam in Fig. 1, the eigenvalue problem is given by

$$\frac{\partial^2}{\partial x^2}\left(EI(x)\frac{\partial^2 \phi_j}{\partial x^2}\right) + P\frac{\partial^2 \phi_j}{\partial x^2} = m\omega_j^2 \phi_j,\ j = 1, 2, 3\ldots. \qquad (28)$$

with the boundary conditions

$$\phi_j = \frac{\partial^2 \phi_j}{\partial x^2} = 0 \text{ for } x = 0, \ell \tag{29}$$

that hold for the simply supported case.

The normalized eigenfunctions for Eqs. (28) – (29) for a prismatic beam can be written in the form

$$\phi_j = \sqrt{(2/m\ell)} \sin\frac{j\pi x}{\ell}, \; j = 1,2,\ldots \tag{30}$$

As a result of substitution of Eq. (30) into Eq. (28), we have

$$\omega_j = \pm \begin{cases} j\mu\gamma_j & \text{if } p < j, \; j = 1, 2, \ldots \\ ij\mu\gamma_j & \text{if } p > j, \; j = 1, 2, \ldots \end{cases} \tag{31}$$

where

$$p = \frac{P}{P_E}, \qquad \gamma_j = \sqrt{j^2 - p} \tag{32}$$

$$\mu = \left(\frac{\pi^2}{\ell^2}\right)\sqrt{\frac{EI}{m}}, \qquad i = \sqrt{-1}$$

and P_E denotes the Euler load of the beam–column, i.e., $P_E = \pi^2 EI/\ell^2$.

When p is smaller than 1, i.e., the axial load P is smaller than P_E, the beam is stable and the control law expressed in Eq. (29) is still valid with the ω_j, j = 1, 2, 3. . . . given by Eq. (31). When the beam is in critical condition, i.e. p = 1, then ω_1 is zero. However, because $\lim_{p \to 1} \frac{\sin(\omega_1 t)}{\omega_1} = t$, and $\lim_{p \to 1} \cos(\omega_1 t) = 1$, one obtains from Eq. (11)

$$\begin{aligned} \underset{\sim}{S}_c(t) &= \text{diag}\left[t, \; \omega_j^{-1} \sin(\omega_j t)\right], \quad j = 2, 3 \ldots N_c \\ \underset{\sim}{C}_c(t) &= \text{diag}\left[1, \cos(\omega_j t)\right], \quad j = 2, 3, \ldots N_c \end{aligned} \tag{33}$$

Following the same procedure described in previous sections, one can obtain the same control law that is already shown in Eqs. (23) – (24), in which $\underset{\sim}{S}$ and $\underset{\sim}{C}$ are now defined by Eq. (33).

Similarly, when $1 < p < 2$, then the control force and the system behavior can be given by Eqs. (23) – (26) wherein $\underset{\sim}{S}$, $\underset{\sim}{C}$ can be written in the form

$$\underset{\sim}{S}_c(t) = \text{diag}\left[|\omega_1|^{-1} \sinh|\omega_1|t,\ \omega_j^{-1} \sin(\omega_j t)\right], \qquad j = 2, 3, \ldots N_c$$

$$\underset{\sim}{C}_c(t) = \text{diag}\left[\cosh|\omega_1|t,\ \cos(\omega_j t)\right], \qquad j = 2, 3, \ldots\ldots N_c \tag{34}$$

For numerical evaluation the beam with the same parameters as shown in Fig. 1 is used. The system is subjected to an impulse at midspan. The control intervals are set to 0.01 and 0.05 seconds to compare results. The behavior of the beam–column is assumed to be adequately represented by its first ten modes and of these the lowest five are controlled. Fig. 4 shows the response of the system over a 10 seconds interval. The axial compressive load ratio $p = P/P_E$ is set to 0.9. The deflection of the beam remains bounded and is accompanied by control force spilling over into the higher modes. The time histories of the commanded forces for the actuator A_2 – one of the middle two actuators – are shown in Fig. 5. The larger control interval causes higher beginning values for the force and lower tail end values. Thus it may be concluded that the control is effective at the axial load of 0.9 of buckling load.

Next, we consider the control of beam–column at the buckling case, i.e., $p = 1$. In this case, the $\underset{\sim}{E}$ defined in Eq. (24) can be partitioned as

$$\underset{\sim}{E} = \begin{bmatrix} E_{11} & \underset{\sim}{E}_{12} \\ \underset{\sim}{E}_{21} & \underset{\sim}{E}_{22} \end{bmatrix} = \begin{bmatrix} \begin{Bmatrix} 0 \\ \\ 0 \end{Bmatrix} & \begin{bmatrix} (1-C_2)\dfrac{B_{21}}{R_1} & \cdots & (1-C_{N_c})\dfrac{B_{N_c1}}{R_1} \\ (1-C_2)\dfrac{B_{22}}{R_2} & \cdots & (1-C_{N_c})\dfrac{B_{N_c2}}{R_2} \\ (1-C_2)\dfrac{B_{2n}}{R_n} & \cdots & (1-C_{N_c})\dfrac{B_{N_cn}}{R_n} \end{bmatrix} \end{bmatrix} \tag{35}$$

where

$$C_i: \cos(\omega_i(t - t_0)),\ i = 1 \ldots N_c$$

Also the $\underset{\sim}{B}_c$, $\underset{\sim}{D}$ matrices and the $\underset{\sim}{a}_c$ vector can be partitioned as

$$\underset{\sim}{B}_c = \begin{bmatrix} B_{11} & \underset{\sim}{B}_{12} \\ \underset{\sim}{B}_{21} & \underset{\sim}{B}_{22} \end{bmatrix},\ \underset{\sim}{D} = \begin{bmatrix} D_{11} & \underset{\sim}{D}_{12} \\ \underset{\sim}{D}_{21} & \underset{\sim}{D}_{22} \end{bmatrix},\ \underset{\sim}{a}_c = \begin{Bmatrix} a_1 \\ \underset{\sim}{a}_2 \end{Bmatrix} \tag{36}$$

Then the first modal equation, that corresponds to the buckling mode of the beam–column in Eq. (25), becomes

$$\ddot{a}_1 + Q_1 B_{11} D_{11} \dot{a}_1 = f_1 - Q_1 \underset{\sim}{B}_{12} \underset{\sim}{D}_{12} \dot{\underset{\sim}{a}}_2 - Q_2 (B_{11} \underset{\sim}{E}_{12} + \underset{\sim}{B}_{12} \underset{\sim}{E}_{22}) \underset{\sim}{a}_2 \qquad (37)$$

with appropriate initial conditions at t_0. From Eq. (37), it is clear that the feedback control law can only provide a certain amount of damping to the system and the coupled modal control force acts as a counter force to balance f_1. Because a_1 is coupled with $\underset{\sim}{a}_2$, it is difficult to find the analytical solution for a_1. Therefore the same numerical procedure employed previously is used to find approximate results. Fig. 6 shows the results of the control of the beam–column at its buckling load when the control intervals are set to 0.01 and 0.05 seconds. The control spillover is present and the system acquires a permanent deflection that depends on the control interval in both cases. Different control efforts are pumped into the system for the different control intervals, and this is responsible for the different levels of permanent deformations shown in Fig. 6. As observed form Fig. 7 the commanded force for smaller control intervals have higher initial and lower final values.

Fig. 8 and Fig. 9 show the control of the beam–column when $p = 1.1$. In Fig. 8, the system appears unstable with deflection growing continually for both control intervals of 0.01 and 0.05 seconds. The deflection of the control interval of 0.05 seconds have a smaller rate of growth than that of the control interval of 0.01 seconds.

Fig. 9 shows the A_2 the actuator's commanded force histories for both cases. The importance of the beginning stages of control effort, being a function of the control interval, shows up prominently in these cases. In the case of the short control, i.e., $t-t_0 = 0.01$ seconds, the commanded force of the actuator A_2 increases with the unbounded response of the controlled system. However, in the long control interval case, i.e., $t-t_0 = 0.05$ seconds, the commanded force grows with a more moderate rate compared to that of the short interval. Therefore, it is clear that the longer control intervals are better in that they provide slower rate of growth of deflections.

Conclusion. The control algorithm developed here determines the control forces in the time domain explicitly in terms of the coupled modal amplitudes and velocities. The algorithm is optimal in that it minimizes the total energy of the system instantly and spatially rather than over a time domain. Consequently the algorithm is causal and can be implemented on the basis of what has transpired prior to implementation. Additionally the algorithm allows participation by a limited number of modes with the remainder being subject to spillover effect. The instantaneous nature of the algorithm limits its applicability to the small neighborhood of the initial point which however is arbitrary. This arbitrariness allows integration over a sequence of intervals whose initial data is determined by the end conditions of the previous interval.

The coupled–modal control algorithm can successfully be used to suppress the vibration of the beam subjected to a resonant cyclic load and a moving load as shown in this paper. The algorithm is causal, i.e. it only depends on the past history of the system and such that the control interval dominates the system's behavior. The maximum control interval as given by Eq. (27) is a function of the period of the highest controlled mode. The response of controlled system is significantly affected by the control interval. Therefore, a proper choice of the control interval is necessary for the algorithm. The algorithm can be applied to control of a beam–column before buckling. However, the controller can only provide a damping resistance to the buckled mode, and the algorithm is not well suited for controlling a beam after buckling.

Acknowledgement. The work reported here was conducted in the course of research supported by the Army Research Office under Contract DAAL 03–89–K–0002 with Rensselaer Polytechnic Institute. This support is gratefully acknowledged.

References.

Abdel–Rohman, M. and Nayfeh, A.H., 1987, "Active Control of Nonlinear Oscillations in Bridges," ASCE, Journal of Engineering Mechanics, Vol. 113, pp. 335–348.

Balas, M.J., 1978, "Active Control of Flexible Systems," Journal of Optimization Theory and Applications, Vol. 25, pp. 415–436.

Balas, M.J., 1978, "Feedback Control of Flexible Systems," IEEE Transaction on Automatic Control, Vol. AC–23, pp. 415–436.

Barnes, E.R., 1971, "Necessary and Sufficient Optimality Conditions for a class of Distributed Parameter Control Systems," SIAM Journal on Control, Vol. 9, No. 1, pp. 62–82.

Komkov, V., 1972, "Optimal Control Theory for the Damping of Vibrations of Simple Elastic Systems," Lecture Notes in Mathematics No. 253, Springer Verlag.

Leipholz, H.H.E. and Abdel–Rohman, M., 1986, Control of Structure, Martinus Nijhoff Publishers.

Luenberger, D. 1968, "An Introduction to Observers," IEEE Trans. Automat. Contrl, Vol AC–16 No. 6 pp. 316–353.

Meirovitch, L., 1969, Analytical Methods in Vibrations, Macmillan Publishing Co.

Meirovitch, L., and Oz, H., 1980, "Modal Space Control of Distributed Gyroscopic Systems," Journal of Guidance and Control, Vol. 2, No. 3, pp. 140–150.

Meirovitch, L. and Baruh, H., 1982, "Control of Self–Adjoint Distributed Parameter Systems," Journal of Guidance, Control, and Dynamics Vol. 5, No. 1, pp. 60–66.

Meirovitch, L., and Baruh, H., 1983, "Robustness of the Independent Modal Space Control Method," Journal of Guidance, Control, and Dynamics Vol. 6, No. 1, pp. 20–25.

Meirovitch, L., and Baruh, H., and Oz, H., 1983, "A Comparison of Control Techniques for Large FLexible Systems," Journal of Guidance, Control, and Dynamics Vol. 6, No. 4, pp. 302–310.

Meirovitch, L., and Baruh, H., 1985, "The Implementation of Modal Filters for Control of Structures," Journal of Guidance, Vol. 8, No. 6, pp. 707–716.

Meirovitch, L. 1987, "Some Problems Associated with the Control of Distributed Structures," Journal of Optimization Theory and Applications, Vol. 54, No. 1, pp. 1–21.

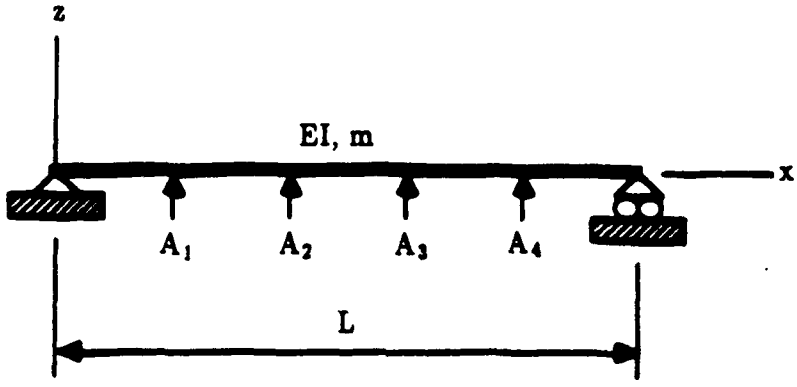
Soong, T.T. and Chang, J.C.H., 1982, "Active Vibration Control of Large Flexible Structures," Shock Vibration Bulletin, Vol. 52, Part IV, pp. 47–54.

Yang, J.N., Akbarpour, A., and Ghaemmaghami, P., 1987, "New Optimal Control Algorithms for Structural Control," ASCE, Journal of Engineering Mechanics, Vol. 113, No. 9, pp. 1369–1386.

Yang, J.N., Akbarpour, A., and Ghaemmaghami, P., 1985, "Optimal Control Algorithms for Earthquake–Excited Building," Proceeding of Second International Symposium on Structural Control, University of Waterloo, Waterloo, Canada, July 1985, Edited by Leipholz, H.H.E., Martinus Nijhoff, Amsterdam, 1987.

TABLE I

Mode	Max. $(t-t_o)$, sec.	Mode	Max. $(t-t_o)$, sec.
1	1.621	6	0.045
2	0.405	7	0.033
3	0.180	8	0.025
4	0.101	9	0.020
5	0.065	10	0.016



System Parameters:

L : 60.m

EI : $5 * 10^5$ N/m^2

m : 1. kg/m

$x_1 = 13.$ m

$x_2 = 25.$ m

Control Parameters:

Q_1 : 5000.

Q_2 : 5000.

R_i : 1., i=1,2,3,4

$x_3 = 35.$ m

$x_4 = 47.$ m

Fig. 1 Simply supported beam with 4 discrete actuators, A_i, i = 1,..4.

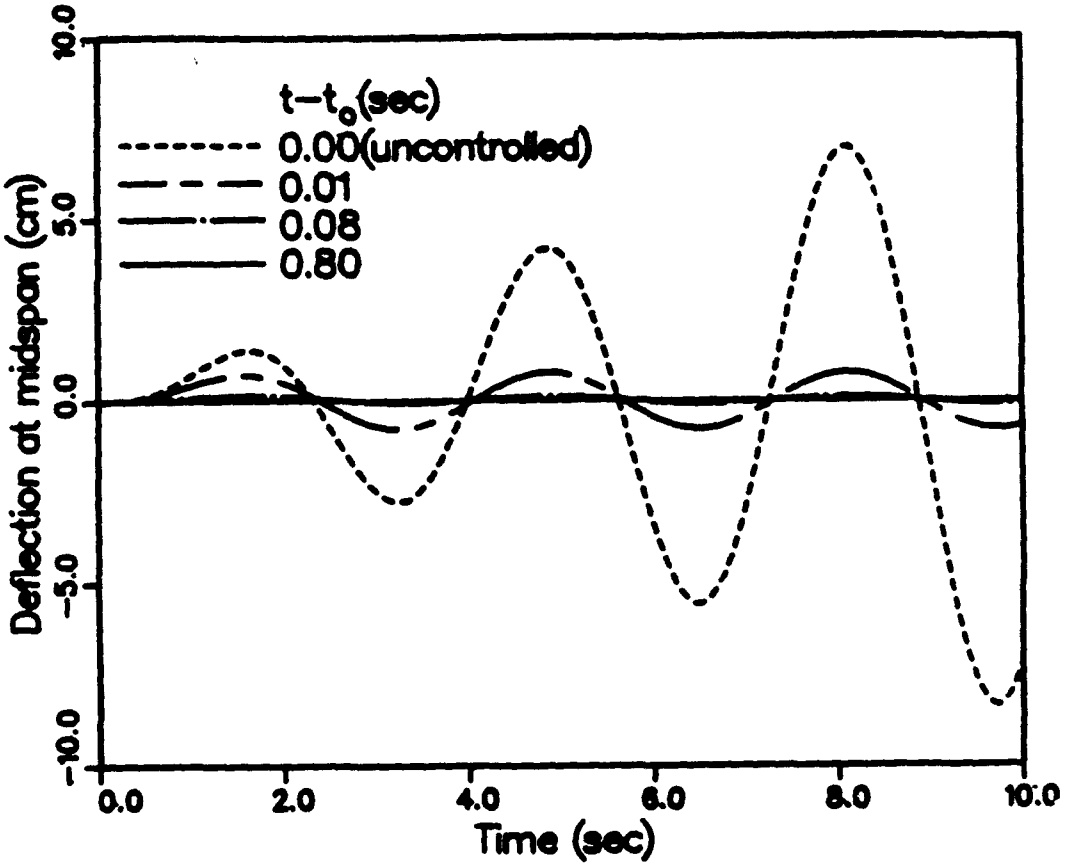


Fig. 2 The midspan deflection of the beam subjected to an unit periodic load resonant with the first fundamental frequency of the beam, $t-t_o$ = 0.01, 0.08, 0.80 seconds.

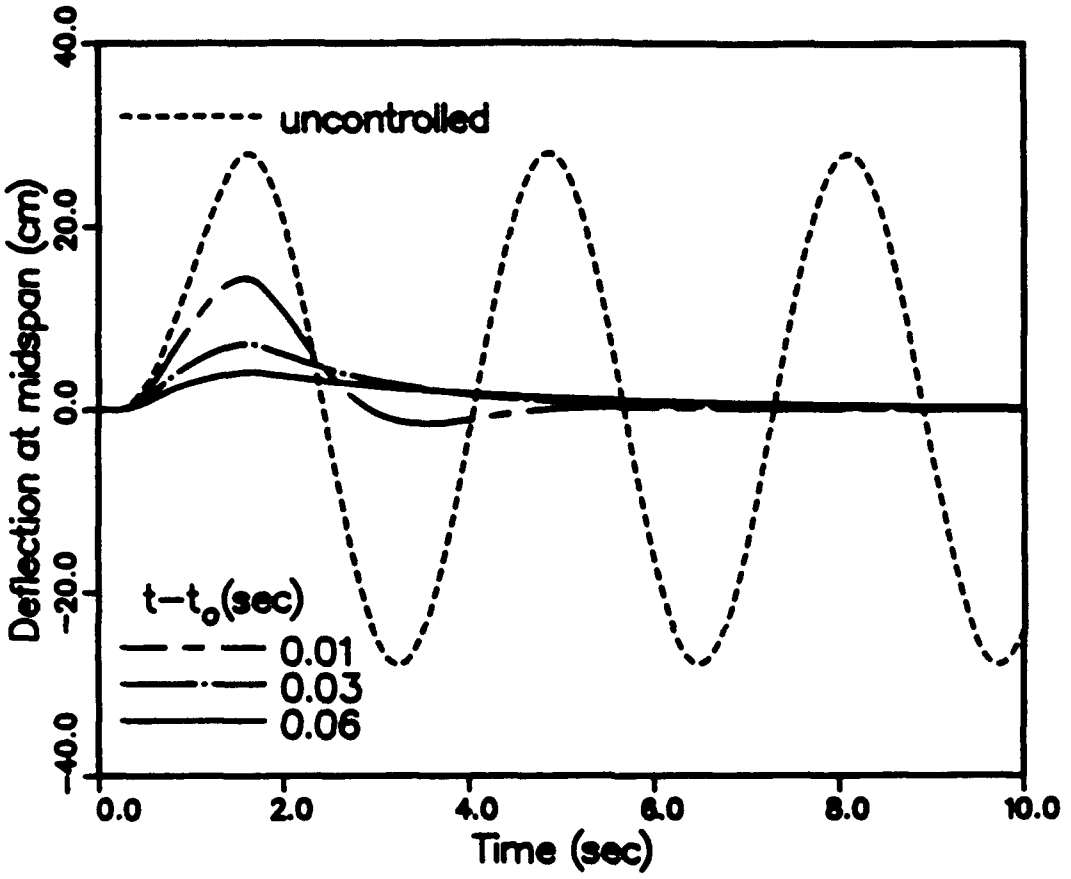


Fig. 3 The midspan deflection of the beam subjected to a moving load traveling at 37 m/second speed, $t-t_o$ = 0.01, 0.03 and 0.06 seconds.

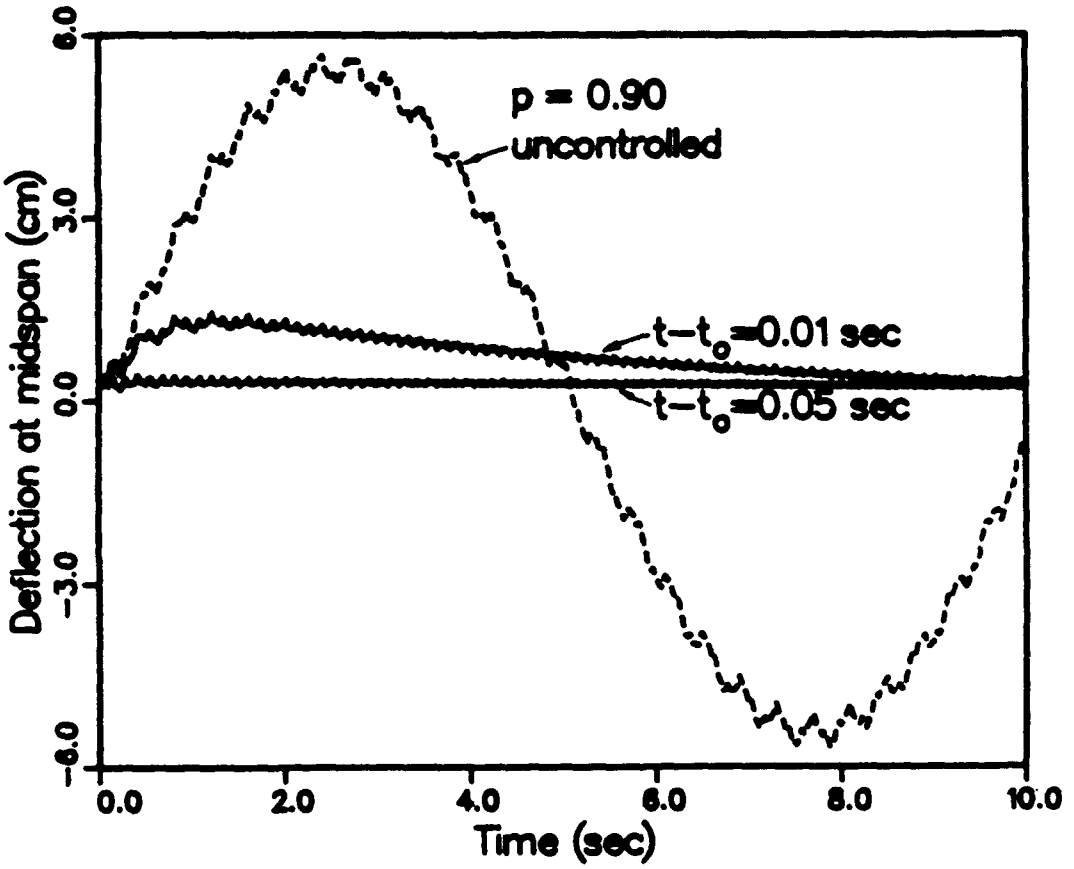


Fig. 4 The midspan deflection of the controlled beam–column subjected to an unit impulse at midspan, p = 0.9, $t-t_o$ = 0.01, 0.05 seconds.

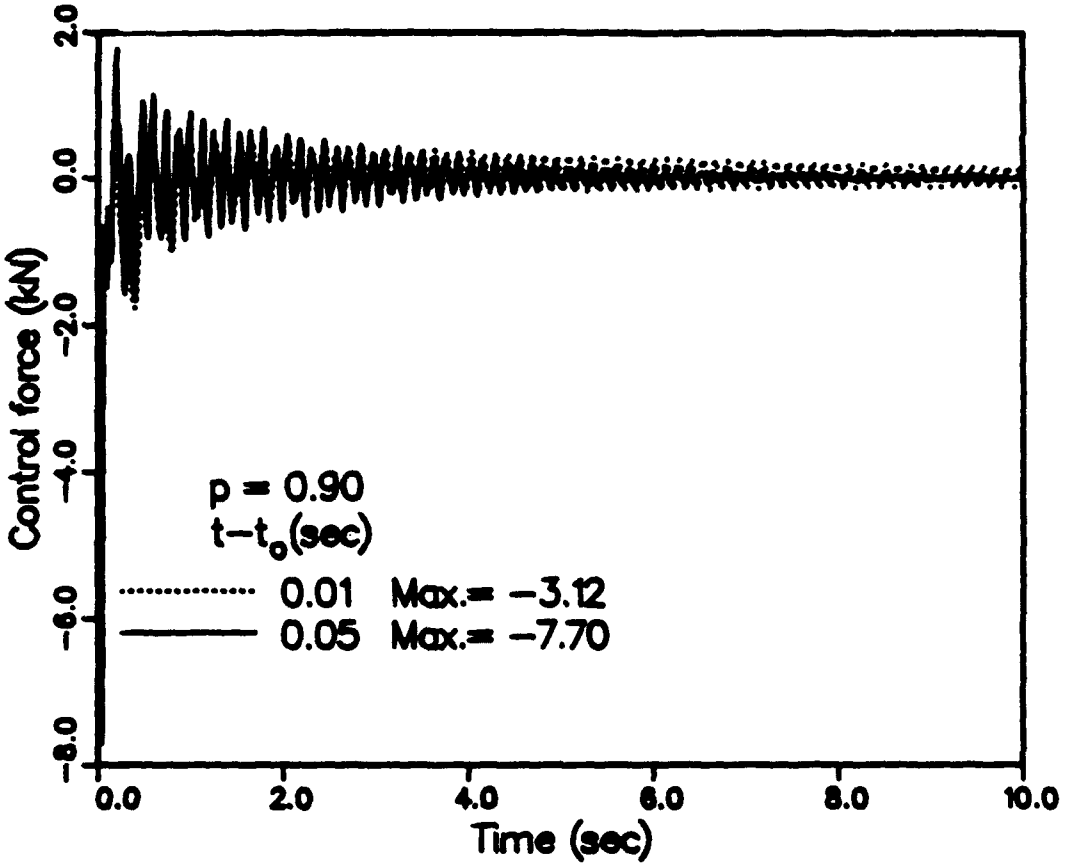


Fig. 5 Commanded force of A_2 actuator for the control of the beam–column, $p = 0.9$, $t-t_0 = 0.01, 0.05$ seconds.

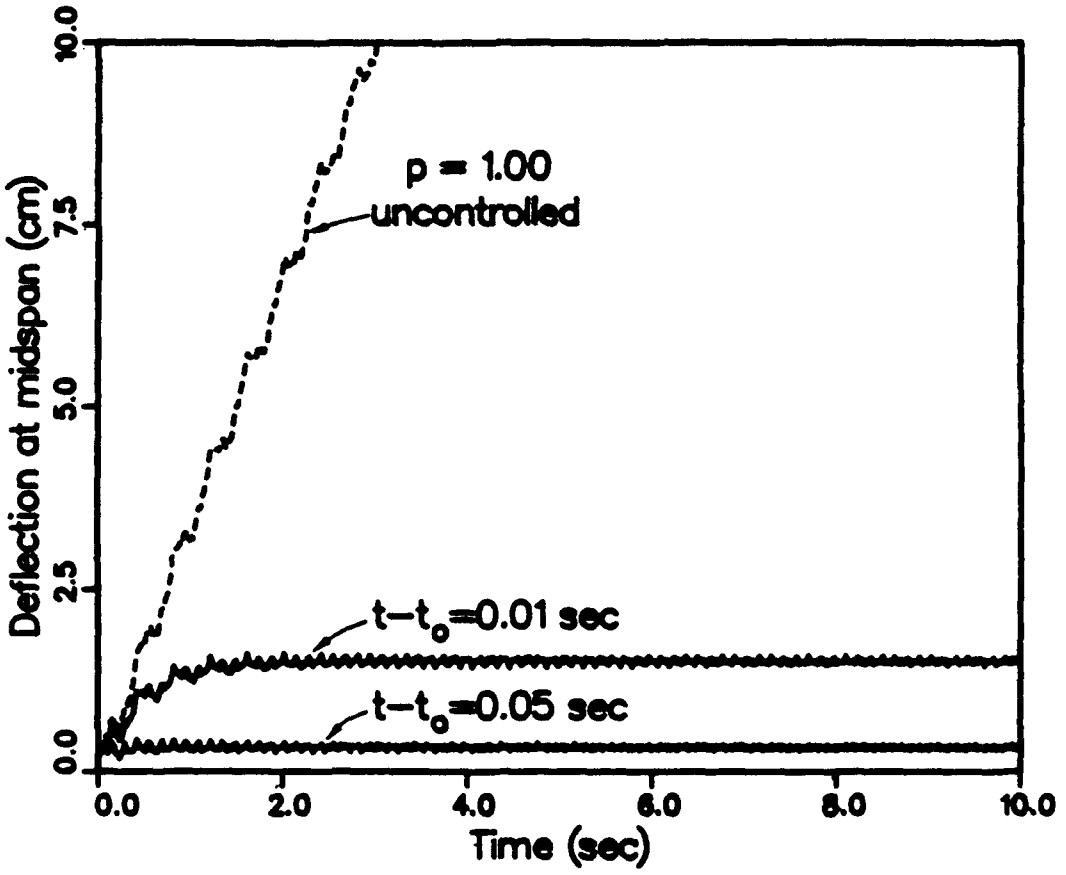


Fig. 6 The midspan deflection of the controlled beam–column subjected to an unit impulse at midspan, p = 1.0, $t-t_o$ = 0.01, 0.05 seconds.

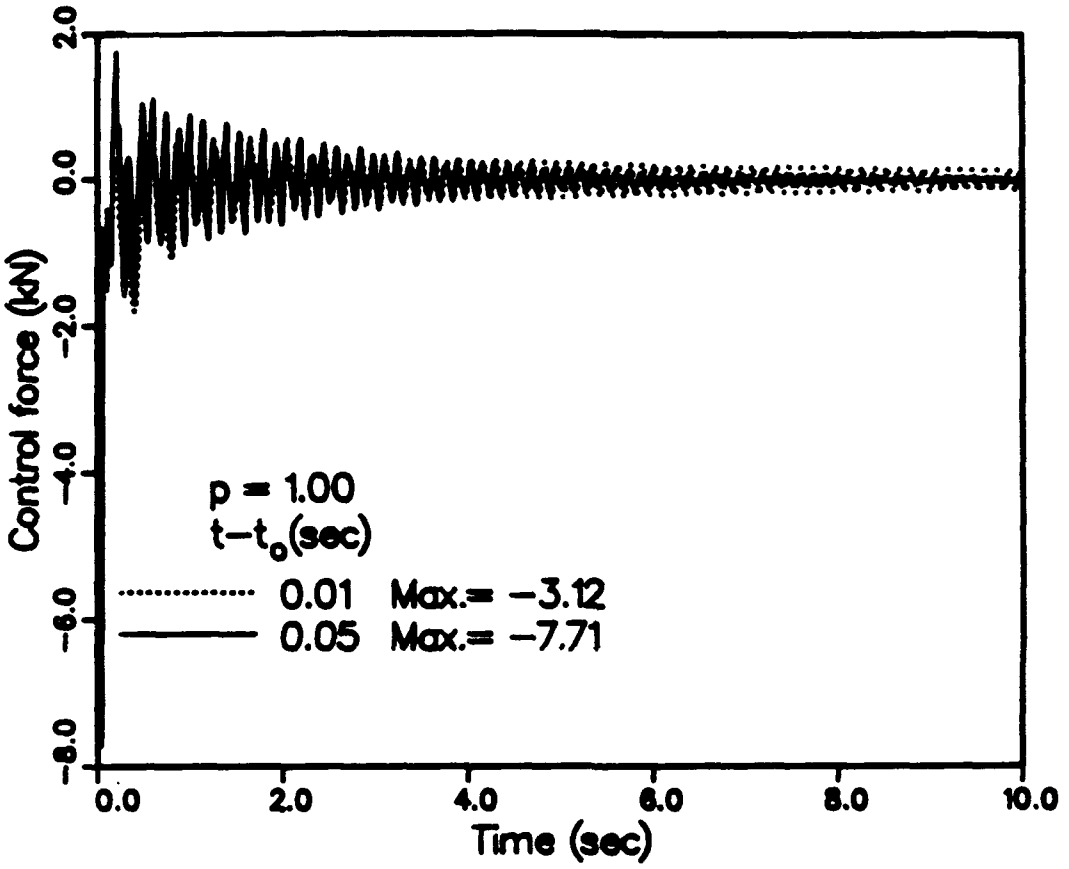


Fig. 7 Commanded force of A_2 actuator for the control of the beam–column, $p = 1.0$, $t-t_0 = 0.01, 0.05$ seconds.

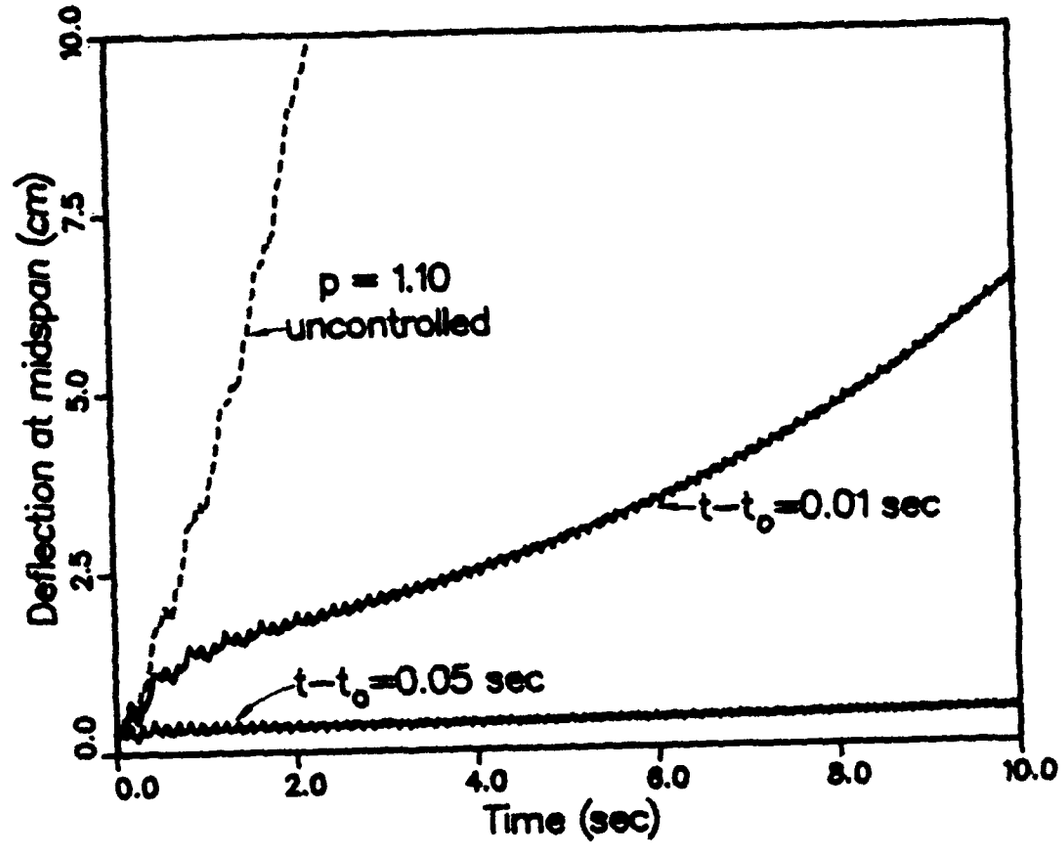


Fig. 8 The midspan deflection of the controlled beam–column subjected to an unit impulse at midspan, p = 1.1, $t-t_o$ = 0.01, 0.05 seconds.

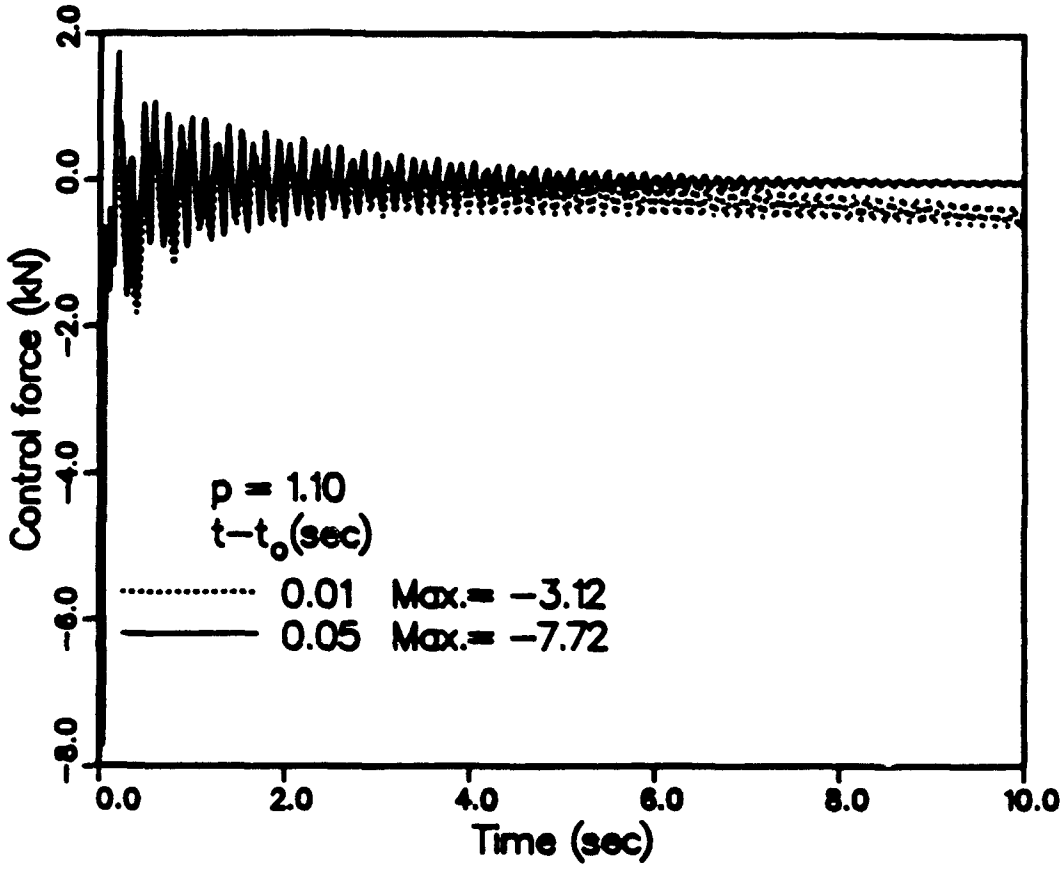


Fig. 9 Commanded force of A_2 actuator for the control of the beam–column, $p = 1.1$, $t-t_o = 0.01$, 0.05 seconds.

A CENTRAL LIMIT THEOREM FOR INTEGRAL FUNCTIONALS OF A STATIONARY GAUSSIAN PROCESS*

Simeon M. Berman

Courant Institute of Mathematical Sciences

New York University

New York, NY 10012

ABSTRACT. Let $X(t)$, $t \geq 0$, be a real stationary Gaussian process with covariance function $r(t)$. Let $f(x)$ be a function in $L_2(\phi)$, where $\phi(z)$ is the standard normal density, and assume that $\int x\, f(x)\, \phi(x)\, dx \neq 0$. It is shown that the central limit theorem holds for the functional $\int_0^t f(X(s))\, ds$, for $t \to \infty$, under the sole assumptions $r(t) \geq 0$ and $r(t) \to 0$ for $t \to \infty$.

1. Summary.

Let $X(t)$, $t \geq 0$, be a real stationary Gaussian process with mean 0 and covariance function $r(t) = EX(0)\, X(t)$. For a Borel function $f(x)$ and $t > 0$, consider the functional

$$\int_0^t f(X(s))\, ds\ . \tag{1.1}$$

There has been a sustained interest in proving the Central Limit Theorem, for $t \to \infty$, for such functionals, that is, determining the limiting distribution of the normed random variable

$$\frac{\int_0^t f(X(s))\, ds - E[\int_0^t f(X(s))\, ds]}{\{Var \int_0^t f(X(s))\, ds\}^{1/2}}\ . \tag{1.2}$$

* Supported by the U. S. Army Research Office.

Let $\phi(x)$ represent the standard normal density, and let $H_k(x)$, $k = 0, 1, \ldots$, be the family of Hermite polynomials. The assumption commonly used for the function f is that it belong to $L_2(\phi)$, that is, $\int |f(x)|^2 \, \phi(x) \, dx < \infty$. Every such function has an expansion

$$f(x) = \sum_{k=0}^{\infty} f_k H_k(x)/\sqrt{k!} \, , \tag{1.3}$$

where

$$f_k = \frac{1}{\sqrt{k!}} \int_{-\infty}^{\infty} f(x) \, H_k(x) \, \phi(x) \, dx \, . \tag{1.4}$$

The Hermite rank of f is the smallest positive integer k for which $f_k \neq 0$.

Limit theorems for (1.2) are of two categories. The first is characterized by convergence of the distribution to a normal distribution under a hypothesis of mixing, involving sufficiently rapid convergence of $r(t)$ to 0 for $t \to \infty$. The best results in this area up to now are those of Breuer and Major (1983) for a discrete time process and Chambers and Slud (1989) for a continuous time process. The general result is that if the function f is of Hermite rank k, and $r \in L_k(-\infty, \infty)$, then (1.2) has a limiting normal distribution.

The second category of results is a collection of "non-Central Limit Theorems", where (1.2) has a limiting distribution which, except for the case of a function of Hermite rank 1, is not a normal distribution. Such theorems state in their hypotheses that the process is assumed to have "long range dependence". More precisely, it is assumed that i) The function f is of Hermite rank k, for some $k \geq 1$, and ii) For some $\alpha < 1/k$, $r(t)$ is regularly varying of index $-\alpha$ for $t \to \infty$ (Dobrusin and Major (1979), and Taqqu (1979)).

Our main result represents an extension of previous results for the case of a function of Hermite rank 1, in which the limiting distribution is normal under either the mixing condition $r \in L_1$ or the long range dependence condition of the regular variation of r for index $-\alpha > -1$. Our only assumptions on $r(t)$ are that it be nonnegative, and converge to 0 for $t \to \infty$. The limiting distribution of (1.2) is always normal; however, the variance of the limiting distribution has two forms, depending on whether $r \in L_1$ or $r \notin L_1$.

The result is:

THEOREM 1.1. *If* $r(t) \geq 0$ *for* $t \geq 0$, *and* $r(t) \to 0$ *for* $t \to \infty$, *then, for any function* $f \in L_2(\phi)$ *such that*

$$\gamma = \int_{-\infty}^{\infty} y\,\phi(y)\,f(y)\,dy \neq 0\ , \tag{1.5}$$

the random variable

$$\frac{\int_0^t f(X(s))\,ds - t\int_{-\infty}^{\infty} f(y)\,\phi(y)\,dy}{\left(2\gamma^2\int_0^t (t-s)\,r(s)\,ds\right)^{1/2}} \tag{1.6}$$

has a limiting normal distribution with mean 0. *The variance of the limiting distribution is equal to* 1 *if* $r \notin L_1$ *and is equal to*

$$\frac{\sum_{k=1}^{\infty} f_k^2 \int_0^{\infty} r^k(s)\,ds}{\gamma^2 \int_0^{\infty} r(s)\,ds}\ , \tag{1.7}$$

if $r \in L_1$.

The main idea of the proof is that $\int_0^{\infty} r(t)\,dt$ is either finite or $+\infty$. In the former case, the result follows from that of Chambers and Slud (1989). In the latter case the process has a form of long range dependence, and the contribution of the theorem is that the limiting distribution exists without a precisely assumed form of $r(t)$ for $t \to \infty$.

2. Proof of the Theorem.

Let $\phi(x, y; r)$ represent the standard bivariate normal density with correlation r. A direct calculation shows that the variance of $\int_0^t f(X(s))\,ds$ is equal to

$$2\int_0^t (t-s)\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x)\,f(y)\big[\phi(x,y;r(s)) - \phi(x)\,\phi(y)\big]\,dx\,dy\,ds\ .$$

By the diagonal expansion of $\phi(x, y; r(s))$ in Hermite polynomials (Cramér 1946 page 290) the integral above is equal to

$$2\int_0^t (t-s)\sum_{k=1}^{\infty} f_k^2 r^k(s)\,ds\ , \tag{2.1}$$

with f_k as in (1.4).

Suppose $r \in L_1$; then, for every $k \geq 1$,

$$t^{-1} \int_0^t (t-s)\, r^k(s)\, ds \to \int_0^\infty r^k(s)\, ds \ ,$$

for $t \to \infty$, and so the expression (2.1), divided by t, converges to

$$2 \sum_{k=1}^{\infty} f_k^2 \int_0^\infty r^k(s)\, ds \ .$$

Since, by definition, $H_1(x) = x$, we have $f_1 = \gamma \neq 0$ under (1.5). Therefore, the assertion of the theorem in the case $r \in L_1$ is the special case of the result of Chambers and Slud (1989) for a function of Hermite rank 1.

Next consider the case $r \notin L_1$. Since $\gamma = f_1$, (2.1) is equal to the sum of

$$2\gamma^2 \int_0^t (t-s)\, r(s)\, ds \tag{2.2}$$

and

$$2 \int_0^t (t-s) \Big[\sum_{k=2}^{\infty} f_k^2\, r^k(s) \Big]\, ds \ . \tag{2.3}$$

Our first aim is to show that the expression (2.3) is of order smaller than (2.2) for $t \to \infty$, and this will establish that the denominator in (1.6) is asymptotically equal to the standard deviation of the numerator:

$$Var \int_0^t f(X(s))\, ds \sim 2\gamma^2 \int_0^t (t-s)\, r(s)\, ds \ . \tag{2.4}$$

For arbitrary fixed $T > 0$, take $t > T$ in (2.3), and write the latter as the sum of two terms

$$2 \int_0^T (t-s) \Big[\sum_{k=2}^{\infty} f_k^2\, r^k(s) \Big]\, ds \tag{2.5}$$

and

$$2 \int_T^t (t-s) \Big[\sum_{k=2}^{\infty} f_k^2\, r^k(s) \Big]\, ds \ . \tag{2.6}$$

The term (2.5) is at most equal to

$$2tT\sum_{k=2}^{\infty} f_k^2 = \text{ constant } \times t \ .$$

This is of order smaller than that of the right-hand member of (2.4) because, by L'Hospital's Rule and the assumption $r \notin L_1$,

$$\lim_{t\to\infty} \frac{t}{\int_0^t (t-s)\, r(s)\, ds} = \lim_{t\to\infty} \frac{1}{\int_0^t r(s)\, ds} = 0 \ .$$

The term (2.6) is at most equal to

$$2\int_0^t (t-s)\, r(s)\, ds \times \sum_{k=2}^{\infty} f_k^2 \times \sup_{s\geq T} r(s) \ . \tag{2.7}$$

The ratio of the latter expression to the right hand member of (2.4) is at most

$$\gamma^{-2} \sum_{k=2}^{\infty} f_k^2 \times \sup_{s\geq T} r(s) \ .$$

Since T is arbitrary, and $r(t) \to 0$ for $t \to \infty$, the expression above can be made arbitrarily small by choosing T sufficiently large. This concludes the argument that (2.3) is of smaller order than (2.2), and this confirms (2.4).

By expansion of f in Hermite polynomials, we find that

$$\int_0^t f(X(s))\, ds - \int_{-\infty}^{\infty} f(x)\, \phi(x)\, dx$$

is representable as

$$\sum_{k=1}^{\infty} f_k \int_0^t \frac{1}{\sqrt{k!}} H_k(X(s))\, ds \ . \tag{2.8}$$

It is known that the random variables

$$\int_0^t \frac{1}{\sqrt{k!}} H_k(X(s))\, ds \ , \quad k = 1, 2, \ldots$$

are uncorrelated and have means 0. The first term in (2.8) is

$$\gamma \int_0^t X(s)\, ds \ , \tag{2.9}$$

which has a normal distribution with mean 0 and variance (2.2). According to the method of Berman (1979), the fact that the variance of the sum (2.8) is asymptotically equal to the variance of the first term implies, in this case, that the limiting distribution of the sum (2.8) is equal to the limiting distribution of the first term in (2.9). □

REFERENCES

BERMAN, S. M. (1979) High level sojourns for strongly dependent Gaussian processes. *Z. Wahrscheinlichkeitstheorie Verw. Geb.* **50** 223-236.

BREUER, P. & MAJOR, P. (1983) Central limit theorems for nonlinear functionals of Gaussian fields. *J. Multivar. Anal.* **13** 425-441.

CHAMBERS, D. & SLUD, E. (1989) Central limit theorems for nonlinear functionals of stationary Gaussian processes. *Probability Theory Rel. Fields* **80** 323-346.

CRAMER, H. (1946) *Mathematical Methods of Statistics.* Princeton Univ. Press, Princeton.

DOBRUSIN, R. L. & MAJOR, P. (1979) Non-central limit theorems for nonlinear functionals of Gaussian fields. *Z. Wahrscheinlichkeitstheorie Verw. Geb.* **50** 27-52.

TAQQU, M. S. (1979) Convergence of integrated processes of arbitrary Hermite rank. *Z. Wahrscheinlichkeitstheorie Verw. Geb.* **50** 53-83.

The Voronoi Diagram for The Euclidean Traveling Salesman Problem is Piecemeal Quartic and Hyperbolic

T.M. Cronin
CECOM Center for Signals Warfare
Warrenton VA 22186-5100

Abstract. It is shown that the Voronoi diagram for the Euclidean traveling salesman problem is piecemeal quartic and hyperbolic. Previous attempts to leverage the traditional (linear) Voronoi diagram upon the problem have failed; in particular, counterexamples have demonstrated that the optimal tour need not traverse the Voronoi dual. In this paper, the shortest tour is treated as the union of a set of perturbations of the convex hull, with interior cities added incrementally, one at a time. A perturbation is defined to be the union of a new city with a subpath which connects two adjacent hull vertices to a set (possibly null) of previously entered interior cities. The length of a perturbation is therefore equal to the sum of two variable distances, minus the sum of a set of fixed distances. This length is called the elliptic length of the perturbation. Beginning with the convex hull, a single city is randomly added to the interior, and the hull is perturbed to capture the new city in optimal fashion. For a perturbation of a specific elliptic length, this quantity determines an ellipse symmetric about a hull segment, with foci at the segment endpoints. Any other hull perturbation of the same length defines another ellipse symmetric about some other hull segment. As the perturbation length is allowed to vary continuously from zero to infinity, a set of confocal ellipses is produced about each hull segment, and the intersection across all other such sets produces a set of quartics. For the special case in which hull segments share an endpoint (focus), the locus is a hyperbola. Each hull segment is bounded by those quartics for which the segment is the source of minimal perturbation length. The region of the plane thus bounded is called the quartic Voronoi cell for that segment. There is a quartic cell corresponding to each hull segment, and the union of all such cells forms the Voronoi diagram of the hull. Now, if a random city is injected into the hull, and the city is observed to lie in a specific Voronoi cell, we know that to produce the minimal tour, the city must be connected to the endpoints of the hull segment corresponding to the cell, and in turn the endpoints of the hull segment must be disconnected. If one maintains the proper canonical forms to alter the topology of the perturbation space when a new city is added, the technique may be extended to accommodate multiple interior cities. The quartic Voronoi diagram is shown to differ from the traditional Voronoi diagram in three distinct ways: it depicts shortest tour connectivity rather than point-to-point proximity; its cell boundaries are quartic and hyperbolic rather than linear; and the diagram is bounded by the convex hull rather than being unbounded (although this last constraint may be relaxed to add new cities outside the hull). A naively derived ceiling function demonstrates that an unsupervised perturbation approach is of exponential complexity, with a scaling factor as a function of the size of the hull. By resorting to an algorithm which exploits the canonical forms, it is shown that this bound may be diminished to $O[n^3]$. The algorithm is demonstrated for a database consisting of the forty-eight capitals of the contiguous United States. Open research issues include whether the technique may be extended to accommodate a hull which encloses an arbitrary number of cities, and whether the ceiling function may be further reduced

Statement of the Problem. The Euclidean traveling salesman problem (ETSP) is a special case of the general traveling salesman problem (TSP). Given a set of cities and the associated costs between pairs of cities, the goal of the TSP is to find the optimal tour which visits every city exactly once, except for the start city, which is revisited at tour's end. Unlike the TSP which utilizes a general cost function to link cities, the ETSP employs the Euclidean distance between cities as the metric, and equates optimality with shortest tour length. The objective of this research is to attempt to rigorously characterize the

underlying geometry of ETSP tour construction, and subsequently to pursue an algorithm for an exact solution to the problem for city databases of modest size.

Background. The traveling salesman problem has been an outstanding research issue for over a century, and has been approached computationally since the end of the second world war [L1]. It is important to differentiate between the TSP and TSP decision; the former requests a list of cities ordered as they appear in the optimal tour, whereas the latter seeks a yes or no answer to the question "is there a tour of cost k or less"? In 1972, it was proven that TSP decision is NP-complete [K2]; and in 1976 it was shown that ETSP decision with discretized distance is also NP-complete [P4, G1]. The ETSP with non-discretized distance is NP-hard in the strong sense [G1]. The failure of the ETSP to yield to known problem-solving strategies has caused the vast majority of researchers to abandon the search for an exact algorithm, and instead strive for fast approximation techniques. Many heuristic algorithms have been developed to date; they include: *k-opt edge exchange* [L3, J4], *branch-and-bound* [L4], *simulated annealing* [M1, J4], *neural networks* [F1, J4], *genetic algorithms* [B4]; *and elastic bands* [D4]. A preeminent researcher in the field is of the opinion that for consistently high quality solutions on databases of very large scale, the Lin-Kernighan edge exchange algorithm has few competitors [J5]. Chapters 5-7 of reference [L1] provide valuable suggestions for evaluating the performance of some of the heuristic methods.

An Historical Perspective of the Euclidean Traveling Salesman Problem and the Voronoi Diagram.

Operations research has been the historical forum for ETSP. There have been very few efforts possessing a computational geometry flavor. In the seventies, the issue was raised about whether a ETSP optimal tour must necessarily traverse adjacent cells of the Voronoi diagram [S2]. This conjecture has subsequently been answered in the negative. A counterexample for a degenerate case was discovered in 1983 [K1], and one for the general case was skillfully crafted five years later [D3]; the latter counterexample is portrayed at Figure 1. It will be shown in this paper that the fundamental reason for the difficulty in applying the traditional Voronoi diagram to ETSP is that the search space imposed by a perturbation of Euclidean distances is non-linear (in particular, it is quartic), whereas the traditional Voronoi cell possesses linear boundaries. It will also be shown that with modifications, the traditional Voronoi diagram may be extended to portray the optimal tour for an n-city problem, given the optimal tour for n-1 cities.

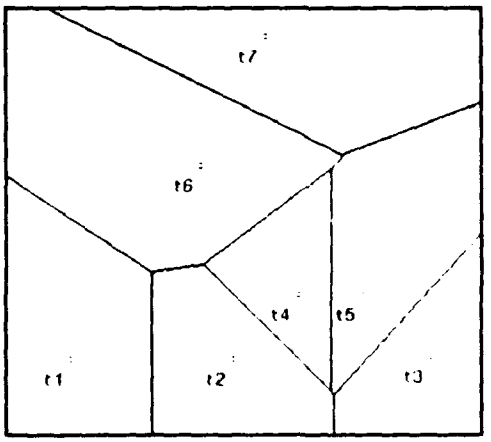


Figure 1. The traditional Voronoi diagram, computed for the Dillencourt dataset. In the optimal tour, t4 and t3 are connected, but their respective Voronoi cells are not, which counterindicates Shamos' conjecture that an optimal tour must traverse the Voronoi dual.

The Euclidean Shortest Tour as a Perturbation of the Convex Hull.

In 1957, Barachet proved that there exists an optimal tour which preserves the relative order of the points on the convex hull [B1]. This result implies that the shortest tour may be expressed as a hull deformation produced by an excursion into the interior, to capture points which do not lie on the hull (Figure 2). In 1977, a heuristic was developed to utilize the hull as an initial starting tour, and to attach interior cities based on a two step procedure [S4]. First, the sum of the distances from an interior city to the endpoints of an existing segment are computed, and the length of the segment subtracted; across all existing segments, the minimal such expression associates the city with a particular segment. The next step involves selecting a city to be inserted based on the maximal angle formed with its associated segment. The procedure iterates until a Hamiltonian cycle is formed utilizing all interior cities. Finally, an arbitrating function decides if the resultant cycle is sufficiently accurate. Analysis of the method has indicated that it is superior to some methods which do not utilize the hull [G3]. Nevertheless, due to the fact that the approach is only approximate, the tour produced is generally suboptimal, and it is not well understood why the heuristic performs as it does.

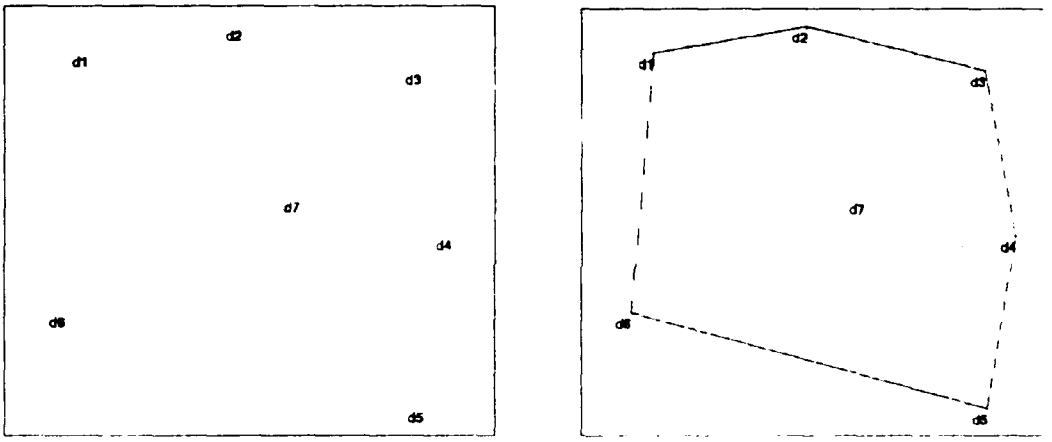


Figure 2. What is the shortest tour connecting cities d1-d7? We know that the tour must preserve the order of the cities lying on the the convex hull, so a natural way to proceed is to perturb the hull. The problem thus reduces to finding the optimal (shortest) way to attach city d7 to a pair of adjacent hull vertices.

Terminology and Notation.

In the following discussion, we shall call any excursion into the interior from two adjacent hull vertices a *perturbation* of the hull. It is important to note that a perturbation of the hull entails the corresponding loss of the segment which connects the two adjacent hull vertices. If a hull segment is unperturbed it is called a *null perturbation*. A *tour* is defined to be the union of a set of perturbations in counterclockwise order, as they appear about the hull. A convention will be adopted to represent certain perturbation concepts: a perturbation is denoted by the letter "π", a tour is denoted by the letter "τ"; the length of perturbation π_i or tour τ_j is denoted respectively "len π_i" or "len τ_j"; the Euclidean distance between points p, q is denoted "d(p, q)"; the set of cities lying on the convex hull is denoted "H"; the convex hull itself is denoted "τ_H"; the number of cities in set S, also called the *order of S*, is denoted [S]. In concluding this introductory section, we formalize the definition of a perturbation, and prove three minor counting theorems.

Definition. *Hull Perturbation.* Given convex hull τ_H ordered with counterclockwise orientation, and the set I of interior cities. A *perturbation of the hull* π_k is an ordered subpath $\pi_k = h_k \cup I_j \cup h_{k+1}$; $I_j \subseteq I$. [Note that I_j may be the null set, in which case π_k is a null perturbation].

Theorem. When computing the Euclidean shortest tour, the number of perturbations of the convex hull cannot exceed the rank of the hull. Proof: By definition, a hull perturbation is an excursion into the interior of the hull which connects two adjacent hull vertices to a subset of the interior. Without regard to order, there are n ways to connect n adjacent hull vertices (the first to the second; the second to the third; ..., the nth to the first), producing a set of n hull segments. Each of these segments may be the source of a perturbation.

Theorem. If the size of the set of interior cities [I] exceeds the size of the hull [H], then the shortest Euclidean tour must contain a hull segment perturbation of order at least [I] - [H] + 1. Proof: from the pigeonhole principle, since there are more interior cities than hull segments, some hull segment must be assigned at least [I] - [H] + 1 interior cities.

Theorem. If the size of the hull [H] exceeds the size of the set of inner cities [I], then there must exist at least [H - I] hull segments which remain unperturbed when constructing the shortest Euclidean tour. Proof: again, from the pigeonhole principle, since there are fewer interior cities than hull segments, at least [H - I] hull segments must be null perturbations.

An Arbitrary Hull Enclosing a Single Interior City.

Since we know intuitively that the shortest Euclidean tour may be represented as a perturbation of the hull, let us proceed with the simplest case by introducing a single interior city into an arbitrary hull. It is natural to derive under what conditions a perturbation initiated from a given hull segment to the new city results in the shortest tour, versus one initiated from another segment. If len τ_H represents the sum of the lengths of the segments which comprise the convex hull, and p is an arbitrary point introduced into the hull interior, then to produce the shortest tour, one is interested in minimizing the expression len τ_H + [d(p, h_i) + d(p, h_{i+1}) - d(h_i, h_{i+1})] $\forall$ $h_i \in H$. The boundaries of equal hull perturbation are those for which len π_i = len π_j, for distinct elements h_i, $h_j \in H$. To formally characterize the boundaries of equal perturbation requires that the expressions for perturbations initiated from two distinct hull segments be set equal to each other, and the resulting equation solved.

The Elliptic Distance of a Point to Two Other Points.

During traveling salesman problem solving, the operation which decrements the length of a segment from the sum of the the distances from the segment endpoints to an arbitrary point is sufficiently fundamental to be given a special name, which we will call the *elliptic distance*.

Definition. The *elliptic distance* of a point p to two points q, r, denoted $d_e(p,q,r)$, is defined to be:

$$d_e(p,q,r) = d(p, q) + d(p, r) - d(q, r) \qquad [1]$$

Derivation of the Quartic Locus for the Single Interior City Problem.

Theorem. In general, the Voronoi diagram of the convex hull of the set of cities for the Euclidean traveling salesman problem has quartic edges.

Proof: Let p be an arbitrary point on the interior of convex hull τ_H, and let π_i, π_j be perturbations of two distinct hull segments such that len π_i = len π_j.

$$\text{len } \pi_i = [\, d(p, h_i) + d(p, h_{i+1}) - d(h_i, h_{i+1})] \text{ for some } h_i \in H, \text{ and}$$
$$\text{len } \pi_j = [\, d(p, h_j) + d(p, h_{j+1}) - d(h_j, h_{j+1})] \text{ for some } h_j \in H.$$

Let len π_i = len π_j = k_{ij}, which represents some specific elliptic length. Thus,

$$d_e[p, h_i, h_{i+1}] = d_e[p, h_j, h_{j+1}] = k_{ij} \qquad [2]$$

Equation [2] describes two ellipses, the first with major axis aligned with the hull segment having endpoints h_i, h_{i+1}, and the second aligned with the hull segment having endpoints h_j, h_{j+1}. The endpoints of the hull segments are the respective foci of the ellipses. The distances involving point p are variable, while the distances on the hull are constant. Let us represent the two ellipses as follows:

$$x^2/a^2 + y^2/b^2 = 1 \qquad [3]$$
$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0 \qquad [4]$$

Equation [3] denotes one of the two ellipses of interest after it has been rotated and translated to be in standard form about the origin. Equation [4] represents the second ellipse with the coefficients A, B, C, D, and E determined in the coordinate system of [3]. To characterize the locus of equal perturbation, we are required to simultaneously solve [3] and [4].

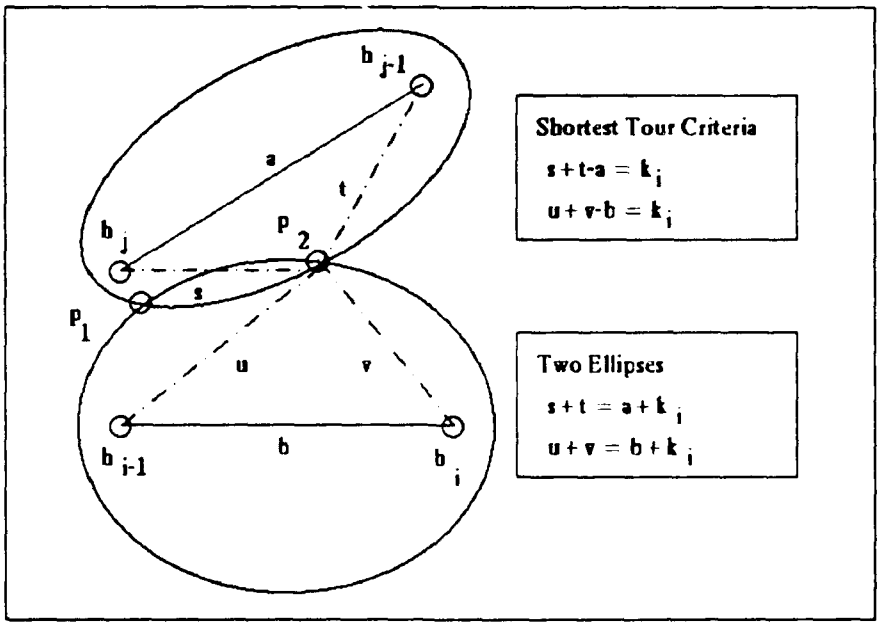


Figure 3. The locus of equal perturbation between hull segments is obtained by intersecting two ellipses.

From [3], we obtain: $y = +/- (b/a)\sqrt{a^2 - x^2}$ [5]

Substituting the positive root for y in [4] yields

$$Ax^2 + Bx(b/a)\sqrt{a^2 - x^2} + C(b/a)^2(a^2 - x^2) + Dx + E\sqrt{a^2 - x^2} + F = 0 \quad [6]$$

Factoring, and moving the radical to the right side of the equation produces

$$[A - (b^2/a^2)C]x^2 + Dx + b^2C + F = -(b/a)\sqrt{a^2 - x^2}\,(Bx + E) \quad [7]$$

Squaring both sides of [7] to clear the radical, and gathering coefficients with respective powers of x results in the equation:

$$\begin{aligned}
&[A^2 - (2b^2AC/a^2) + b^4C^2/a^4 + (b^2/a^2)B^2] && x^4\\
&+ [2AD - (2b^2CD/a^2) + (2b^2BE/a^2)] && x^3\\
&+ [2Ab^2C + 2AF - (2b^4C^2/a^2) - (2b^2CF/a^2) + b^2E^2/a^2 - b^2B^2 + D^2] && x^2\\
&+ [2b^2CD + 2DF - 2b^2BE] && x\\
&+ b^4C^2 + 2b^2CF - b^2E^2 + F^2 = 0. && [8]
\end{aligned}$$

QED. Thus the locus of equal perturbation for inserting a random city into the hull is defined by a quartic equation, with coefficients expressed in terms of the parameters for two ellipses, where the ellipses are symmetric about segments formed by linking two cities.

A Graphic Depiction of a Simple Quartic Space.

Figure 4 illustrates an example of a quartic space imposed on a four city database. The segment containing z1 and z2 is fixed in the plane. The segment containing z3 and z4 is allowed to pivot about z4, with z3 being rotated counterclockwise ninety degrees through an angle θ, in increments of ten degrees. We are trying to find the locus such that a perturbation from the segment z1-z2 is equal to a perturbation from segment z4-z3. Initially, when z3 is on the x-axis, the locus is a horizontal line lying halfway between the two segments. For the sake of argument, when the angle $\theta = 0$, let the locus be the line $y = k$. At ten degrees, the locus lifts slightly from the horizontal and develops curvature. At about forty-five degrees, the locus develops a prominent maximum, and also mainfests two inflection points; it visually resembles the planar curve known as the Witch of Agnesi. Beyond forty-five degrees, the locus gradually loses its smooth maximum and develops a pronounced cusp; the fact that four distinct roots exist is now apparent. Finally, at ninety degrees of rotation, the locus becomes a diagonal line connecting z1 with the initial position of z3.

Note that a peculiar phenomenon has occurred. Although the segment containing z3 has been allowed to rotate ninety degrees (from the line $y = 0$ to the line $x = 0$), the corresponding locus has rotated only forty-five degrees (from the line $y = k$ to the line $y = -x + 2k$). Therefore, the angular range of the output is only half that of the input. It should also be noted that realistically, the behavior which produces the cusp does not occur, for when city z3 is rotated beyond a certain critical angle, it is absorbed by segment z1-z2, and the shortest tour becomes z1-z3-z2-z4-z1, rather than z1-z2-z3-z4-z1.

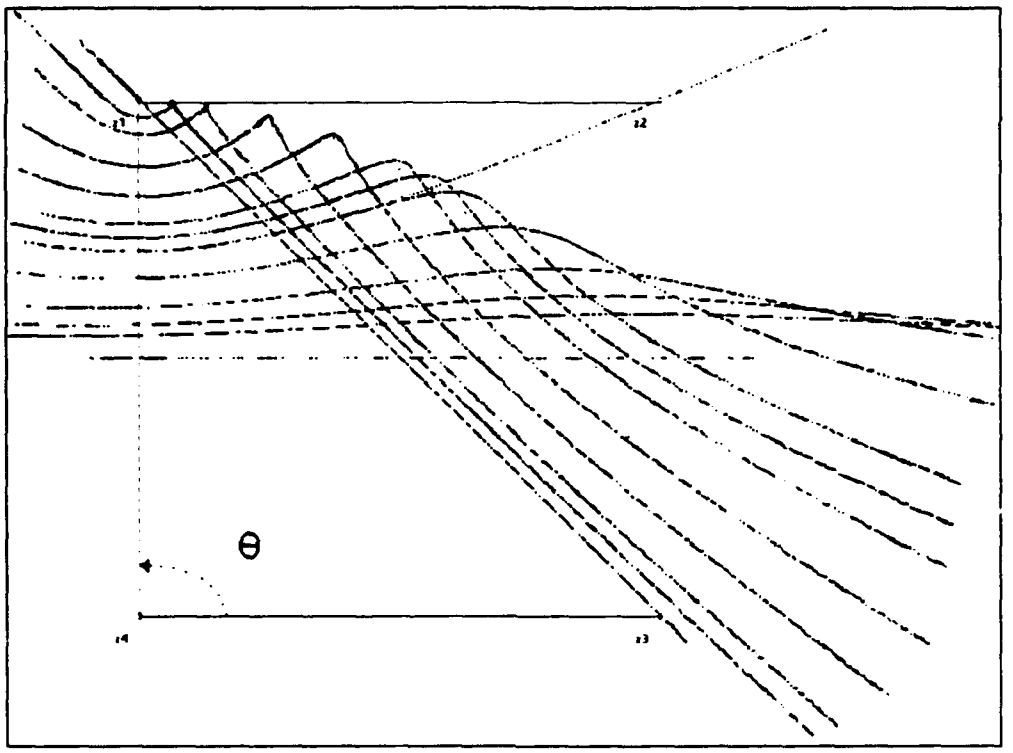


Figure 4. A plot of a quartic locus, using z4 as a pivot, while rotating z3 counterclockwise.

Two Hull Segments Sharing an Endpoint Produce a Hyperbolic Locus.

In this section, a corollary to the theorem is proven to show that in a special case which frequently occurs during traveling salesman problem solving, the locus of equal perturbation is hyperbolic rather than quartic. The simplest hypothesis to maintain when inserting another city into the hull is that the current set of perturbations will merely be extended by the new city, without radically altering the topology. As will become apparent below, the simple extension of a perturbation is arbitrated by an intrapath hyperbolic discriminator; the more complex operation of reasoning between perturbations requires a quartic discriminator.

Corollary. When two hull segments share a city, the Voronoi edge is a semi-hyperbola

Proof: Refer to Figure 5 below. Let the endpoints of the segments be respectively h_{i-1}, h_i and h_i, h_{i+1}. Let a be the length of the hull segment connecting h_i and h_{i+1}, and b the length of the segment connecting h_i and h_{i-1}. Let p be an arbitrary point on the locus. Let x be the distance from h_{i+1} to p, y the distance from h_i to p, and z the distance from h_{i-1} to p. We proceed to derive the locus:

$$\begin{aligned} d_e(h_i, h_{i-1}) &= d_e(h_i, h_{i+1}) \\ x + y - a &= y + z - b \\ x - a &= z - b \\ x - z &= a - b \end{aligned} \qquad [9]$$

QED Equation [9] represents a semi-hyperbola passing through hull vertex h_i, with foci at h_{i-1} and h_{i+1}. It is bowed toward the longer of the two hull segments. In the case

when the two hull segments are of equal length, the semi-hyperbola degenerates to a line.

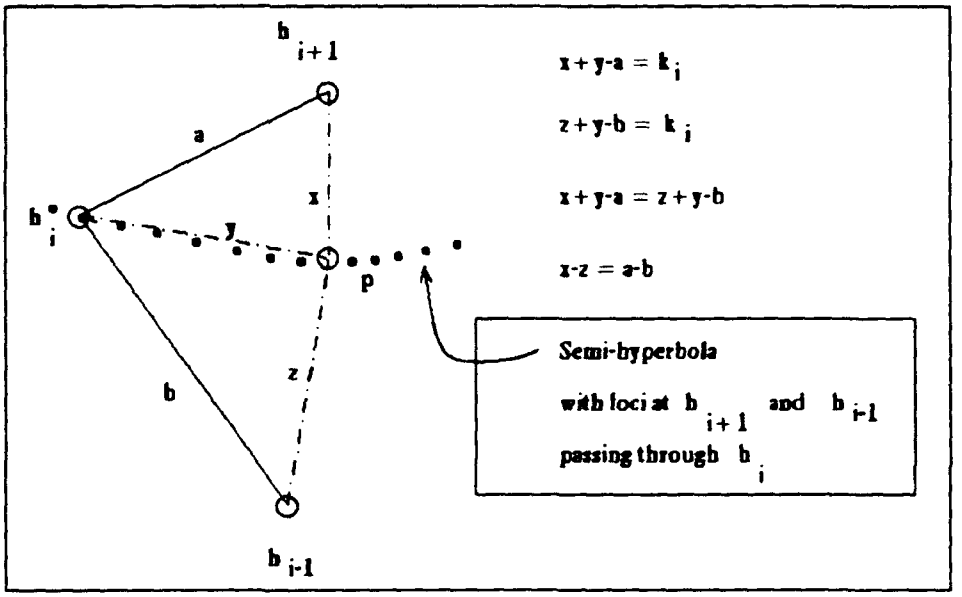


Figure 5. A hyperbolic locus results when two hull segments share an endpoint.

The Quartic Voronoi Diagram to Determine an Exact Solution to the Single Interior City Problem.

It is clear that to develop the delimiters of equal hull perturbation, we are required to develop a subset of the union of quartics and hyperbolas induced by the elliptic distance between pairs of hull segments. This subset is called the quartic Voronoi diagram of the hull, and is defined as follows:

Definition. Given convex hull τ_H, the quartic Voronoi diagram of the hull, VorQ (τ_H), is defined to be:

$$\text{VorQ}(\tau_H) = \{ x \mid d_e(x, h_i, h_{i+1}) = d_e(x, h_j, h_{j+1}) \text{ for some } h_j \in H, \text{ and } d_e(x, h_i, h_{i+1}) < d_e(x, h_k, h_{k+1}) \ \forall k \neq j \}.$$

For the seven city example introduced above, VorQ (τ_H) is displayed at Figure 6. City d7 is properly contained within the quartic Voronoi cell corresponding to the segment connecting d5 and d6, implying the segment must be perturbed to capture d7. Note that prior to introducing city d7, the existing optimal tour is the convex hull. The Voronoi diagram consists of some edges passing through the existing tour's vertices (the cities on the hull), and some which do not pass through any of the cities in the search space. In the former case, the edges are composed of hyperbolas, whereas in the latter case the edges are pieces of quartic curves. It is important to keep this concept in mind, because it will be revisited when the diagram is extended in general to accommodate a new city deposited into an arbitrary optimal tour space. The generalization will be seen to function in the following manner: if the existing optimal tour is extended simply by inserting a new city between two cities in the tour, the locus of equal perturbation which arbitrates the decision is a semi-hyperbola; otherwise, a more complex decision process must be invoked to reason across the perturbation space, and the locus is a quartic polynomial.

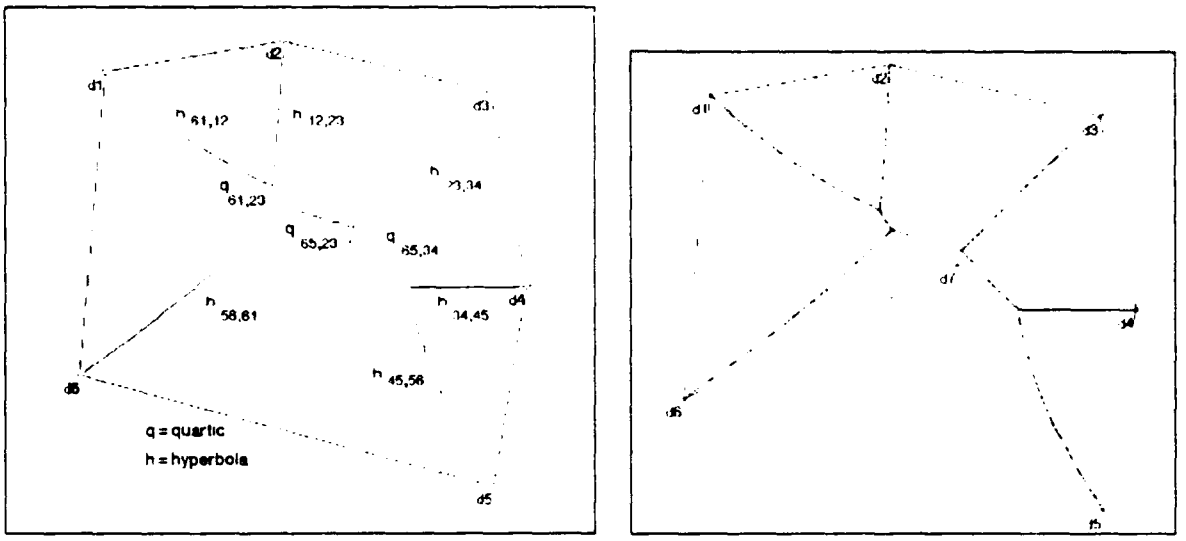


Figure 6 (computed quartics and hyperbolas). The quartic Voronoi diagram of the hull as a connectivity map. The curves passing through the hull vertices are hyperbolas; whereas the others are quartics. City d7 resides within the quartic Voronoi cell corresponding to the segment connecting d5 and d6, and therefore the optimal connection is as shown at the right.

A Two-City-in-a-Hull Example.

We will continue in this vein by commiting to computer memory the optimal perturbation for city d7, and introducing yet another city into the perturbed space. Figure 7 depicts an instance of a *two-city-in-a-hull quartic Voronoi diagram*. City d7 has been fixed, after having discovered its optimal perturbation (d6-d7-d5) in a previous step. A new city (p) is about to be introduced. At the left, the quartic edges demarcating optimal connectivity of p with d2, d3, d5, d6, and d7 are displayed, with the globally relevant pieces highlighted. Similar plots may be obtained for other pairs of vertices; for the complete interaction between segment d5-d6 and each of the other five hull segments, please refer to the Appendix (where d7 appears under the alias of d8). When computed across all relevant pairs, the quartic Voronoi diagram emerges (right). As an example, if p happens to fall in the quartic cell labeled with the descriptor "6p75", the optimal tour must insert p between cities d6 and d7, in the already established perturbation d6-d7-d5. However, if p happens to fall in the cell labeled "1p6;675", two distinct perturbations are required: the one involving p is issued from segment d1-d6; whereas the one involving d7 is issued from segment d6-d5.

Note that in the vicinity of city d7, a single perturbation from some hull segment is sufficient to ensure optimality. However, if city p happens to reside in one of the quartic cells beyond this region, it is necessary to perturb two hull segments to achieve optimality. Perhaps the most intriguing aspect of the Voronoi diagram as a connectivity map is that it partitions the plane into cells which indicate precisely how to maintain optimality when inserting an arbitrary city into the current tour. What this really means is that one can predict how to attach a new city to an optimal tour, without specifying the coordinates of the city ahead of time. The implications are profound, for if an efficient algorithm can be designed to construct (or perhaps merely reflect) the quartic diagram for arbitrarily large sets of cities, it follows that a dynamic programing approach is sufficient to solve the problem exactly. The

fourth-order complexity inherent to the loci of tour constraints in large part explains why those approaches which subscribe to Dantzig's linear simplex have to date failed to solve the problem.

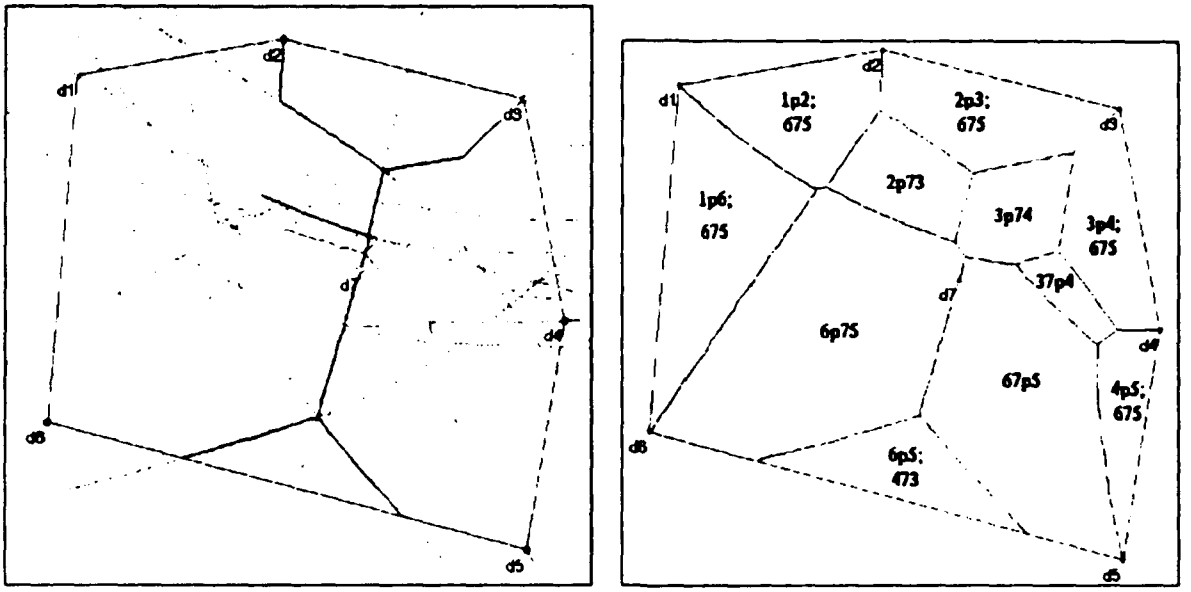


Figure 7 (computed quartics and hyperbolas). The quartic Voronoi diagram for two interior cities.

Nested Hull Traversal, Outside-in vs. Inside-out.

Since the theory is based on a perturbation of the convex hull, there is an obvious requirement to secure an algorithm which efficiently computes the hull. It is shown in reference [K3] that in the plane, the hull may be optimally computed in O [n * log h] time. With the traveling salesman problem, we interpret n as the total number of cities, and h as the number of cities on the convex hull. For our implementation, we will compute the entire nested hull decomposition, sometimes called the "onion" [E2]. The purpose of computing the onion is to gain control of the search space by attempting to insert the cities uniformly into the hull, to limit generation of "greedy" perturbations. A greedy perturbation occurs when by mere virtue of having probed sufficiently far into the hull, a perturbed hull segment continues to absorb cities which rightfully belong to another perturbation. Reference [C2] demonstrates that a planar nested hull structure may be constructed in O [n * log n] time. However, in the implementation described below, we will utilize an algorithm due to Eddy [E1], with time complexity O [n^2], but with average run time O [n].

Recall that by convention we order a hull with counterclockwise orientation, smallest ordinate first. If we label the outer hull with index 0, and label each inner hull with an ordinal number formed by incrementing the index by 1, it is seen that during processing, a city will be inserted based on a primary key equal to the ordinal number of its hull, and a secondary key equal to its relative counterclockwise position within its hull. The exception to this rule is to reject the insertion if it causes some unprocessed city to be bypassed.

A rather startling twist to the outside-in approach is based on the fact that the quartic loci of equal perturbation extend both inside and outside a hull. If we start with the innermost hull (the core of the onion), in theory we should be able to probe outward one hull at a time, maintaining optimality

as we proceed. This technique, which we will call the *inside-out* approach, is in fact as valid as the other. An experiment detailed below demonstrates that both approaches are indeed capable of producing the optimal tour.

The Topology of Quartic Voronoi Space, in the Context of the Shortest Tour.

In this section we develop the canonical forms required to maintain quartic incremental optimality. Three primitive operations will be informally introduced, and then developed more rigorously.

The quartic Voronoi diagram partitions the plane into cells, the boundaries of which demarcate the locus of the shortest tour among various combinations of subtours. It is intuitively obvious that if a newly introduced city lies within an arbitrarily small neighborhood of an existing optimal tour, the new tour can be formed simply by extending the old space to the new city. This topological structure, which we shall call ***hyperbolic extension space***, is computationally the simplest hypothesis to be entertained when introducing a new city. Hyperbolic extension space preserves an existing tour by extending an existing perturbation to encompass the new city.

A markedly different topology is manifested when an extended perturbation interplays with another perturbation, which is located two hull segments backward (forward) to compel the issuing of a new perturbation at the preceding (subsequent) hull segment, which is called a *shunt to the left (shunt to the right)*. This topology is called ***quartic shunt space***. It addresses the issue of maintaining optimality in a radial fashion; i.e., in a manner roughly orthogonal to the convex hull which defines the baseline tour. An intuitive way to describe this canonical form is that it acts as a monitor of flanking behavior on both sides of a perturbation, and cedes the flank to a neighboring hull segment when necessary to maintain optimality.

Because of the existence of the two distinct topologies, it is necessary to maintain separate computational hypotheses in parallel (Figure 8). An extension occurs if a new city lies in one of the extension cells in the lower portion of the diagram at the top. However, if the city lies within a Voronoi shunt cell as indicated at the top, a transition to quartic shunt space occurs when two existing perturbations are bridged (diagram at bottom). This not uncommon spatial phenomenon may radically alter the global shape of the tour, and must be hypothesized every time a new city is processed, to guarantee tour optimality.

The final topology deals with the issue of perturbation encroachment. There are instances when a perturbation probes suffciently far into the hull that for the sake of optimality it is necessary for it to claim cities from another perturbation. This spatial phenomenon produces the third topology which we call *quartic interchange space*. Quartic interchange space consists of those Voronoi cells which indicate that cities from one or more perturbations are to be exchanged into an extended or shunted perturbation. Quartic interchange is invoked after hyperbolic extension and quartic shunting, which are performed in parallel. It can be an operation of quadratic complexity, because an existing perturbation may be broken into sections and nullified by the operation, with separate sections being absorbed by separate perturbations. Quartic interchange is particularly relevant when using the outside-in nested hull approach, because at certain moments in time perturbations from across the hull begin to collide with those on the near side, and the interaction must be arbitrated to preserve optimality.

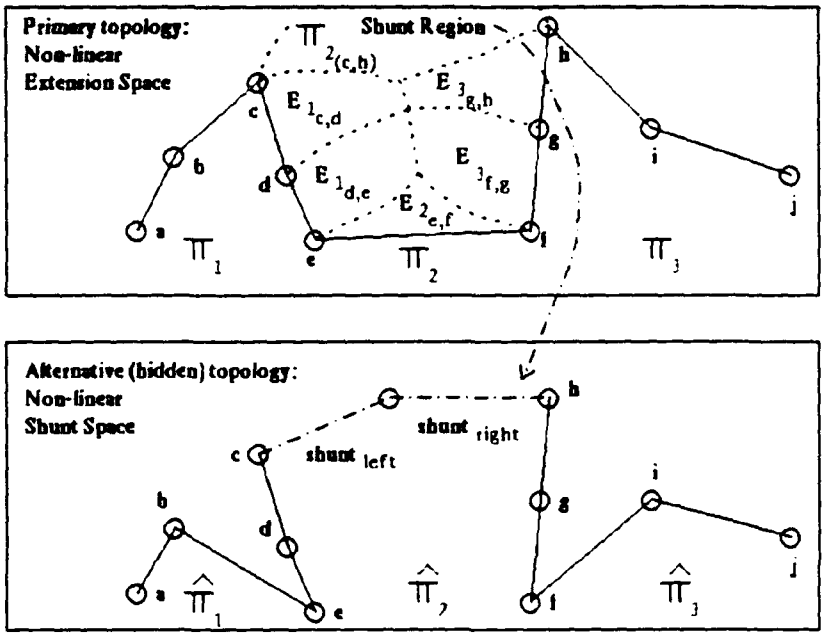


Figure 8. Maintaining two quartic topologies in parallel.

Assume that after k cities have been optimally connected to the convex hull, we would like to know under what conditions it is possible to simply extend the tour to a new city, vs. radically altering the tour by permitting the new city to link others which are currently non-adjacent. It is obvious that if the new city is within a small spatial neighborhood of the existing tour, optimality is preserved by simply inserting the city into the tour between two cities. The question of which two cities is governed by a set of hyperbolas which pass through the endpoints of the segments that connect the ordered list of cities defining the current optimal tour. Whether or not the new city is within a suitable neighborhood of the current tour is arbitrated by a set of quartic curves which discriminate if a shunting operation is in order.

Hyperbolic Extension Space.

It is a simple matter to connect a new city to an existing perturbation, if that is what is desired (it will be seen below in the section on quartic shunting that optimality is not always preserved by simply extending a perturbation). The city is connected to those two cities in the perturbation for which the elliptic distance is minimal. In other words, an existing perturbation should be extended to a new point if and only if the length of the perturbation plus the elliptic length of the optimal extension to the point is less than the corresponding sum for all other perturbations. For example, referring to Figure 9, if a new city is found to reside in quartic Voronoi cell "jk", it must be connected to cities j and k, while at the same time the segment joining j to k must be deleted. In this way, perturbation π_3 is extended to capture the new city.

There are cases when a specific perturbation requires reordering to maintain optimality: namely, when some city is nearer to the city to be inserted than either of the endpoints of the best segment found when minimizing the elliptic distance across all segments in the perturbation. However, the algorithm required to implement this operation is of linear time complexity, and does not detract from the performance of the general extension philosophy.

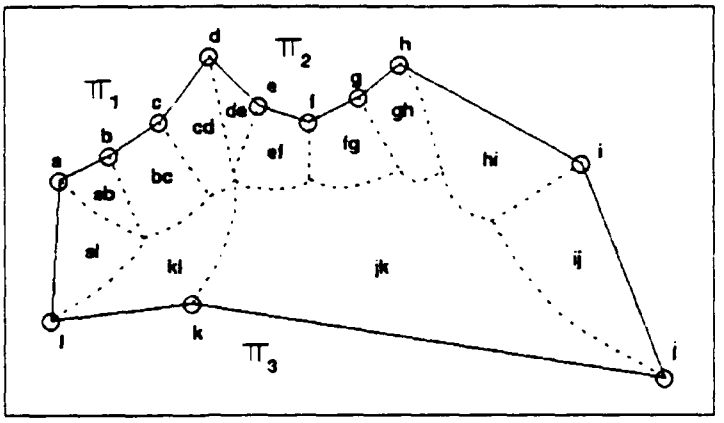


Figure 9 (estimated quartics and hyperbolas). Extending an existing perturbation is straightforward. When a new city enters the system, it is connected to the perturbation for which the elliptic distance to a segment is minimal, across all perturbations.

Quartic Shunt Space.

A shunt to the left is a bridging operation which connects a new city to the hull segment to the left of its extended perturbation, whereas a shunt to the right connects the city to the hull segment succeeding it. The left shunt is formed by connecting the city to its nearest neighbor two perturbations to the left, and then following the respective perturbation subpaths down to the hull vertices of the perturbation at the left. Any cities which become detached by this process must be reconnected to the perturbation space. The quartic shunt operator is a powerful tool, useful for merging two perturbations of the same parity into one with opposite parity, lying between the other two.

An example of a quartic shunt to the left is shown at Figure 10 (the data is a handcrafted approximation of a graphic depicted on p. 224, reference [P5]. At the left side of the figure, city c12 has just been introduced. Hyperbolic extension space calls for a perturbation of hull segment c5-c4, indicated by the dotted lines. However, quartic shunt space calls for a shunt to be formed between c12 and c16, which is the nearest neighbor two hull segments to the left of c12's extended perturbation. The endpoints of the shunt are followed down from c12 and c16 respectively to c5 and c6. City c13, which is left dangling by the shunt operation, is optimally reattached to the perturbation space by connecting it to segment c6-c7.

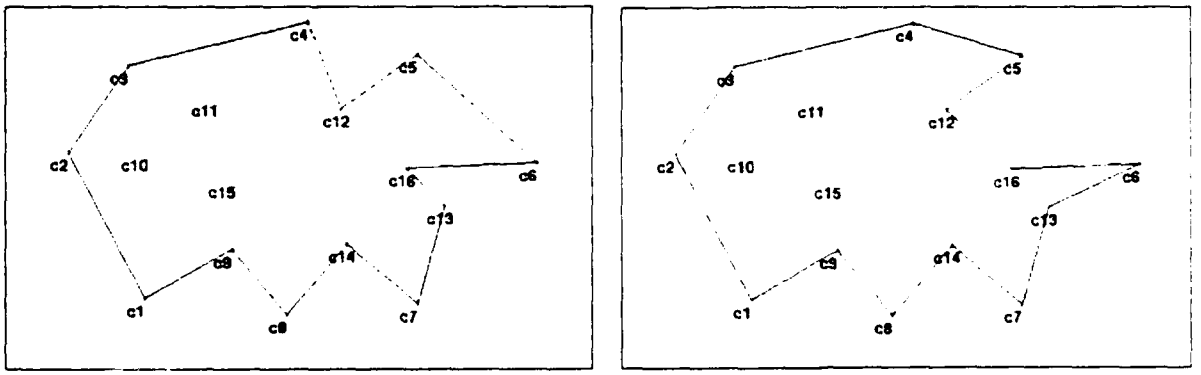


Figure 10. A quartic snunt to the left, using the Preparata and Shamos dataset.

Quartic Interchange Space.

Quartic interchange space dictates when a new city's perturbation, whether it be an extension or shunt, has encroached sufficiently far into the hull to encompass cities which earlier were optimally installed in some other perturbation. Every time a new city is processed, it must be hypothesized that the extending perturbation may now have encroached deep enough into the hull to begin influencing perturbations on the other side. While at some time in the past it may have been legitimate to have constructed a cross hull perturbation to maintain optimality, it may now be time to partially or completely "undo" the perturbation by swapping some of its cities to the near side of the hull.

Quartic interchange is iterative. The currently extended perturbation is compared to all other perturbations in the space. If an exchange of cities is warranted, it is permitted to occur, and the revised perturbation space is subjected to interchange once again. This action is repeated until no improvement is obtained.

The General Voronoi Diagram for the Euclidean Traveling Salesman Problem.

Earlier, we proved that the Voronoi diagram of the convex hull has quartic edges, but possesses hyperbolic edges between adjoining hull segments. For the general case, this concept may be extended. The generalized Voronoi diagram partitions the plane into three types of cells: hyperbolic extension cells; quartic shunt cells; and quartic interchange cells. Before the k^{th} city is introduced, one computes the Voronoi diagram for the set of previously introduced k-1 cities. As in the convex hull case, computation must once again resort to an elliptic distance comparison, except now three different types of tour topologies must be hypothesized, rather than the single hypothesis entertained by introducing a single city into the hull. The space is once again quartic, because to obtain the boundaries of equal perturbation to the k^{th} city, two variable distances are added, and the sum of a set of fixed distances (the length of a specific hypothesized subpath) is subtracted. Rather than reasoning with perturbed segments on the hull, one must reason with tangible segments which are part of an existing tour, hypothesized segments which form shunts between perturbations, and hypothesized segments which form interchange links with other perturbations. To formalize, the quartic Voronoi diagram indicated by the optimal tour for k-1 cities, denoted $\mathrm{VorQ}(\tau_{k-1})$, is a function ψ of five arguments:

$$\mathrm{VorQ}(\tau_{k-1}) = \psi[\tau_{k-1}, \mathrm{Ext_H}(\tau_{k-1}), \mathrm{Shunt_L}(\tau_{k-1}), \mathrm{Shunt_R}(\tau_{k-1}), \mathrm{Inter}(\tau_{k-1})]$$

where:

τ_{k-1}	=	the optimal tour for k-1 cities;
$\mathrm{Ext_H}(\tau_{k-1})$	=	the hyperbolic extension space induced by τ_{k-1}.
$\mathrm{Shunt_L}(\tau_{k-1})$	=	the left quartic shunt space induced by τ_{k-1}.
$\mathrm{Shunt_R}(\tau_{k-1})$	=	the right quartic shunt space induced by τ_{k-1}.
$\mathrm{Inter}(\tau_{k-1})$	=	the quartic interchange space induced by τ_{k-1}

The Principle of Quartic Incremental Optimality.

It is clear we are proceeding with a strategy akin to dynamic programming. Namely, when adding a new city to the interior, we attach the city to an existing optimal tour in a way indicated by the quartic Voronoi diagram computed just prior to the city's introduction. Formally, the shortest tour τ_k for k cities is a function Δ of two arguments:

$$\tau_k = \Delta\left((c_{k_x}, c_{k_y}), \mathrm{VorQ}(\tau_{k-1}) \right); \qquad k = [H] + 1, \ldots, n;$$

where:

τ_k	=	the optimal tour containing k cities;
(c_{k_x}, c_{k_y})	=	the coordinates of the k^{th} city to be introduced;
τ_{k-1}	=	the optimal tour containing k-1 cities;
$\mathrm{VorQ}(\tau_{k-1})$	=	the quartic Voronoi diagram prescribed by k-1 cities;
$[H]$	=	the order of the convex hull H;
n	=	the total number of cities to be processed.

Unique vs. Multiple Numbers of Distinct Optimal Tours as a Function of the Quartic Space.

Proper containment within a quartic Voronoi cell guarantees a unique tour. In nondegenerate cases, there can be no more than three unique tours because quartic Voronoi edges converge in groups of three just as linear Voronoi edges do. If a newly introduced city is situated at a Voronoi junction (a point where three quartics come together), three optimal tours exist; whereas if the city lies on a quartic but not on a junction, two optimal tours exist. In the degenerate case when cities are equispaced in the plane, there may be more tours than in the nondegenerate case. For example, a hull consisting of a regular polygon containing k sides produces k optimal tours if the introduced city lies at the center of the polygon, because the center is at the intersection of k Voronoi edges (degenerate hyperbolas). As a point of interest, it can be seen that if one allows the number of vertices of a regular polygon to approach infinity, so does the number of optimal tours connecting the hull to the center. However, in this case, the limiting form of the polygon is a circle, and a paradox arises, because the Euclidean distance between adjacent hull vertices approaches zero as the number of distinct optimal tours rises to infinity.

Future Work on A Proof of the Admissibility of Δ.

Mathematical induction will be used in an attempt to show that Δ is admissable. For the case of inserting a single city into the hull (i.e., k = 1), the shortest tour is trivially depicted by VorQ (τ_H), so the initial step of the inductive proof is satisfied. What remains to be shown is that the sequencing of the operations of hyperbolic extension, quartic shunting, and quartic interchange preserves optimality.

Summary of the Generalization of the Voronoi Diagram to ETSP.

During the first decade of the century, Voronoi's intention was to develop a mathematical structure which could be used to rapidly associate a query point with the nearest point contained within a known two-dimensional constellation of points [V2]. In decades of subsequent work, the dimensionality constraint has been relaxed, as well as the specification that both the query object and known objects be points [P5, E2]. This paper has focused on an exact solution to the Euclidean traveling salesman problem, and consequently has introduced a new distance metric, known as the elliptic distance, used to compute the distance of a floating query point from two fixed points. The limiting form of this metric, when intersected with that from another perturbed segment, induces a quartic structure on the Voronoi diagram. This non-linear search space permits a feasible testbed arena for the problem which the traditional Voronoi diagram cannot provide.

Dillencourt's nondegenerate counterexample [D3] to Shamos' conjecture that the shortest tour must traverse the Voronoi dual is shown at the left of Figure 11. At the right is the quartic Voronoi

diagram, which depicts connectivity of the three interior cities to the convex hull. All the curves indicated are hyperbolas, except the short one plotted between segment t1-t7 and segment t2-t3, which is a quartic locus. The shortest tour is t1-t2-t4-t3-t5-t7-t6-t1, but the linear Voronoi dual does not permit t3 and t4 to be connected. Note that in this instance, if one merely attaches each of the three interior cities to the hull segment indicated by the quartic Voronoi cell in which it resides, using the hyperbolic extension operator, the shortest tour is obtained. Of course, the actual computer run invokes the processes of quartic shunting and quartic interchange, but in this case the fourth-order operators fail to improve the tour produced by hyperbolic extension space.

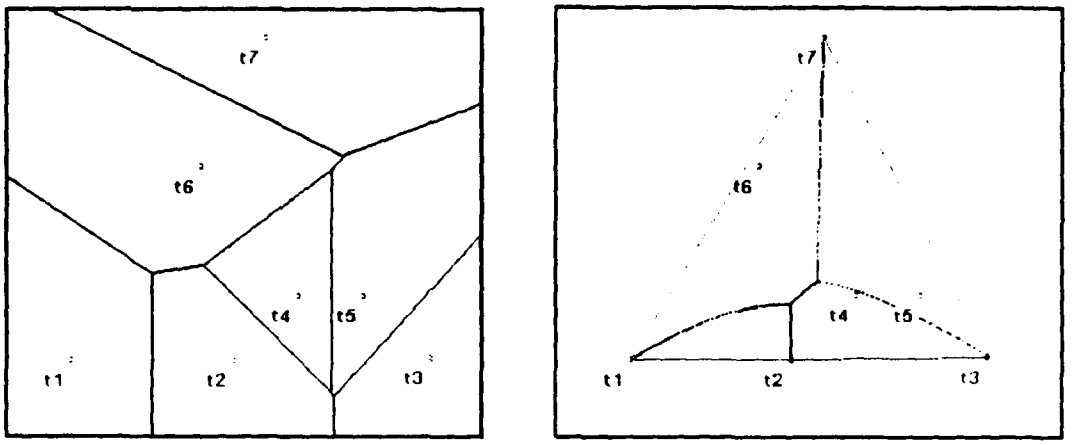


Figure 11 (computed edges). The Voronoi diagram for the Dillencourt data (left), and its one-city-in-a-hull quartic Voronoi diagram (right). This data is the first known nondegenerate counterexample to Shamos' conjecture that the shortest Euclidean tour must traverse adjacent Voronoi cells. In the optimal tour, t4 is connected to t3. It is apparent that t4 and t3 can be connected in the qurtic diagram, but not in the linear one.

The traditional Voronoi diagram is a proximity map, where at a glance it can be seen which object in a search space is nearest to a query point. The quartic Voronoi diagram is a connectivity map, which displays shortest tour connectivity information for the k^{th} city, as a function of a constellation of k-1 fixed cities. It has been shown that the process of intersecting an infinite set of confocal ellipses symmetric about an existing ETSP link with those about another link produces a quartic curve. The quartic curve, which in practice frequently reduces to a hyperbola because of the tendency of extension space to dominate during nested hull traversal, serves the same role as the perpendicular bisector does in the traditional Voronoi diagram. Thus, instead of being piecemeal linear, the extended Voronoi structure for ETSP is piecemeal quartic. The final discrepancy between the traditional Voronoi diagram and the quartic diagram deals with the issue of boundedness. Traditional Voronoi diagrams are unbounded; i.e., cells on the perimeter of the diagram are permitted to extend to infinity. However, for the Euclidean traveling salesman problem, the quartic Voronoi diagram is bounded by the convex hull of cities, so that no cell is unbounded. Nevertheless, this is not to say that the boundedness constraint cannot be loosened to incrementally add new cities exterior to the hull, which is the philosophy behind the inside-out nested hull approach (an example of this technique will be elaborated upon in an example appearing below, in which the innermost hull is used as a baseline optimal tour from which to add cities incrementally to the exterior). Table 1 summarizes the three distinctions between the traditional, linear Voronoi diagram, and its quartic counterpart designed for exact solution of the Euclidean traveling salesman problem.

Traditional Usage ⟶	Extension to ETSP
Proximity Map	Shortest-tour Connectivity Map
Cell Boundaries are Line Segments	Cell Boundaries are Piecemeal Hyperbolic and Quartic
Perimeter of Diagram is Unbounded	Perimeter of Diagram is Bounded by Convex Hull

Table 1. Extension of the Voronoi Diagram to the Euclidean Traveling Salesman Problem.

The Computational Complexity of the Hull Perturbation Approach.

a. The Time Complexity of Blind Search.

The convex hull of a set of cities serves as a control structure from which to initiate perturbations. In this section we naively derive an expression for the time complexity of the approach, if the tack is taken to blindly generate perturbations from hull segments in an arbitrary fashion. The derivation hinges upon making a substitution at an opportune moment when the binomial coefficients are manifested. It is hoped that future research will lend insight into techniques to improve the naive bound. The number of perturbations in the optimal tour cannot exceed the size of the hull, because a perturbation is defined to be an excursion into the interior from a hull segment, the number of which is equal to the number of hull vertices. Let H be the set of cities on the convex hull, and I be the set of cities lying on the interior of the hull. Let the rank of H be h, and the rank of I be i. If n is the total number of cities, then n = h + i. From each hull segment, the set of interior cities may be visited zero at a time, one at a time, two at a time, ... or i at a time. Thus the total number of computations required to find the shortest tour is:

$$h * {}_iC_1 + h * {}_iC_2 + \ldots + h * {}_iC_i =$$

$$h * ({}_iC_1 + {}_iC_2 + \ldots + {}_iC_i) =$$

$$h * (2^i) =$$

$$h * (2^{n-h}) =$$

$$\frac{h * 2^n}{2^h}$$

Some observations may be made about this naive bound. Note that a large hull is desirable, because the effect of the denominator is to diminish the 2^n term. Although the complexity is exponential, it is an order-of-magnitude improvement over a brute force approach, which is of factorial complexity.

b. The Time Complexity of Quartically-Controlled Search, using Nested Hull Traversal.

As a city is processed during nested hull traversal, there are three general phases of computing which must be performed in sequence. The first is a linear time operation to extend the current topology by minimizing the elliptic distance from all perturbations to the new city, which includes reordering a perturbation if necessary. The second phase involves two linear time operations, to construct the left and right quartic shunt topologies, after which they are compared with the extension topology to render the one with shortest tour length. Finally, the quartic interchange space is computed, which is a quadratic operation, because the left and right tour edges produced by the insertion of the new city act as windows to possibly absorb whole groups of cities from perturbations on the far side of the hull. Therefore, to process each new city, the worst case time complexity is quadratic. The sum of a set of quadratic expressions in k, where k ranges from 0 to n, is a closed form expression equal to $n * (n + 1) * (2n + 1) / 6$.

In summary, a computer implementation of the principle of quartic incremental optimality requires $O[n * \log n]$ preprocessing time to compute the nested hull decomposition, $O[n]$ storage for intercity distances and optimal partial tours, and $O[n^3]$ time complexity to maintain incremental optimality.

An Example: The Forty-eight Capital Problem.

The shortest tour connecting the forty-eight capitals of the contiguous United States remained an intriguing open problem until Shen Lin obtained an optimal solution in 1985 [A1, A2]. Each coordinate of the database represents the location of a Bell telephone office in the capital of a state. The principles of nested hull traversal and quartic incremental optimality were leveraged against this database. The nested hull structure for this data is exhibited at Figure 12. The implementation to derive the convex hull is based on an iterative enhancement made by the author to an algorithm developed in the seventies by W.F. Eddy [E1].

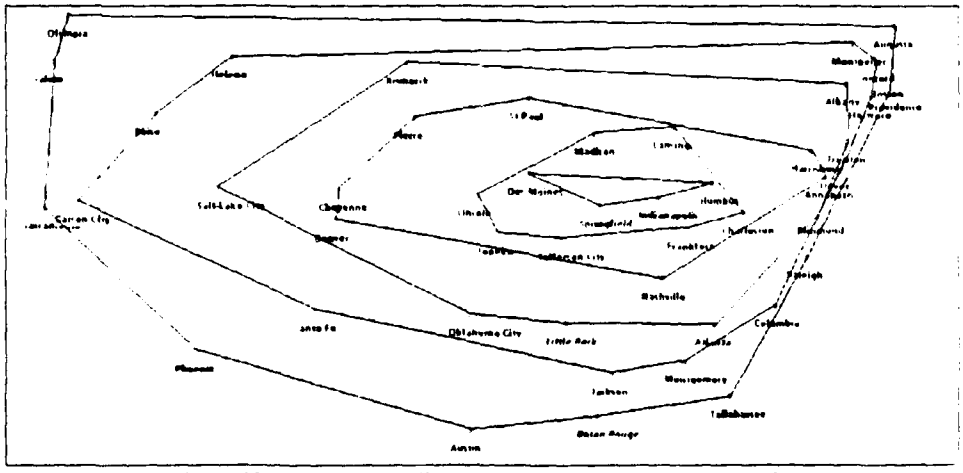


Figure 12. The Nested Hull Structure of the Forty-eight Capitals.

Working from the Outer Hull to the Inner.

First, the problem was attacked by starting with a baseline tour consisting of the outer convex hull, and probing inwards. Each nested hull is traversed in counterclockwise order to insert new cities, before the next inner hull is processed. A temporal history of incremental optimality is shown in Figure 13. The inserted city and the quartic function triggered are listed below each graphic. The interesting cases are those which are not mere hyperbolic extensions, but those which also involve quartic shunts and exchanges. The most dramatic quartic shunt occurs in row five, column seven, when the introduction of Springfield, Illinois produces a shunt to the right. Springfield is originally processed by the hyperbolic extension operator, which compels attachment to the perturbation which contains Lincoln, Nebraska. However, quartic shunt space produces a shorter tour by conjoining Springfield with a perturbation to the right containing Frankfort, Kentucky. Another interesting iteration occurs in row four, column six, when the introduction of Cheyenne, Wyoming into hyperbolic extension space causes the transposition of Bismarck, North Dakota with Pierre, South Dakota. Subsequently, quartic interchange causes Salt Lake City , Utah to be drawn out of its perturbation with Carson City, Nevada into the perturbation containing Cheyenne. The algorithm correctly terminates with Lin's optimal tour, shown in row six, column two.

Working from the Inner Hull to the Outer.

Next, the same data was processed by starting with the innermost nested hull and probing outward. Because the quartic Voronoi edges extend through the hull vertices both on the inside and the outside, the nested hull technique is theoretically valid in either direction. Figure 14 illustrates the optimal subtours produced by the algorithm when starting with the innermost hull and probing successively outward through the outer hulls. In this case, the innermost hull contains only four cities, so the original number of perturbations is four. There is an interesting tradeoff on time complexity when working with fewer perturbations. Again, note that the optimal tour is produced.

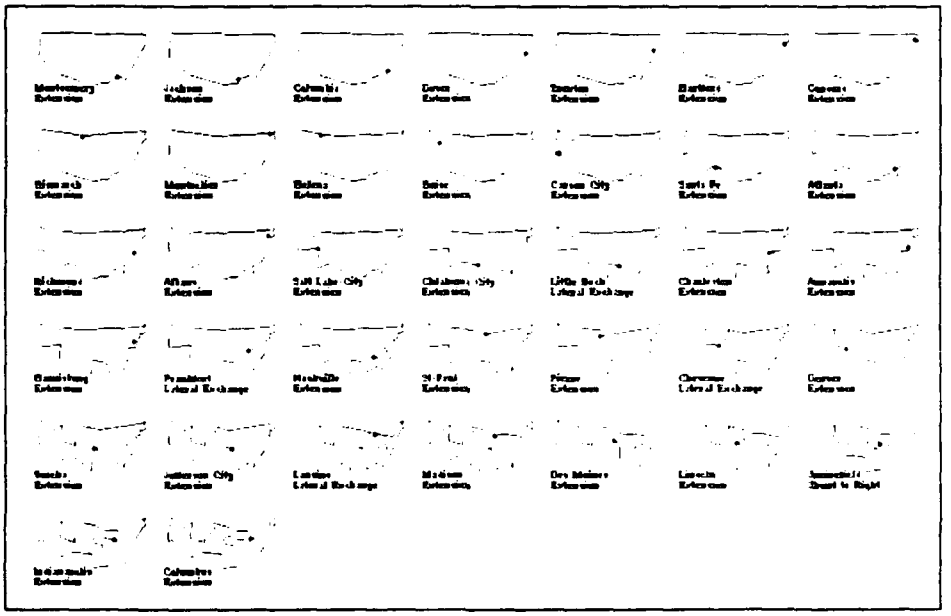


Figure 13. Working inwards from the outer hull, employing quartic nested hull traversal.

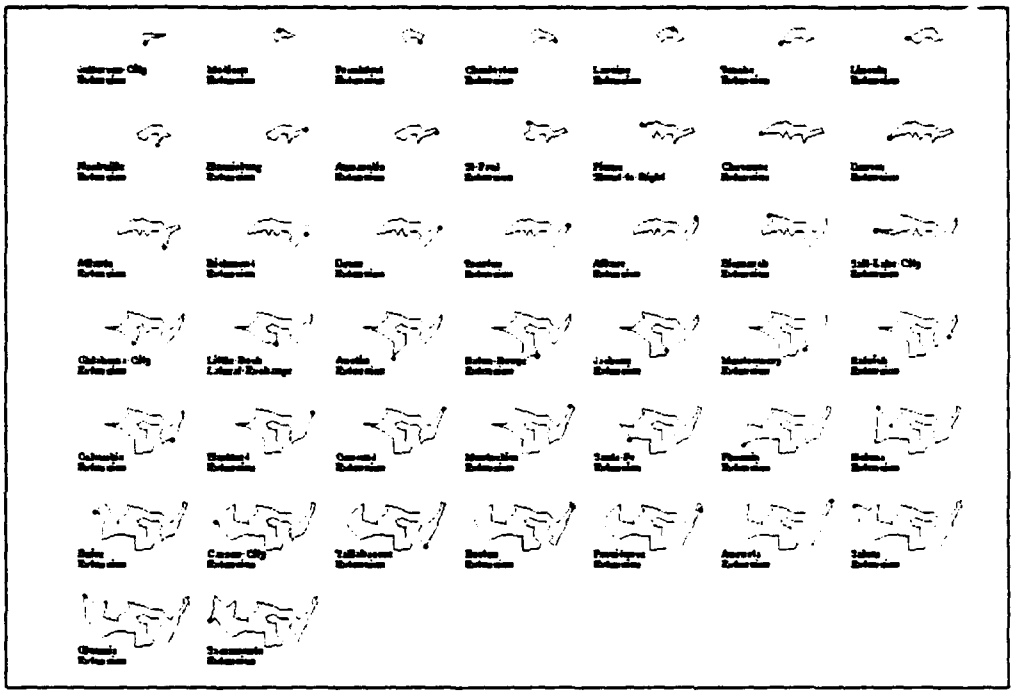


Figure 14. Working outwards from the inner hull, employing quartic nested hull traversal.

Summary.

The chief result of the research to date is a proof that the underlying search space (the Voronoi diagram) for the Euclidean traveling salesman problem is non-linear; specifically, the space is quartic when reasoning across subtours, and hyperbolic when reasoning within subtours. These facts become apparent when one realizes that reasoning about shortest tours is a process which inherently involves the intersection of a pair of ellipses, the foci of which are defined by pairs of cities. Ellipse intersection is an operation which in the worst case produces a fourth-order equation (quartic). In the special case in which two ellipses share a focus, the locus is a semi-hyperbola. The discovery of the non-linear search space has prompted the author to devise an algorithm which utilizes three operators to constrain search: hyperbolic extension; quartic shunting; and quartic interchange. To limit the generation of greedy perturbations, cities are gradually inserted in an incremental fashion, according to their position within the nested hull structure of the city database. The new knowledge about the non-linear search space has resulted in an O[n^3] solution to optimality of a forty-eight city problem. The solution is obtained both by beginning with the outer convex hull and probing inward, or by starting with the innermost hull and probing outwards.

Future Directions of the Research.

It is desirable to pursue a rigorous proof of the theory of quartic incremental optimality; the proof will proceed by induction. Also, to facilitate further empirical analysis, the theory as it currently stands will continue to be developed and leveraged against several large databases of cities for which the optimal tour is known, in an attempt to find examples which counterindicate the algorithm. Short-term plans include runs against a 127-city database [R1], and a 532-city benchmark for which the optimal solution has been developed [P1]. Subsequently, the runtime for the experimental data will be

plotted as a function of the number of cities, to determine if the algorithmic ceiling function is of cubic order as predicted by the analysis.

Acknowledgments

The research has greatly benefited from discussions with Gerald Andersen, Richard Antony, Jacob Barhen, Chris Bogart, Paul Broome, Jagdish Chandra, Douglas Chubb, Michael Dillencourt, Martin Groetschel, Andrew Harrell, Robert Hein, William Jackson, Shen Lin, Sanjoy Mitter, Al Pollard, John Pfaltz, Carl Russell, Paul Tseng, Franz-Erich Volter, and David Willshaw. I am once again indebted to Richard Antony for his customarily excellent job of technical editing and commentary. A special note of gratitude to David S. Johnson of AT&T Bell Laboratories for incisive, constructive criticism of the principle of incremental optimality, and for providing the coordinate data for the forty-eight capital problem. I am grateful to Geoffrey Fox and Robert Bixby for an invitation to a very productive workshop entitled "Computational Aspects of the Traveling Salesman Problem", held at Rice University, 22-24 April 1990. Herbert Edelsbrunner was a source of valuable insights into issues arising when computing the convex hull. Thanks to Gerhard Reinelt for discussions on the Voronoi diagram, and for establishing a library of traveling salesman problem city databases, which will allow researchers worldwide to share a common testbed of benchmark problems for which the optimal solutions are either known or bounded.

APPENDIX

The appendix consists of a series of five computer plots which graphically portray the quartic (fourth-order polynomial) loci which exist naturally when hypothesizing a solution to the two-city-in-a-hull Euclidean traveling salesman problem described in the main body of text. These loci were discovered empirically by the author during the summer of 1989; it was only later after several months of research that a proof was obtained to demonstrate algebraically that the loci are actually comprised of quartic and hyperbolic curves. It may be of interest to some readers to know how the plots were obtained. An algorithm was designed to capture the knowledge about the possible ways (permutations) to connect city d8 and one other arbitrary city to the convex hull (d1-d2-d3-d4-d5-d6). In the eight city example, there are six possible topologies to compare between any two hull segments: the two ways to attach d8 and the arbitrary city to each of the two hull segments (which yields a subtotal of four), and the two ways to attach one city to one segment and the second to the other segment. In general, this means that there are fifteen quartic loci (the combination of six topologies taken two at a time) among which to arbitrate when hypothesizing a shortest tour. The algorithm was encoded in Lisp and run on an artificial intelligence computer workstation. A set of experiments were conducted as follows: the computer mouse was moved about its pad on the desk, which caused the cursor to move about the monitor screen displaying the constellation of cities. If the length of a specific arrangement of cities was within one unit of that of another arrangement, a black dot was plotted to the screen at the position of the cursor. This action provided positive feedback to the author, who dynamically readjusted the position of the mouse to obtain "more black dots" in a continuous fashion, until an entire quartic curve manifested itself. When a point in time was reached in which it became obvious that no more loci were forthcoming, the session was terminated, and another pair of segments was selected for experimentation. In the set of graphics selected for exhibit here, one of the pair is always segment d6-d5. Also note that what in the text was referred to as city "d7" is here called city "d8". It should also be pointed out that in exhibits A-3, A-4, and A-5, the quartic plots are superimposed over the solution to the one-city-in-a-hull problem discussed in the text.

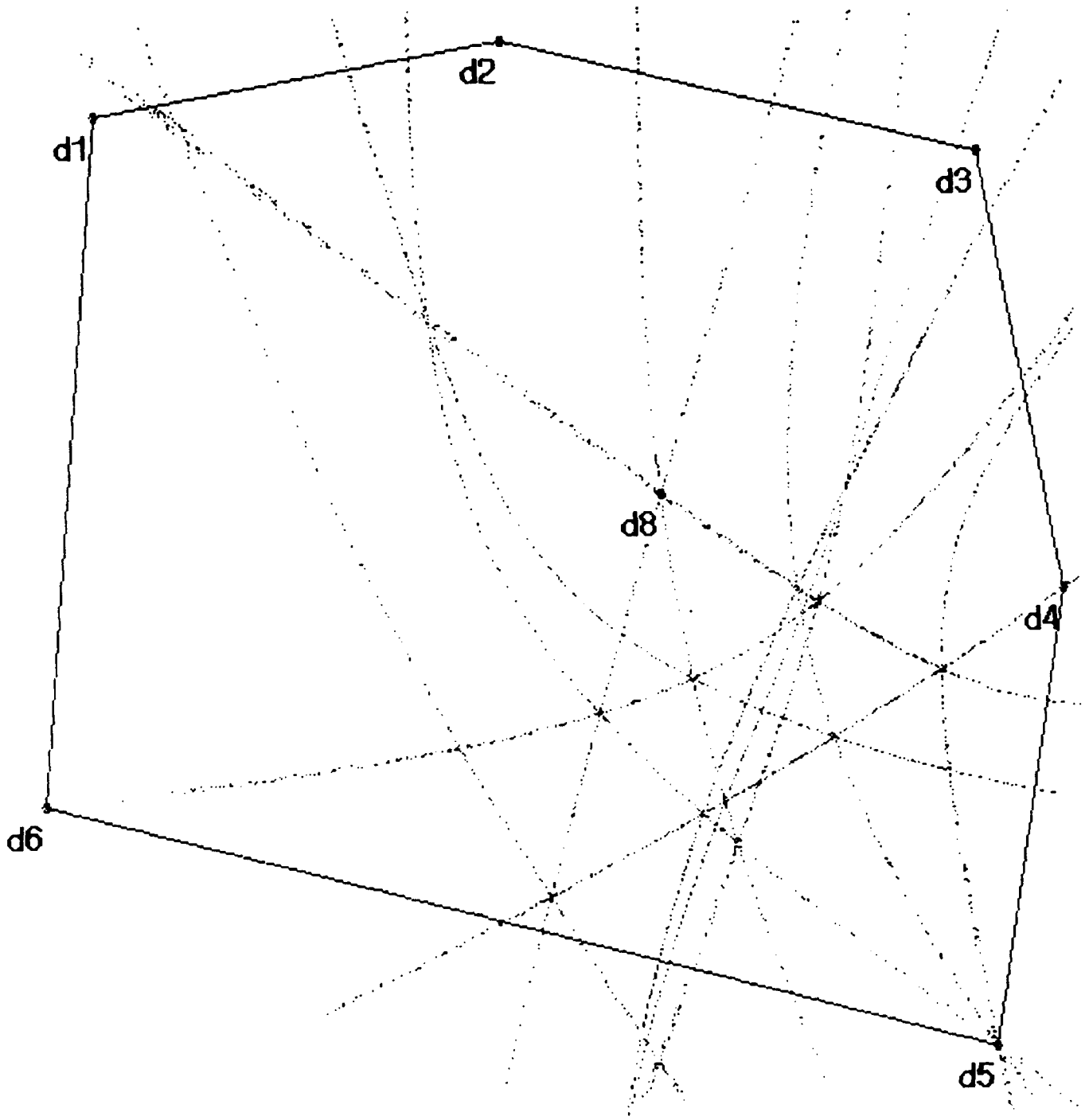


Exhibit A-1. The quartic interplay between segment d6-d5 and segment d5 d4, when attaching d8 and an arbitrary city to the convex hull to produce the shortest Euclidean tour.

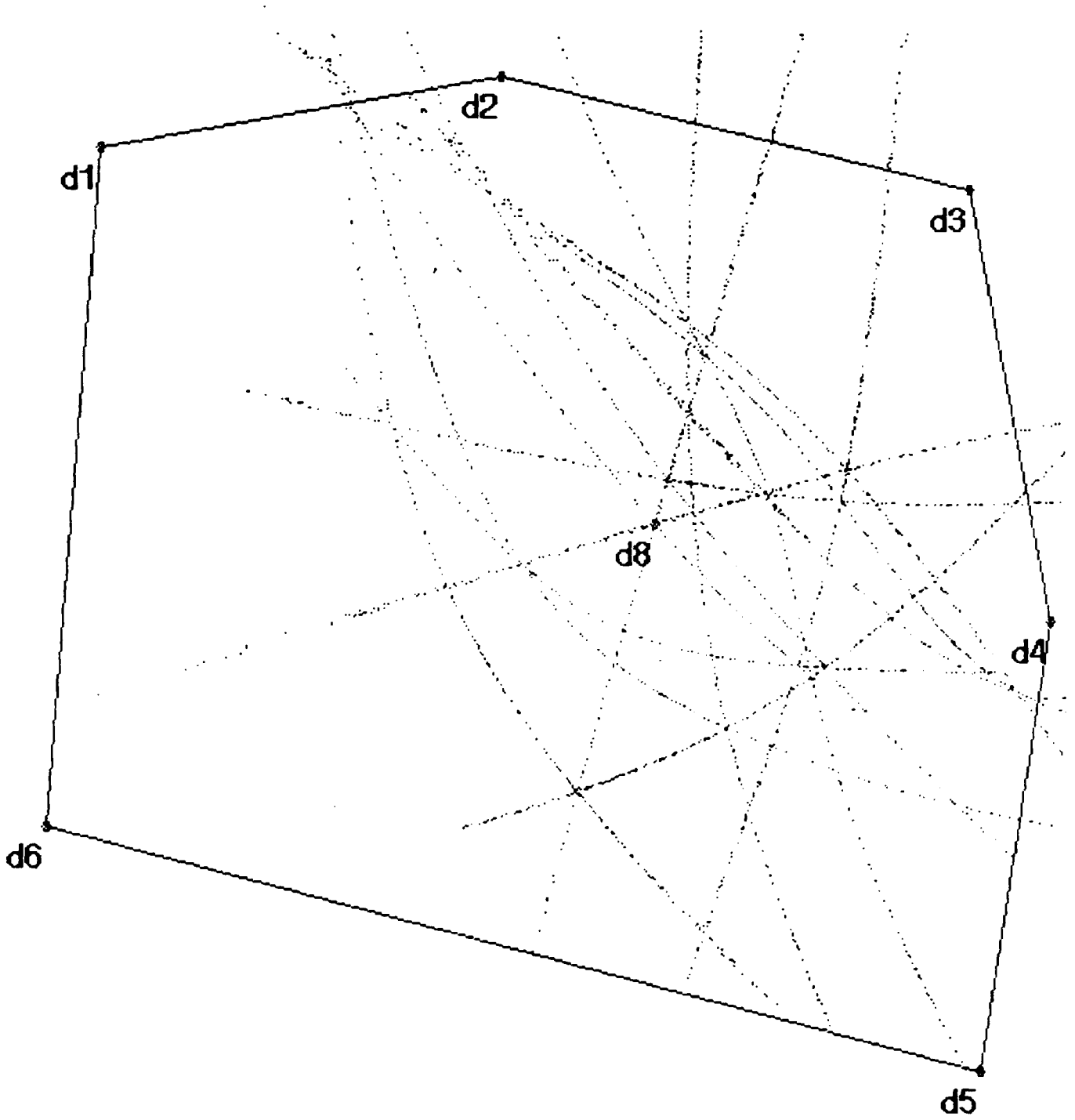


Exhibit A-2. The quartic interplay between segment d6-d5 and segment d4-d3, when attaching d8 and an arbitrary city to the convex hull to produce the shortest Euclidean tour.

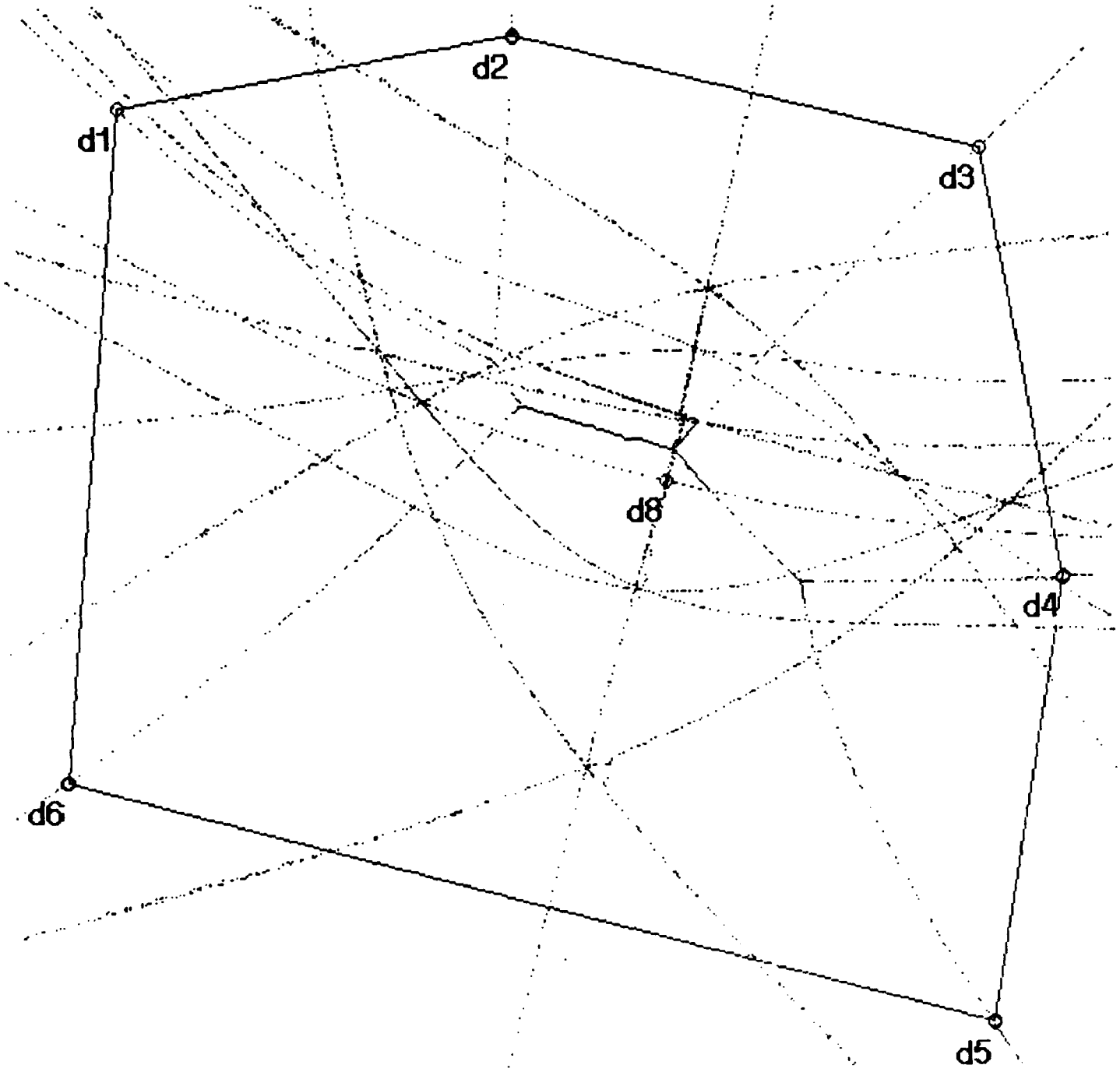


Exhibit A-3. The quartic interplay between segment d6-d5 and segment d3-d2, when attaching d8 and an arbitrary city to the convex hull to produce the shortest Euclidean tour.

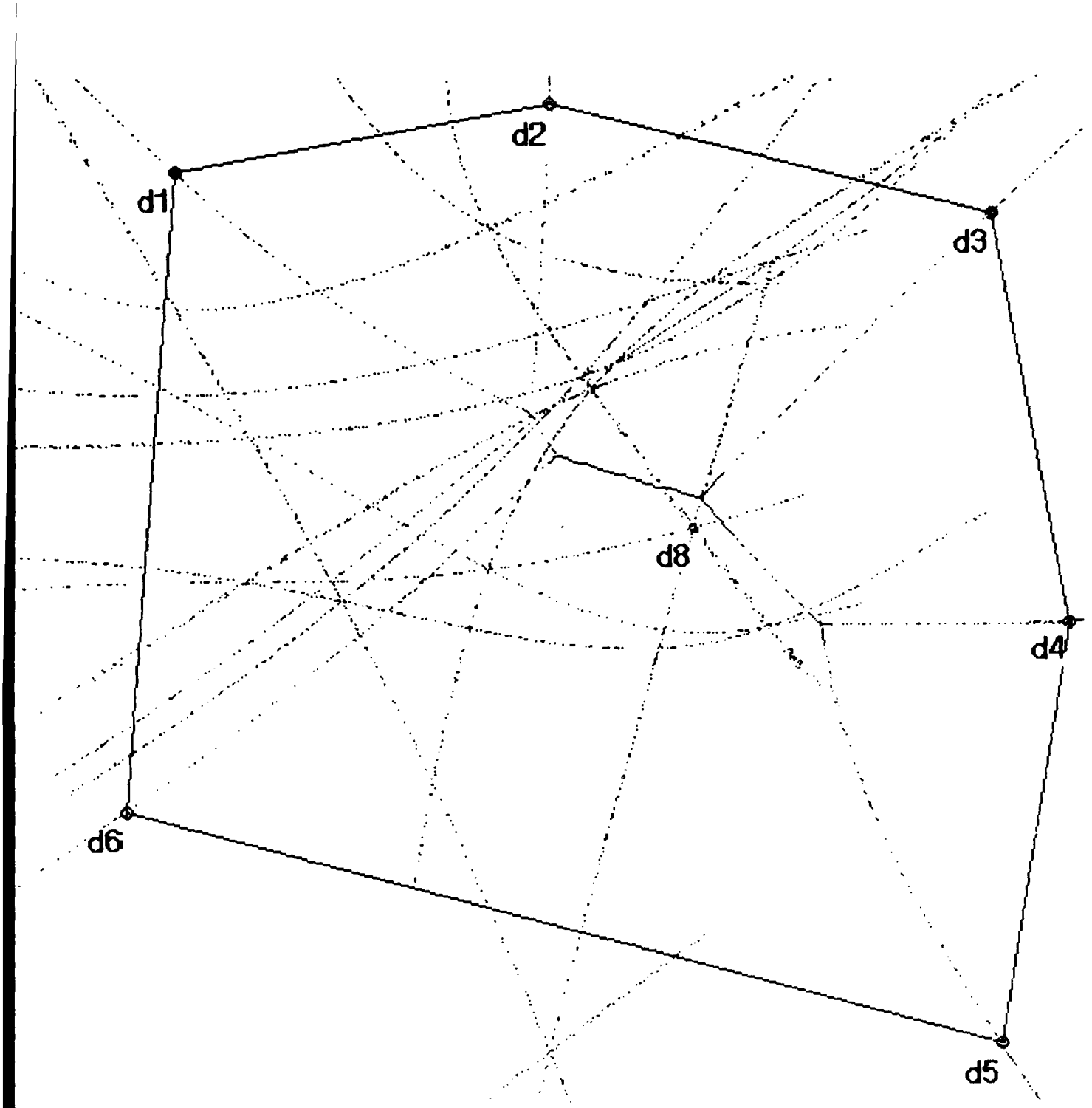


Exhibit A-4. The quartic interplay between segment d6-d5 and segment d2-d1, when attaching d8 and an arbitrary city to the convex hull to produce the shortest Euclidean tour.

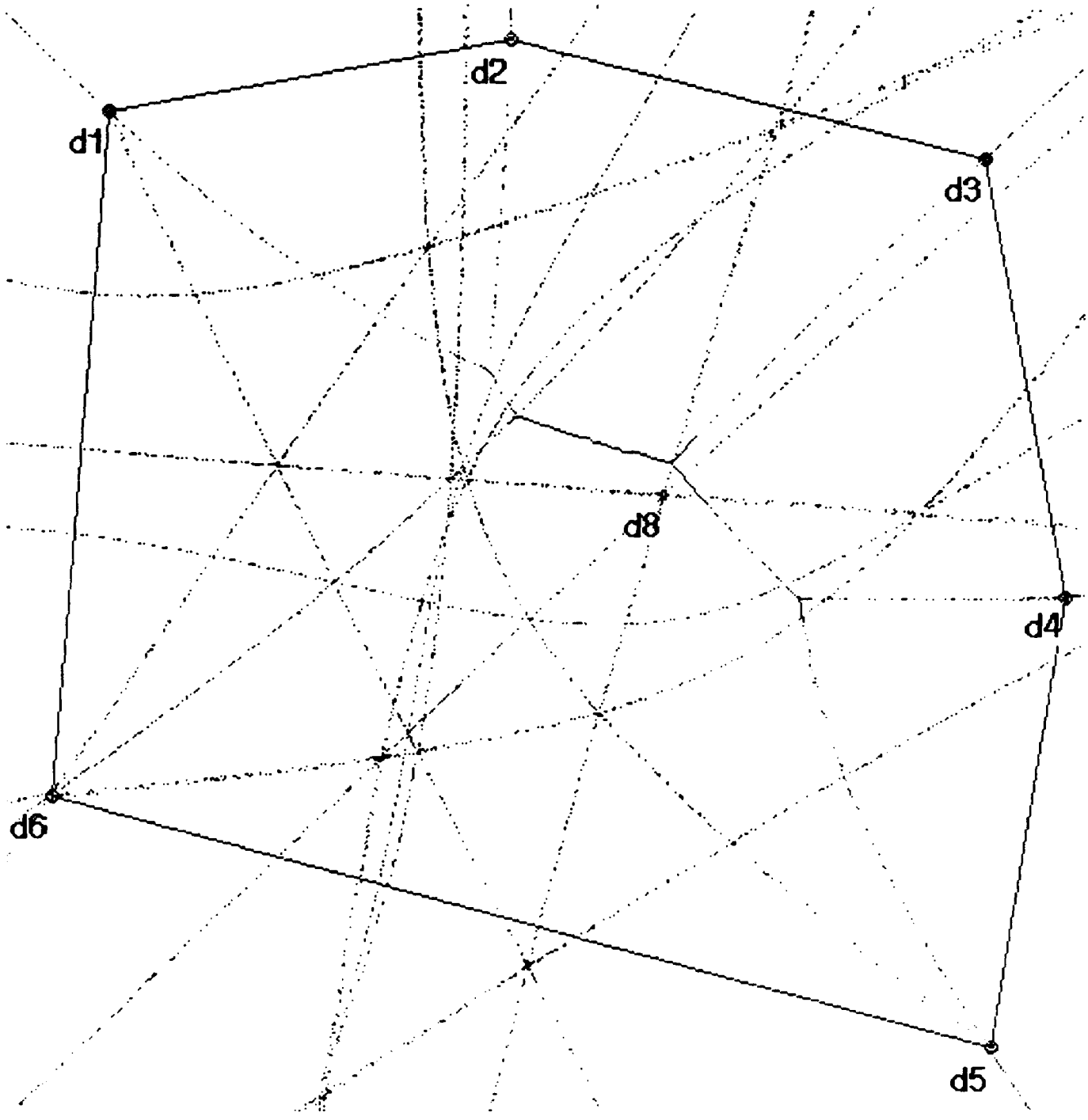


Exhibit A-5. The quartic interplay between segment d6-d5 and segment d1-d6, when attaching d8 and an arbitrary city to the convex hull to produce the shortest Euclidean tour.

Bibliography

[A1] Anonymous, "Just Keep Going", Los Angeles Times, Section II, p. 4, col. 1, 17 June 1985.

[A2] Anonymous, "Here's one for the Road, and then Some", Discover, July 1985.

[B1] Barachet, L.L. "Graphic Solution of the Traveling Salesman Problem", Operations Research 5, 1957, pp. 841-845.

[B2] Bellmore, M., and G.L. Nemhauser, "The Traveling Salesman Problem: A Survey", Operations Research 16, 1968, pp. 538-558.

[B3] Bern, M.W., and R.L. Graham, "The Shortest Network Problem", Scientific American, January 1989, pp. 84-89.

[B4] Brady, R.M., "Optimization strategies gleaned from biological evolution", Nature 317, 1985.

[C1] Christofides, N., "Worst-case Analysis of a New Heuristic for the Travelling Salesman Problem", Report 388, Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh PA, 1976.

[C2] Chazelle, B., "On the Convex Layers of a Convex Set", IEEE *Trans. Inform. Theory* IT-31, 1985.

[C3] Crowder, H., and M.W. Padberg, "Solving Large-Scale Travelling Salesman Problems to Optimality", Management Science 26, 1980, pp. 495-509.

[D1] Dantzig, G.B., D.R. Fulkerson, and S.M. Johnson, "Solution of a Large-scale Traveling-Salesman Problem", Operations Research 2, 1954, pp. 393-410.

[D2] Dantzig, G.B., D.R. Fulkerson, and S.M. Johnson, "On a Linear Programming, Combinatorial Approach to the Traveling-Salesman Problem", Operations Research 7, 1959, pp. 58-66.

[D3] Dillencourt, M.B., "Graph-Theoretical Properties of Algorithms Involving Delaunay Triangulations", University of Maryland, Center for Automation Research, Technical Report TR-369, September 1988.

[D4] Durbin, R., and D. Willshaw, "An analogue approach to the travelling salesman problem using an elastic net method", Nature 326, 16 April 1987.

[E1] Eddy, W.F., "A new convex hull algorithm for planar sets", ACM Transactions on Mathematical Software 3, 1977.

[E2] Edelsbrunner, H., Algorithms in Combinatorial Geometry, Springer-Verlag, Berlin Germany, 1987.

[E3] Edelsbrunner, H., G. Rote, and E. Welzl, "Testing the Necklace Condition for Shortest Tours and Optimal Factors in the Plane", Proceedings of the 14th International Colloquium on Automata, Languages, and Programming, T. Ottmann, ed., Karlsruhe, Federal Republic of Germany, July 1987.

[E4] Eisenhart, L.P., Coordinate Geometry, Dover Publications Inc., New York NY, 1939.

[F1] Fox, G., Private communication, Semi-annual Concurrent Processing technology review of the Joint Tactical Fusion Office, Jet Propulsion Laboratory, Pasadena CA, March 1990.

[G1] Garey, M.R., R.L. Graham, and D.S. Johnson, "Some NP-complete Geometric Problems", Eighth Annual Symp. on Theory of Comput., May 1976, pp. 10-22.

[G2] Garey, M.R., and D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman and Company, New York NY, 1979.

[G3] Golden, B.L., and W.R. Stewart, "Empirical Analysis of Heuristics", in The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization, eds. E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, and D.B. Shmoys, John Wiley & Sons, Chichester UK, 1985.

[H1] Harel, D., The Science of Computing: Exploring the Nature and Power of Algorithms, Addison-Wesley, Reading MA, 1987, 1989.

[H2] Held, M., and R.M. Karp, "The Traveling Salesman Problem and Minimum Spanning Trees: Part II", Mathematical Programming 1, 1971, pp. 6-25.

[J1] Johnson, D.S., and C.H. Papadimitriou, "Computational Complexity", in The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization, eds. E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, and D.B. Shmoys, John Wiley & Sons, Chichester UK, 1985.

[J2] Johnson, D.S., and C.H. Papadimitriou, "Performance Guarantees for Heuristics", in The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization, eds. E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, and D.B. Shmoys, John Wiley & Sons, Chichester UK, 1985.

[J3] Johnson, D.S., "More approaches to the travelling salesman guide", Nature 330, December 1987, p. 525.

[J4] Johnson, D. S., Set of viewgraphs entitled "The Traveling Salesman Problem", AT&T Bell Laboratories, Murray Hill NJ, September 1989.

[J5] Johnson, D. S., Private communication, and set of viewgraphs entitled "How to Beat Lin-Kernighan", AT&T Bell Laboratories, Murray Hill NJ, April 1990.

[K1] Kantabutra, V., "Traveling salesman cycles are not always subgraphs of Voronoi duals", Information Processing Letters 16(1):11-12, January 1983.

[K2] Karp, R.M., "Reducibility among Combinatorial Problems", in Complexity of Computer Computations, ed. R.E. Miller and J.W. Thatcher, Plenum Press, New York NY, 1972.

[K3] Kirkpatrick, D.G., and R. Seidel, "The Ultimate Planar Convex Hull Algorithm?", SIAM Journal Computing 15, 1986.

[K4] Krolak, P., and W. Felts, "A Man-Machine Approach Toward Solving the Traveling Salesman Problem", Communications of the ACM 14, May 1971, pp. 327-335.

[L1] Lawler, E.L., J.K. Lenstra, A.H.G. Rinnooy Kan, and D.B. Shmoys, eds., The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization, John Wiley & Sons, Chichester UK, 1985.

[L2] Lin, S., "Computer Solutions of the Traveling Salesman Problem", Bell Systems Technical Journal 44, 1965, pp. 2245-2269.

[L3] Lin, S., and B.W. Kernighan, "An Effective Heuristic Algorithm for the Traveling Salesman Problem", Operations Research 21, 1973, pp. 498-516.

[L4] Little, J.D.C., K.G. Murty, D.W. Sweeney, and C. Karel, "An algorithm for the traveling salesman problem", Operations Research 11, 1963, pp. 972-989.

[M1] Mitter, S., Private communication, Semi-annual U.S. Army review of the Center for Intelligent Control, Massachusetts Institute of Technology, Cambridge MA, September 1989.

[P1] Padberg, M., and G. Rinaldi, "Optimization of a 532-city Symmetric Traveling Salesman Problem By Branch and Cut", Operations Research Letters, Vol. 6, No. 1, March 1987.

[P2] Padberg, M., and G. Rinaldi, "A Branch and Cut Algorithm for the Resolution of Large-scale Symmetric Traveling Salesman Problems", New York University technical report, April 1989.

[P3] Papadimitriou, C.H., "The Euclidean Traveling Salesman Problem is NP-complete", Theoretical Computer Science 4, 1977, pp. 237-244.

[P4] Papadimitriou, C.H., and K. Steiglitz, "Some Complexity Results for the Traveling Salesman Problem", Eighth Annual Symp. on Theory of Comput., May 1976, pp. 1-9.

[P5] Preparata, F.P., and M.I. Shamos, Computational Geometry: An Introduction, Springer-Verlag, New York NY, 1985.

[R1] Reinelt, G., "TSPLIB - A Traveling Salesman Problem Library", Institute of Mathematics, University of Augsburg, Augsburg Germany, 20 March 1989.

[S1] Sedgewick, R., Algorithms, Addison-Wesley, Reading MA, 1983.

[S2] Shamos, M.I., "Computational Geometry", Ph.D. Thesis, Department of Computer Science, Yale University, New Haven CT, 1978.

[S3] Steiglitz, K., An Introduction to Discrete Systems, John Wiley & Sons, New York NY, 1974.

[S4] Stewart, W.R., "A Computationally Efficient Heuristic for the Traveling Salesman Problem", Proceedings 13th Annual Meeting of S.E. TIMS, 1977.

[V1] Varaiya, P.P., Notes on Optimization, Von Nostrand Reinhold Company, New York NY, 1972.

[V2] Voronoi, G., "Nouvelles applications des parametres continus a la theorie des formes quadratiques", J. Reine Angew. Math. 134, 1908, pp. 97-178.

[Y1] Young, J.W., T. Fort, and F.M. Morgan, Analytic Geometry, Houghton Mifflin, Cambridge MA, 1936.

Optimal Digital Redesign of Continuous-time systems

Leang S. Shieh and Jian L. Zhang

Department of Electrical Engineering, University of Houston,

University Park, Houston, TX 77204-4793

Norman P. Coleman

U.S. Army Armament Center, Picatinny Arsenal, Dover, New Jersey 07806-5000

Abstract

This paper proposes a new optimal digital redesign technique for finding a dynamic digital control law from the available analog counter part and simultaneously minimizing a quadratic performance index. The proposed technique can be applied to a system with a more general class of reference inputs, and the developed digital regulator can be implemented via low cost microcomputers.

1. Introduction

Many practical dynamic systems are described by continuous-time state equations for which a state-feedback gain and a forward gain are designed based upon some specific desired goals. Advances in digital control theory and industrial electronics have made a dramatic extension in the possibilities of replacing these analog controllers by the equivalent digital controllers so that they can be implemented via high performance, low cost microprocessors and associated microelectronics. The conversion of the designed continuous-time controller (analog controller) to an equivalent discrete-time controller (digital controller) so that the responses of the redesigned equivalent digital system closely match those of the original analog system for the same input and initial conditions is a digital redesign problem [1]. The digital redesign problem can be described as follows.

Consider the linear controllable continuous-time system described by

$$\dot{x}_c(t) = Ax_c(t) + Bu_c(t); \quad x_c(0) \tag{1}$$

where $x_c(t)$ and $u_c(t)$ are an $n \times 1$ state vector and an $m \times 1$ input vector, respectively, and A and B are constant matrices of appropriate dimensions. Let the state-feedback control law be

$$u_c(t) = -K_c x_c(t) + E_c r(t) \tag{2}$$

where K_c is an $m \times n$ feedback gain, E_c is an $m \times m$ forward gain, and $r(t)$ is an $m \times 1$ reference input. The resulting closed-loop system becomes

$$\dot{x}_c(t) = (A - BK_c)x_c(t) + BE_c r(t); \quad x_c(0) \tag{3}$$

Let the state equation of a continuous-time system which contains the same system matrix A and input matrix B of the system in (1), with a different input, be represented by

$$\dot{x}_d(t) = Ax_d(t) + Bu_d(t); \quad x_d(0) \tag{4a}$$

where $u_d(t)$ is an $m \times 1$ piecewise-constant input function,

$$u_d(t) = u_d(kT) \qquad \text{for} \quad kT \le t < (k+1)T \tag{4b}$$

and T is the sampling period. A zero-order hold is utilized in (4). The solution of the state equation in (4) is

$$x_d(t) = e^{A(t-kT)}x_d(kT) + \int_{kT}^{t} e^{A(t-\lambda)}B\,d\lambda\,u_d(kT) \quad \text{for} \quad kT \le t \le kT+T \qquad (5)$$

For $t = kT + T$, the equivalent discrete-time model of the continuous-time system in (4) can be written as

$$x_d(kT+T) = Gx_d(kT) + Hu_d(kT); \quad x_d(0) \qquad (6a)$$

where

$$G \triangleq e^{AT} \quad \text{and} \quad H \triangleq \int_0^T e^{A\lambda}B\,d\lambda = [G - I_n]A^{-1}B \qquad (6b)$$

Let the discretized state-feedback control law for the system in (4) be

$$u_d(kT) = -K_d x_d(kT) + E_d r(kT) \qquad (7)$$

where K_d is an $m \times n$ digital feedback gain, E_d is an $m \times m$ digital forward gain, and $r(kT)$ is an $m \times 1$ discrete-time reference input. The resulting closed-loop system becomes

$$\dot{x}_d(t) = Ax_d(t) - BK_d x_d(kT) + BE_d r(kT); \quad x_d(0) \quad \text{for} \quad kT \le t < (k+1)T \qquad (8)$$

Now, the static digital redesign problem reduces to finding the digital constant state-feedback gain K_d and forward gain E_d in (7) from the continuous state-feedback gain K_c and forward gain E_c in (2) so that the states of the digital model in (8) are approximately equal to the states of the analog system in (3) for $x_c(0) = x_d(0)$ and the same reference input.

In Kuo's pioneer work [1], a discrete-state matching method was proposed to solve the static digital redesign problem and successfully applied to a simplified one-axis skylab satellite system [1]. In their work [1], they have assumed that the continuous-time reference input $r(t)$ in (2) can be closely approximated by the piecewise-constant input $r(kT)$ in (8) and the continuous-time state $x_c(t)$ in (3) can be closely matched the continuous-time state $x_d(t)$ in (8) at each sampling instant, $t = kT$, with a sufficiently small sampling period T. The values of $x_c(t)$ in (3) and $x_d(t)$ in (8) between each sampling instant are not considered in their work. In this paper, a new optimal digital regesign technique is

proposed for finding a dynamic digital control law, instead of the static digital control law as shown in (7), from the available analog control law in (2) for a continuous-time reference input $r(t)$. It is optimal in the sense that the quadratic performance index of the errors between $x_c(t)$ in (3) and the digital redesigned state, controlled by a dynamic digital control law, is minimized.

2. Optimal Digital Redesign

Consider the dynamic system described as in (3) and (4) with $x_c(0) = x_d(0)$. Let the quadratic cost function be

$$J = \frac{1}{2}\int_0^\infty [x_d(t) - x_c(t)]^T Q[x_d(t) - x_c(t)]\, dt \tag{9}$$

where $Q \in \mathcal{R}^{n\times n}$ is a positive definite symmetric weighting matrix, $x_c(t)$ is the state of the system in (3), and $x_d(t)$ is the state of the system in (4) to be redesigned. It is desirable to find a dynamic digital control law for the system in (6a) such that J in (9) is minimized. The above optimization problem is slightly different from an optimal state tracking problem [2] in the sense that the states of interest in (9) are those of the dynamic systems in (3) and (4) which involve the same system matrix A and input matrix B with different input functions.

An alternative expression of J in (9) is

$$\begin{aligned} J &= \sum_{k=0}^{\infty} \frac{1}{2}\int_{kT}^{kT+T} [x_d(t) - x_c(t)]^T Q[x_d(t) - x_c(t)]\, dt \\ &\triangleq \sum_{k=0}^{\infty} J_k \end{aligned} \tag{10a}$$

where

$$J_k \triangleq \frac{1}{2}\int_{kT}^{kT+T} [x_d(t) - x_c(t)]^T Q[x_d(t) - x_c(t)]\, dt \tag{10b}$$

Assume that the continuous-time reference input $r(t)$ in (3) can be realized via zero-input state equations (an example of this method is shown in Appendix 7.1) as follows:

$$\dot{y}_r(t) = A_r y_r(t); \quad y_r(0) \tag{11a}$$

$$r(t) = C_r y_r(t) \tag{11b}$$

where $y_r(t)$ is an $p_r \times 1$ state vector, $r(t)$ in (11b) is an $m \times 1$ output vector (the reference input of the system in (3)), and A_r and C_r are constant matrices of appropriate dimensions. Combining the state equations in (3) and (11) leads to

$$\begin{bmatrix} \dot{x}_c(t) \\ \dot{y}_r(t) \end{bmatrix} = \begin{bmatrix} A_c & BE_cC_r \\ 0 & A_r \end{bmatrix} \begin{bmatrix} x_c(t) \\ y_r(t) \end{bmatrix}; \quad \begin{bmatrix} x_c(0) \\ y_r(0) \end{bmatrix} \in \mathcal{R}^{n_1 \times 1} \tag{12a}$$

or

$$\dot{q}(t) = A_1 q(t); \qquad q(0) \tag{12b}$$

where $n_1 \triangleq n + p_r$, $A_c \triangleq A - BK_c$, and

$$A_1 \triangleq \begin{bmatrix} A_c & BE_cC_r \\ 0 & A_r \end{bmatrix} \in \mathcal{R}^{n_1 \times n_1}, \quad q(t) \triangleq \begin{bmatrix} x_c(t) \\ y_r(t) \end{bmatrix} \in \mathcal{R}^{n_1 \times 1}$$

The solution of the state equation in (12) is given by

$$q(t) = e^{A_1(t-kT)} q(kT) \quad \text{for} \quad kT \le t \le kT + T \tag{13a}$$

The equivalent discrete-time model of the system in (13a) is

$$q(kT + T) = G_1 q(kT); \quad q(0) \tag{13b}$$

where

$$G_1 \triangleq e^{A_1 T} \triangleq \begin{bmatrix} G_c & H_c \\ 0 & G_r \end{bmatrix} \in \mathcal{R}^{n_1 \times n_1}, \quad q(kT) = \begin{bmatrix} x_c(kT) \\ y_r(kT) \end{bmatrix} \in \mathcal{R}^{n_1 \times 1}$$

with $G_c \triangleq e^{A_c T}$, $G_r \triangleq e^{A_r T}$, and $H_c \triangleq G_c \int_0^T e^{-A_c \lambda} BE_cC_r e^{A_r \lambda} \, d\lambda$ (Note that H_c can be solved using the method developed in Appendix 7.2). Also, by combining the dynamic systems in (4) and (12),we obtain

$$\begin{bmatrix} \dot{x}_d(t) \\ \dot{q}(t) \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & A_1 \end{bmatrix} \begin{bmatrix} x_d(t) \\ q(t) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u_d(kT) \tag{14a}$$

The solution of the state equation in (14a) can be obtained by combining the solutions in (5) and (13a) as

$$\begin{bmatrix} x_d(t) \\ q(t) \end{bmatrix} = \begin{bmatrix} e^{A(t-kT)} & 0 \\ 0 & e^{A_1(t-kT)} \end{bmatrix} \begin{bmatrix} x_d(kT) \\ q(kT) \end{bmatrix} + \begin{bmatrix} \int_{kT}^{t} e^{A(t-\lambda)} B \, d\lambda \\ 0 \end{bmatrix} u_d(kT) \tag{14b}$$
$$\text{for} \quad kT \le t \le kT + T$$

For $t = kT + T$, the equivalent discrete-time model of the augmented system in (14) becomes

$$z(kT+T) = \hat{G}z(kT) + \hat{H}u_d(kT); \quad z(0) \tag{15}$$

where

$$z(kT) = [x_d^T(kT), q^T(kT)]^T \in \mathcal{R}^{(n+n_1)\times 1}$$

$$\hat{G} \triangleq \text{block diag}[G, G_1] \in \mathcal{R}^{(n+n_1)\times(n+n_1)} \quad \text{with} \quad G = e^{AT} \quad \text{and} \quad G_1 = e^{A_1 T}$$

and $$\hat{H} \triangleq [H^T, 0]^T \in \mathcal{R}^{(n+n_1)\times m} \quad \text{with} \quad H = \int_0^T e^{A\lambda} B \, d\lambda = [G - I_n]A^{-1}B$$

Note that the augmented system in (15) contains the reference subsystem in (11), whereas the cost function in (9) does not include the state of the reference subsystem in (11). To include the state of the reference subsystem in (11) into the cost function in (9), we modify the cost function in (10) as follows:

$$\begin{aligned} J_k &= \frac{1}{2}\int_{kT}^{kT+T} [x_d^T(t), x_c^T(t)] \begin{bmatrix} Q & -Q \\ -Q & Q \end{bmatrix} \begin{bmatrix} x_d(t) \\ x_c(t) \end{bmatrix} dt \\ &= \frac{1}{2}\int_{kT}^{kT+T} z^T(t)\tilde{Q}z(t)\, dt \end{aligned} \tag{16}$$

where

$$z(t) \triangleq [x_d^T(t), q^T(t)]^T = [x_d^T(t), x_c^T(t), y_r^T(t)]^T \in \mathcal{R}^{(n+n_1)\times 1}$$

$$\tilde{Q} \triangleq \begin{bmatrix} Q & -Q & 0 \\ -Q & Q & 0 \\ 0 & 0 & 0 \end{bmatrix} \in \mathcal{R}^{(n+n_1)\times(n+n_1)}$$

Substituting (14b) into (16) and making some algebraic simplifications results in

$$J_k = \frac{1}{2}z^T(kT)\hat{Q}z(kT) + z^T(kT)Mu_d(kT) + \frac{1}{2}u_d^T(kT)Ru_d(kT) \tag{17a}$$

where

$$\hat{Q} \triangleq \begin{bmatrix} \hat{Q}_{11} & \hat{Q}_{12} \\ \hat{Q}_{12}^T & \hat{Q}_{22} \end{bmatrix} \in \mathcal{R}^{(n+n_1)\times(n+n_1)} \tag{17b}$$

with

$$\hat{Q}_{11} \triangleq \int_0^T e^{A^T t} Q e^{At}\, dt \in \mathcal{R}^{n\times n}$$

$$\hat{Q}_{12} \triangleq \int_0^T e^{A^T t}[-Q, 0] e^{A_1 t}\, dt \in \mathcal{R}^{n\times n_1}$$

$$\hat{Q}_{22} \triangleq \int_0^T e^{A_1^T t} \begin{bmatrix} Q & 0 \\ 0 & 0 \end{bmatrix} e^{A_1 t}\, dt \in \mathcal{R}^{n_1\times n_1}$$

and

$$M \triangleq \begin{bmatrix} M_1 \\ M_2 \end{bmatrix} \in \mathcal{R}^{(n+n_1)\times m} \tag{17c}$$

with

$$M_1 \triangleq \int_0^T \{e^{A^T t} Q \int_0^t e^{A\lambda} B \, d\lambda\} dt \in \mathcal{R}^{n\times m}$$

$$M_2 \triangleq \int_0^T \{e^{A_1^T t}[-Q, 0]^T \int_0^t e^{A\lambda} B \, d\lambda\} dt \in \mathcal{R}^{n_1\times m}$$

and

$$R \triangleq \int_0^T \{[\int_0^t e^{A\lambda} B \, d\lambda]^T Q [\int_0^t e^{A\lambda} B \, d\lambda]\} dt \in \mathcal{R}^{m\times m} \tag{17d}$$

If the matrices A and A_1 satisfy certain conditions (see Appendix 7.2), the weighting matrices $\hat{Q}$, M, and R can be solved from a set of Lyapunov equations. Thus, the quadratic cost function in (10) can be rewritten as

$$J = \sum_{k=0}^{\infty} [\frac{1}{2} z^T(kT)\hat{Q} z(kT) + z^T(kT) M u_d(kT) + \frac{1}{2} u_d^T(kT) R u_d(kT)] \tag{18}$$

Now, we can easily identify that the cost function in (18) and the dynamic equation in (15) constitute a standard discrete-time optimal regulator problem [1,2]. The optimal control law is given [1,2] by

$$u_d(kT) = -(R + \hat{H}^T P \hat{H})^{-1}(\hat{H}^T P \hat{G} + M^T) z(kT) \tag{19a}$$

where $P \in \mathcal{R}^{(n+n_1)\times(n+n_1)}$ is the positive definite symmetric solution of the discrete-time Riccati equation:

$$P = \hat{G}^T P \hat{G} + \hat{Q} - (\hat{G}^T P \hat{H} + M)(R + \hat{H}^T P \hat{H})^{-1}(\hat{G}^T P \hat{H} + M)^T \tag{19b}$$

Since the adjoint system in (15) is not completely controllable, it is not always possible to find a positive semidefinite symmetric matrix P from (19b). However, for a stable subsystem matrix G_1, there exists a positive definite symmetric matrix P [3] which can be solved as follows.

Define the matrix P as

$$P \triangleq \begin{bmatrix} P_{11} & P_{12} \\ P_{12}^T & P_{22} \end{bmatrix} \tag{20a}$$

where $P_{11} \in \mathcal{R}^{n\times n}$, $P_{12} \in \mathcal{R}^{n\times n_1}$, and $P_{22} \in \mathcal{R}^{n_1\times n_1}$. The Riccati equation in (19b) can be partitioned into separate equations for P_{11}, P_{12}, and P_{22}:

$$P_{11} = G^T P_{11} G + \hat{Q}_{11} - (G^T P_{11} H + M_1)(R + H^T P_{11} H)^{-1}(G^T P_{11} H + M_1)^T \quad (20b)$$

$$P_{12} = G^T P_{12} G_1 + \hat{Q}_{12} - (G^T P_{11} H + M_1)(R + H^T P_{11} H)^{-1}(G_1^T P_{12}^T H + M_2)^T \quad (20c)$$

$$P_{22} = G_1^T P_{22} G_1 + \hat{Q}_{22} - (G_1^T P_{12}^T H + M_2)(R + H^T P_{11} H)^{-1}(G_1^T P_{12}^T H + M_2)^T \quad (20d)$$

Equation (20b) is a discrete-time algebraic Riccati equation and can be solved via eigenvalue-eigenvector approach [4] or sign algorithm [5]. Once P_{11} has been found, it is substituted into (20c), which can be rearranged into the following Lyapunov equation:

$$\begin{aligned}&[G - H(R + H^T P_{11} H)^{-1}(H^T P_{11} G + M_1^T)]^T P_{12} - P_{12} G_1^{-1} \\ &\quad + [\hat{Q}_{12} - (G^T P_{11} H + M_1)(R + H^T P_{11} H)^{-1} M_2^T] G_1^{-1} = 0 \end{aligned} \quad (21)$$

Equation (21) can be solved via a matrix direct-product method [6]. The desired optimal digital control law in (19a) becomes

$$u_d(kT) = -K_d x_d(kT) + K_q q(kT) \quad (22)$$

where

$$K_d = (R + H^T P_{11} H)^{-1}(H^T P_{11} G + M_1^T)$$

$$K_q = -(R + H^T P_{11} H)^{-1}(H^T P_{12} G_1 + M_2^T)$$

Because $q(kT)$ in (22) is generated from the dynamic system in (12) or (13), *i.e.*,

$$q(kT+T) = \begin{bmatrix} x_c(kT+T) \\ y_r(kT+T) \end{bmatrix} = \begin{bmatrix} G_c & H_c \\ 0 & G_r \end{bmatrix} \begin{bmatrix} x_c(kT) \\ y_r(kT) \end{bmatrix}; \quad \begin{bmatrix} x_c(0) \\ y_r(0) \end{bmatrix} \quad (23)$$

We decompose the dynamic gain K_q in (22) as $K_q = [\hat{K}_c, \hat{K}_r]$ where $\hat{K}_c \in \mathcal{R}^{m\times n}$ and $\hat{K}_r \in \mathcal{R}^{m\times p_r}$. Hence the desired optimal dynamic digital control law in (22) can be rewritten as

$$u_d(kT) = -K_d x_d(kT) + \hat{K}_c x_c(kT) + \hat{K}_r y_r(kT) \quad (24a)$$

where

$$x_c(kT) = G_c^k x_d(0) + \sum_{i=0}^{k-1} G_c^{k-i-1} H_c y_r(iT) \quad (24b)$$

and

$$y_r(kT) = G_r^k y_r(0) \tag{24c}$$

Thus, the digital redesigned system via the optimal dynamic digital controller in (24) becomes

$$\dot{x}_d(t) = Ax_d(t) - BK_d x_d(kT) + B\hat{K}_c x_c(kT) + B\hat{K}_r y_r(kT); \quad x_d(0) \tag{25}$$

The digital redesigned closed-loop system is shown in Fig. 1.

If $y_r(t)$ in (11) is measurable, or the initial vector $y_r(0)$ is available, the control law in (24a) can be realized via a microcomputer. However, in practice, it is quite possible that only an incoming signal $r(t)$ is available. In this case, an estimator can be constructed with $r(t)$ as an input, and the estimated state $\hat{y}_r(t)$ of $y_r(t)$ as an output [2] provided that the pair $[A_r,\ C_r]$ are observable.

When $r(t)$ in (11) is a step function, then $C_r = I_m$, $y_r(t) = r(t)$, and $y_r(kT) = r(kT)$. The optimal dynamic digital control law in (24) reduces to

$$\begin{aligned} u_d(kT) &= -K_d x_d(kT) + \hat{K}_c x_c(kT) + \hat{K}_r r(kT) \\ &= -K_d x_d(kT) + \hat{K}_c G_c^k x_d(0) + [\hat{K}_c \sum_{i=0}^{k-1} G_c^{k-i-1} H_c + \hat{K}_r] r(kT) \end{aligned} \tag{26}$$

3. Illustrative Example

Consider an unstable system in (1) with

$$A = \begin{bmatrix} 0.809 & -2.060 & 0.325 & 0.465 & 0.895 \\ 6.667 & 0.200 & 1.333 & 0.000 & 0.667 \\ -1.291 & 0.458 & -1.072 & -2.326 & -0.199 \\ -0.324 & 0.824 & 1.670 & -1.186 & -0.358 \\ -3.509 & -4.316 & -0.702 & 0.000 & -8.351 \end{bmatrix} \tag{27a}$$

$$B = \begin{bmatrix} 0.955 & -0.379 \\ -1.667 & -1.667 \\ -0.212 & 1.195 \\ 0.618 & 0.052 \\ 0.877 & 1.403 \end{bmatrix}; \quad x_c(0) = 0 \tag{27b}$$

and the eigenvalues of A are $\sigma(A) = \{0.2 \pm j4.0, -1.0 \pm j2.0, -8.0\}$.

Using the optimal pole-placement method proposed in [7], the optimal state feedback gain K_c in (2) is found as

$$K_c = \begin{bmatrix} 7.871 & -0.563 & 3.255 & -0.137 & 0.754 \\ 1.625 & -1.247 & 1.297 & -1.003 & 0.182 \end{bmatrix} \tag{28}$$

Utilizing the feedback gain K_c, the eigenvalues of the closed-loop system in (3) are placed within the common region of an open sector (with a sector angle $\pm 45°$ from the negative real axis) and the left-hand side of a -1.1 vertical line on the negative real axis in the complex s-plane, and $\sigma(A - BK_c) = \{-4.6789 \pm j4.6518, -1.8983 \pm j1.898, -8.0\}$.

Assume $E_c = I_2$ in (2), and let that the reference input $r(t)$ in (2) contain a sine function ($\sin(\omega t)$) with an angular frequency $\omega = 3.0$ and an unit-step function, that is

$$r(t) = \begin{bmatrix} r_1(t) \\ r_2(t) \end{bmatrix} = \begin{bmatrix} \sin(3.0t) \\ 1 \end{bmatrix}; \quad t \geq 0 \tag{29}$$

The reference input $r(t)$ can be represented by a zero-input state equation in (11) with

$$A_r = \begin{bmatrix} 0.0 & 3.0 & 0.0 \\ -3.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{bmatrix} \tag{30a}$$

$$C_r = \begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \tag{30b}$$

and an initial vector $y_r(0) = [0.0 \quad 1.0 \quad 1.0]^T$.

Using the method proposed in this paper, we obtain the dynamic digital control law in (24) with a sampling period $T = 0.5(sec.)$ as

$$u_d(kT) = -K_d x_d(kT) + \hat{K}_c x_c(kT) + \hat{K}_r y_r(kT) \tag{31}$$

where

$$K_d = \begin{bmatrix} 0.7940 & -1.0970 & -0.2206 & 0.3500 & 0.0574 \\ -2.2280 & -0.2333 & 0.0002 & -0.7134 & -0.1575 \end{bmatrix} \tag{32a}$$

$$\hat{K}_c = \begin{bmatrix} -0.4472 & -0.2190 & -0.7267 & 0.2998 & -0.0587 \\ -0.0501 & 0.0972 & 0.5231 & -0.2748 & 0.0399 \end{bmatrix} \tag{32b}$$

$$\hat{K}_r = \begin{bmatrix} 0.2112 & 0.2314 & -0.2938 \\ -0.1482 & -0.0498 & 0.5082 \end{bmatrix} \tag{32c}$$

The simulation results of the closed-loop systems in (3) and (25) are shown in Fig.2 for both $x_c(t)$ in (3) and $x_d(t)$ in (25), and those of the controls $u_c(t)$ in (2) and $u_d(t)$ in (24) are shown in Fig.3. The simulation results have shown that $x_d(t)$ is very close to $x_c(t)$ even with a rather larger sampling period (considering the dynamics of the given system and the frequency of the reference input).

4. Conclusion

A new optimal digital redesign technique has been developed for finding a dynamic digital control law in (24) from the available analog counterpart in (2) and simultaneously minimizing a quadratic performance index in (9). First, an augmented system in (14a), which consists of a reference model in (11), an orginal closed-loop system in (3) and the digital controlled original open-loop system in (4*a*), is constructed and converted into a discrete-time dynamic system in (15) to be designed. Next, a quadratic performance index in (18) is established from that in (9) using a set of weighting matrices in (17). A set of Lyapunov equations have been developed in Appendix 7.2 to find the weighting matrices for the quadratic performance index. Then, a standard discrete-time optimal regulator in (22) is determined by solving a small dimensional Riccati equation in (20*b*) and a Lyapunov equation in (21). Finally, the desired digital state-feedback gain and forward gain in (24*a*) can be computed from those in (22), (24*b*) and (24c). An illustrative example has been presented to demonstrate the effectiveness of the proposed method. The developed dynamic digital redesigned control law enables an optimally close matching of the states of the digital redesigned closed-loop system as compared to the states of the original closed-loop system and it can be implemented via low cost microcomputers. The proposed technique can be applied to a system with a more general class of reference inputs having a relatively large sampling period.

5. Acknowledgements

This work was supported in part by the U.S. Army Research Office under contract DAAL-03-87-K-0001.

6. References

[1.] KUO, B.C. :"Digital Control Systems" (Holt. Rinehart and Winston, 1980).

[2.] ANDERSON, B.D.O. and Moore, J.B. :"Linear Optimal Control" (Prentice-Hall, Englewood Cliffs, New Jersey, 1971)

[3.] DRESSLER, R.M., and Larson, R.E. :"Computation of Optimal Control in Partially Controlled Linear Systems", *Proceedings of the 1968 Joint Automatic Control Conference*, 1968, pp.711–715.

[4.] PATEL, R.V., and Munro, N. :"Multivariable System Theory and Design" (Pergamon Press, 1982).

[5.] SHIEH, L.S., Wang, C.T., and Tsay, Y.T. :"Fast Suboptimal State-space Self-tuner for Linear Stochastic Multivariable Systems", *IEE Proc. D. Control Theory and Appl.*, 1983, (4), pp.143–154.

[6.] BELLMAN, R. :"Introduction to Matrix Analysis" (McGraw-Hill, 1970).

[7.] SHIEH, L.S., Dib, H.M., and Ganesan, S. :"Continuous-time Quadratic Regulators and Pseudo-continuous-time Quadratic Regulators with Pole Placement in a Specific Region", *IEE Proc., D.*, 1987, (5), pp.138–346.

[8.] SHIEH, L.S., Chen, C.F., and Huang, C.J. :"On Identifying Transfer Functions and State Equations for Linear Syaytems", *IEEE Trans. on Aerospace and Electronic Systems*, 1972, AES–8, pp.811–820.

7. Appendices

Appendix 7.1

Let an $m \times 1$ output rational function $R(s)$, which is the product of a transfer function matrix and an input function, be represented by an irredu ble left matrix fraction description [4] as

$$R(s) = [I_m s^p + D_1 s^{p-1} + \ldots + D_p]^{-1}[N_1 s^{p-1} + \ldots + N_p] \qquad (A.1)$$

where $D_i \in \mathcal{R}^{m\times m}$ and $N_i \in \mathcal{R}^{m\times m}$ for $i = 1, 2, \ldots, p$.

The left matrix fraction description can be realized by the following zero-input state equtation:

$$\dot{y}_r(t) = A_r y_r(t); \quad y_r(0) \qquad (A.2)$$

$$r(t) = C_r y_r(t) \qquad (A.3)$$

where $y_r(t) \in \mathcal{R}^{pm\times 1}$, $r(t) \in \mathcal{R}^{m\times 1}$

$$A_r = \begin{bmatrix} 0_m & I_m & 0_m & \ldots & 0_m \\ 0_m & 0_m & I_m & \ldots & 0_m \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -D_p & -D_{p-1} & -D_{p-2} & \ldots & -D_1 \end{bmatrix}, \quad C_r^T = \begin{bmatrix} I_m \\ 0_m \\ \vdots \\ 0_m \end{bmatrix}$$

and

$$y_r(0) = \begin{bmatrix} I_m & 0_m & 0_m & \ldots & 0_m & 0_m \\ D_1 & I_m & 0_m & \ldots & 0_m & 0_m \\ D_2 & D_1 & I_m & \ldots & 0_m & 0_m \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ D_{p-1} & D_{p-2} & D_{p-3} & \ldots & D_1 & I_m \end{bmatrix}^{-1} \begin{bmatrix} N_1 \\ N_2 \\ N_3 \\ \vdots \\ N_p \end{bmatrix}$$

The above result can be obtained by following the method shown in [8]

Appendix 7.2

Some useful formulas for computing the matrices $\hat{Q}_{11}$, $\hat{Q}_{12}$, M_1, M_2, and R are given as follows.

Let the matrix $\hat{Q}_{11}$ be defined as

$$\hat{Q}_{11} \triangleq \int_0^T e^{A^T t} Q e^{At}\, dt = \int_0^T F_Q(t)\, dt \tag{A.4a}$$

where

$$F_Q(t) \triangleq e^{A^T t} Q e^{At} \tag{A.4b}$$

Taking the derivative of (A.4b) with respect to t gives

$$\dot{F}_Q(t) = A^T e^{A^T t} Q e^{At} + e^{A^T t} Q e^{At} A = A^T F_Q(t) + F_Q(t) A \tag{A.5}$$

Integrating (A.5) on both sides from 0 to T yields

$$\begin{aligned} \int_0^T \dot{F}_Q(t)\, dt = F_Q(t)\Big|_0^T &= e^{A^T T} Q e^{AT} - Q \\ &= A^T \int_0^T F_Q(t)\, dt + \int_0^T F_Q(t)\, dt\, A \end{aligned}$$

Thus

$$A^T \hat{Q}_{11} + \hat{Q}_{11} A = G^T Q G - Q \tag{A.6}$$

where $G \triangleq e^{AT}$.

When A is nonsingular, the unique solution $\hat{Q}_{11}$ can be solved from the Lyapunov equation in (A.6) via the matrix direct-product method [6].

Let $\hat{Q}_{12}$ be defined as

$$\hat{Q}_{12} \triangleq \int_0^T e^{A^T t} \tilde{Q} e^{A_1 t}\, dt \tag{A.7}$$

where $\tilde{Q} \triangleq [-Q, 0] \in \mathcal{R}^{n \times n_1}$. Similar to the above derivation, if A and A_1 are nonsingular and $\sigma_i(A) + \sigma_j(A_1) \neq 0$ for all i, j, where $\sigma(\cdot)$ denotes the eignspectrum of $(\cdot)$, then the unique solution $\hat{Q}_{12}$ can be obtained from the following Lyapunov equation:

$$A^T \hat{Q}_{12} + \hat{Q}_{12} A_1 = G^T \tilde{Q} G_1 - \tilde{Q} \tag{A.8}$$

where $G \triangleq e^{AT}$ and $G_1 \triangleq e^{A_1 T}$.

Let M_1 be defined as

$$M_1 \triangleq \int_0^T F_M(t)\,dt \tag{A.9a}$$

where

$$F_M(t) \triangleq e^{A^T t} Q \int_0^t e^{A\lambda} B\,d\lambda \tag{A.9b}$$

Carrying out the differentiation of $(A.9b)$ with respect to t, we obtain

$$\begin{aligned} \dot{F}_M(t) &= A^T e^{A^T t} Q \int_0^t e^{A\lambda} B\,d\lambda + e^{A^T t} Q e^{At} B \\ &= A^T F_M(t) + F_Q(t) B \end{aligned} \tag{A.10}$$

Integrating both sides of (A.10) from 0 to T gives

$$\begin{aligned} F_M(t)\Big|_0^T &= e^{A^T T} Q \int_0^T e^{A\lambda} B\,d\lambda = G^T Q H \\ &= A^T \int_0^T F_M(t)\,dt + \int_0^T F_Q(t)\,dt\,B = A^T M_1 + \hat{Q}_{11} B \end{aligned}$$

where $H \triangleq \int_0^T e^{A\lambda} B\,d\lambda$. Thus,

$$A^T M_1 + \hat{Q}_{11} B = G^T Q H \tag{A.11}$$

If A is nonsingular, then

$$M_1 = (A^T)^{-1}[G^T Q H - \hat{Q}_{11} B] \tag{A.12}$$

Define

$$M_2 \triangleq \int_0^T \left\{ e^{A_1^T t} \bar{Q}^T \Big[\int_0^t e^{A\lambda} B\,d\lambda \Big] \right\} dt$$

Similar to the derivations of (A.9) through (A.12), if A_1 is nonsingular, M_2 can be found as

$$M_2 = (A_1^T)^{-1}[G_1^T \bar{Q}^T H - \hat{Q}_{12}^T B]$$

Let R be defined as

$$R \triangleq \int_0^T \left\{ \Big[\int_0^t e^{A\lambda} B\,d\lambda \Big]^T Q \Big[\int_0^t e^{A\lambda} B\,d\lambda \Big] \right\} dt \tag{A.13}$$

If A is nonsigular, then

$$\int_0^t e^{A\lambda} B\,d\lambda = [e^{At} - I_n] A^{-1} B \tag{A.14}$$

Substitute (A.14) into (A.13), we have

$$\begin{aligned} R &= (A^{-1}B)^T \int_0^T \left[e^{A^T t} Q e^{At} - e^{A^T t} Q - Q e^{At} + Q\right] dt\, (A^{-1}B) \\ &= (A^{-1}B)^T [\hat{Q}_{11} - (G - I_n)^T (A^T)^{-1} Q - Q(G - I_n) A^{-1} + QT](A^{-1}B) \end{aligned} \tag{A.15}$$

If A and/or A_1 are singular, the matrices $\hat{Q}_{11}$, $\hat{Q}_{12}$, M_1, M_2, and R can be computed by any numerical integration method.

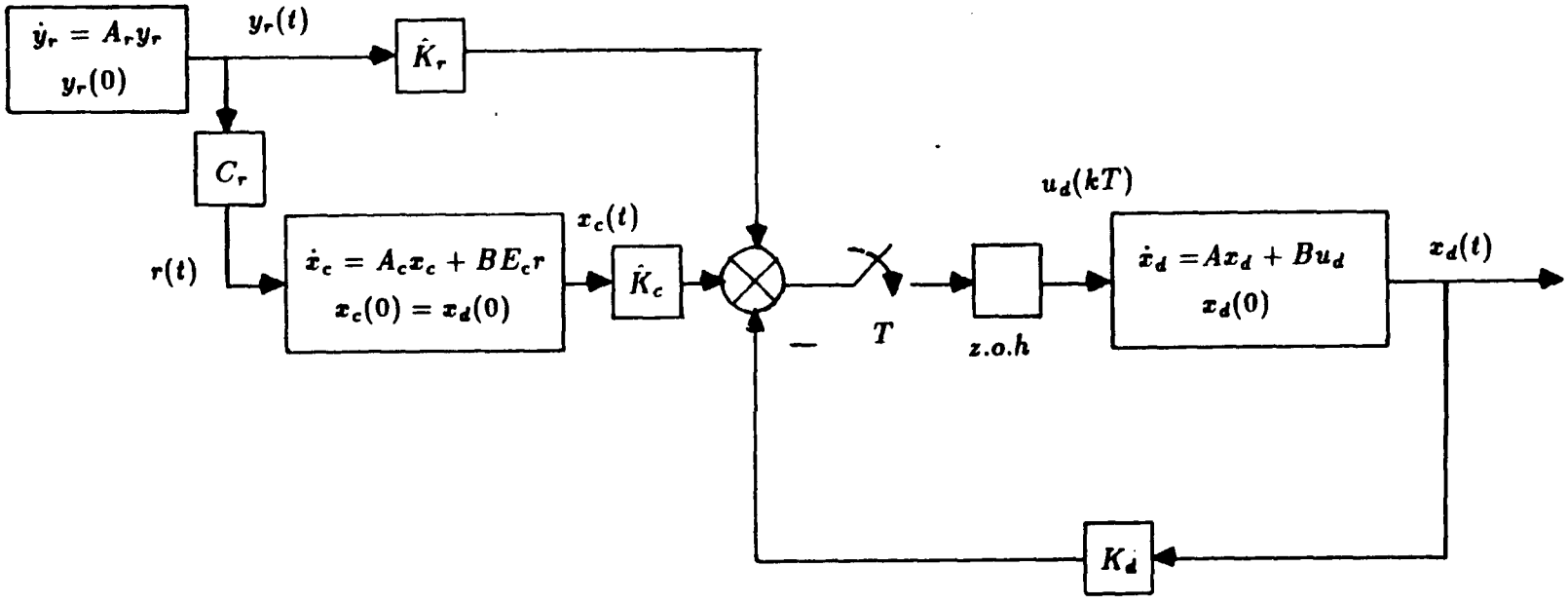


Fig.1 The digital redesigned closed-loop system.

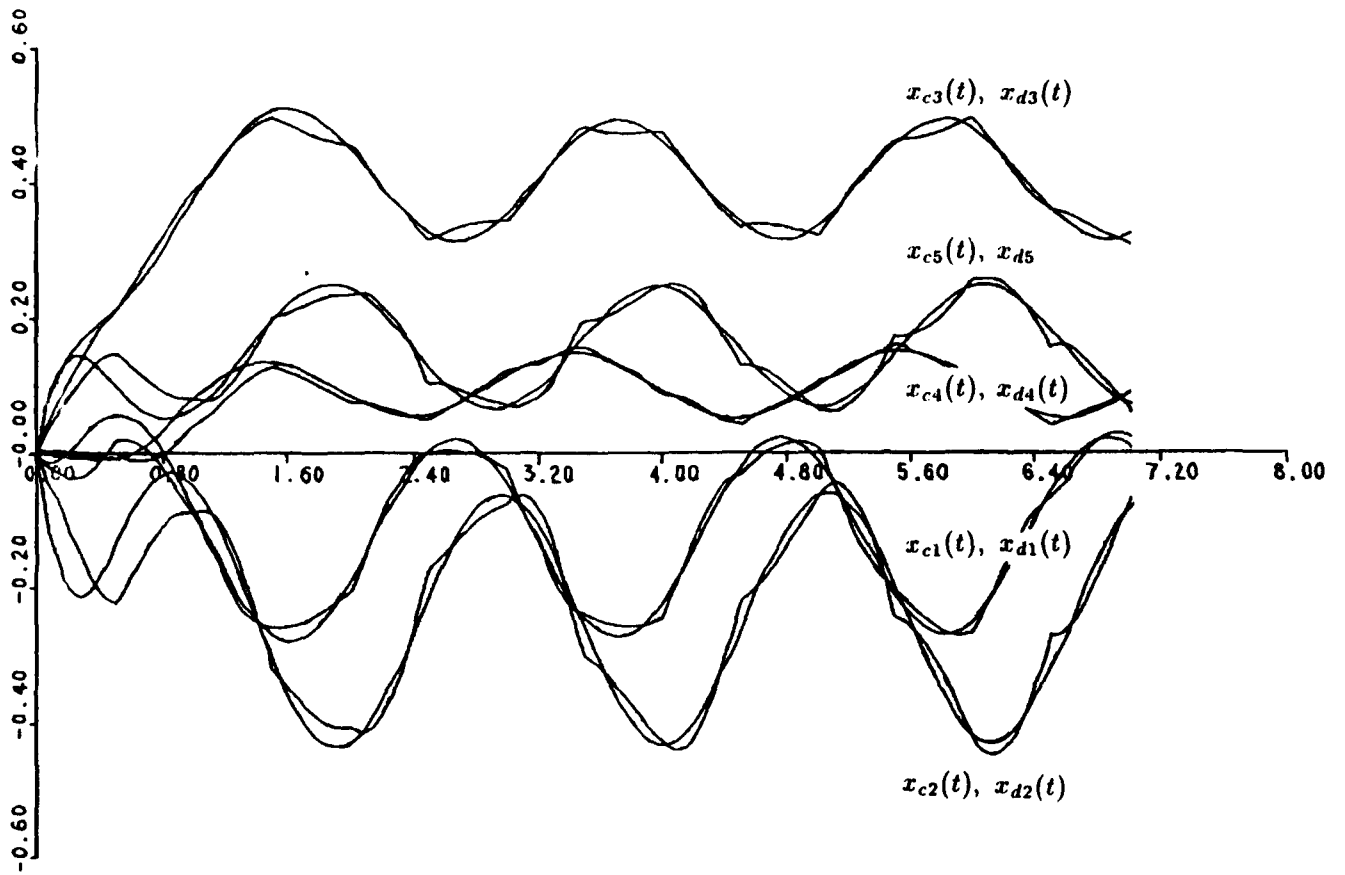


Fig.2 The state responses of the original closed-loop system in (3) and the digital redesigned closed-loop system in (25).

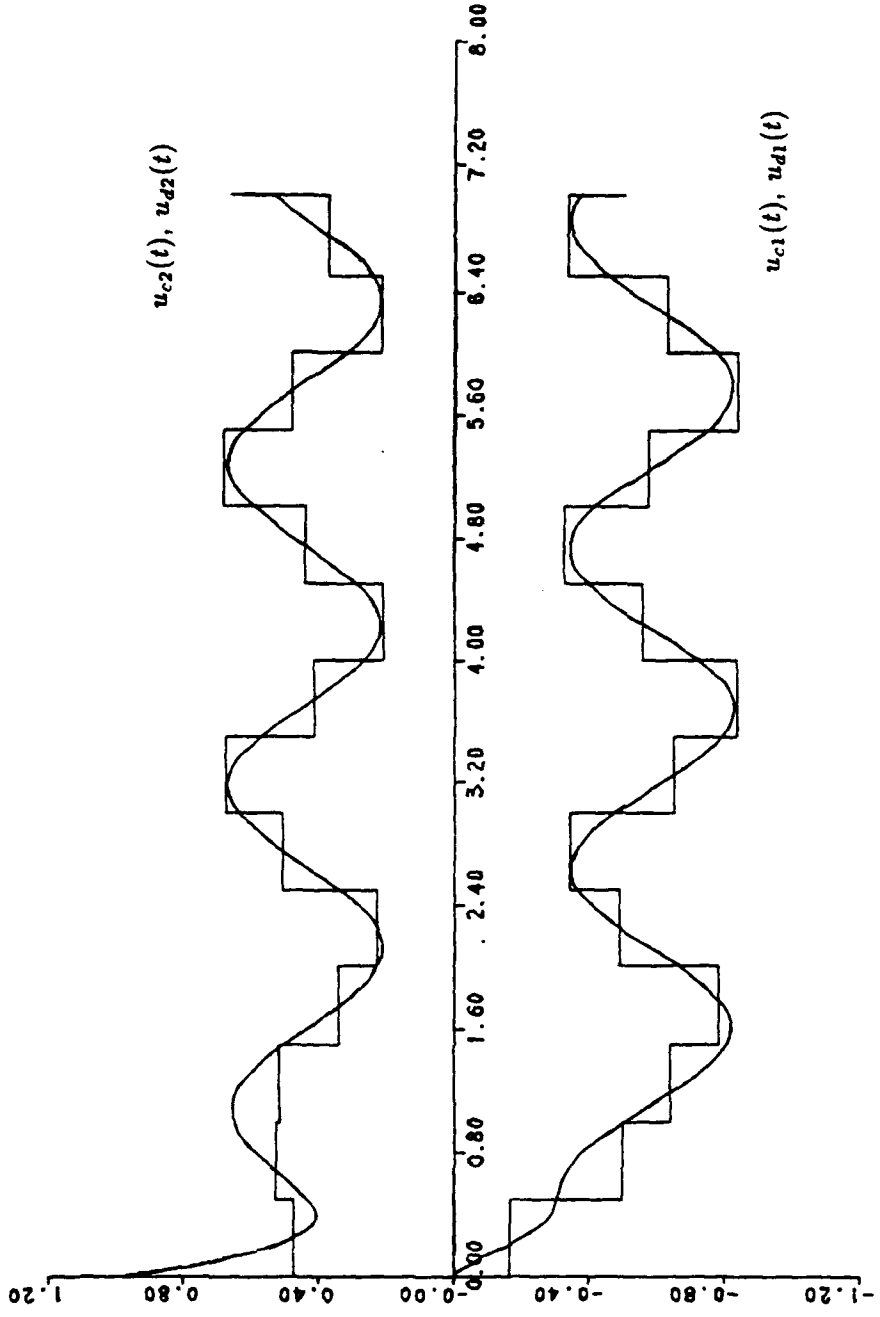


Fig.3 Controls $u_c(t)$ in (2) and $u_d(t)$ in (24).

Effectiveness of A Class of Smart Munitions:

A Stochastic Model

B. D. Sivazlian and K. Gakis
Department of Industrial and Systems Engineering
University of Florida
Gainesville, Florida 32611

ABSTRACT

A methodology is developed to assess the effectiveness of a class of smart munition system. Following a formal analysis of the aggregate problem and the characterization of the elements of the system, a prototype mathematical model is formulated. This model describes the temporal operation of the weapon system in the battlefield in the presence of threats and countermeasures. Simultaneously, it captures the uncertainty element typically arising in such problems. The solution results derived may be used to develop cost-free and cost-related measures of effectiveness to evaluate and select smart munitions weapon systems.

I. INTRODUCTION

Over the last decade, various DOD agencies have been involved in developing a family of weapon systems known as smart munitions (SM) which could significantly enhance the U.S. capability in the battlefield, while simultaneously improving mission survivability.

SMs have the autonomous capability to search, detect, acquire and engage targets. They can be delivered by a variety of means such as rockets, guns, dispensers, etc..., in large quantities over a large arrays of land-mobile targets. They can simultaneously engage multiple targets and be accurately delivered on selected targets without requiring an operator on the loop. The development of a methodology to assess the effectiveness of this new weapon system in the battlefield

while incorporating threats and countermeasures, becomes an important problem to be analyzed and studied.

II. OBJECTIVES OF THE RESEARCH EFFORT

As part of an ongoing program, the Analysis Division of the Air Force Armament Laboratory at Eglin AFB has requested its Technology Assessment Branch to provide an assessment for a Smart Submunition Technology program.

The objective of this program is to integrate advanced technologies for the next generation of smart submunitions (SM). The technologies would be advanced technology sensors, warhead, and maneuvering, compatible with advanced aircraft and dispenser delivery systems, and capable of providing substantial increases in effectiveness over current weapons against ground mobile targets. The target set being considered for the program spans the spectrum of ground mobile targets, and includes heavy armour, softer vehicles ranging from air defense targets to light armor, and rail transport.

The technology assessment necessitated the development of an appropriate methodology for evaluating the proposed system. As a result, the following tasks had to be undertaken:

1. Study the operational characteristics of the Smart Submunition Weapon Systems;
2. Analyze the system by identifying the various components in the operation of the system as well as the targets;
3. Characterize each component. Develop appropriate descriptive parameters which may be used as inputs in an effectiveness model;
4. Formulate a mathematical model which describes the operation of the weapon systems under combat condition by incorporating threat and by capturing the stochastic nature of the problem;
5. Provide a systematic methodology to solve the mathematical model;
6. Develop appropriate measures of effectiveness.

A brief review of items (1), (2) and (3) is provided. The emphasis of the present report is

1) to formulate a stochastic model and to provide a method of solution to a prototype model describing the operations of the weapon system;

2) to provide selected measures of effectiveness of the operation of the weapon system.

III. THE SMART SUBMUNITION WEAPON SYSTEMS (SSWS)

The effectiveness in the use of a SSWS depends on the assumption that the ultimate engagement in the battlefield is a "many-on-many". A number of smart submunitions (SS) are delivered in the proximity of an area where several targets are located such as tanks in a tank company. Through its sensors, each submunition is capable of locating, detecting, acquiring and engaging a target of a given type. The sensors have the capability of identifying the kind of targets (e.g. tanks or APC) that they are to engage. In general, the design of a SS is governed by the environment within which it will be deployed and the characteristics of the targets. As many factors as possible are accounted for in developing the configuration of the SS. Even then, the effectiveness of the SS may be enhanced or deterred depending upon the mode by which ch weapons are delivered in the vicinity of the target. For example, a parachute-suspended SS is typically not highly maneuverable and uses a small search footprint. As such, it would not be very effective against moving targets. A better design would be a parafoil-suspended SS or an inflatable-wing SS both of which are highly maneuverable and have a larger footprint. They have the capability of guiding the SS more rapidly towards the target, thus being more effective against moving targets. In addition, a larger footprint increases the probability of acquiring a target.

For the engagement to be successful, it is necessary to deliver a large number of weapons over a given area, and to provide each weapon with the capability to search and locate a target with a high probability of success. This requires that a large number of targets be available within the footprint of the delivery weapon. The larger the footprint, the more likely the weapon will acquire a target during its search, assuming the same target density in the area. Once acquired,

the discriminating sensors carried by the weapon will identify the target, select the ones to be attacked and thus pair each munition to a target of a kind. An obvious disadvantage of this system is the likelihood of more than one weapon attacking one particular target unless specific algorithms are built into the weapon system to preclude such situations. A second disadvantage is that in order to be effective, the SS must be placed in the vicinity of the target by a SS delivery system. Finally, in order to acquire and to precision guide towards the target, it becomes necessary to slow down the search and acquisition process. These disadvantages do not appear in most of the existing weapons involving one-on-one engagement in which the weapon is delivered from a much longer distance at a very high speed.

So far, of these three disadvantages, a solution has been found only to the second one in which an unmanned carrier or dispenser launched from a platform at a standoff position is used to place the weapons in the vicinity of the target. This however requires the use of a data link system which provides the platform with the necessary information about target area coordinates and target movement so that the carrier is launched and directed towards the vicinity of the target area. Additional information may have to be continuously provided to the dispenser regarding target location and the specific time at which weapons are to be released. Aerial and ground sensors are typical means to collect information on target location and movement. Aerial sensors may be in the form of remotely piloted vehicles (RPV) or unmanned air vehicles (UAV) or AWACS. A combination of aerial and ground sensors may be used. The information provided is transmitted to a C^3I post which then relays it to the launch platform. The launch platform function is to transport the dispenser and to utilize the relayed information from the C^3I post to aim and launch the dispenser from an appropriate location at a given time. The dispenser transports the SS subpacks to a given location and drops them so as to create a dispersal pattern which results in the best engagement opportunities for the SS. It must be noted that once the SS is dispensed, it depends solely on its own seekers and sensors to guide it terminally towards the target.

In detailing the effectiveness of the aggregate weapon system in the context of the mission it is supposed to be performing, one may not

neglect the contribution of the intelligence gathering system used in support of the mission as well as the contribution of C^3. The reliability of the mission is as good as the reliability of its components and C^3I must be considered as an integral part of the overall system.

When developing the appropriate equations to compute the overall mission reliability as a function of time, it may be assumed that the system is made up of two subsystems. The first subsystem consists of the C^3I components providing the data link. The second subsystem consists of the smart submunition weapon system (SSWS) whose components are the platform, the dispenser, the parafoil and the submunition. Assuming independence of operation of these two subsystems, the overall mission reliability is the product of the reliability of the first subsystem and the second subsystem. Mission reliability is sometimes used as one of the measures of effectiveness.

A very important consideration at this stage is that time becomes an important parameter to be accounted for in the development of appropriate measures of effectiveness. This is insignificant when studying the mission performance of traditional weapons since they rely on their high speed for delivery and on the element of surprise when attacking targets. However, in the case of SSWS, weapon delivery time is much longer. This in turn eliminates the element of surprise and provides the enemy significant more time to react to the attacking weapon system. Thus the enemy will have increased capability to perform such actions as:

- maneuvering out of the range of incoming weapons;
- visually acquiring and destroying the guiding vehicle of the weapon;
- initiating countermeasures to minimize or eliminate the effectiveness of the weapon;
- securing positions by scattering or scrambling so as to decrease the target density in the area of attack.

Typical components of the SSWS are:

1. The air platform;
2. The dispenser;

3. The parafoil;
4. The submunition warhead;
5. The submunition sensors;

Each of these components is characterized next. Following some remarks concerning the operational effectiveness of the weapon system, the target element is characterized.

1. The Air Platform

In the delivery of smart submunitions, it is envisioned that an aircraft will carry a number of dispensers, each loaded with several subpacks of submunitions. From a standoff position, and following a process of target area acquisition and location, the aircraft will fire the dispensers either in salvo or in sequence so that each dispenser is capable of maneuvering towards the target area.

Descriptive Parameters

- Range and speed;
- Average time to release dispenser from the moment the aircraft enters enemy territory;
- Intensity of threat encounter;
- Probability of aircraft being killed;
- Intensity of electronic jamming encounter;
- Probability of aircraft's communication and data link being jammed and loosing its mission capability.

2. The Dispenser

The dispenser is a container capable of self propulsion. It can either be preprogrammed to move from its launch platform towards the target area or it can be directed towards the area through a data link. Alternatively, it is conceivable that through its own sensors it has its own capability of homing towards the target area (autopilot). Once within target area vicinity, it releases the submunition subpacks either in salvo, or in sequence though an intervalometer setting. The release time of the subpacks from the time of the dispenser's release from the air platform is a variable and is constrained by the range of the dispenser. Once the subpacks are released, the dispenser being not programmed to be recovered is allowed to crash on landing and/or self-destruct.

The choice of the dispenser depends on the mission to be accomplished. This will dictate its capacity, load, speed, range and other characteristics. For example a dispenser with a longer range is desired if the weapons have to be carried from the air platform in a distant position to second echelon supply lines in an interdiction role. In addition the dispenser may have to generate a larger footprint.

Descriptive Parameters

- Range and speed;
- Average time from instant of dispenser release from air platform to instant of subpack releases;
- Intensity of threat encounter;
- Probability of dispenser being killed;
- Intensity of electronic jamming encounter;
- Probability of dispenser's data link being jammed and loosing its mission capability.

3. The Parafoil

The parafoil is the element used for the indirect delivery of the submunition in widely dispersed area of target elements. It is a wide-area search and control system of the targets with its own guidance. It generates the search of a target over a relatively large footprint and once the target is acquired by the submunition sensors, it controls and directs its motion towards the target so as to achieve a range from which the warhead could be fired. The payload consists of the sensor and the warhead. At the desired altitude, the subpacks in the dispenser are ejected and the submunitions released. The parafoils are then deployed and the search mode initiated. The wide area scan greatly increases the search area to compensate for large delivery errors. Once the fuze ignites the charge and an explosion is set up, the slug is formed. Simultaneously the submunition sensors and the parafoil are destroyed. The ability of a parafoil-controlled submunition to glide provides an increased search area and control to target. With an ability to change horizontal to vertical velocity ratio and to brake, the parafoil can be programmed to provide a simple terminal homing capability.

Descriptive Parameters

- Average time to search and acquire a target.

4. The Submunition Warhead

The objective of the warhead is to achieve mobility kill in ground mobile targets including heavy armor (tanks), softer vehicles (APCs, air defense targets, light armor) and rail transport. The means of attaining this objective is a modular Explosively Formed Penetrator (EFP). The warheads are used in shoot-to-kill sensor fuzed munitions(SFM).

In EFP warheads, the fuze ignites the explosive material (chemical energy warheads). Some of the explosive energy is used to reshape the liner and accelerate it towards the target.

5. The Submunition Sensors

The submunition sensors perform five basic functions:

a. A search function involving target search, detection, identification, discrimination, classification and acquisition;

b. A target location function involving the location, relative speeds, coordinates and other dynamic characteristics of the target with respect to the submunition for target engagement;

c. A maneuvering function so that once target is acquired, the submunition will maneuver to an optimum lethal range and position to fire a warhead at a predetermined aimpoint;

d. Auxiliary functions such as false target rejection, false alarm elimination, warhead mode selection, etc....

Sensors perform their functions by receiving electromagnetic radiation emitted by targets and their surrounding environment. Variations in electrical pulses due to radiation changes are sent to a signal processor which perform the above functions. Sensors are characterized by their operating mode (passive, active or dual) and their operating waveband (infrared, millimeter wave, etc...).

In a passive mode, sensors receive radiation through a receiver, emitted or reflected by objects on the battlefield. In an active mode, sensors transmit radiation through a transmitter and receive the associated reflections as well as radiation from other sources through a receiver. Sensors operating in a dual mode include typically an active

mode for target acquisition and tracking and a passive mode for the terminal phase.

We shall discuss three types of sensors: the infrared (IR) sensors, the millimeter wave (MMW) sensors, and the electro-optical (EO) sensors.

i. IR Sensors

IR sensors capture the radiant energy emitted by heated objects. In their simplest type, IR sensors operating in the passive mode, scan optically the target area for IR radiation in a single waveband by all bodies. A more complex design involves IR sensors that detect target signature in two different wavebands, thus allowing discrimination between a true target (e.g. tank) and a decoy (e.g. flare). Finally, for high angular resolution for target detection and tracking, imaging infrared (IIR) resulting in image like properties of the target, may be used.

ii. MMW Sensors

MMW sensors capture the radiant energy emitted by the reflection of metal objects.

iii. EO Sensors

EO sensors typically integrate optics and lasers and employ advanced forward-looking infrared (FLIR) sensors and image-processing computers for

- automatic target recognition;
- intelligent tracking;
- prioritization of multiple targets;
- sensor input integration.

They contain a laser designator for directing laser-guided weapons and for helping see at night where smoke, dust, haze and smog are present.

Descriptive Parameters of the Submunition

- Number of submunitions;
- Average time to search and detect a target;
- Probability of acquiring a target given that it is detected.
- Probability of acquiring a false target.

6. Remarks

i. The use of multi-mode (active and passive) and multispectral (MMW and IR) systems present several advantages such as:

- provide greater accuracy in target hit;
- reduce false alarms;
- improve target detectability and acquisition;
- defeat enemy countermeasures.

ii. The final disposition of the submunition is a critical one particularly if it is not paired to a target and its explosive is not activated. In such a case, if the submunition is designed to self-explode at a given altitude, it can create a source of heat which could capture other incoming live submunitions. If it is not designed to self-explode or if the self-destruct mechanism fails to be activated, it is liable to fall into enemy hands and thus be technologically accessible for the development of countermeasures.

7. The Target

Primary targets are ground mobile targets including heavy armors (tanks) as well as softer vehicles ranging from air defense targets to light armor and rail transport.

The distinguishing feature of smart munitions from other types of anti-armor weapons is that they home on their targets and/or are activated by them. They also attack targets at their most vulnerable point namely the top. Typically, a large number of smart munitions will be needed to insure defeat of a massive armored assault consisting of many targets. A barrage of thousands of these munitions would blunt armored assault and reinforcing columns.

Descriptive Parameters

- Total number of targets (true and decoys);
- Proportion of decoys to total number of targets;
- Probability of target being hit;
- Probability of target being killed given that it is hit.

IV. THE MATHEMATICAL MODEL

1. The Problem

The development of a mathematical model depends on several factors governing the actual conditions under which the weapon system operates. This may include for example, environment, combat scenario, operation

sequence, mode of weapon delivery, weapon technology, etc... In general, a model should be able to capture the uncertainty element present in an actual combat situation together with the evolution of the combat state at successive time epochs. Very often the objective is not on simply winning a battle, but on how quickly to win a battle. From that point of view, time becomes an important parameter to be incorporated in the model.

To illustrate the methodology, we consider a situation in which an aircraft releases a single dispenser from a standoff location just before penetrating enemy lines. The dispenser carries M submunitions to be released over an area A containing N targets (M > N).

2. Assumptions

We assume that:

a. the dispenser carries M submunitions to be released over an area containing N targets;

b. the number of targets is reduced only by the number of targets killed by the submunitions. No other weapons are fired against these targets and the targets are considered to remain within the scanned area;

c. if a submunition does not acquire a target, it searches for another target;

d. once a submunition acquires a target, it is locked onto the target to be shot and killed;

e. no two or more submunitions may acquire the same target;

f. each submunition acts independently of any other submunition;

g. the submunition sensors are not subject to threats;

h. a submunition can kill only one target;

i. a submunition may kill only the target that it acquires. That means that a target cannot be killed "by mistake";

j. a submunition may not acquire a false or dead target;

k. the dispenser releases the M submunitions in salvo;

If any of the above assumptions are changed, the mathematical model will have to be modified accordingly. The present model may be viewed as a prototype which may be used to construct other model variants.

The stochastic aspect of the problem is characterized by $P(M-m, N-n, t)$, the probability that at time t following release from the dispenser, there are exactly M-m remaining live submunitions ($m = 0, 1, \ldots, M$) and N-n remaining live targets ($n = 0, 1, \ldots, N$). One can obtain $P(M-m, N-n, t)$ by developing appropriate differential - difference equations subject to a set of initial conditions. One way of obtaining the solution of these equations is through a recursive approach. Once $P(M-m, N-n, t)$ is derived, one can obtain such characteristics as the expected number of targets killed, the expected number of submunitions to kill a given number of targets, the mission reliability, as well as other measures of effectiveness.

2. The Symbols

$\lambda dt + o(dt)$ - probability that the dispenser will release the submunitions in salvo in the time interval $(t, t + dt)$;

$\nu dt + o(dt)$ - probability that the dispenser will encounter an enemy threat in the time interval $(t, t + dt)$;

p_1 - probability that the dispenser will be killed given that it encounters an enemy threat;

$\omega dt + o(dt)$ - probability that the dispenser will encounter enemy countermeasures (e.g. jamming) in the time interval $(t, t + dt)$;

p_2 - probability that the dispenser will be neutralized by countermeasures and will not be able to accomplish its mission;

$P(i,t)$ - probability that at time t, the dispenser is in a state i, $i = 0, 1$. State $i = 0$ corresponds to the state "dispenser killed"; state $i = 1$ corresponds to the state "dispenser not killed";

τ - time at which dispenser releases the submunitions, measured from time origin at which dispenser is ejected;

M - number of submunitions released in salvo by the dispenser;

N - total number of targets in footprint area (includes decoys);

N_1 = total number of decoys in footprint area;

$\mu dt + o(dt)$ = probability that a submunition will detect a target in the time interval $(t, t + dt)$; (τ is time origin);

p_A = probability that a submunition will acquire a target once detected;

p_K = probability that a submunition will kill a target once acquired;

$P(M-m, N-n, t)$ = probability that at time t, there are M-m remaining live submunitions and n remaining targets, $m = 0, 1, \ldots, M$ and $n = 0, 1, \ldots, N$.

$E[\cdot]$ = the expectation operator.

3. The Model

i. The Dispenser

We have

$$P(1, t + dt) = P(1, t)(1 - \nu dt)(1 - \omega dt) + P(1, t)\nu dt\,(1 - \omega dt)\,(1 - p_1) + P(1, t)\,\omega dt\,(1 - \nu dt)(1 - p_2) + o(dt)$$

$$\text{or } \frac{dP(1,t)}{dt} = -\,(\nu p_1 + \omega p_2)\,P(1,t)$$

subject to the initial condition $P(1, 0) = 1$

This yields $P(1, t) = e^{-(\nu p_1 + \omega p_2)t}$

The probability that the submunitions are released in the time interval $(\tau, \tau + d\tau)$ following dispenser ejection at time origin is:

$$f(\tau)d\tau = P(1, \tau)e^{-\lambda\tau}\,\lambda d\tau$$

$$= e^{-(\lambda + \nu p_1 + \omega p_2)\tau}\,\lambda d\tau \qquad (1)$$

ii. The Submunitions

In general the total number M of submunitions exceeds the total number N of targets. Let $P(M-m, N-n, t)$ denote the probability that at time t there are (M-m) remaining live submunitions and (N-n) remaining

live targets (n targets killed). Note that always $m \geq n$. For $t = 0$, $P(M, N, 0) = 1$.

Now for $0 \leq m \leq M$ and $0 \leq n \leq N$, $n \leq m$

$$P(M-m, N-n, t+dt) =$$

$$= [1 - (M-m)(N-n)\ \mu p_A\ dt]\ P(M-m, N-n, t)$$

$$+ (M-m+1)(N-n+1)\ \mu p_A p_k\ dt\ P(M-m+1, N-n+1, t)$$

$$+ (M-m+1)(N-n)\ \mu p_A\ (1-p_k)\ dt\ P(M-m+1, N-n, t)$$

By expanding the first term, bringing $P(M-m, N-n, t)$ to the left side, dividing by dt and then taking the limit as $dt \to 0$ we obtain:

$$\frac{d\,P(M-m, N-n, t)}{dt} = -\,(M-m)(N-n)\ \mu p_A\ P(M-m, N-n, t)$$

$$+ (M-m+1)(N-n+1)\ \mu p_A p_k\ P\ (M-m+1, N-n+1, t)$$

$$+ (M-m+1)(N-n)\ \mu p_A\ (1-p_k)\ P(M-m+1, N-n, t)$$

for $m = 0, 1, \ldots, M$, $n = 0, 1, \ldots, N$

The solution to this system of differential-difference equations is

$$P(M-m, N-n, t) = \frac{M!}{(M-m)!}\ \frac{N!}{(N-n)!}\ \frac{1}{(m-n)!}\ p_k^{\,n}\ (1-p_k)^{m-n}$$

$$\sum_{i=0}^{m-n} (-1)^{(m-n-i)}\ \frac{(m-n)!}{(m-n-i)!\,i!}\ \sum_{j=0}^{n}\ \frac{e^{-(M-i-j)(N-j)\ \mu p_A t}}{\prod_{\substack{\ell=0 \\ \ell \neq j}}^{n} [(M-i-\ell)(N-\ell)-(M-i-j)(N-j)]}$$

Now as $t \to \infty$, we have

For $0 \leq n < m < M$ and $n < N$

$$\lim_{t\to\infty} P(M-m, N-n, t) = 0$$

Also that for $0 \leq n \leq m = M$ and $n \leq N$

$$P_1(n) = \lim_{t\to\infty} P(0, N-n, t)$$

$$= \frac{M!}{(M-n)!n!} p_k^n (1-p_k)^{M-n} = \binom{M}{n} p_k^n (1-p_k)^{M-n}$$

For $n = N \leq m \leq M$

$$P_2(m) = \lim_{t\to\infty} P(M-m, 0, t)$$

$$= p_k^N (1-p_k)^{m-N} \sum_{i=0}^{m-N} (-1)^{(m-N-i)} \binom{M}{i} \binom{M-N-i}{M-m}$$

Expressions for the probability that all targets are killed and the expected number of targets killed are as follows:

1) for $M \geq N$

$$P\{\text{all targets killed}\} = 1 - \sum_{r=0}^{N-1} \binom{M}{r} p_k^r (1-p_k)^{M-r} = P_1(N)$$

$$\text{Expected number of targets killed} = \sum_{n=0}^{N} nP_1(n)$$

2) for $M = N$

$$P\{\text{all targets killed}\} = p_k^M$$

$$\text{Expected number of targets killed} = Mp_k$$

3) for $M < N$

$$P\{\text{all targets killed}\} = 0$$

$$\text{Expected number of targets killed} = Mp_k$$

Steady state results for the probability of kill and the expected number of targets killed are presented in Figures 1 and 2 for $N = 5$, $p_k = 0.90$

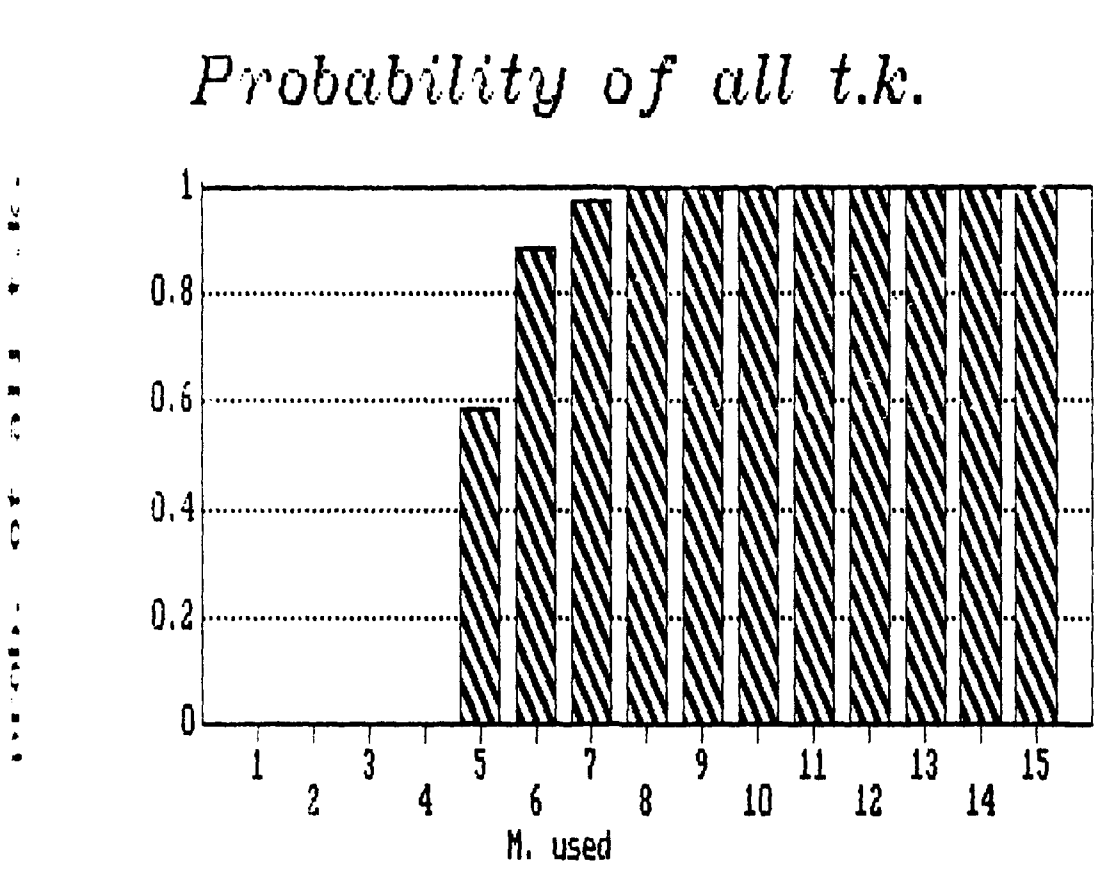


Figure 1: Probability of all targets killed as a function of munitions used (N = 5)

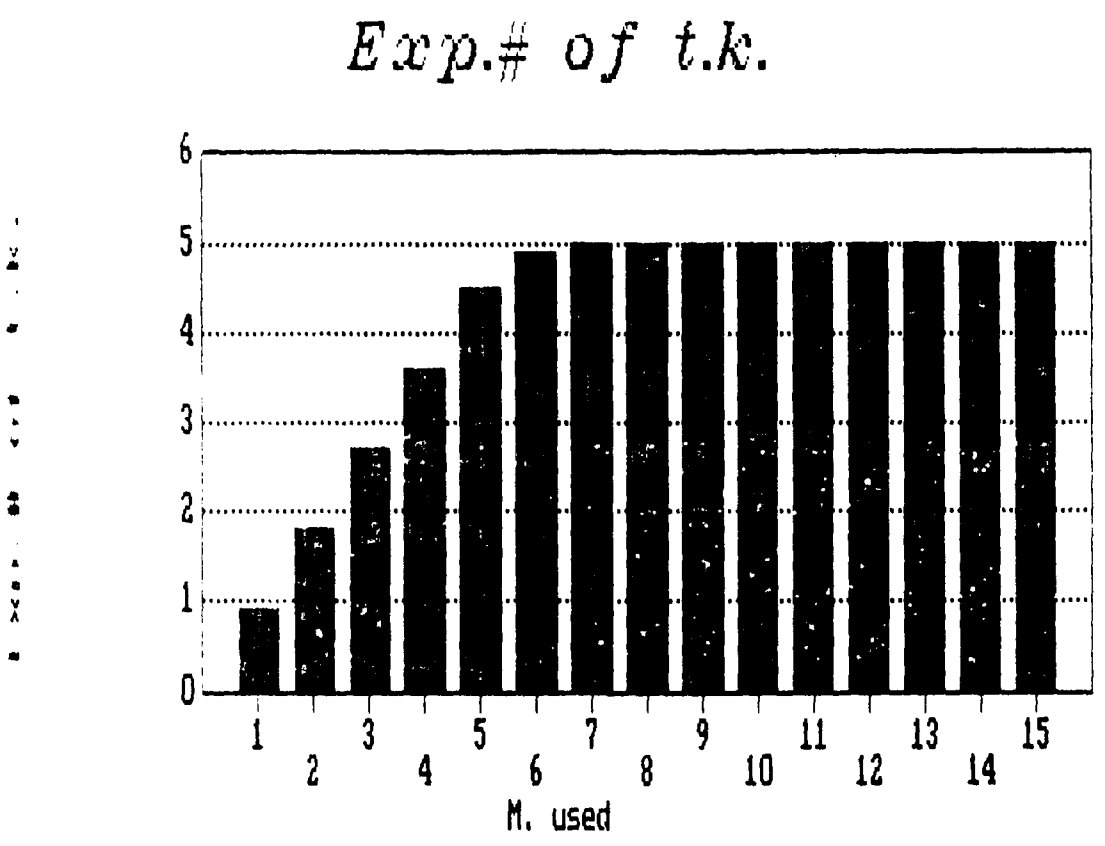


Figure 2: Expected number of targets killed as a function of munitions used (N = 5)

and $M = 1, \ldots, 15$.

V. MEASURES OF EFFECTIVENESS

We develop five measures of effectiveness

1. Expected number of targets killed at time t, $\hat{n}(t)$:

$$E[\hat{n}(t)] = \sum_{m=0}^{M} \sum_{n=0}^{\min(m,N)} n\, P(M-m, N-n, t)$$

2. Expected number of submunitions to kill n targets at time t, $\hat{m}(t)$:

$$E[\hat{m}(t)] = \sum_{m=n}^{M} m\, P(M-m, N-n, t)$$

3. Probability that all targets are killed at time t:

$$\sum_{m=N}^{M} P(M-m, 0, t)$$

4. Mission reliability for the dispenser

This is the probability that the dispenser successfully releases all submunitions. Let

$\hat{P}(1, 0, t)$ = probability that at time t the dispenser is operating successfully and the submunitions are not released;

$\hat{P}(1, 1, t)$ = probability that at time t the dispenser is operating successfully and the submunitions are released.

It is then easy to verify that

$$\frac{d\hat{P}(1, 0, t)}{dt} = -(\mu + \lambda p_1 + \nu p_2)\, \hat{P}(1, 0, t)$$

and $$\frac{d\hat{P}(1, 1, t)}{dt} = \mu\, P(1, 0, t)$$

subject to $\hat{P}(1, 0, 0) = 1$ and $\hat{P}(1, 1, 0) = 0$

From these two equations one obtains for mission reliability of the dispenser

$$\hat{P}(1, 1, t) = \frac{\mu}{\mu + \lambda p_1 + \nu p_2} [1 - e^{-(\mu + \lambda p_1 + \nu p_2)t}]$$

5. Expected duration of the battle

Lt T be the random variable denoting the duration of the battle. The battle will terminate if either all targets are killed or there are no munitions left. Hence

$$P(T > t) = 1 - \sum_{m=N}^{M} P(M-m, N-N, t) - \sum_{n=0}^{N-1} P(M-M, N-n, t)$$

$$E[T] = \int_0^\infty P(T > t)\, dt$$

After some algebraic manipulation, one obtains

$$E[T] = \sum_{m=N}^{M} \frac{M!\, N!}{(M-m)!} p_k^n (1-p_k)^{m-N} \sum_{i=0}^{m-N} \frac{(-1)^{m-N-i}}{(m-N-i)!\, i!}$$

$$\sum_{j=0}^{N-1} \frac{1/(M-i-j)(N-j)\mu p_A}{\prod_{\substack{\ell=0 \\ \ell \neq j}}^{N} [(M-i-\ell)(N-\ell) - (M-i-j)(N-j)]}$$

$$+ \sum_{n=0}^{N-1} \frac{M!\, N!}{(N-n)!} p_k^n (1-p_k)^{M-n} \sum_{i=0}^{M-n-1} \frac{(-1)^{M-n-i}}{(M-n-i)!\, i!}$$

$$\sum_{j=0}^{N-1} \frac{1/(M-i-j)(N-j)\mu p_A}{\prod_{\substack{\ell=0 \\ \ell \neq j}}^{N} [(M-i-\ell)(N-\ell)-(M-i-j)(N-j)]}$$

$$+ \sum_{n=0}^{N-1} \frac{M!\ N!}{(M-n)!(N-n)!} p_k^n (1-p_k)^{M-n}$$

$$\sum_{j=0}^{n-1} \frac{1/(n-j)(N-j)\mu p_A}{\prod_{\ell=0}^{n} [(n-\ell)(N-\ell)-(n-j)(N-j)]}$$

Remark: Case When Decoys Are Present

Let

N = total number of targets including decoys

N_1 = total number of true targets

$N-N_1$ = total number of decoys

$P(r|n)$ = probability that r true targets are killed given that n total targets are killed

Then $$P(r|n) = \frac{\binom{N_1}{r}\binom{N-N_1}{n-r}}{\binom{N}{n}} \quad r = 0, 1, 2, \ldots, n$$

The probability that r true targets are killed, n total targets are killed and m submunitions are used is $[P(r|n).P(M-m, N-n, t)]$. Using this expression one obtains the following

i. Expected number of true targets killed, $\hat{r}(t)$.

$$E[\hat{r}(t)] = \sum_{m=0}^{M} \sum_{n=0}^{\min(m,N)} \sum_{r=0}^{n} r.\ P(r|n)\ P(M-m, N-n, t)$$

ii. Expected number of submunitions used to kill r true targets, $\hat{s}(t)$.

$$E[\hat{s}(t)|r] = \sum_{m=r}^{M} \sum_{n=r}^{\min(m,N)} m\, P(r|n)\, .\, P(M-m,\, N-n,\, t)$$

VI. CONCLUSIONS

The effective use of the smart submunition weapon system depends to a considerable extent on the environment in which it will be operating, whether such environment is natural or man-created. It becomes thus imperative to factor out all the elements which may exist in an actual battlefield situation and which contribute adversely on the performance of these weapons, either by neutralizing or degrading their effectiveness. Among such elements, one has to consider the following:

1. Hostile threats against the carriage aircraft, the dispenser, the submunitions and the supporting C^3I.
2. Adverse weather conditions, particularly rain, snow, sleet and fog.
3. Use of various passive and active countermeasures on the submunition sensors, such as smoke, corner reflectors, IR sources, decoys and camouflage.
4. Use of ECM on the C^3I system supporting the delivery of the submunition. This includes air and/or ground sensors for target area location and target movements, which transmit information through data link to the air platform and/or the dispenser.
5. Use of reactive countermeasures by the enemy particularly as a result of:
   i. the visual acquisition of incoming submunitions delivered by slow moving devices, such as parafoils;
   ii. the activation of target mounted sensors warning of laser, radar and IR acquisition;
   iii. the possibility of acquisition by the enemy in the battlefield of partly damaged dispensers and undetonated submunitions.
6. The potential use of presently unknown types of countermeasures.
7. The acquisition by submunition sensors of false targets.
8. The acquisition by submunition sensors of dead targets.

9. The engagement of one target, single or dead, by several submunitions.

Consideration needs to be given to IR/MMW submunition sensors homing on targets which have been incapacitated by previous attacks. Such a situation can considerably deter the effective engagement and pairing of a given submunition with a given live target.

Second, it is possible that test results may have demonstrated acceptable performance level of weapons under the condition that a single target is present in the test area. However, the main characteristic of a smart submunition weapon system is that it is a many-on-many engagement system. Therefore, tests must be conducted simulating actual combat conditions when several targets are present and when they are being simultaneously engaged by several submunitions.

Finally, when assessing the performance of a smart munition system either mathematically or by simulation, it is essential

1. to study and analyze the aggregate system including the carriage aircraft, the dispenser, the submunitions and the supporting C^3I;
2. to incorporate in the model all pertinent and significant factors that bear on the performance of the system, including environment, threats and countermeasures;
3. to define likely engagement scenarios and to perform the analysis for each scenario in order to assess and compare their performance. These scenarios should be the ones that are the most likely to occur in a realistic combat situation using existing weapon technologies for target acquisition, discrimination and classification.

VII. ACKNOWLEDGEMENTS

The authors wish to thank the Air Force Systems Command, the Air Force Office of Scientific Research, the Air Force Armament Laboratory, and Universal Energy Systems for sponsoring this research supported under Contract No. F49620-88-C-0053.

The support, encouragement and help of Mr. George C. Crews, Mr. Gus Gesselman and Mr. John Gagliano, all from the Technology Assessment

Branch, Analysis and Strategic Defense Division of the Armament Laboratory, is acknowledged.

VIII. REFERENCES

1. Cohen, S., "Don't Think They Can't be Outwitted", Editorial Article, The Wall Street Journal, June 21, 1989.
2. Lee, S. M. (ed), "Proceedings of the Sixth Annual KRC Symposium on Ground Vehicle Signatures", August 21-22, 1984, Keeweenaw Research Center, Houghton, MI.
3. Okorkiewicz, R. M., "Countermeasures for Tanks, Beating Smart Munitions" International Defense Review, Vol. 22, pp. 53-57, Jan. 1989.
4. "Proceedings of the Precision Guided Weapons Symposium" GACIAC PR 88-05, IIT Research Institute, Chicago, IL, (1988).
5. Sivazlian, B. D., "Aircraft Sortie Effectiveness Model", Naval Research Logistics, Vol. 36, pp. 127-137 (1989).
5. "Smart Munitions", GACIAC SR-87-08, IIT Research Institute, Chicago, IL., (1987).

ELECTROMAGNETISM AND GRAVITY

Richard A. Weiss
U. S. Army Engineer Waterways Experiment Station
Vicksburg, Mississippi 39180

ABSTRACT. Space and time coordinates exhibit broken symmetries that are carried over long distances by gravitation. Electromagnetism in space and time with broken internal symmetries has electric and magnetic field vectors and potentials that have internal phase angles. These phase angles enter the calculation of the measurable electric and magnetic fields. Maxwell's equations with broken internal symmetries are solved for simple physical cases and the general form of the internal phase angles of the field vectors are determined. The integral equation development of electromagnetism is formulated in asymmetric spacetime, and the Gauss theorem and Ampere's law with broken symmetry are developed. The elementary calculations of electromagnetism are examined to determine the effects of the broken symmetry of space and time on electrostatics, magnetostatics, electric currents in wires and the propagation of electromagnetic waves in matter. Over long distances the broken symmetry of space and time is due to gravitation so that the elementary calculations and measurements of electromagnetism are expected to depend on gravity. Simple electromagnetic experiments are suggested to determine the internal phase angles of space and time in the vicinity of the earth. The results of this paper will have applications to the theory of the propagation of electromagnetic waves in the neighborhood of the earth, the measurement of the electromagnetic properties of matter, and high temperature superconductivity.

1. INTRODUCTION. Electromagnetism is a gauge field.[1-3] There are also other gauge fields in nature such as gravity and the strong and weak nuclear forces. All of these forces may individually represent various degrees of the broken gauge symmetry of a single unified force.[4] Gravity is described by a metric tensor gauge theory as being the manifestation of the curvature of a four dimensional spacetime which is determined by the mass distribution in space.[5] Electromagnetism is a vector gauge theory whose forces are mediated by a massless spin one photon.[1-4] These apparently distinct forces may be related through gauge field theory.[6]

Before the advent of gauge field theory it was already known that gravitational forces interacted with electromagnetic forces. This was predicted by Einstein's theory of general relativity in the concept of the gravitational red shift of spectral lines from the sun, and in the bending of star light in the gravitational field of the sun.[7-10] The interaction of gravity with electromagnetism is weak and it requires a body as massive as the sun to bend star light sufficiently to be measured. The corresponding effect due to the earth's gravity is negligible and has no practical effects.

This paper considers another way in which gravity affects electromagnetism. This occurs through the broken symmetry of space and time that is associated with

a pressure field in matter.[11,12] Ultimately the pressure field in the solid earth, ocean and atmosphere is due to gravity, and therefore the dominant effect producing the internal phase angles of space and time in the vicinity of the earth is due to gravity. The method of determining the internal phase angles of the space coordinates from the internal phase angle of the pressure has been presented for gravitating stars and planets.[12] It involves the solution of a series of coupled differential equations. Numerical values of the internal phase angle associated with the radial coordinate can be determined experimentally through the apparent deviation from Newtonian gravity in the earth, and from the Pound-Rebka-Snider photon red shift experiment.[12] It has already been shown that the field vectors of the electromagnetic field have a broken symmetry which is represented by internal phase angles.[12]

The internal phase angles of the electric and magnetic field vectors and potentials will be shown in this paper to depend on the internal phase angles of the space and time coordinates. For a relatively weak electromagnetic field in the vicinity of the earth, the internal phase angles of the electromagnetic field vectors will depend on the gravity induced internal phase angles of the space and time coordinates. In this way gravity will have a direct and measurable effect on the character of an electromagnetic field.

The broken symmetry of space and time requires that coordinates be complex numbers of the form for cartesian coordinates[12]

$$\bar{x} = xe^{j\theta x} \qquad \bar{y} = ye^{j\theta y} \qquad \bar{z} = ze^{j\theta z} \tag{1}$$

while for cylindrical coordinates

$$\bar{r} = re^{j\theta r} \qquad \bar{z} = ze^{j\theta z} \qquad \bar{\phi} = \phi e^{j\theta \phi} \tag{2}$$

and for spherical coordinates

$$\bar{r} = re^{j\theta r} \qquad \bar{\psi} = \psi e^{j\theta \psi} \qquad \bar{\phi} = \phi e^{j\theta \phi} \tag{3}$$

The internal phase angles of the coordinates depend on the local energy density and pressure.[12]

The broken symmetry sine and cosine functions are

$$\sin \bar{\psi} = S_{\psi} e^{j\theta s\psi} \tag{4}$$

$$\cos \bar{\psi} = C_{\psi} e^{-j\theta c\psi} \tag{5}$$

where

$$S_{\psi} = [\sin^2(\psi \cos \theta_{\psi}) + \sinh^2(\psi \sin \theta_{\psi})]^{1/2} \tag{6}$$

$$C_{\psi} = [\cos^2(\psi \cos \theta_{\psi}) + \sinh^2(\psi \sin \theta_{\psi})]^{1/2} \tag{7}$$

$$\tan \theta_{s\psi} = \cot(\psi \cos \theta_{\psi}) \tanh(\psi \sin \theta_{\psi}) \tag{8}$$

$$\tan \theta_{c\psi} = \tan(\psi \cos \theta_{\psi}) \tanh(\psi \sin \theta_{\psi}) \tag{9}$$

The following angles are used to describe the variation of the internal phase angles with the magnitudes of the corresponding coordinates

$$\tan \beta_{xx} = x\partial\theta_x/\partial x \tag{10}$$

$$\tan \beta_{yy} = y\partial\theta_y/\partial y \tag{11}$$

$$\tan \beta_{zz} = z\partial\theta_z/\partial z \tag{12}$$

$$\tan \beta_{rr} = r\partial\theta_r/\partial r \tag{13}$$

$$\tan \beta_{\phi\phi} = \phi\partial\theta_{\phi}/\partial\phi \tag{14}$$

$$\tan \beta_{\psi\psi} = \psi\partial\theta_{\psi}/\partial\psi \tag{15}$$

The corresponding measured values of these coordinates are given by

$$x_m = x \cos \theta_x \qquad y_m = y \cos \theta_y \qquad z_m = z \cos \theta_z \tag{16}$$

$$r_m = r \cos \theta_r \qquad z_m = z \cos \theta_z \qquad \phi_m = \phi \cos \theta_{\phi} \tag{17}$$

$$r_m = r \cos \theta_r \qquad \psi_m = \psi \cos \theta_{\psi} \qquad \phi_m = \phi \cos \theta_{\phi} \tag{18}$$

From these equations it follows immediately that

$$\partial x_m/\partial x = \cos \theta_x - x_m \tan \theta_x \, \partial\theta_x/\partial x \tag{19}$$

$$\partial y_m/\partial y = \cos \theta_y - y_m \tan \theta_y \, \partial\theta_y/\partial y \tag{20}$$

$$\partial z_m/\partial z = \cos \theta_z - z_m \tan \theta_z \, \partial\theta_z/\partial z \tag{21}$$

$$\partial r_m/\partial r = \cos \theta_r - r_m \tan \theta_r \, \partial\theta_r/\partial r \tag{22}$$

$$\partial \psi_m/\partial\psi = \cos \theta_{\psi} - \psi_m \tan \theta_{\psi} \, \partial\theta_{\psi}/\partial\psi \tag{23}$$

$$\partial \phi_m/\partial\phi = \cos \theta_{\phi} - \phi_m \tan \theta_{\phi} \, \partial\theta_{\phi}/\partial\phi \tag{24}$$

The skewed nature of the coordinates is due to the skewed nature of the pressure[12]

$$\bar{P} = Pe^{j\theta_P} \tag{25}$$

Whenever an internal phase angle appears in the electromagnetic calculations of this paper it represents mainly the effects of gravitationally induced broken symmetry associated with the pressure and spacetime coordinates. The measured

value of the pressure is given by

$$P_m = P \cos \theta_P \tag{26}$$

Time also has a broken symmetry represented by an internal phase angle, so that time must be written as a complex number as follows

$$\bar{t} = te^{j\theta_t} \tag{27}$$

The internal phase angle of time depends on the local energy density and pressure $\theta_t = \theta_t(\bar{E},P)$. The components of particle velocity can then be written as[12]

$$\bar{v}_\alpha = v_\alpha e^{j\theta v\alpha} = d\bar{\alpha}/d\bar{t} \tag{28}$$

where v_α and $\theta_{v\alpha}$ are calculated in Reference 12 for $\alpha = x, y, z$.

This paper determines the electric and magnetic field vectors for space and time that has broken internal symmetries. The vacuum has intrinsically broken space and time coordinate symmetries.[12] Thus the electromagnetic field has intrinsically broken symmetry associated with the field vectors.[12] But in the presence of matter it is predominantly gravity (with its infinite range) that determines the broken symmetry of the coordinates and the electromagnetic field vectors. Section 2 treats asymmetric gravity, Section 3 considers Maxwell's equations with broken internal symmetries, Section 4 deals with broken symmetry electrostatics, Section 5 studies asymmetric magnetostatics, and Section 6 treats electromagnetic waves in a gravitational field.

2. ASYMMETRIC GRAVITY. The force of gravity for radially symmetric stars or planets composed of matter with broken internal symmetries is described by the following two equations[12]

$$-2\theta_r^G + \pi = \theta_P + \beta_{Pr} \tag{29}$$

$$\cos \beta_{rr}^G \; 1/r^2 \; \partial/\partial r(r^2/\rho \; \partial P/\partial r \cos \beta_{rr}^G \sec \beta_{Pr}) = -4\pi G\rho \tag{30}$$

where

$$\tan \beta_{Pr} = (P\partial\theta_p/\partial r)/(\partial P/\partial r) \tag{31}$$

$$\tan \beta_{rr}^G = r\partial\theta_r^G/\partial r \tag{32}$$

where ρ = density magnitude (not the measured density) and G = Newtonian gravitation constant. It is easy to show that equations (29) and (30) are equivalent to[12]

$$\cos \beta_{rr} \; \partial\bar{P}/\partial r = -GM\rho/\bar{r}^2 \tag{33}$$

$$\cos \beta_{rr} \; \partial M/\partial r = 4\pi r^2\rho \tag{34}$$

Thus θ_r^G is related to the pressure field in matter which over macroscopic distances is produced by gravity. In general $\theta_r^G = \theta_r^G(P)$. For geometrically asymmetric planets, such as the earth, the internal phase angles θ_ϕ and θ_ψ also enter the calculation of the gravitational field.[12] Note that β_{Pr} is in the third quadrant and can be written as $\beta_{Pr} = \pi + \beta'_{Pr}$ where β'_{Pr} is in the first quadrant, then equation (29) can be written as

$$- 2\theta_r^G = \theta_P + \beta'_{Pr} \tag{35}$$

$$\tan \beta_{Pr} = \tan \beta'_{Pr} \tag{35A}$$

For small angles equations (31) and (35) give

$$- 2\theta_r^G = \theta_P + (P\partial\theta_p/\partial r)/(\partial P/\partial r) \tag{36}$$

This is the value of θ_r that is associated with a gravitating body. For a particle falling, or in a circular orbit, in a gravitational field the acceleration phase angle condition $\theta_r^G - 2\theta_t^G \sim - 2\theta_r^G$ gives

$$\theta_t^G \sim 3/2\ \theta_r^G \tag{36A}$$

$$= - 3/4\ [\theta_P + (P\partial\theta_p/\partial r)/(\partial P/\partial r)]$$

For an elliptical orbit the situation is more complicated.[12]

In general the pressure at a point in matter is due to other forces in addition to gravitation such as the strong and weak nuclear forces and electromagnetism. The total radial coordinate internal phase angle can be written as

$$\theta_r = \theta_r^G + \theta_r^{SN} + \theta_r^{WN} + \theta_r^{EM} + \theta_r^v \tag{37}$$

where θ_r , θ_r^G , θ_r^{SN} , θ_r^{WN} , θ_r^{EM} and θ_r^v = total, gravitational, strong nuclear, weak nuclear, electromagnetic and vacuum values respectively of the internal phase angle of the radial coordinate. Over long distances in matter one has approximately

$$\theta_r \sim \theta_r^G + \theta_r^{EMR} + \theta_r^v \tag{38}$$

where θ_r^{EMR} = internal phase angle for electromagnetic radiation. Static electric fields are shielded in matter and rapidly attenuate with distance. For matter in a star or planet where gravitation is the dominant force equation (38) becomes

$$\theta_r \sim \theta_r^G \tag{39}$$

In tenuous isolated matter one expects

$$\theta_r \sim \theta_r^{EMR} + \theta_r^{v} \tag{40}$$

In empty space $\theta_r = \theta_r^{v}$. The approximation in equation (39) is expected to be valid in the vicinity of the earth. It should be mentioned that rest mass is considered to be a scalar having zero internal phase angle.[12] For the internal phase angle of time the same general argument gives

$$\theta_t = \theta_t^{G} + \theta_t^{SN} + \theta_t^{WN} + \theta_t^{EM} + \theta_t^{v} \tag{41}$$

Similar results hold for the internal phase angles of the angular coordinates.

A. Density and Mass

The definition of the mass density in broken symmetry space for spherical polar coordinates described by equation (3) is[12]

$$\bar{\rho} = d^3M/(\bar{r}^2 \sin\bar{\psi}\, d\bar{\psi}\, d\bar{\phi}\, d\bar{r}) \tag{42}$$

$$\rho = \cos\beta_{rr} \cos\beta_{\phi\phi} \cos\beta_{\psi\psi}\, d^3M/(S_\psi r^2\, d\psi\, d\phi\, dr) \tag{43}$$

$$\theta_\rho = -3\theta_r - \theta_{s\psi} - \theta_\psi - \theta_\phi - \beta_{rr} - \beta_{\psi\psi} - \beta_{\phi\phi} \tag{44}$$

where S_ψ and $\theta_{S\psi}$ are given by equations (6) and (8) respectively. For spherical symmetry

$$\bar{\rho}^1 = (4\pi\bar{r}^2\bar{L}_s)^{-1} dM/d\bar{r} \tag{45}$$

$$\rho^1 = \cos\beta_{rr}\, (4\pi r^2 L_s)^{-1} dM/dr \tag{46}$$

$$\theta_\rho^1 = -3\theta_r - \theta_{Ls} - \beta_{rr} \tag{47}$$

where[12]

$$\bar{L}_s = L_s e^{j\theta Ls} = 1/2 e^{j\langle\theta_\phi\rangle}[1 - \cos(\pi e^{j\langle\theta_\psi\rangle})] \tag{48}$$

is the complex number angular factor for spherical geometry defined by

$$\bar{L}_s = 1/(4\pi)\oint \sin\bar{\psi}\, d\bar{\psi}\, d\bar{\phi} \tag{49}$$

The average values of internal phase angles are given by $\langle\theta_\psi\rangle = \theta_\psi(\pi)$ and $\langle\theta_\phi\rangle = \theta_\phi(2\pi)$. The real and imaginary parts of equation (48) determine L_s and θ_{LS} . For applications of the divergence form of Gauss's law (subsection B) it is necessary to have a density independent of the angular factor $\bar{L}_s$ because this factor cancels in the definition of the divergence. Therefore an often used definition of complex number density for spherical symmetry will be

$$\bar{\rho} = (4\pi\bar{r}^2)^{-1} dM/d\bar{r} \tag{50}$$

$$\rho = \cos \beta_{rr} \, (4\pi r^2)^{-1} dM/dr \tag{51}$$

$$\theta_\rho = -3\theta_r - \beta_{rr} \tag{52}$$

The measured mass density is given by

$$\rho_m = \rho \cos \theta_\rho = f_{3r} \rho_c \tag{53}$$

where the conventionally calculated mass density is given by

$$\rho_c = (4\pi r_m^2)^{-1} dM/dr_m \tag{54}$$

and combining equations (51) and (53) gives

$$f_{3r} = \cos \beta_{rr} \cos(3\theta_r + \beta_{rr}) \cos^2 \theta_r \, dr_m/dr \tag{55}$$

where dr_m/dr is given by equation (22). The mass is calculated as follows

$$M = 4\pi \int \rho \sec \beta_{rr} \, r^2 dr = 4\pi \int \rho_c r_m^2 dr_m \tag{56}$$

$$= 4\pi \int \rho_m / f_{3r} r_m^2 dr_m$$

where ρ is given by equation (51).

For a cylindrical mass distribution the complex number density is given by

$$\bar{\rho} = \rho e^{j\theta\rho} = d^3M/(\bar{r} \, d\bar{\phi} \, d\bar{r} \, d\bar{z}) \tag{57}$$

$$\rho = \cos \beta_{\phi\phi} \cos \beta_{rr} \cos \beta_{zz} \, d^3M/(r \, d\phi \, dr \, dz) \tag{58}$$

$$\theta_\rho = -2\theta_r - \theta_\phi - \theta_z - \beta_{rr} - \beta_{\phi\phi} - \beta_{zz} \tag{59}$$

For cylindrical symmetry

$$\bar{L}_c = L_c e^{j\theta Lc} = 1/(2\pi) \oint d\bar{\phi}$$

so that

$$\bar{L}_c = e^{j\langle\theta_\phi\rangle} \qquad L_c = 1 \qquad \theta_{Lc} = \langle\theta_\phi\rangle \tag{61}$$

where the average value $\langle\theta_\phi\rangle = \theta_\phi(2\pi)$. Then

$$\bar{\rho}^1 = (2\pi\bar{r}\bar{L}_c)^{-1} d^2M/(d\bar{r}\, d\bar{z}) \tag{62}$$

$$\rho^1 = \cos\beta_{rr} \cos\beta_{zz} /(2\pi r) d^2M/(dr\, dz) \tag{63}$$

$$\theta_\rho^1 = -2\theta_r - \langle\theta_\phi\rangle - \theta_z - \beta_{rr} - \beta_{zz} \tag{64}$$

As will be described in subsection B, the density that appears in the divergence form of Gauss's law for axial symmetry does not contain the internal phase factor $\langle\theta_\phi\rangle$ and for this usage the following density is used for a mass distribution with cylindrical symmetry

$$\bar{\rho} = 1/(2\pi\bar{r}) d^2M/(d\bar{r}\, d\bar{z}) \tag{65}$$

$$= 1/(2\pi\bar{r}) d\bar{\rho}_\ell/d\bar{r} \tag{66}$$

$$\rho = \cos\beta_{rr} \cos\beta_{zz} /(2\pi r) d^2M/(dr\, dz) \tag{67}$$

$$\theta_\rho = -2\theta_r - \theta_z - \beta_{rr} - \beta_{zz} \tag{67}$$

where the mass per unit length is given by

$$\bar{\rho}_\ell = dM/d\bar{z} \tag{68}$$

$$\rho_\ell = \cos\beta_{zz}\, dM/dz \qquad \theta_{\rho\ell} = -\theta_z - \beta_{zz} \tag{69}$$

$$\rho_{\ell m} = \rho_\ell \cos\theta_{\rho\ell} \tag{70}$$

The measured mass density is given by

$$\rho_m = \rho \cos\theta_\rho = f_{2r}\rho_c \tag{71}$$

where the conventionally calculated mass density is

$$\rho_c = (2\pi r_m)^{-1} d^2M/(dr_m\, dz_m) \tag{72}$$

$$= (2\pi r_m)^{-1} d\rho_{\ell c}/dr_m$$

where

$$\rho_{\ell c} = dM/dz_m \tag{73}$$

and where combining equations (66) and (71) gives

$$f_{2r} = \cos\beta_{rr} \cos\beta_{zz} \cos\theta_r \cos\theta_\rho\, dz_m/dz\, dr_m/dr \tag{74}$$

The mass is obtained from equation (66), (71) and (72) as

$$M = 2\pi\int\int \rho \sec \beta_{rr} \sec \beta_{zz} \, dr \, dz \tag{75}$$

$$= 2\pi\int\int \rho_c r_m \, dr_m \, dz_m$$

$$= 2\pi\int\int \rho_m / f_{2r} r_m \, dr_m \, dz_m$$

For cartesian coordinates the mass density is given by[12]

$$\bar{\rho} = d^3M/(d\bar{x}\, d\bar{y}\, d\bar{z}) \tag{76}$$

$$\rho = \cos \beta_{xx} \cos \beta_{yy} \cos \beta_{zz} \, d^3M/(dx\, dy\, dz) \tag{77}$$

$$\theta_\rho = -\theta_x - \theta_y - \theta_z - \beta_{xx} - \beta_{yy} - \beta_{zz} \tag{78}$$

where $\beta_{\alpha\alpha}$ is given by equations (10) through (12). The measured mass density is

$$\rho_m = \rho \cos \theta_\rho = f_{3xyz} \rho_c \tag{79}$$

where the conventionally calculated mass density is

$$\rho_c = d^3M/(dx_m\, dy_m\, dz_m) \tag{80}$$

and where

$$f_{3xyz} = \cos \theta_\rho \cos \beta_{xx} \cos \beta_{yy} \cos \beta_{zz} \, dx_m/dx \, dy_m/dy \, dz_m/dz \tag{81}$$

where $d\alpha_m/d\alpha$ is given by equations (19) through (21), and where

$$(\rho_c - \rho_m)/\rho_c = 1 - f_{3xyz} \tag{82}$$

Finally, the mass is given by

$$M = \int \rho \sec \beta_{xx} \sec \beta_{yy} \sec \beta_{zz} \, dx \, dy \, dz \tag{83}$$

$$= \int \rho_c \, dx_m \, dy_m \, dz_m$$

$$= \int \rho_m / f_{3xyz} \, dx_m \, dy_m \, dz_m$$

B. Newton's Gravitation Law for Asymmetric Matter

The gravity field of radially symmetric spherical mass composed of matter with broken internal symmetry can be obtained from Gauss's law and the diver-

gence theorem which is given by the following broken symmetry generalization of the standard result[13]

$$\bar{\vec{\nabla}} \cdot \bar{\vec{F}} = \bar{\nabla}_r \bar{F}_r = - \bar{\rho} G \tag{84}$$

where $\bar{F}_r$ = broken symmetry gravity force on a unit mass = gravitational acceleration, and $\bar{\rho}$ = complex number mass density as given in equations (50), (65) or (76). These densities do not contain $<\theta_\phi>$ and $<\theta_\psi>$ because the factors $\bar{L}_s$ and $\bar{L}_c$ given by equations (48) and (61) respectively drop out of the calculation of the divergence and so must not appear in the right hand side of equation (84). For a system with spherical symmetry equation (84) becomes the following generalization of the standard scalar result[13-17]

$$1/\bar{r}^2 \; d/d\bar{r}(\bar{r}^2 \bar{F}_r) = - \bar{\rho} G \tag{85}$$

Combining equations (50) and (85) gives

$$\bar{F}_r = F_r e^{j\theta_{Fr}} = - GM/(4\pi\bar{r}^2) = - GM/(4\pi r^2) e^{-2j\theta_r} \tag{86}$$

$$F_r = - GM/(4\pi r^2) \qquad \theta_{Fr} = - 2\theta_r \tag{87}$$

The broken symmetry gravity potential is given by

$$\bar{W}_r = W_r e^{j\theta_{Wr}} = - GM/(4\pi\bar{r}) = - GM/(4\pi r) e^{-j\theta_r} \tag{88}$$

$$W_r = - GM/(4\pi r) \qquad \theta_{Wr} = - \theta_r \tag{89}$$

The measured gravity force and potential for spherical symmetry is given by[12]

$$F_{rm} = F_r \cos \theta_{Fr} = - GM/(4\pi r^2) \cos(2\theta_r) \tag{90}$$

$$= - GM/(4\pi r_m^2) \cos^2 \theta_r \cos(2\theta_r)$$

$$W_{rm} = W_r \cos \theta_{Wr} = - GM/(4\pi r) \cos \theta_r \tag{91}$$

$$= - GM/(4\pi r_m) \cos^2 \theta_r$$

where θ_r is generally a function of the radial coordinate magnitude.

For the two dimensional axisymmetric case the following is the broken symmetry generalization of the standard scalar result[13-17]

$$1/\bar{r} \; d/d\bar{r}(\bar{r}\bar{F}_r) = - G\bar{\rho} \tag{92}$$

Using equations (65) and (92) gives

$$\bar{F}_r = F_r e^{j\theta_{Fr}} = - G\bar{\rho}_\ell / (2\pi\bar{r}) \tag{93}$$

$$F_r = - G\rho_\ell/(2\pi r) \tag{94}$$

$$\theta_{Fr} = - \theta_r - \theta_z - \beta_{zz} \tag{95}$$

$$\bar{W}_r = W_r e^{j\theta_{Wr}} = G/(2\pi)\bar{\rho}_\ell \; \ell n \; \bar{r} = G/(2\pi)\bar{\rho}_\ell(\ell n \; r + j\theta_r) \tag{96}$$

$$W_r = G/(2\pi)\rho_\ell (\ell n^2 \; r + \theta_r^2)^{1/2} \tag{97}$$

$$\theta_{Wr} = \theta_{\rho\ell} + \tan^{-1}(\theta_r/\ell n \; r) \tag{98}$$

where $\bar{\rho}_\ell$, ρ_ℓ and $\theta_{\rho\ell}$ are given by equations (68) and (69). The measured gravity force is given by

$$F_{rm} = F_r \cos \theta_{Fr} = - G\rho_\ell/(2\pi r)\cos(\theta_r + \theta_z + \beta_{zz}) \tag{99}$$

$$= - G\rho_{\ell m}/(2\pi r_m)\cos \theta_r \cos(\theta_r + \theta_z + \beta_{zz})/\cos(\theta_z + \beta_{zz})$$

The measured gravity potential is given by

$$W_{rm} = W_r \cos \theta_{Wr} \tag{100}$$

where W_r and θ_{Wr} are given by equations (97) and (98) respectively.

The case of constant density (ρ = constant and θ_ρ = constant) is easily treated and gives the following generalization of the standard scalar result for a sphere of radius R

$$\bar{W}_r = \rho/6(3R^2 - r^2)e^{-j\theta_r} \qquad r \leqslant R \tag{101}$$

$$= \rho/6(3R^2 - \bar{r}^2 e^{-2j\theta_r})e^{-j\theta_r}$$

where $\theta_r = \theta_R$ = constant. The gravitational force is then given by

$$\bar{F}_r = - \partial\bar{W}_r/\partial\bar{r} = - 1/3\rho r e^{-2j\theta_r} \tag{102}$$

Note that for constant density $\bar{\rho}$, equations (50) through (52) give ρ = constant, θ_r = constant, $\beta_{rr} = 0$ and

$$M(r) = 4\pi/3r^3\rho \qquad M(R) = 4\pi/3R^3\rho \qquad r \leqslant R \tag{103}$$

$$\bar{F}_r = - M(R)r/(4\pi R^3)e^{-2j\theta_r} \qquad r \leqslant R \tag{104}$$

$$F_r = - 1/3\rho r \qquad r \leqslant R \tag{105}$$

$$\theta_{Fr} = - 2\theta_r = - 2\theta_r(R) = - 2\theta_r(0) \qquad r \leqslant R \tag{106}$$

$$F_{rm} = F_r \cos\theta_{Fr} = -\rho r/3 \cos(2\theta_r) \qquad r \leq R \qquad (107)$$

$$= -1/3\rho_m r_m \cos(2\theta_r)/[\cos\theta_r \cos(3\theta_r)]$$

$$\rho_m = \rho \cos(\theta_\rho) = \rho \cos(3\theta_r) \qquad r \leq R \qquad (108)$$

C. Gauss Law for Broken Symmetry Matter

The form of Gauss' law for matter with broken internal symmetry is obtained from equation (84) to be[12]

$$\oint \vec{\bar{F}} \cdot \vec{\bar{n}} d\bar{A} = -MG\bar{L} \qquad (109)$$

where

$$\bar{L} = \bar{L}_s \text{ for spherical coordinates} \qquad (110)$$

$$\bar{L} = \bar{L}_c \text{ for cylindrical coordinates} \qquad (111)$$

where $\bar{L}_s$ and $\bar{L}_c$ are given respectively by equations (48) and (61). For broken symmetry space a factor $\bar{L}(\theta_\phi,\theta_\psi)$ must be included in the right hand side of the Gauss law given by equation (109).

3. MAXWELL'S EQUATIONS. Maxwell's equations are a set of vector differential equations that characterize the electromagnetic field.[14-28] For broken symmetry space and time the coordinates, differential operators and the field vectors have internal phase angles, and Maxwell's equations must be written as[12]

$$\vec{\bar{\nabla}} \times \vec{\bar{E}} = -\partial\vec{\bar{B}}/\partial\bar{t} \qquad (112)$$

$$\vec{\bar{\nabla}} \times \vec{\bar{H}} = \vec{\bar{j}} + \partial\vec{\bar{D}}/\partial\bar{t} \qquad (113)$$

$$\vec{\bar{\nabla}} \cdot \vec{\bar{B}} = 0 \qquad (114)$$

$$\vec{\bar{\nabla}} \cdot \vec{\bar{D}} = \bar{\rho}_q \qquad (115)$$

where $\vec{\bar{E}}$ = renormalized complex number electric field vector, $\vec{\bar{B}}$ = renormalized complex number magnetic induction vector, $\vec{\bar{H}}$ = renormalized complex number magnetic field vector, $\vec{\bar{j}}$ = renormalized complex number current density vector, $\vec{\bar{D}}$ = renormalized complex number electric displacement vector, and $\bar{\rho}_q$ = complex number electric charge density. The components of the electric and magnetic fields can be written as

$$\bar{E}_\alpha = E_\alpha e^{j\theta E\alpha} \qquad \theta_{E\alpha} = \theta_\alpha - 2\theta_t + 2(\beta_{\alpha\alpha} - \beta_{tt}) \qquad (116)$$

$$\bar{D}_\alpha = D_\alpha e^{j\theta D\alpha} \qquad \theta_{D\alpha} = \theta_\varepsilon + \theta_{E\alpha} \qquad \theta_\varepsilon = 2\theta_t - \theta_x - \theta_y - \theta_z \qquad (117)$$

$$= 2\theta_t - 3\theta_r$$

$$\bar{H}_\alpha = H_\alpha e^{j\theta H\alpha} \qquad \theta_{H\alpha} = -\theta_t - \beta_{tt} - \theta_\delta \tag{118}$$

$$\bar{B}_\alpha = B_\alpha e^{j\theta B\alpha} \qquad \theta_{B\alpha} = \theta_\mu + \theta_{H\alpha} \qquad \theta_\mu = \theta_r = (\theta_x + \theta_y + \theta_z)/3 \tag{119}$$

where α and δ = x, y, z or r , ϕ , ψ and δ may or may not be equal to α . The phase angles in equations (116) and (118) are the simplest forms. In fact $\theta_{H\alpha}$ and $\theta_{E\alpha}$ are mutually involved because they are determined from equations (112), and (113). For instance, this is shown in equations (448), (452), (458) and (462) for electromagnetic waves. The measured field vectors are

$$E_{\alpha m} = E_\alpha \cos \theta_{E\alpha} \qquad D_{\alpha m} = D_\alpha \cos \theta_{D\alpha} \tag{120}$$

$$H_{\alpha m} = H_\alpha \cos \theta_{H\alpha} \qquad B_{\alpha m} = B_\alpha \cos \theta_{B\alpha} \tag{121}$$

A. Measured Coordinates and Field Vectors in Maxwell's Equations.

The component equations corresponding to Maxwell's equations (112) through (115) appear in Reference 12. For instance, the x component equations corresponding to equation (112) are

$$\begin{matrix}\cos \\ \sin\end{matrix} \phi_{Ezy} \cos \beta_{yy} \sec \beta_{Ezy}\, \partial E_z/\partial y - \begin{matrix}\cos \\ \sin\end{matrix} \phi_{Eyz} \cos \beta_{zz} \sec \beta_{Eyz}\, \partial E_y/\partial z \tag{122}$$

$$= - \begin{matrix}\cos \\ \sin\end{matrix} \phi_{Bxt} \cos \beta_{tt} \sec \beta_{Bxt}\, \partial B_x/\partial t$$

with similar equations for the y and z vector components, where

$$\phi_{Ezy} = \theta_{Ez} + \beta_{Ezy} - \theta_y - \beta_{yy} \tag{123}$$

$$\phi_{Eyz} = \theta_{Ey} + \beta_{Eyz} - \theta_z - \beta_{zz} \tag{124}$$

$$\phi_{Bxt} = \theta_{Bx} + \beta_{Bxt} - \theta_t - \beta_{tt} \tag{125}$$

$$\tan \beta_{Ezy} = (E_z\, \partial\theta_{Ez}/\partial y)/(\partial E_z/\partial y) \tag{126}$$

$$\tan \beta_{Eyz} = (E_y\, \partial\theta_{Ey}/\partial z)/(\partial E_y/\partial z) \tag{127}$$

$$\tan \beta_{Bxt} = (B_x\, \partial\theta_{Bx}/\partial t)/(\partial B_x/\partial t) \tag{128}$$

From equations (16) through (24) and equations (120) and (121) it follows that the derivatives in equation (122) can be written in terms of measured quantities by

$$\partial E_z/\partial y = g_{Ezy}\, \partial E_{zm}/\partial y_m + h_{Ezy} E_{zm} \tag{129}$$

$$\partial E_y/\partial z = g_{Eyz}\, \partial E_{ym}/\partial z_m + h_{Eyz} E_{ym} \tag{130}$$

$$\partial B_x/\partial t = g_{Bxt}\ \partial B_{xm}/\partial t_m + h_{Bxt} B_{xm} \tag{131}$$

where

$$g_{Ezy} = \sec \theta_{Ez}\ \partial y_m/\partial y \tag{132}$$

$$g_{Eyz} = \sec \theta_{Ey}\ \partial z_m/\partial z \tag{133}$$

$$g_{Bxt} = \sec \theta_{Bx}\ \partial t_m/\partial t \tag{134}$$

$$h_{Ezy} = \sec \theta_{Ez} \tan \theta_{Ez}\ \partial\theta_{Ez}/\partial y \tag{135}$$

$$h_{Eyz} = \sec \theta_{Ey} \tan \theta_{Ey}\ \partial\theta_{Ey}/\partial z \tag{136}$$

$$h_{Bxt} = \sec \theta_{Bx} \tan \theta_{Bx}\ \partial\theta_{Bx}/\partial t \tag{137}$$

In this way equation (122) can be written in terms of measured quantities

$$(a^x_c \partial/\partial y_m + d^x_c) E_{zm} - (b^x_c \partial/\partial z_m + e^x_c) E_{ym} = -(f^x_c \partial/\partial t_m + k^x_c) B_{xm} \tag{137A}$$

$$(a^x_s \partial/\partial y_m + d^x_s) E_{zm} - (b^x_s \partial/\partial z_m + e^x_s) E_{ym} = -(f^x_s \partial/\partial t_m + k^x_s) B_{xm} \tag{137B}$$

where

$$a^x_{c,s} = \begin{matrix}\cos\\ \sin\end{matrix} \Phi_{Ezy} \cos \beta_{yy} \sec \beta_{Ezy} \sec \theta_{Ez}\ \partial y_m/\partial y \tag{137C}$$

$$b^x_{c,s} = \begin{matrix}\cos\\ \sin\end{matrix} \Phi_{Eyz} \cos \beta_{zz} \sec \beta_{Eyz} \sec \theta_{Ey}\ \partial z_m/\partial z \tag{137D}$$

$$d^x_{c,s} = \begin{matrix}\cos\\ \sin\end{matrix} \Phi_{Ezy} \cos \beta_{yy} \sec \beta_{Ezy} \sec \theta_{Ez} \tan \theta_{Ez}\ \partial\theta_{Ez}/\partial y \tag{137E}$$

$$e^x_{c,s} = \begin{matrix}\cos\\ \sin\end{matrix} \Phi_{Eyz} \cos \beta_{zz} \sec \beta_{Eyz} \sec \theta_{Ey} \tan \theta_{Ey}\ \partial\theta_{Ey}/\partial z \tag{137F}$$

$$f^x_{c,s} = \begin{matrix}\cos\\ \sin\end{matrix} \Phi_{Bxt} \cos \beta_{tt} \sec \beta_{Bxt} \sec \theta_{Bx}\ \partial t_m/\partial t \tag{137G}$$

$$k^x_{c,s} = \begin{matrix}\cos\\ \sin\end{matrix} \Phi_{Bxt} \cos \beta_{tt} \sec \beta_{Bxt} \sec \theta_{Bx} \tan \theta_{Bx}\ \partial\theta_{Bx}/\partial t \tag{137H}$$

Equations (137A) and (137B) represent the x-component of Faraday's induction law equation (112) in terms of measured field components and measured space and time coordinates. The conventional Maxwell equation corresponding to equations (137A) and (137B) is

$$\partial E_{zc}/\partial y_m - \partial E_{yc}/\partial z_m = -\partial B_{xc}/\partial t_m \tag{137I}$$

The remainder of Maxwell's equations are handled in a similar way. For instance, equation (115) becomes the following two equations

$$\begin{matrix}\cos \\ \sin\end{matrix} \Phi_{Dxx} \cos \beta_{xx} \sec \beta_{Dxx} \, \partial D_x/\partial x + \begin{matrix}\cos \\ \sin\end{matrix} \Phi_{Dyy} \cos \beta_{yy} \sec \beta_{Dyy} \, \partial D_y/\partial y \tag{138}$$

$$+ \begin{matrix}\cos \\ \sin\end{matrix} \Phi_{Dzz} \cos \beta_{zz} \sec \beta_{Dzz} \, \partial D_z/\partial z = \rho_q \begin{matrix}\cos \\ \sin\end{matrix} \theta_{\rho q}$$

where

$$\Phi_{Dxx} = \theta_{Dx} + \beta_{Dxx} - \theta_x - \beta_{xx} \tag{139}$$

$$\Phi_{Dyy} = \theta_{Dy} + \beta_{Dyy} - \theta_y - \beta_{yy} \tag{140}$$

$$\Phi_{Dzz} = \theta_{Dz} + \beta_{Dzz} - \theta_z - \beta_{zz} \tag{141}$$

$$\tan \beta_{Dxx} = (D_x \, \partial\theta_{Dx}/\partial x)/(\partial D_x/\partial x) \tag{142}$$

$$\tan \beta_{Dyy} = (D_y \, \partial\theta_{Dy}/\partial y)/(\partial D_y/\partial y) \tag{143}$$

$$\tan \beta_{Dzz} = (D_z \, \partial\theta_{Dz}/\partial z)/(\partial D_z/\partial z) \tag{144}$$

The derivatives appearing in equation (138) can be rewritten in terms of measured quantities as

$$\partial D_x/\partial x = g_{Dxx} \, \partial D_{xm}/\partial x_m + h_{Dxx} D_{xm} \tag{145}$$

$$\partial D_y/\partial y = g_{Dyy} \, \partial D_{ym}/\partial y_m + h_{Dyy} D_{ym} \tag{146}$$

$$\partial D_z/\partial z = g_{Dzz} \, \partial D_{zm}/\partial z_m + h_{Dzz} D_{zm} \tag{147}$$

where

$$g_{Dxx} = \sec \theta_{Dx} \, \partial x_m/\partial x \tag{148}$$

$$g_{Dyy} = \sec \theta_{Dy} \, \partial y_m/\partial y \tag{149}$$

$$g_{Dzz} = \sec \theta_{Dz} \, \partial z_m/\partial z \tag{150}$$

$$h_{Dxx} = \sec \theta_{Dx} \tan \theta_{Dx} \, \partial\theta_{Dx}/\partial x \tag{151}$$

$$h_{Dyy} = \sec \theta_{Dy} \tan \theta_{Dy} \, \partial\theta_{Dy}/\partial y \tag{152}$$

$$h_{Dzz} = \sec \theta_{Dz} \tan \theta_{Dz} \, \partial\theta_{Dz}/\partial z \tag{153}$$

Placing equations (145) through (147) into equations (138) gives the divergence equation in terms of measured electric displacement and measured coordinates

$$(\ell_c \partial/\partial x_m + t_c)D_{xm} + (m_c \partial/\partial y_m + u_c)D_{ym} + (r_c \partial/\partial z_m + v_c)D_{zm} = \rho_q \cos \theta_{\rho q} \quad (153A)$$

$$(\ell_s \partial/\partial x_m + t_s)D_{xm} + (m_s \partial/\partial y_m + u_s)D_{ym} + (r_s \partial/\partial z_m + v_s)D_{zm} = \rho_q \sin \theta_{\rho q} \quad (153B)$$

where

$$\ell_{c,s} = \begin{matrix}\cos \\ \sin\end{matrix} \Phi_{Dxx} \cos \beta_{xx} \sec \beta_{Dxx} \sec \theta_{Dx} \, \partial x_m/\partial x \quad (153C)$$

$$m_{c,s} = \begin{matrix}\cos \\ \sin\end{matrix} \Phi_{Dyy} \cos \beta_{yy} \sec \beta_{Dyy} \sec \theta_{Dy} \, \partial y_m/\partial y \quad (153D)$$

$$r_{c,s} = \begin{matrix}\cos \\ \sin\end{matrix} \Phi_{Dzz} \cos \beta_{zz} \sec \beta_{Dzz} \sec \theta_{Dz} \, \partial z_m/\partial z \quad (153E)$$

$$t_{c,s} = \begin{matrix}\cos \\ \sin\end{matrix} \Phi_{Dxx} \cos \beta_{xx} \sec \beta_{Dxx} \sec \theta_{Dx} \tan \theta_{Dx} \, \partial \theta_{Dx}/\partial x \quad (153F)$$

$$u_{c,s} = \begin{matrix}\cos \\ \sin\end{matrix} \Phi_{Dyy} \cos \beta_{yy} \sec \beta_{Dyy} \sec \theta_{Dy} \tan \theta_{Dy} \, \partial \theta_{Dy}/\partial y \quad (153G)$$

$$v_{c,s} = \begin{matrix}\cos \\ \sin\end{matrix} \Phi_{Dzz} \cos \beta_{zz} \sec \beta_{Dzz} \sec \theta_{Dz} \tan \theta_{Dz} \, \partial \theta_{Dz}/\partial z \quad (153H)$$

Equations (153A) and (153B) represent the divergence form of the Gauss law equation (115) expressed in terms of the measured field components and coordinates. The conventional Maxwell equation corresponding to equations (153A) and (153B) is

$$\partial D_{xc}/\partial x_m + \partial D_{yc}/\partial y_m + \partial D_{zc}/\partial z_m = \rho_m \quad (153I)$$

The Maxwell equations (113) and (114) are handled in a similar way.

B. Charge Density

The complex number charge density is given by

$$\bar{\rho}_q = \rho_q e^{j\theta \rho q} = dQ/d\bar{V} \quad (154)$$

where charge is taken to be a scalar having a zero internal phase angle. From equation (154) it follows that

$$\rho_q = dQ/|d\bar{V}| = \cos \beta_{VV} \, dQ/dV \quad (155)$$

$$\theta_{\rho q} = -\theta_V - \beta_{VV} \quad (156)$$

$$\tan \beta_{VV} = V \partial \theta_V/\partial V \quad (157)$$

$$Q = \int \rho_q |d\bar{V}| = \int \rho_q \sec \beta_{VV} \, dV \quad (158)$$

Case 1: Spherical Coordinates

$$\bar{\rho}_q = d^3Q/(\bar{r}^2 \sin \bar{\psi} \, d\bar{\psi} \, d\bar{\phi} \, d\bar{r}) \quad (159)$$

$$\rho_q = \cos \beta_{rr} \cos \beta_{\phi\phi} \cos \beta_{\psi\psi} \, d^3Q/(S_\psi r^2 \, d\psi \, d\phi \, dr) \quad (160)$$

$$\theta_{\rho q} = -3\theta_r - \theta_{s\psi} - \theta_\psi - \theta_\phi - \beta_{rr} - \beta_{\psi\psi} - \beta_{\phi\phi} \tag{161}$$

where S_ψ and $\theta_{s\psi}$ are given by equations (6) and (8) respectively, and $\beta_{\alpha\alpha}$ are given by equations (13) through (15). For spherical symmetry

$$\bar{\rho}_q^1 = (4\pi\bar{r}^2\bar{L}_s)^{-1}\, dQ/d\bar{r} \tag{162}$$

$$\rho_q^1 = \cos\beta_{rr}\,(4\pi r^2 L_s)^{-1}\, dQ/dr \tag{163}$$

$$\theta_{\rho q}^1 = -3\theta_r - \theta_{Ls} - \beta_{rr} \tag{164}$$

where $\bar{L}_s$ is defined in equation (48). The charge density that is used in the right hand side of the divergence form of Gauss' law given in equation (115) does not contain the factor $\bar{L}_s$ because this factor cancels from the numerator and denominator of the surface and volume integrals that define the divergence of a vector. For use in the divergence equation (115) the following definition of charge density for spherical symmetry is used

$$\bar{\rho}_q = (4\pi\bar{r}^2)^{-1}\, dQ/d\bar{r} \tag{165}$$

$$\rho_q = \cos\beta_{rr}\,(4\pi r^2)^{-1}\, dQ/dr \tag{166}$$

$$\theta_{\rho q} = -3\theta_r - \beta_{rr} \tag{167}$$

$$\rho_{qm} = \rho_q \cos\theta_{\rho q} = f_{3r}\rho_{qc} \tag{168}$$

where ρ_{qc} = conventionally calculated charge density given by

$$\rho_{qc} = (4\pi r_m^2)^{-1}\, dQ/dr_m \tag{169}$$

and where f_{3r} is given by equation (55). From equation (168) it follows that

$$(\rho_{qc} - \rho_{qm})/\rho_{qc} = 1 - f_{3r} \tag{170}$$

Case 2. Cylindrical Coordinates

$$\bar{\rho}_q = d^3Q/(\bar{r}\, d\bar{\phi}\, d\bar{r}\, d\bar{z}) = \rho_q e^{j\theta\rho q} \tag{171}$$

$$\rho_q = \cos\beta_{\phi\phi}\cos\beta_{rr}\cos\beta_{zz}\, d^3Q/(r\, d\phi\, dr\, dz) \tag{172}$$

$$\theta_{\rho q} = -2\theta_r - \theta_z - \theta_\phi - \beta_{rr} - \beta_{zz} - \beta_{\phi\phi} \tag{173}$$

For cylindrical symmetry

$$\bar{\rho}_q^1 = (2\pi\bar{r}\bar{L}_c)^{-1}\, d^2Q/(d\bar{r}\, d\bar{z}) \tag{174}$$

$$\rho_q^1 = \cos \beta_{rr} \cos \beta_{zz} / (2\pi r) \, d^2Q/(dr\, dz) \tag{175}$$

$$\theta_{\rho q}^1 = -2\theta_r - \theta_z - \langle\theta_\phi\rangle - \beta_{rr} - \beta_{zz} \tag{176}$$

For cylindrical symmetry the charge density that appears in the divergence equation (115) does not contain $\bar{L}_c$ and must be written in the following form

$$\bar{\rho}_q = (2\pi\bar{r})^{-1} \, d^2Q/(d\bar{r}\, d\bar{z}) = (2\pi\bar{r})^{-1} \, d\bar{\rho}_{q\ell}/d\bar{r} \tag{177}$$

where

$$\rho_q = \cos \beta_{rr} \cos \beta_{zz} / (2\pi r) \, d^2Q/(dr\, dz) \tag{178}$$

$$\theta_{\rho q} = -2\theta_r - \theta_z - \beta_{rr} - \beta_{zz} \tag{179}$$

$$\bar{\rho}_{q\ell} = dQ/d\bar{z} \tag{180}$$

$$\rho_{q\ell} = \cos \beta_{zz} \, dQ/dz \qquad \theta_{\rho q\ell} = -\theta_z - \beta_{zz} \tag{181}$$

where $\bar{\rho}_{q\ell}$ = complex number line charge per unit length. The measured charge density is given by

$$\rho_{qm} = \rho_q \cos \theta_{\rho q} = f_{2r}\rho_{qc} \tag{182}$$

where ρ_{qc} = conventionally calculated charge density given by

$$\rho_{qc} = (2\pi r_m)^{-1} \, d^2Q/(dr_m\, dz_m) \tag{183}$$

where f_{2r} is given by equation (74). The measured line charge density is given by

$$\rho_{q\ell m} = \rho_{q\ell} \cos \theta_{\rho q\ell} = \rho_{q\ell c} \, dz_m/dz \cos \beta_{zz} \cos(\theta_z + \beta_{zz}) \tag{184}$$

where the conventional line charge density is

$$\rho_{q\ell c} = dQ/dz_m \tag{185}$$

Case 3: Rectangular Coordinates

$$\bar{\rho}_q = d^3Q(d\bar{x}\, d\bar{y}\, d\bar{z}) = \rho_q e^{j\theta\rho q} \tag{186}$$

$$\rho_q = \cos \beta_{xx} \cos \beta_{yy} \cos \beta_{zz} \, d^3Q/(dx\, dy\, dz) \tag{187}$$

$$\theta_{\rho q} = -\theta_x - \theta_y - \theta_z - \beta_{xx} - \beta_{yy} - \beta_{zz} \tag{188}$$

$$\rho_{qm} = \rho_q \cos \theta_{\rho q} = f_{3xyz}\rho_{qc} \tag{189}$$

where the conventionally calculated charge density is

$$\rho_{qc} = d^3Q/(dx_m\, dy_m\, dz_m) \tag{190}$$

and

$$f_{3xyz} = \cos \theta_{\rho q} \cos \beta_{xx} \cos \beta_{yy} \cos \beta_{zz}\, dx_m/dx\, dy_m/dy\, dz_m/dz \tag{191}$$

where $d\alpha_m/d\alpha$ is given by equations (19) through (21).

In general the following relationships are valid

$$\bar{\rho}_q = q\bar{n} \qquad \rho_q = qn \qquad \theta_{\rho q} = \theta_n \tag{192}$$

where q = particle charge and $\bar{n}$ = complex particle number density. For use in Maxwell's equations $\bar{\rho}_q$ in equation (192) does not contain the internal angular factors $\bar{L}_s$ and $\bar{L}_c$ and $\bar{\rho}_q$ is defined by equations (165) and (177). For other uses (subsection D) the definition of $\bar{\rho}_q$ given in equations (162) and (174) must be used in equation (192) and $\bar{n}$ does contain the internal angular factors $\bar{L}_s$ and $\bar{L}_c$ for spherical and cylindrical systems respectively.

C. Electric Current

The electric current for broken symmetry spacetime is given by

$$\bar{I} = Ie^{j\theta I} = dQ/d\bar{t} \tag{193}$$

$$I = \cos \beta_{tt}\, dQ/dt \qquad \theta_I = -\theta_t - \beta_{tt} \tag{194}$$

The current density is given by the following vector component equations

$$\bar{j}_\alpha = d\bar{I}/d\bar{A}_\alpha = d^2Q/(d\bar{t}\, d\bar{A}_\alpha) = \bar{\rho}_q\bar{v}_\alpha \tag{195}$$

where $\alpha = x, y, z$; r, ϕ, ψ or r, ϕ, z ; $\bar{A}_\alpha$ = broken symmetry area normal to the α direction, and $\bar{v}_\alpha$ = broken symmetry electron velocity along the α axis. Equation (195) can be written in vector form as

$$\vec{\bar{j}} = \bar{\rho}_q\vec{\bar{v}}_\alpha \tag{196}$$

Writing the current density component as

$$\bar{j}_\alpha = j_\alpha e^{j\theta j\alpha} \tag{197}$$

gives

$$j_\alpha = \rho_q v_\alpha \tag{198}$$

$$\theta_{j\alpha} = \theta_{\rho q} + \theta_{v\alpha} = \theta_n + \theta_{v\alpha} \tag{199}$$

where v_α and $\theta_{v\alpha}$ are given in equation (28).

For a rectangular geometry equation (195) gives

$$\bar{j}_z = d^2\bar{I}/(d\bar{x}\, d\bar{y}) \tag{200}$$

$$j_z = \cos \beta_{xx} \cos \beta_{yy}\, d^2I/(dx\, dy) \tag{201}$$

$$\theta_{jz} = -\theta_t - \theta_x - \theta_y - \beta_{tt} - \beta_{xx} - \beta_{yy} \tag{202}$$

$$\sim -\theta_t - \theta_x - \theta_y$$

which combined with equation (199) gives

$$\theta_{vz} = \theta_z + \beta_{zz} - \theta_t - \beta_{tt} \tag{203}$$

which is the result obtained in Reference 12 by kinematical reasoning. It is easy to see that for the current in the x and y directions

$$j_x = \cos \beta_{yy} \cos \beta_{zz}\, d^2I/(dy\, dz) \tag{204}$$

$$\theta_{jx} = -\theta_t - \theta_y - \theta_z - \beta_{tt} - \beta_{yy} - \beta_{zz} \tag{205}$$

$$j_y = \cos \beta_{xx} \cos \beta_{zz}\, d^2I/(dx\, dz) \tag{206}$$

$$\theta_{jy} = -\theta_t - \theta_x - \theta_z - \beta_{tt} - \beta_{xx} - \beta_{zz} \tag{207}$$

For spherical systems of current flow the current density is

$$\bar{j} = d^2\bar{I}/(\bar{r}^2 \sin \bar{\psi}\, d\bar{\phi}\, d\bar{\psi}) \tag{208}$$

$$j = \cos \beta_{\phi\phi} \cos \beta_{\psi\psi}\, d^2I/(r^2 S_\psi d\phi\, d\psi) \tag{209}$$

$$\theta_j = \theta_I - 2\theta_r - \theta_{s\psi} - \theta_\phi - \theta_\psi - \beta_{\phi\phi} - \beta_{\psi\psi} \tag{210}$$

where θ_I is given in equation (194). For spherically symmetric systems equation (208) becomes

$$\bar{j}^1 = \bar{I}/(4\pi\bar{r}^2\bar{L}_s) \tag{211}$$

$$j^1 = I/(4\pi r^2 L_s) \qquad \theta_j^1 = \theta_I - 2\theta_r - \theta_{Ls} \tag{212}$$

But the current that appears in the curl equation (113) does not contain the internal angular factor $\bar{L}_s$ because this factor drops out of the definition of the curl of a vector in broken symmetry space. Therefore for applications to Maxwell's equations the following definition of radially symmetric current density must be used

$$\bar{j} = \bar{I}/(4\pi\bar{r}^2) \tag{213}$$

$$j = I/(4\pi r^2) \qquad \theta_j = \theta_I - 2\theta_r \tag{214}$$

where θ_I is given by equation (194).

Similarly, for a cylindrical geometry the current density is given by

$$\bar{j} = d^2\bar{I}/(\bar{r}\, d\bar{\phi}\, d\bar{r}) \tag{215}$$

$$j = \cos\beta_{\phi\phi} \cos\beta_{rr}\, d^2I/(r\, d\phi\, dr) \tag{216}$$

$$\theta_j = \theta_I - 2\theta_r - \theta_\phi - \beta_{rr} - \beta_{\phi\phi} \tag{217}$$

For axial symmetry this becomes

$$\bar{j}^1 = d\bar{I}/(2\pi\bar{r}\bar{L}_c d\bar{r}) \tag{218}$$

$$j^1 = dI/(2\pi r dr) \qquad \theta_j^1 = \theta_I - 2\theta_r - \beta_{rr} - \langle\theta_\phi\rangle \tag{219}$$

where θ_I is given by equation (194). Again, this is not the current that appears in Maxwell's equation (113) which does not contain the internal angle $\langle\theta_\phi\rangle$. For application in equation (113) the following current is used for axial symmetry.

$$\bar{j} = d\bar{I}/(2\pi\bar{r}d\bar{r}) \tag{220}$$

$$j = dI/(2\pi r dr) \qquad \theta_j = \theta_I - 2\theta_r - \beta_{rr} \tag{221}$$

D. Asymmetric Constitutive Equations

Constitutive equations relate $\vec{\bar{D}}$ to $\vec{\bar{E}}$ for electric fields in broken symmetry matter and $\vec{\bar{B}}$ to $\vec{\bar{H}}$ for magnetic fields located in broken symmetry matter.

Case 1. Electric Fields

The generalization of the standard scalar result is[18-28]

$$\vec{\bar{D}} = \varepsilon_o \vec{\bar{E}} + \vec{\bar{P}} = \bar{\varepsilon}\vec{\bar{E}} \tag{222}$$

where ε_o = permittivity of the vacuum, $\vec{\bar{P}}$ = complex number polarization vector for broken symmetry matter, and $\bar{\varepsilon}$ = complex number permittivity for broken symmetry matter. The complex number polarization vector is given by

$$\vec{\bar{P}} = \varepsilon_o \bar{\chi}_e \vec{\bar{E}} \tag{223}$$

where $\bar{\chi}_e$ = complex number electric susceptibility. Then

$$\bar{P}_\alpha = P_\alpha e^{j\theta P\alpha} = \varepsilon_o \bar{\chi}_e \bar{E}_\alpha \tag{224}$$

$$P_\alpha = \varepsilon_o \chi_e E_\alpha \tag{225}$$

$$\theta_{P\alpha} = \theta_{\chi e} + \theta_{E\alpha} \tag{226}$$

where $\alpha = x , y , z$. From equations (222) and (223) it follows that

$$\bar{\varepsilon} = \varepsilon_o (1 + \bar{\chi}_e) \tag{227}$$

Writing

$$\bar{\varepsilon} = \varepsilon e^{j\theta\varepsilon} \tag{228}$$

$$\bar{\chi}_e = \chi_e e^{j\theta\chi e} \tag{229}$$

allows equation (227) to be written as

$$\varepsilon \cos \theta_\varepsilon = \varepsilon_o + \varepsilon_o \chi_e \cos \theta_{\chi e} \tag{230}$$

$$\varepsilon \sin \theta_\varepsilon = \varepsilon_o \chi_e \sin \theta_{\chi e} \tag{231}$$

Equations (230) and (231) give

$$\tan \theta_\varepsilon = (\chi_e \sin \theta_{\chi e})/(1 + \chi_e \cos \theta_{\chi e}) \tag{232}$$

$$(\varepsilon/\varepsilon_o)^2 = 1 + \chi_e^2 + 2\chi_e \cos \theta_{\chi e} \tag{233}$$

In general $\theta_\varepsilon \neq \theta_{\chi e} \neq 0$.

In order to understand why $\theta_\varepsilon \neq \theta_{\chi e}$ one can use the fact that the polarization vector is equal to the dipole moment per unit volume as follows for broken symmetry matter

$$\bar{P}_\alpha = \bar{n} q \bar{d}_\alpha = \bar{\rho}_q \bar{d}_\alpha \tag{234}$$

where $\bar{d}_\alpha$ = broken symmetry value of the α component of the atomic dipole distance. Combining equations (224) and (234) gives

$$\varepsilon_o \chi_e E_\alpha = nqd_\alpha \tag{235}$$

$$\theta_{\chi e} + \theta_{E\alpha} = \theta_{\rho q} + \theta_{d\alpha} \tag{236}$$

Combining equations (188) and (236) and choosing the dipole to be oriented in the z direction gives the following approximation

$$\theta_{\chi e} \sim - \theta_{Ez} - \theta_x - \theta_y + (\theta_{dz} - \theta_z) \tag{237}$$

From the theory of the simple capacitor the following simple generalization holds for broken symmetry matter[18-28]

$$\bar{D}_z = \bar{\varepsilon}\bar{E}_z = d^2Q/(d\bar{x}\, d\bar{y}) \tag{238}$$

from which

$$D_z = \varepsilon E_z = \cos \beta_{xx} \cos \beta_{yy}\, d^2Q/(dx\, dy) \tag{239}$$

$$\theta_{Dz} = \theta_\varepsilon + \theta_{Ez} = - \theta_x - \theta_y - \beta_{xx} - \beta_{yy} \sim - \theta_x - \theta_y \tag{240}$$

Combining equations (237) and (240) gives

$$\theta_{\chi e} \sim \theta_\varepsilon + (\theta_{dz} - \theta_z) \tag{241}$$

Therefore because θ_{dz} on the atomic scale (due to electromagnetic forces) differs from θ_z on the macroscopic scale (due to gravity) it follows that $\theta_{\chi e} \neq \theta_\varepsilon$.

Case 2. Magnetic Fields

The generalization of the scalar constitutive equation for magnetic fields in broken symmetry matter can be written as[18-28]

$$\vec{\bar{B}} = \mu_o(\vec{\bar{H}} + \vec{\bar{M}}) = \bar{\mu}\vec{\bar{H}} \tag{242}$$

where μ_o = magnetic permeability for the vacuum, $\vec{\bar{M}}$ = magnetic polarization vector for broken symmetry matter, and $\bar{\mu}$ = complex number magnetic permeability for broken symmetry matter. The complex number magnetization vector is given by[18-28]

$$\vec{\bar{M}} = \bar{\chi}_M \vec{\bar{H}} \tag{243}$$

where $\bar{\chi}_M$ = complex number magnetic susceptibility. Equation (243) can be written as

$$\bar{M}_\alpha = M_\alpha e^{j\theta_{M\alpha}} = \bar{\chi}_M \bar{H}_\alpha = \chi_M H_\alpha e^{j(\theta_{\chi M} + \theta_{H\alpha})} \tag{244}$$

$$M_\alpha = \chi_M H_\alpha \tag{245}$$

$$\theta_{M\alpha} = \theta_{\chi M} + \theta_{H\alpha} \tag{246}$$

where $\alpha = r$, ϕ , z for cylindrical coordinates. From equations (242) and (243) it follows that

$$\bar{\mu} = \mu_o(1 + \bar{\chi}_M) \tag{247}$$

Writing $\bar{\mu}$ and $\bar{\chi}_M$ in complex form as

$$\bar{\mu} = \mu e^{j\theta_\mu} \tag{248}$$

$$\bar{\chi}_M = \chi_M e^{j\theta_{\chi M}} \tag{249}$$

allows equation (247) to be written as

$$\tan \theta_\mu = (\chi_M \sin \theta_{\chi M})/(1 + \chi_M \cos \theta_{\chi M}) \tag{250}$$

$$(\mu/\mu_o)^2 = 1 + \chi_M^2 + 2\chi_M \cos \theta_{\chi M} \tag{251}$$

From equation (242) it follows that

$$B_\alpha = \mu H_\alpha \tag{252}$$

$$\theta_{B\alpha} = \theta_\mu + \theta_{H\alpha} \tag{253}$$

Equation (250) shows that $\theta_\mu \neq \theta_{\chi M}$. This can be understood from an atomic basis by considering the fact that the magnetization vector is equal to the magnetic dipole moment per unit volume, or in component form

$$\bar{M}_\alpha = \bar{n}\bar{m}_\alpha^d = \bar{\chi}_M \bar{H}_\alpha \tag{254}$$

where $\bar{m}_\alpha^d$ = components of the atomic magnetic dipole moment $\bar{\vec{m}}^d$, and where $\bar{n}$ = complex number of dipole moments per unit volume where now the angular factors $\langle\theta_\phi\rangle$ and $\langle\theta_\psi\rangle$ are included for spherically symmetric systems and the factor $\langle\theta_\phi\rangle$ is included for axially symmetric systems. Therefore for spherically symmetric systems the $\bar{n}$ that appears in equation (254) is given by

$$\bar{n}^1 = ne^{j\theta_n} = dN/d\bar{V} = (4\pi\bar{r}^2\bar{L}_s)^{-1}\, dN/d\bar{r} \tag{255}$$

$$n^1 = (4\pi r^2 L_s)^{-1} \cos \beta_{rr}\, dN/dr \tag{256}$$

$$\theta_n = -3\theta_r - \theta_{Ls} - \beta_{rr} \tag{257}$$

and for axial symmetry by

$$\bar{n}^1 = dN/d\bar{V} = (2\pi\bar{r}\bar{L}_c)^{-1}\, d^2N/(d\bar{r}\, d\bar{z}) \tag{258}$$

$$n^1 = (2\pi r)^{-1} \cos\beta_{rr} \cos\beta_{zz}\, d^2N/(dr\, dz) \tag{259}$$

$$\theta_n^1 = -2\theta_r - \theta_z - \langle\theta_\phi\rangle - \beta_{rr} - \beta_{zz} \tag{260}$$

The general case for the number density of a spherical system is

$$\bar{n} = d^3N/(\bar{r}^2 \sin\bar{\psi}\, d\bar{\psi}\, d\bar{\phi}\, d\bar{r}) \tag{261}$$

$$n = \cos\beta_{\psi\psi} \cos\beta_{\phi\phi} \cos\beta_{rr}\, d^3N/(r^2 S_\psi d\psi\, d\phi\, dr) \tag{262}$$

$$\theta_n = -3\theta_r - \theta_{s\psi} - \theta_\psi - \theta_\phi - \theta_r - \beta_{\psi\psi} - \beta_{\phi\phi} - \beta_{rr} \tag{263}$$

while for cylindrical coordinates

$$\bar{n} = d^3N/(\bar{r}\, d\bar{\phi}\, d\bar{r}\, d\bar{z}) \tag{264}$$

$$n = \cos\beta_{\phi\phi} \cos\beta_{rr} \cos\beta_{zz}\, d^3N/(r\, d\phi\, dr\, dz) \tag{265}$$

$$\theta_n = -2\theta_r - \theta_z - \theta_\phi - \beta_{\phi\phi} - \beta_{zz} - \beta_{rr} \tag{266}$$

and for rectangular coordinates

$$\bar{n} = d^3N/(d\bar{x}\, d\bar{y}\, d\bar{z}) \tag{267}$$

$$n = \cos\beta_{xx} \cos\beta_{yy} \cos\beta_{zz}\, d^3N/(dx\, dy\, dz) \tag{268}$$

$$\theta_n = -\theta_x - \theta_y - \theta_z - \beta_{xx} - \beta_{yy} - \beta_{zz} \tag{269}$$

Consider a current in a wire situated in the z direction of a broken symmetry spacetime. The magnetic field is in the azimuthal direction and equation (254) can be written as

$$\bar{M}_\phi = M_\phi e^{j\theta M\phi} = \bar{n}^1 \bar{m}_\phi^d = \bar{\chi}_M \bar{H}_\phi \tag{270}$$

where the atomic magnetic moment is given by the following generalization of the standard result[18-28]

$$\bar{m}_\phi^d = \pi\bar{r}_d^2 \bar{L}_{cd} \bar{I}_d = \pi\bar{r}_d^2 \bar{I}_d e^{j\langle\theta_\phi^d\rangle} \tag{271}$$

where $\bar{r}_d$ = complex number atomic radius, $\bar{L}_{cd}$ = internal angular factor for atomic current loop given by equation (61), $\bar{I}_d$ = complex number atomic electron current, and $\langle\theta_\phi^d\rangle$ = average value of the internal phase angle of the broken symmetry azimuthal angle of the atomic current loop. Equation (271) is based on the fact that the broken symmetry area of the atomic electron orbit is[12]

$$\bar{A}_d = \pi \bar{r}_d^2 \bar{L}_{cd} = \pi \bar{r}_d^2 e^{j\langle\theta_\phi^d\rangle} \tag{272}$$

From equations (270) and (271) it follows that

$$M_\phi = \chi_M H_\phi = \pi r_d^2 n^1 I_d \tag{273}$$

$$\theta_{M\phi} = \theta_{\chi M} + \theta_{H\phi} = \theta_n^1 + 2\theta_r^d + \theta_I^d + \langle\theta_\phi^d\rangle \tag{274}$$

where the following relations were used

$$\bar{r}_d = r_d e^{j\theta_r^d} \tag{275}$$

$$\bar{I}_d = I_d e^{j\theta_I^d} \tag{276}$$

and where from equation (194) $\theta_I^d = -\theta_t^d$ where θ_t^d = internal phase angle of the atomic time. Combining equations (273) and (274) with equations (260), (407) and (408) gives

$$\chi_M = \pi r_d^2 n^1 (2\pi r) I_d / I_z \tag{277}$$

$$\theta_{\chi M} = \theta_{M\phi} - \theta_{H\phi} \tag{278}$$

$$= (2\theta_r^d - \theta_r - \theta_z) + (\langle\theta_\phi^d\rangle - \langle\theta_\phi\rangle) + (\theta_t - \theta_t^d)$$

where the following results derived later in equations (407) and (408) were used

$$H_\phi = I_z/(2\pi r) \tag{279}$$

$$\theta_{H\phi} = \theta_{Iz} - \theta_r \sim -\theta_t - \theta_r \tag{280}$$

From equation (278) it is clear that $\theta_{\chi M} \neq 0$ only if the internal phase angles of the coordinates at the atomic scale are different from the internal phase angles at macroscopic distances. At the atomic scale θ_r^d , θ_ϕ^d and θ_t^d are determined mainly by electric forces, while at macroscopic distances θ_r , θ_ϕ and θ_t are determined by gravitation. Note for superconductors it is expected that the internal phase angles over macroscopic distances θ_r , θ_ϕ and θ_t are determined mainly by coherent electrical forces and $\theta_{\chi M} \sim 0$. The analysis in this section shows the importance of axially symmetric and spherically symmetric densities that appear in equations (162), (174), (255) and (258). The measured value of χ_M is given generally by $\chi_{Mm} = \chi_M \cos\theta_{\chi M}$.

E. Broken Symmetry Form of Ohm's Law.

The current in a wire is driven by a potential difference (as produced by a battery for instance) which is related to the electric field by the following broken symmetry generalization of the standard scalar result[18-28]

$$\bar{W}_\alpha = W_\alpha e^{j\theta W\alpha} = \int \vec{\bar{E}} \cdot d\vec{\bar{s}} = \bar{E}_\alpha \bar{\alpha} \tag{281}$$

for the case of a constant electric field in the α direction. From equation (281) it follows that

$$W_\alpha = E_\alpha \alpha \qquad \theta_{W\alpha} = \theta_{E\alpha} + \theta_\alpha \tag{282}$$

$$= [\theta_\alpha - 2\theta_t + 2(\beta_{\alpha\alpha} - \beta_{tt})] + \theta_\alpha$$

$$= 2(\theta_\alpha + \beta_{\alpha\alpha} - \theta_t - \beta_{tt})$$

$$\sim 2(\theta_\alpha - \theta_t)$$

The current density is then given by the broken symmetry form of Ohm's law

$$\bar{j}_\alpha = \bar{\sigma}_\alpha \bar{E}_\alpha \tag{283}$$

where the complex number electric conductivity is

$$\bar{\sigma}_\alpha = \sigma_\alpha e^{j\theta\sigma\alpha} \tag{284}$$

so that

$$j_\alpha = \sigma_\alpha E_\alpha \tag{285}$$

$$\theta_{j\alpha} = \theta_{\sigma\alpha} + \theta_{E\alpha} \sim \theta_{\sigma\alpha} + \theta_\alpha - 2\theta_t \tag{286}$$

Combining equations (199) and (286) gives

$$\theta_{\sigma z} = \theta_{jz} - \theta_{Ez} = \theta_n + \theta_{vz} - \theta_{Ez} \tag{287}$$

for the current in the z direction. Combining equations (203), (266), (269), (282) and (287) gives the internal phase angle of the conductivity in the z direction as

$$\theta_{\sigma z} \sim -\theta_t + \theta_z - \theta_x - \theta_y - \theta_{Wz} \tag{288}$$

$$= \theta_t - \theta_x - \theta_y - \theta_z$$

$$= \theta_t + \theta_n$$

where $\beta_{\alpha\alpha}$ has been neglected. For current in the x and y directions respectively

$$\theta_{\sigma x} \sim -\theta_t + \theta_x - \theta_y - \theta_z - \theta_{Wx} \tag{289}$$

$$= \theta_t - \theta_x - \theta_y - \theta_z$$

$$= \theta_t + \theta_n$$

$$\theta_{\sigma y} \sim -\theta_t + \theta_y - \theta_x - \theta_z - \theta_{Wy} \tag{290}$$

$$= \theta_t - \theta_x - \theta_y - \theta_z$$

$$= \theta_t + \theta_n$$

Therefore $\theta_{\sigma x} = \theta_{\sigma y} = \theta_{\sigma z}$ and their common value can be called ϑ_σ . The measured voltage, current, current density and conductivity are given respectively by

$$W_{\alpha m} = W_\alpha \cos \theta_{W\alpha} \sim W_\alpha \cos[2(\theta_\alpha - \theta_t)] \tag{291}$$

$$I_{\alpha m} = I_\alpha \cos \theta_{I\alpha} = I_\alpha \cos \theta_t \tag{292}$$

$$j_{zm} = j_z \cos \theta_{jz} \tag{293}$$

$$\sigma_{\alpha m} = \sigma_\alpha \cos \theta_{\sigma\alpha} \tag{294}$$

where θ_{jz} and $\theta_{\sigma\alpha}$ are given by equations (202) and (288) through (290) respectively.

The resistance of a wire located in a broken symmetry spacetime is given by the following generalization of the standard result[18-28]

$$\bar{R}_\alpha = R_\alpha e^{j\theta R\alpha} = \bar{L}_\alpha/(\bar{\sigma}_\alpha \bar{A}_\alpha) = L_\alpha/(\sigma_\alpha A_\alpha) e^{j(\theta L\alpha - \theta_{\sigma\alpha} - \theta_{A\alpha})} \tag{295}$$

where $\bar{R}_\alpha$ = complex number resistance of a wire of complex number length $\bar{L}_\alpha$ situated in the α direction, and complex number cross sectional area $\bar{A}_\alpha$ perpendicular to the α direction. Equation (295) is equivalent to

$$R_\alpha = L_\alpha/(\sigma_\alpha A_\alpha) \tag{296}$$

$$\theta_{R\alpha} = \theta_{L\alpha} - \theta_{\sigma\alpha} - \theta_{A\alpha} \tag{297}$$

and the measured value of the resistance of a wire situated in the α direction is given by

$$R_{\alpha m} = R_\alpha \cos \theta_{R\alpha} \tag{298}$$

Using equations (288) through (290) and the following simple results

$$\theta_{Lx} = \theta_x \qquad \theta_{Ly} = \theta_y \qquad \theta_{Lz} = \theta_z \tag{299}$$

$$\theta_{Ax} = \theta_y + \theta_z \qquad \theta_{Ay} = \theta_x + \theta_z \qquad \theta_{Az} = \theta_x + \theta_y \tag{300}$$

in equation (297) shows that

$$\theta_{R\alpha} = \theta_{W\alpha} + \theta_t \sim 2(\theta_\alpha - \theta_t) + \theta_t = 2\theta_\alpha - \theta_t \tag{301}$$

for $\alpha = x , y , z$. Equation (301) becomes clear when it is realized that in general

$$\bar{R}_\alpha = \bar{W}_\alpha / \bar{I}_\alpha \tag{302}$$

so that

$$R_\alpha = W_\alpha / I_\alpha \qquad \theta_{R\alpha} = \theta_{W\alpha} - \theta_{I\alpha} \sim \theta_{W\alpha} + \theta_t \sim 2\theta_\alpha - \theta_t \tag{303}$$

The measured resistance is given by equations (291), (292) and (303) to be

$$\begin{aligned} R_{\alpha m} &= R_\alpha \cos \theta_{R\alpha} = W_\alpha / I_\alpha \cos(\theta_{W\alpha} + \theta_t) \\ &= W_{\alpha m} / I_{\alpha m} (1 - \tan \theta_t \tan \theta_{W\alpha}) \cos^2 \theta_t \\ &= W_{\alpha m} / I_{\alpha m} \{1 - \tan \theta_t \tan[2(\theta_\alpha - \theta_t)]\} \cos^2 \theta_t \end{aligned} \tag{304}$$

where $W_{\alpha m}$ and $I_{\alpha m}$ are given by equations (291) and (292). Equation (304) is a broken symmetry form of Ohm's law.

The conventionally calculated resistance is given by

$$R_{\alpha c} = W_{\alpha m} / I_{\alpha m} \tag{305}$$

and therefore a comparison of the measured and conventionally calculated resistance is given by

$$\xi_\alpha = (R_{\alpha c} - R_{\alpha m}) / R_{\alpha c} = \sin \theta_t \cos \theta_t (\tan \theta_t + \tan \theta_{W\alpha}) \tag{306}$$

$$\sim \theta_t (\theta_t + \theta_{W\alpha}) = \theta_t \theta_{R\alpha} = \theta_t (2\theta_\alpha - \theta_t) \qquad \} \text{ small angles}$$

A measurement of ξ_α may possibly be used to determine θ_t and θ_α for the local spacetime in the vicinity of the experiment. The internal phase angle of a battery voltage $\theta_{W\alpha}$ for the potential difference $\bar{W}_\alpha$ applied in the α direction is associated with the operation of a battery in matter with a gravitational field present. The measured resistance given in equation (304) depends on θ_t and θ_α , and therefore the measured resistance will depend on the orientation of the wire relative to a gravitational field even when $W_{\alpha m}/I_{\alpha m}$ have the same values for $\alpha = x, y, z$. For gravity at the earth's surface, the Pound-Rebka-Snider experiment gives the values $\theta_z = -5.7°$ and $\theta_t = -3/2(5.7°) = -8.55°$.[12] Accurate measurements of $R_{\alpha m}$, $W_{\alpha m}$ and $I_{\alpha m}$ for various orientations of a wire relative to the direction of the earth's gravity field may possibly be used to determine θ_α and θ_t through the application of equation (304) and the relation $\theta_t = 3/2\theta_r$ which holds for a massive particle falling in a gravitational field. In fact even the measured voltage of a battery itself determined by equation (291) for various inclinations relative to the earth's gravity field may be used to determine θ_α and θ_t . On the other hand the measured conductivities given by equation (294) will have the same values irrespective of the orientation of the wire with respect to the gravity field if $\sigma_x = \sigma_y = \sigma_z$ because their internal phase angles satisfy the relation $\theta_{\sigma x} = \theta_{\sigma y} = \theta_{\sigma z}$.

An examination of equation (304) shows that it is possible for the measured resistance to have a zero value resulting in a superconducting state. This occurs when $R_{\alpha m} = 0$ in equation (304) and

$$\tan \theta_t = \cot \theta_{W\alpha} \qquad \theta_t = \pi/2 - \theta_{W\alpha} \tag{306A}$$

$$2\theta_\alpha - \theta_t = \pi/2 \qquad = \pi/2 - 2(\theta_\alpha - \theta_t)$$

Therefore in a broken symmetry state the degree of coherency of time in matter may be sufficiently large so as to allow equation (306A) to be satisfied. A form of superconductivity is therefore possible if time in matter can form a coherent state. This may occur for all matter when the gravitational field is so large, as in the case of compact stars such as white dwarfs and neutron stars, that a coherent time state described by equation (306A) can form. For a vertical wire, the result of combining equation (36A) and (306A) gives the following values of the internal phase angles for gravitationally induced superconductivity.

$$\theta_r = -\pi \qquad \theta_t = -3\pi/2 \tag{306B}$$

On the other hand, even at ordinary pressures and densities it may be possible for the atomic and molecular structure to be of such a form that the space and time fields are coherent. For this case the charge carrying Cooper pairs are essentially free particles with $\theta_\alpha = 2\theta_t$, and this combined with equation (306A) yields the following condition for structurally induced superconductivity

$$\theta_\alpha = \pi/3 \qquad \theta_t = \pi/6 \tag{306C}$$

The planar structure of the copper oxides that exhibit high temperature superconductivity may in fact be a structure that allows a large value of θ_t to arise from electromagnetic fields as in equation (41). In this case, time may exhibit a coherent state due to electrical forces and equation (306C) may describe high temperature superconductivity if θ_t is temperature dependent and

becomes large below a transition temperature. For this coherent time state an increment of time would correspond to an internal phase rotation of the form $\Delta t = t\theta_t$. The two time scales t and $t\theta_t$ correspond to different energy scales. The relative energies of superconductors are described by the normalized superconductivity energy gap

$$\Delta' = 2\Delta/(kT_c) \tag{306D}$$

where Δ = superconductivity gap corresponding to the transition temperature T_c. Then a simple analysis using the Heisenberg uncertainty principle shows that the relationship between the normalized superconductivity energy gap for the coherent time state Δ'_{ct} and the conventional normalized energy gap for incoherent time Δ'_{it} is given by

$$\Delta'_{ct} = \Delta'_{it}/\theta_t = 6/\pi\ \Delta'_{it} \sim 1.91\Delta'_{it} \tag{306E}$$

corresponding to $\theta_t = \pi/6$ for superconductivity in a collection of free particles or holes. The result in equation (306E) agrees with measured superconductivity gaps for the planar cuprous oxide high T_c structures.[30] It should be noted that the normalized energy gap for a gravitationally induced superconductor is much smaller and is obtained from equation (306B) to be

$$\Delta'_{ct} = 2/(3\pi)\Delta'_{it} \sim 0.21\Delta'_{it} \tag{306F}$$

Equations (304) and (305) show that conventional superconductivity occurs when $R_{\alpha c} = 0$.

F. Scalar and Vector Potentials.

The complex number scalar potential $\bar{\phi}$ and the complex number vector potential $\vec{\bar{A}}$ are defined by the following generalizations

$$\vec{\bar{B}} = \vec{\bar{\nabla}} \times \vec{\bar{A}} \tag{307}$$

$$\vec{\bar{E}} = -\vec{\bar{\nabla}}\bar{\phi} - \partial\vec{\bar{A}}/\partial\bar{t} \tag{308}$$

where

$$\bar{A}_\alpha = A_\alpha e^{j\theta A\alpha} \qquad \bar{\phi} = \phi e^{j\theta\phi} \tag{310}$$

For cartesian coordinates equation (307) is written as

$$\bar{B}_x = \partial\bar{A}_z/\partial\bar{y} - \partial\bar{A}_y/\partial\bar{z} \tag{310}$$

$$\bar{B}_y = \partial\bar{A}_x/\partial\bar{z} - \partial\bar{A}_z/\partial\bar{x} \tag{311}$$

$$\bar{B}_z = \partial\bar{A}_y/\partial\bar{x} - \partial\bar{A}_x/\partial\bar{y} \tag{312}$$

The real and imaginary parts of equation (310) are written as

$$B_x \cos\theta_{Bx} = \cos\Phi_{Azy} \cos\beta_{yy} \sec\beta_{Azy}\ \partial A_z/\partial y - \cos\Phi_{Ayz} \cos\beta_{zz} \sec\beta_{Ayz}\ \partial A_y/\partial z \tag{313}$$

$$B_x \sin \theta_{Bx} = \sin \Phi_{Azy} \cos \beta_{yy} \sec \beta_{Azy} \partial A_z/\partial y \qquad (314)$$

$$- \sin \Phi_{Ayz} \cos \beta_{zz} \sec \beta_{Ayz} \partial A_y/\partial z$$

$$\Phi_{Azy} = \theta_{Az} + \beta_{Azy} - \theta_y - \beta_{yy} \qquad (315)$$

$$\Phi_{Ayz} = \theta_{Ay} + \beta_{Ayz} - \theta_z - \beta_{zz} \qquad (316)$$

$$\tan \beta_{Azy} = (A_z \partial\theta_{Az}/\partial y)/(\partial A_z/\partial y) \qquad (317)$$

$$\tan \beta_{Ayz} = (A_y \partial\theta_{Ay}/\partial z)/(\partial A_y/\partial z) \qquad (318)$$

Equations (311) and (312) can be handled in a similar manner. From equation (308) it follows that

$$\bar{E}_\alpha = - \partial\bar{\phi}/\partial\bar{\alpha} - \partial\bar{A}_\alpha/\partial\bar{t} \qquad (319)$$

where $\alpha = x , y, z$. For $\alpha = x$, the real and imaginary parts of equation (319) are

$$E_x \cos \theta_{Ex} = - \cos \Phi_{\phi x} \cos \beta_{xx} \sec \beta_{\phi x} \partial\phi/\partial x \qquad (320)$$

$$- \cos \Phi_{Axt} \cos \beta_{tt} \sec \beta_{Axt} \partial A_x/\partial t$$

$$E_x \sin \theta_{Ex} = - \sin \Phi_{\phi x} \cos \beta_{xx} \sec \beta_{\phi x} \partial\phi/\partial x \qquad (321)$$

$$- \sin \Phi_{Axt} \cos \beta_{tt} \sec \beta_{Axt} \partial A_x/\partial t$$

where

$$\Phi_{\phi x} = \theta_\phi + \beta_{\phi x} - \theta_x - \beta_{xx} \qquad (322)$$

$$\Phi_{Axt} = \theta_{Ax} + \beta_{Axt} - \theta_t - \beta_{tt} \qquad (323)$$

$$\tan \beta_{\phi x} = (\phi\partial\theta_\phi/\partial x)/(\partial\phi/\partial x) \qquad (324)$$

$$\tan \beta_{Axt} = (A_x \partial\theta_{Ax}/\partial t)/(\partial A_x/\partial t) \qquad (325)$$

The y and z components of equation (319) can be handled in a similar fashion. Equations (307) through (325) are the equations of the electromagnetic field in a broken symmetry spacetime such as may be induced by a gravitational field.

The scalar and vector potentials for broken symmetry spacetime can also

be written as the following modifications of the standard results[18-28]

$$\bar{\phi} = (4\pi\bar{\varepsilon}\bar{L})^{-1} \int \bar{\rho}'/|\vec{\bar{r}} - \vec{\bar{r}}'| e^{-i\vec{\bar{k}}\cdot(\vec{\bar{r}} - \vec{\bar{r}}')} d\bar{V}' \tag{326}$$

$$\vec{\bar{A}} = \bar{\mu}/(4\pi\bar{L}) \int \vec{\bar{j}}'/|\vec{\bar{r}} - \vec{\bar{r}}'| e^{-i\vec{\bar{k}}\cdot(\vec{\bar{r}} - \vec{\bar{r}}')} d\bar{V}' \tag{327}$$

where

$$\bar{L} = \bar{L}_s \qquad \text{spherically symmetric system} \tag{328}$$

$$\bar{L} = \bar{L}_c \qquad \text{axially symmetric system} \tag{329}$$

where $\bar{L}_s$ and $\bar{L}_c$ are given by equations (48) and (61) respectively in terms of $\langle\theta_\phi\rangle$ and $\langle\theta_\psi\rangle$. These average values are defined as follows

$$\bar{L}_c = e^{j\langle\theta_\phi\rangle} = 1/(2\pi)\int_0^{2\pi} d\bar{\phi} = e^{j\theta_\phi(2\pi)} \tag{330}$$

$$= 1/(2\pi)\int_0^{2\pi} \sec\beta_{\phi\phi}\, e^{j(\theta_\phi + \beta_{\phi\phi})} d\phi$$

and therefore

$$\langle\theta_\phi\rangle = \theta_\phi(2\pi) \tag{331}$$

for cylindrical coordinates. For spherical coordinates it follows that

$$\bar{L}_s = 1/(4\pi)\int_0^{2\pi}\int_0^{\pi} \sin\bar{\psi}\, d\bar{\phi}\, d\bar{\psi} \tag{332}$$

$$= 1/(4\pi)\int_0^{2\pi}\int_0^{\pi} S_\psi \sec\beta_{\phi\phi} \sec\beta_{\psi\psi}\, e^{j\Phi_{\phi\psi}} d\phi\, d\psi$$

$$= 1/2\; e^{j\theta_\phi(2\pi)}[1 - \cos(\pi e^{j\theta_\psi(\pi)})]$$

$$= 1/2\; e^{j\langle\theta_\phi\rangle}[1 - \cos(\pi e^{j\langle\theta_\psi\rangle})]$$

where

$$\Phi_{\phi\psi} = \theta_{s\psi} + \theta_\phi + \theta_\psi + \beta_{\phi\phi} + \beta_{\psi\psi} \tag{333}$$

$$\langle\theta_\psi\rangle = \theta_\psi(\pi) \tag{334}$$

The integration over the volume elements $d\bar{V}'$ produces the factor $\bar{L}$ so that this factor cancels in the numerator and denominator in equations (326) and (327), and $\bar{\phi}$ and $\vec{\bar{A}}$ are independent of $\langle\theta_\phi\rangle$ and $\langle\theta_\psi\rangle$.[12] This means that the electric and magnetic fields are also independent of $\langle\theta_\phi\rangle$ and $\langle\theta_\psi\rangle$. This will be examined in more detail in Sections 4 and 5.

G. Boundary Conditions in Broken Symmetry Spacetime.

For electromagnetic fields in matter located in a gravitational field the standard four boundary conditions become complex number equations due to the internal phase angles of the field vectors, current density and charge density.

Case 1. Tangential Components of the Electric Fields.

The tangential component of the electric field vector is continuous at the boundary between two materials

$$\bar{E}_{t1} = \bar{E}_{t2} \tag{335}$$

which gives

$$E_{t1} = E_{t2} \qquad \theta_{Et1} = \theta_{Et2} \tag{336}$$

Thus the magnitude and the internal phase angle of the tangential component of the electric field vector are continuous across a boundary.

Case 2. Tangential Components of the Magnetic Field.

The tangential components of the magnetic field vector on either side of a boundary are related to the complex number surface current density by

$$\bar{j}_s = \bar{H}_{1t} - \bar{H}_{2t} \tag{337}$$

which gives

$$j_s \cos \theta_{js} = H_{1t} \cos \theta_{H1t} - H_{2t} \cos \theta_{H2t} \tag{338}$$

$$j_s \sin \theta_{js} = H_{1t} \sin \theta_{H1t} - H_{2t} \sin \theta_{H2t} \tag{339}$$

$$j_s^2 = H_{1t}^2 + H_{2t}^2 - 2H_{1t}H_{2t} \cos(\theta_{H2t} - \theta_{H1t}) \tag{340}$$

$$\tan \theta_{js} = (H_{1t} \sin \theta_{H1t} - H_{2t} \sin \theta_{H2t})/(H_{1t} \cos \theta_{H1t} - H_{2t} \cos \theta_{H2t}) \tag{341}$$

In a gravitational field, the internal angles of the field vector must be considered in the boundary conditions. Equation (338) shows that the measured surface current density is equal to the difference of the measured values of the tangential components of the magnetic field vectors.

Case 3. Normal Components of the Magnetic Induction Vector.

The normal component of the magnetic induction vector is continuous across a boundary between two materials

$$\bar{B}_{n1} = \bar{B}_{n2} \tag{342}$$

and therefore

$$B_{n1} = B_{n2} \qquad \theta_{Bn1} = \theta_{Bn2} \tag{343}$$

Both magnitude and internal phase angle are continuous across a boundary.

Case 4. Normal Components of the Electric Displacement Vector.

The normal components of the electric displacement vector are related to the complex number surface electric charge density as follows

$$\bar{\rho}_s = \bar{D}_{n1} - \bar{D}_{n2} \tag{344}$$

$$\rho_s \cos \theta_{\rho s} = D_{n1} \cos \theta_{Dn1} - D_{n2} \cos \theta_{Dn2} \tag{344A}$$

$$\rho_s \sin \theta_{\rho s} = D_{n1} \sin \theta_{Dn1} - D_{n2} \sin \theta_{Dn2} \tag{344B}$$

$$\rho_s^2 = D_{n1}^2 + D_{n2}^2 - 2D_{n1}D_{n2} \cos(\theta_{Dn1} - \theta_{Dn2}) \tag{345}$$

$$\tan \theta_{\rho s} = (D_{n1} \sin \theta_{Dn1} - D_{n2} \sin \theta_{Dn2})/(D_{n1} \cos \theta_{Dn1} - D_{n2} \cos \theta_{Dn2}) \tag{346}$$

Equation (344A) states that the measured surface charge density is equal to the difference of the measured values of the normal components of the electric displacement vector.

4. ELECTROSTATICS WITH BROKEN INTERNAL SYMMETRIES. The preceding section suggested that gravitation will have an effect on electromagnetism through the internal phase angles of space and time coordinates. This section considers the effect of a gravity induced broken spacetime symmetry on the elementary calculations of electrostatics.

A. Gauss Law for Broken Symmetry Spacetime.

The differential form of Gauss' law equation (115) can be written in terms of a potential function by

$$\vec{\bar{E}} = - \vec{\nabla}\bar{W} \tag{347}$$

where $\bar{W}$ = potential function with an internal phase angle. Then equation (115) becomes

$$\bar{\nabla}^2\bar{W} = - \bar{\rho}_q/\bar{\varepsilon} \tag{348}$$

It will be shown that $\bar{\rho}_q$ in equation (348) does not contain the angular factor $\bar{L}$. On the other hand, the integral form of the Gauss law for a broken symmetry spacetime is

$$\oint \vec{\bar{D}} \cdot \vec{\bar{n}} d\bar{S} = \bar{L}Q \tag{349}$$

where $\bar{L}$ is given by equations (48) or (332) for spherical polar coordinates and by equations (61) or (330) for cylindrical polar coordinates. The factor $\bar{L}$ occurs in equation (349) on account of the surface integral. In general the factor $\bar{L}$ occurs because of an integration over a surface or a volume of a sphere or cylinder with broken symmetry coordinates. In fact

$$\oint d\bar{V} = \bar{L}_s 4/3\pi\bar{r}^3 \tag{350}$$

spherical coordinates

$$\oint d\bar{S} = \bar{L}_s 4\pi\bar{r}^2 \tag{351}$$

$$\oint d\bar{V} = \bar{L}_c \pi\bar{r}^2\bar{h} \tag{352}$$

$$\oint d\bar{S} = \bar{L}_c 2\pi\bar{r}\bar{h} \tag{353}$$

cylindrical coordinates

$$\oint d\bar{s} = \bar{L}_c 2\pi\bar{r} \tag{354}$$

where $\bar{L}_s$ and $\bar{L}_c$ are given by equations (332) and (330) respectively, and where $\bar{h}$ = complex number height of a cylinder. Because $\bar{L}$ occurs on both sides of equation (349) it follows that $\langle\theta_\phi\rangle$ and $\langle\theta_\psi\rangle$ do not enter the final expressions for the complex number electric field and electric displacement vectors. This conclusion also follows from the definition of the divergence of a vector with broken internal symmetry which is

$$\vec{\bar{\nabla}} \cdot \vec{\bar{D}} = \lim_{V\to o} \oint \vec{\bar{D}} \cdot \vec{\bar{n}}\, d\bar{S}/\oint d\bar{V} = \bar{\rho}_q \tag{355}$$

and because $\bar{L}$ appears in both the integrals over $d\bar{S}$ and $d\bar{V}$ it cancels in the numerator and denominator making $\vec{\bar{D}}$, $\vec{\bar{E}}$ and $\bar{\rho}_q$ independent of $\bar{L}$. Note that in all calculations the charge is a scalar, so that the measured charge, the conventional charge and the fundamental charge are all equal

$$Q_m = Q_c = Q \tag{356}$$

In this respect charge and rest mass are similar.

B. Spherically Symmetric Constant Charge Distribution.

For spherical symmetry equation (115) becomes

$$1/\bar{r}^2 d/d\bar{r}(\bar{r}^2\bar{D}_r) = \bar{\rho}_q \tag{357}$$

or

$$d\bar{D}_r/d\bar{r} + 2/\bar{r}\bar{D}_r = \bar{\rho}_q \tag{358}$$

For $\bar{\rho}_q$ = constant (ρ_q = constant, $\theta_{\rho q}$ = constant)

$$\bar{D}_r = 1/3\bar{\rho}_q\bar{r} = 1/3\rho_q re^{-2j\theta_r} \tag{359}$$

which gives

$$D_r = 1/3\rho_q r \qquad \theta_{Dr} = -2\theta_r \tag{360}$$

where equation (167) was used with θ_r = constant and $\beta_{rr} = 0$ because $\theta_{\rho q}$ = constant. The corresponding complex number potential is given by

$$-\partial\bar{W}_r/\partial\bar{r} = 1/3\bar{\rho}_q\bar{r} \tag{361}$$

$$\bar{W}_r = -1/6\bar{\rho}_q\bar{r}^2 + \bar{c} \tag{362}$$

where the constant $\bar{c}$ is determined from the condition

$$\bar{W}_r(\bar{a}) = Q(a)/(4\pi\bar{a}) = 1/3\rho_q a^2 e^{-j\theta_a} \tag{363}$$

where $\bar{a}$ = complex number radius of the spherical charge. This gives

$$\bar{c} = 1/2\rho_q a^2 e^{-j\theta_a} \tag{364}$$

and therefore

$$\bar{W}_r = \rho_q/6(3a^2 - r^2)e^{-j\theta_a} \qquad r \leqslant a \tag{365}$$

where $\theta_r = \theta_a$ = constant. From equation (166) and $\beta_{rr} = 0$ it follows that

$$Q = 4/3\pi a^3\rho_q \tag{366}$$

and

$$\bar{W}_r = Q/(8\pi a^3)(3a^2 - r^2)e^{-j\theta_a} \qquad r \leqslant a \tag{367}$$

$$W_r = Q/(8\pi a^3)(3a^2 - r^2) \qquad \theta_{Wr} = -\theta_a \qquad r \leqslant a \tag{368}$$

From equation (359) it follows that

$$D_r = Qr/(4\pi a^3) \qquad \theta_{Dr} = -2\theta_a \qquad r \leqslant a \tag{369}$$

The measured value of the electric displacement for the case of constant charge density and for $r \leqslant a$ is

$$D_{rm} = D_r \cos\theta_{Dr} = Qr/(4\pi a^3)\cos(2\theta_a) \qquad r \leqslant a \tag{370}$$

and the measured potential is for $r \leqslant a$

$$W_{rm} = W_r \cos\theta_{Wr} = Q/(8\pi a^3)(3a^2 - r^2)\cos\theta_a \qquad r \leqslant a \tag{371}$$

The conventional calculation of electric displacement and potential is given for constant density and for $r \leqslant a$ by

$$D_{rc} = Qr_m/(4\pi a_m^3) = Qr/(4\pi a^3) \cos^{-2} \theta_a \tag{372}$$

$$W_{rc} = Q/(8\pi a_m^3)(3a_m^2 - r_m^2) = Q/(8\pi a^3)\ (3a^2 - r^2) \cos^{-1} \theta_a \tag{373}$$

and therefore

$$(D_{rc} - D_{rm})/D_{rc} = 1 - \cos^2 \theta_a \cos(2\theta_a) \tag{374}$$

$$(W_{rc} - W_{rm})/W_{rc} = \sin^2 \theta_a \tag{375}$$

where the fact that $\theta_r = \theta_a$ for $r \leqslant a$ was used.

The general case of any spherically symmetric charge distribution is obtained from equations (115) and (165) to be

$$1/\bar{r}^2 \partial/\partial\bar{r}(\bar{r}^2\bar{D}_r) = (4\pi\bar{r}^2)^{-1} dQ/d\bar{r} \tag{376}$$

which gives immediately

$$\bar{D}_r = (4\pi\bar{r}^2)^{-1}Q(r) = (4\pi r^2)^{-1}Q(r)e^{-2j\theta_r} \tag{377}$$

so that in general

$$D_r = (4\pi r^2)^{-1}Q(r) \qquad \theta_{DR} = -2\theta_r \tag{378}$$

For $r \geqslant a$ equations (377) and (378) give

$$\bar{D}_r = Q/(4\pi\bar{r}^2) = Q/(4\pi r^2)e^{-2j\theta_r} \qquad r \geqslant a \tag{379}$$

$$D_r = Q/(4\pi r^2) \qquad \theta_{Dr} = -2\theta_r \qquad r \geqslant a \tag{380}$$

and

$$\bar{W}_r = Q/(4\pi\bar{r}) = Q/(4\pi r)e^{-j\theta_r} \tag{381}$$

$$W_r = Q/(4\pi r) \qquad \theta_W = -\theta_r \qquad r \geqslant a \tag{382}$$

The measured electric displacement and potential for $r \geqslant a$ is

$$D_{rm} = Q/(4\pi r^2) \cos(2\theta_r) \qquad r \geqslant a \tag{383}$$

$$W_{rm} = Q/(4\pi r) \cos \theta_r \qquad r \geqslant a \tag{384}$$

The corresponding conventional calculations give

$$D_{rc} = Q/(4\pi r_m^2) = Q/(4\pi r^2)\ \cos^{-2}\theta_r \qquad r \geqslant a \qquad (385)$$

$$W_{rc} = Q/(4\pi r_m) = Q/(4\pi r)\ \cos^{-1}\theta_r \qquad r \geqslant a \qquad (386)$$

and therefore

$$(D_{rc} - D_{rm})/D_{rc} = 1 - \cos^2\theta_r \cos(2\theta_r) \qquad r \geqslant a \qquad (387)$$

$$(W_{rc} - W_{rm})/W_{rc} = \sin^2\theta_r \qquad r \geqslant a \qquad (388)$$

Equations (379) through (388) agree with equations (367) through (375) for $r = a$. For constant density $Q(r) = 4/3\pi r^3\rho_q$ and θ_r = constant and equations (377) and (378) become the constant density case described in equations (359) and (360) for $r \leqslant a$.

C. Line Charge in Broken Symmetry Space.

The line charge problem in broken symmetry space can be described by equation (115) written in cylindrical coordinates as

$$1/\bar{r}\,d/d\bar{r}(\bar{r}\bar{D}_r) = \bar{\rho}_q = d\bar{\rho}_{q\ell}/(2\pi\bar{r}d\bar{r}) \qquad (389)$$

where $\bar{\rho}_q$ is given by equation (177) and $\bar{\rho}_{q\ell}$ by equation (180). From equation (389) it follows that for a line charge of infinite length

$$\bar{D}_r = \bar{\rho}_{q\ell}/(2\pi\bar{r}) \qquad D_r = \rho_{q\ell}/(2\pi r) \qquad (390)$$

$$D_{rm} = D_r \cos\theta_{Dr} \qquad \theta_{Dr} = -\theta_z - \beta_{zz} - \theta_r \qquad (391)$$

The radial electric field is given by

$$\bar{E}_r = \bar{\rho}_{q\ell}/(2\pi\bar{r}\bar{\varepsilon}) \qquad E_r = \rho_{q\ell}/(2\pi r\varepsilon) \qquad (392)$$

$$F_{rm} = E_r \cos\theta_{Er} \qquad \theta_{Er} = -\theta_z - \beta_{zz} - \theta_r - \theta_\varepsilon \qquad (393)$$

If $\bar{\rho}_{q\ell}$ = constant then from equation (181) it follows that $\rho_{q\ell} = Q/z$ = constant, $\theta_{\rho q\ell} = -\theta_z$ = constant and $\beta_{zz} = 0$. For a line charge of finite length the electric displacement vector is given by the following generalization of the well known result[27]

$$\bar{D}_r = \bar{\rho}_{q\ell}/(2\pi\bar{r})\ \sin\bar{\alpha} \qquad (394)$$

where $\tan\bar{\alpha} = \bar{h}/\bar{r}$, and $2\bar{h}$ = complex number length of the line charge, and

where sin $\bar{\alpha}$ can be written as in equation (4). Equation (394) reduces to equation (390) for an infinite line of charge by noting that

$$S_\alpha(\pi/2) = 1 \qquad \theta_{s\alpha}(\pi/2) = 0 \tag{395}$$

In Reference 12 it is noted that only for the angles $\alpha = \pm \pi/2$ is $\theta_\alpha = 0$ and $\theta_{s\alpha} = 0$.

For a charged cylinder of infinite length a simple analysis gives the electric displacement for $r \leqslant a$ as

$$\bar{D}_r = \bar{\rho}_{q\ell}\bar{r}/(2\pi\bar{a}^2) = \bar{\rho}_q\bar{r}/2 \qquad D_r = \rho_{q\ell}r/(2\pi a^2) \tag{396}$$

$$D_{rm} = D_r \cos\theta_{Dr} \qquad \theta_{Dr} = \theta_r - 2\theta_a - \theta_z - \beta_{zz} \tag{397}$$

where $\bar{a}$ = complex radius of cylinder and $a_m = a \cos\theta_a$, $r_m = r \cos\theta_r$ with $\theta_a \sim \theta_r$.

D. Broken Symmetry Capacitance.

The generalization of the standard definition of capacitance to broken symmetry space is[18-28]

$$\bar{C}_\alpha = Q/\bar{W}_\alpha \tag{398}$$

$$C_\alpha = Q/W_\alpha \qquad \theta_{C\alpha} = -\theta_{W\alpha} = 2(\theta_t - \theta_\alpha) \tag{399}$$

For a parallel plate condenser this becomes[18-28]

$$\bar{C}_z = Q/(\bar{E}\bar{z}) \qquad \bar{D} = \bar{\varepsilon}\bar{E} = Q/\bar{A} \tag{400}$$

and therefore

$$\bar{C}_z = \bar{\varepsilon}\bar{A}_{xy}/\bar{z} \tag{401}$$

$$C_z = \varepsilon A_{xy}/z \qquad \theta_{Cz} = \theta_\varepsilon + \theta_x + \theta_y - \theta_z = 2(\theta_t - \theta_z) \tag{402}$$

with corresponding expressions for C_x and C_y . For a coaxial cylinder of length $\bar{h}$, outer radius $\bar{b}$ and inner radius $\bar{a}$ the capacitance is[18-28]

$$\bar{C} = 2\pi\bar{\varepsilon}\bar{h}/\ell n(\bar{b}/\bar{a}) \tag{403}$$

$$= 2\pi\varepsilon h e^{j(\theta_\varepsilon + \theta_h)}/[\ell n(b/a) + j(\theta_b - \theta_a)]$$

The magnitude C in this case depends on $\theta_b - \theta_a$ because

$$C = 2\pi\epsilon h/[\ell n^2(b/a) + (\theta_b - \theta_a)^2]^{1/2} \tag{404}$$

$$\theta_C = \theta_\epsilon + \theta_h - \tan^{-1}[(\theta_b - \theta_a)/\ell n(b/a)] \tag{405}$$

5. ASYMMETRIC MAGNETOSTATICS. This section considers the basic calculations of magnetostatics in a spacetime that has broken internal symmetries. The broken symmetries are due to the local pressure and energy density of the ambient matter. It is gravity that produces the pressure in the earth's atmosphere and therefore the local broken symmetry of spacetime at the earth's surface is due to gravity. The elementary calculations of magnetostatics are affected by gravity.

A. Ampere's Law for Asymmetric Space and Time.

The broken symmetry form of Ampere's law must be written as

$$\oint \bar{\vec{H}} \cdot d\bar{\vec{\ell}} = \bar{I}\bar{L}_c \tag{406}$$

where $\bar{L}_c$ is given by equation (330). This must be the broken symmetry form of Ampere's law because the factor $\bar{L}_c$ also arises from the line integral on the left hand side of equation (406). For circular symmetry

$$2\pi\bar{r}\bar{H} = \bar{I} \tag{407}$$

$$H = I/(2\pi r) \qquad \theta_H = \theta_I - \theta_r = -\theta_t - \theta_r - \beta_{tt} \tag{408}$$

This result must also follow from Maxwell's equations.

B. Magnetic Field Due to Currents in a Wire, Solenoid and a Toroid with Broken Spacetime Symmetry.

The stationary magnetic field for a current in a wire can be obtained from equations (113) and (220) which can be written as

$$1/\bar{r}d/d\bar{r}(\bar{r}\bar{H}_\phi) = \bar{j}_z = d\bar{I}/(2\pi\bar{r}d\bar{r}) \tag{409}$$

and therefore the azimuthal magnetic field associated with a current in the z direction is given by

$$\bar{H}_\phi = \bar{I}/(2\pi\bar{r}) \tag{410}$$

$$H_\phi = I/(2\pi r) \qquad \theta_{H\phi} = \theta_I - \theta_r = -\theta_t - \beta_{tt} - \theta_r \tag{411}$$

The measured and conventionally calculated magnetic fields are respectively

$$H_{\phi m} = H_\phi \cos\theta_{H\phi} = I/(2\pi r)\cos(\theta_t + \theta_r) \tag{412}$$

$$H_{\phi c} = I_m/(2\pi r_m) = I/(2\pi r)\cos\theta_t / \cos\theta_r \tag{413}$$

Combining equations (411) and (412) gives

$$H_{\phi m} = H_{\phi c}(1 - \tan\theta_r \tan\theta_t)\cos^2\theta_r \tag{414}$$

$$(H_{\phi c} - H_{\phi m})/H_{\phi c} = \sin\theta_r \cos\theta_r (\tan\theta_r + \tan\theta_t) \tag{415}$$

$$\sim \theta_r(\theta_r + \theta_t)$$

From equation (411) and (413) it follows that $H_{\phi m} = 0$ when $\theta_t + \theta_r = \pi/2$, and this combined with the free electron condition $\theta_z = 2\theta_t$ in the wire gives $\theta_z/2 + \theta_r = \pi/2$.

From equation (307) it follows that the magnetic vector potential for a static current in a wire situated in broken symmetry spacetime is given by

$$\bar{B}_\phi = \bar{\mu}\bar{H}_\phi = -\partial\bar{A}_z/\partial\bar{r} \tag{416}$$

Combining equations (410) and (416) gives

$$\bar{A}_z = -\bar{\mu}\bar{I}/(2\pi)\ \ell n\ \bar{r} \tag{417}$$

$$= \bar{\mu}\bar{I}/(2\pi)\ \ell n(1/\bar{r})$$

or

$$\bar{A}_z = A_z e^{j\theta Az} \tag{418}$$

where

$$A_z = -\mu I/(2\pi)[\ell n^2\ r + \theta_r^2]^{1/2} \tag{419}$$

$$\theta_{Az} = \Phi_{\theta r} + \theta_I + \theta_\mu = \Phi_{\theta r} - \theta_t - \beta_{tt} + \theta_\mu \tag{420}$$

$$\tan\Phi_{\theta r} = \theta_r/\ell n\ r \tag{421}$$

The analysis of the coaxial cable in broken symmetry spacetime is a simple extension of the standard analysis.[18-28]

For an infinite solenoid the generalization to broken symmetry spacetime is[18-28]

$$\bar{H}_z = \bar{n}'\bar{I}_\phi \tag{422}$$

$$H_z = n'I_\phi \qquad \theta_{Hz} = -\theta_t - \theta_z - \beta_{tt} \tag{423}$$

where $\bar{n}'$ = number of turns per unit length = $N/\bar{z}$. For a toroid

$$\bar{H}_\phi = N\bar{I}/(2\pi\bar{r}) \tag{424}$$

$$H_\phi = NI/(2\pi r) \qquad \theta_{H\phi} = -\theta_t - \beta_{tt} - \theta_r \tag{425}$$

C. Biot-Savart Law for Broken Symmetry Spacetime.

The Biot-Savart law applied to the calculation of the magnetic field of a current in a thin wire is given by the following complex number extension of a well known result[18-28]

$$\bar{H}_\phi = \bar{I}/(4\pi)\int_{-\infty}^{\infty} \bar{r}/(\bar{r}^2 + \bar{z}^2)^{3/2}\, d\bar{z} \tag{426}$$

Introducing $\bar{z} = \bar{r}\tan\bar{\psi}$ where $\bar{\psi} = \psi\exp(j\theta_\psi)$ gives

$$\bar{H}_\phi = \bar{I}/(4\pi\bar{r})\int_{\bar{\psi}_1}^{\bar{\psi}_2} \cos\bar{\psi}\, d\bar{\psi} \tag{427}$$

$$= \bar{I}/(4\pi\bar{r})(\sin\bar{\psi}_2 - \sin\bar{\psi}_1) \tag{428}$$

$$= \bar{I}/(2\pi\bar{r})\ \sin\bar{\psi}_2$$

where

$$\bar{\psi}_2 = \pi/2\ e^{j\theta_\psi(\pi/2)} \tag{429}$$

$$\bar{\psi}_1 = -\pi/2\ e^{j\theta_\psi(-\pi/2)} \tag{430}$$

But it has been shown in Reference 12 that $\theta_\psi(\pm\pi/2) = 0$, and therefore

$$\bar{H}_\phi = \bar{I}/(2\pi\bar{r}) \tag{431}$$

which agrees with equation (408) which comes immediately from the broken symmetry form of Ampere's law given in equation (406). Because $\theta_\psi(\pm\pi/2) = 0$ it follows that the Biot-Savart does not suggest that internal phase angles of angles should appear in the expression for the magnetic field, and this substantiates the inclusion of the angular factor $\bar{L}_c$ in the right hand side of equation (406) which excludes angular internal phase angles from appearing in the expression for $\bar{H}_\phi$.

It is easy to show that the broken symmetry generalization for the axis value of the magnetic field of a circular current loop of radius $\bar{a}$ at a point $\bar{z}$ on the axis above the loop is given by[18-28]

$$\bar{H}_z = \bar{I}\bar{a}^2/(2\bar{R}^3) \tag{432}$$

where

$$\bar{R}^2 = \bar{a}^2 + \bar{z}^2 \tag{433}$$

Therefore

$$H_z = Ia^2/(2R^3) \tag{434}$$

$$\theta_{Hz} = \theta_I + 2\theta_a - 3\theta_R = -\theta_t - \beta_{tt} + 2\theta_a - 3\theta_R \tag{435}$$

where R and θ_R are obtained from the following two component equations associated with equation (433)

$$R^2 \cos(2\theta_R) = a^2 \cos(2\theta_a) + z^2 \cos(2\theta_z) \tag{436}$$

$$R^2 \sin(2\theta_R) = a^2 \sin(2\theta_a) + z^2 \sin(2\theta_z) \tag{437}$$

D. Asymmetric Hall Effect

A description of the Hall effect is given in detail in the literature.[29] In broken symmetry spacetime the application of transverse electric and magnetic fields to a solid conductor or semiconductor induces a complex number electric field due to charge drift which is given by

$$\bar{E}_x = \bar{\delta}\bar{E}_y\bar{B}_z \tag{438}$$

where $\bar{\delta}$ = complex number ion mobility (Hall mobility) given by

$$\bar{\delta} = \bar{\sigma}/(q\bar{n}) \tag{439}$$

$$\delta = \sigma/(qn) \qquad \theta_\delta = \theta_\sigma - \theta_n = \theta_t \tag{440}$$

From equations (288) through (290) and (438) through (440) it follows that

$$E_x = \delta E_y B_z \tag{441}$$

$$\theta_{Ex} - \theta_{Ey} = \theta_\delta + \theta_{Bz} = \theta_\sigma - \theta_n + \theta_{Bz} \tag{442}$$

$$= \theta_t + \theta_{Bz}$$

$$= \theta_x - \theta_y$$

In fact

$$\theta_{Ex} = \theta_y - \theta_t + \theta_{Bz} = \theta_x - 2\theta_t \tag{443}$$

$$\theta_{Ey} = \theta_y - 2\theta_t \tag{444}$$

The Hall effect, both classical and quantum, may possibly be used to determine θ_x , θ_y and θ_t .

6. ELECTROMAGNETIC WAVES AND BROKEN SYMMETRY. This section considers the effects of broken symmetry spacetime on the propagation of electromagnetic waves in matter and the vacuum. In bulk matter it is gravity which is primarily responsible for the broken symmetry of space and time.[12] Therefore gravity is expected to influence the propagation of electromagnetic waves in the vicinity of the planets and stars. The effects of the broken symmetry of spacetime on the propagation of electromagnetic waves in the vicinity of the earth may be expected to be larger than the effects of a gravitational redshift (general relativity) which is significant only in massive bodies like stars.

A. Asymmetric Electromagnetic Wave Equations.

Consider broken symmetry electromagnetic waves propagating in the z direction. Then equation (113) becomes

$$\partial \bar{H}_y/\partial \bar{z} = - \partial \bar{D}_x/\partial \bar{t} \tag{445}$$

$$\partial \bar{H}_x/\partial \bar{z} = \partial \bar{D}_y/\partial \bar{t} \tag{446}$$

Equation (445) is equivalent to

$$\cos \beta_{zz} \sec \beta_{Hyz} \, \partial H_y/\partial z = - \cos \beta_{tt} \sec \beta_{Dxt} \, \partial D_x/\partial t \tag{447}$$

$$\theta_{Hy} + \beta_{Hyz} - \theta_z - \beta_{zz} = \theta_{Dx} + \beta_{Dxt} - \theta_t - \beta_{tt} \tag{448}$$

where

$$\tan \beta_{Hyz} = (H_y \partial \theta_{Hy}/\partial z)/(\partial H_y/\partial z) \tag{449}$$

$$\tan \beta_{Dxt} = (D_x \partial \theta_{Dx}/\partial t)/(\partial D_x/\partial t) \tag{450}$$

Similarly, equation (446) is written as

$$\cos \beta_{zz} \sec \beta_{Hxz} \, \partial H_x/\partial z = \cos \beta_{tt} \sec \beta_{Dyt} \, \partial D_y/\partial t \tag{451}$$

$$\theta_{Hx} + \beta_{Hxz} - \theta_z - \beta_{zz} = \theta_{Dy} + \beta_{Dyt} - \theta_t - \beta_{tt} \tag{452}$$

where

$$\tan \beta_{Hxz} = (H_x \partial \theta_{Hx}/\partial z)/(\partial H_x/\partial z) \tag{453}$$

$$\tan \beta_{Dyt} = (D_y \partial \theta_{Dy}/\partial t)/(\partial D_y/\partial t) \tag{454}$$

The Maxwell equation (112) for broken symmetry waves becomes

$$\partial \bar{E}_y/\partial \bar{z} = \partial \bar{B}_x/\partial \bar{t} \tag{455}$$

$$\partial \bar{E}_x/\partial \bar{z} = -\partial \bar{B}_y/\partial \bar{t} \tag{456}$$

Equation (455) can be rewritten as the following two equations

$$\cos \beta_{zz} \sec \beta_{Eyz} \, \partial E_y/\partial z = \cos \beta_{tt} \sec \beta_{Bxt} \, \partial B_x/\partial t \tag{457}$$

$$\theta_{Ey} + \beta_{Eyz} - \theta_z - \beta_{zz} = \theta_{Bx} + \beta_{Bxt} - \theta_t - \beta_{tt} \tag{458}$$

where

$$\tan \beta_{Eyz} = (E_y \partial \theta_{Ey}/\partial z)/(\partial E_y/\partial z) \tag{459}$$

$$\tan \beta_{Bxt} = (B_x \partial \theta_{Bx}/\partial t)/(\partial B_x/\partial t) \tag{460}$$

while equation (458) yields

$$\cos \beta_{zz} \sec \beta_{Exz} \, \partial E_x/\partial z = -\cos \beta_{tt} \sec \beta_{Byt} \, \partial B_y/\partial t \tag{461}$$

$$\theta_{Ex} + \beta_{Exz} - \theta_z - \beta_{zz} = \theta_{By} + \beta_{Byt} - \theta_t - \beta_{tt} \tag{462}$$

where

$$\tan \beta_{Exz} = (E_x \partial \theta_{Ex}/\partial z)/(\partial E_x/\partial z) \tag{463}$$

$$\tan \beta_{Byt} = (B_y \partial \theta_{By}/\partial t)/(\partial B_y/\partial t) \tag{464}$$

Combining equations (448) and (462) or equations (452) and (458) gives for electromagnetic waves in matter

$$\theta_\mu + \theta_\varepsilon = 2(\theta_t + \beta_{tt} - \theta_z - \beta_{zz}) \qquad \theta_\varepsilon \sim 2\theta_t - 3\theta_z \qquad \theta_\mu \sim \theta_z \tag{465}$$

where equation (465) corresponds to the wave velocity equation $\bar{v}^2 = 1/(\bar{\varepsilon}\bar{\mu})$ and where uniform material has been assumed so that $\beta_{Dxt} = \beta_{Ext}$, $\beta_{Byt} = \beta_{Hyt}$ and so on. For the vacuum, equation (465) becomes

$$\theta_\mu + \theta_\varepsilon = 0 \tag{466}$$

because the light speed in vacuum has a zero internal phase angle and $\theta_z = \theta_t$ for photons in vacuum. For electromagnetic waves in matter $\theta_\mu + \theta_\varepsilon$ is determined through equation (465) by the values of θ_z and θ_t which depend on the local energy density and pressure and ultimately on gravity.

B. Electromagnetic Waves in Broken Symmetry Matter.

Consider plane electromagnetic waves propagating in the z direction. Assuming $\bar{\varepsilon}$ and $\bar{\mu}$ are constants allows equation (445), (446), (455) and (456) to be written as the following wave equation

$$\partial^2\bar{E}_x/\partial\bar{z}^2 = 1/\bar{v}^2\,\partial^2\bar{E}_x/\partial\bar{t}^2 \tag{467}$$

with similar equations for $\bar{E}_y$, $\bar{H}_x$ and $\bar{H}_y$, where $\bar{v}^2 = 1/(\bar{\varepsilon}\bar{\mu})$. If a product solution of the form $\bar{E}_x = \bar{Z}(\bar{z})\bar{T}(\bar{t})$ is assumed then equation (467) becomes

$$d^2\bar{T}/d\bar{t}^2 + \bar{\omega}^2\bar{T} = 0 \tag{468}$$

$$d^2\bar{Z}/d\bar{z}^2 + \bar{k}^2\bar{Z} = 0 \tag{469}$$

where $\bar{v} = \bar{\omega}/\bar{k}$, and where $\bar{k}$ = broken symmetry wave number generalization of a standard result[27]

$$\bar{k}^2 = \bar{\mu}\bar{\varepsilon}\bar{\omega}^2 + i\bar{\mu}\bar{\sigma}\bar{\omega} \tag{470}$$

and $\bar{k} = \bar{\alpha} + i\bar{\beta}$ where $\bar{\alpha}$ and $\bar{\beta}$ are the following broken symmetry generalizations of the standard equations[27]

$$\bar{\alpha}^2 = \bar{\omega}^2\bar{\mu}\bar{\varepsilon}/2\ \{[1 + \bar{\sigma}^2/(\bar{\varepsilon}^2\bar{\omega}^2)]^{1/2} + 1\} \tag{471}$$

$$\bar{\beta}^2 = \bar{\omega}^2\bar{\mu}\bar{\varepsilon}/2\ \{[1 + \bar{\sigma}^2/(\bar{\varepsilon}^2\bar{\omega}^2)]^{1/2} - 1\} \tag{472}$$

from which α, θ_α, β and θ_β can be obtained. Note that $\bar{\alpha}$ and $\bar{\beta}$ are written as

$$\bar{\alpha} = \alpha e^{j\theta_\alpha} \qquad \bar{\beta} = \beta e^{j\theta_\beta} \tag{473}$$

The solutions of equations (465) and (469) are respectively

$$\bar{T} = T_o e^{-i\bar{\omega}\bar{t}} \tag{474}$$

$$\bar{Z} = Z_o e^{i\bar{k}\bar{z}} \tag{475}$$

Then the broken symmetry electric field is

$$\bar{E}_x = E_x^o e^{-\bar{\beta}\bar{z}}\, e^{i(\bar{\alpha}\bar{z} - \bar{\omega}\bar{t})} \tag{476}$$

Writing the broken symmetry electric field as

$$\bar{E}_x = E_x e^{j\theta Ex} \tag{477}$$

gives the following equations

$$E_x = E_x^o \, e^{-\beta z \cos(\theta_\beta + \theta_z)} \, e^{i[\alpha z \cos(\theta_\alpha + \theta_z) - \omega t \cos(\theta_\omega + \theta_t)]} \tag{478}$$

$$\theta_{Ex} = -\beta z \sin(\theta_\beta + \theta_z) + i[\alpha z \sin(\theta_\alpha + \theta_z) - \omega t \sin(\theta_\omega + \theta_t)] \tag{479}$$

Following Reference 12 $\bar{\omega}\bar{t}$ and $\bar{\alpha}\bar{z}$ must be real numbers for harmonic waves, and therefore equations (478) and (479) become respectively

$$E_x = E_x^o e^{-\beta z \cos(\theta_\beta + \theta_z)} \, e^{i(\alpha z - \omega t)} \tag{480}$$

$$\theta_{Ex} = -\beta z \sin(\theta_\beta + \theta_z) \tag{481}$$

where the following periodicity conditions were used[12]

$$\theta_\omega = -\theta_t \tag{482}$$

$$\theta_\alpha = -\theta_z \tag{483}$$

which correspond to equation (476) of the form

$$\bar{E}_x = E_x^o e^{-\bar{\beta}\bar{z}} \, e^{i(\alpha z - \omega t)} \tag{484}$$

For periodic electromagnetic waves in broken symmetry space and time the complex number propagation constants $\bar{\omega}$ and $\bar{\alpha}$ must adjust themselves to the local spacetime phase structure such that equations (482) and (483) are satisfied. Equation (484) can also be written in terms of measured spacetime coordinates z_m and t_m and measured propagation constants α_m and ω_m .[12]

For the case of periodic waves with zero damping $\bar{\sigma} = 0$ and equations (465), (471) and (472) show that $\beta = 0$ and

$$\theta_\omega = -\theta_t \tag{485}$$

$$\theta_\alpha = \theta_\beta = -\theta_z \tag{486}$$

and

$$E_x = E_x^o e^{i(\alpha z - \omega t)} = E_x^o e^{i(\alpha_m z_m/\cos^2 \theta_z - \omega_m t_m/\cos^2 \theta_t)} \tag{487}$$

$$\theta_{Ex} = 0 \tag{488}$$

which corresponds to undamped waves periodic in spacetime with broken internal symmetries that satisfy equations (485) and (486). Because the local internal phase structure of spacetime in bulk matter is determined primarily by gravi-

tation it follows that equations (478) through (488) describe the effects of gravity on electromagnetic wave propagation.

C. Doppler Effect for Waves with Broken Symmetry.

The generalization of the conventional Doppler formula for the frequency of waves from a source moving with velocity $\bar{v}$ at an angle $\bar{\psi}$ relative to the direction of a stationary observer is given by

$$\bar{f}' = \bar{f}(1 + \bar{v}/c \cos \bar{\psi}) \tag{489}$$

where $\bar{f}'$ and $\bar{f}$ = complex number frequencies at the observer and source respectively, $\bar{v} = ve^{j\theta v}$ = complex number speed of the source and $\bar{\psi}$ = complex number angle between the direction of the source and the direction of the observer. The complex number frequencies are written as

$$\bar{f} = fe^{j\theta f} = 1/\bar{T} \qquad \bar{f}' = f'e^{j\theta f'} = 1/\bar{T}' \tag{490}$$

$$= 1/T\, e^{-j\theta t} \qquad = 1/T'\, e^{-j\theta t'}$$

where T and T' = magnitudes of complex number periods of the wave motion at the source and observer respectively, and $\theta_t = -\theta_f$ and $\theta_t' = -\theta_f'$. The conventional Doppler formula can be written as[27]

$$f_c' = f_c(1 + v_m/c \cos \psi_m) \tag{491}$$

where f_c' = conventionally calculated frequency expected to be seen at the observer, f_c = conventionally calculated frequency at source, v_m = measured source speed, and ψ_m = measured angle between the source direction of motion and the observer's direction. The conventionally calculated frequencies are given by

$$f_c = 1/T_m \qquad f_c' = 1/T_m' \tag{492}$$

where T_m and T_m' = measured wave periods at the source and observer respectively. Frequencies are derived from periods so that for broken symmetry spacetime

$$f = 1/T \qquad f' = 1/T' \tag{493}$$

$$T_m = T \cos \theta_t \qquad T_m' = T' \cos \theta_t' \tag{494}$$

Equation (493) holds only for the magnitudes of the frequency and period, not for their measured values.

From equation (489) it follows that

$$f' \cos \theta_f' = f[\cos \theta_f + vC_\psi/c \cos(\theta_f + \theta_v - \theta_{c\psi})] \tag{495}$$

$$f' \sin \theta_f' = f[\sin \theta_f + vC_\psi/c \sin(\theta_f + \theta_v - \theta_{c\psi})] \tag{496}$$

where C_ψ and $\theta_{c\psi}$ are given by equations (7) and (9) respectively. It is easy to see that equation (495) can be rewritten as

$$f_m' = f_m(1 + v_m/c\ F) \tag{497}$$

where F is given by

$$F = C_\psi \cos(\theta_f + \theta_v - \vartheta_{c\psi})/(\cos\theta_f \cos\theta_v) \tag{498}$$

and where f_m' and f_m = frequency measured at the observer and at the source respectively and are given by[12]

$$\begin{aligned} f_m &= f\cos\theta_f & f_m' &= f'\cos\theta_f' \\ &= 1/T_m \cos^2\theta_t & &= 1/T_m' \cos^2\theta_t' \\ &= f_c \cos^2\theta_t & &= f_c' \cos^2\theta_t' \end{aligned} \tag{499}$$

and

$$(f_c - f_m)/f_c = \sin^2\theta_t \tag{500}$$

$$(f_c' - f_m')/f_c' = \sin^2\theta_t' \tag{501}$$

Combining equations (491) and (497) gives

$$f_c' - f_m' = f_c - f_m + v_m/c\ (f_c \cos\psi_m - f_m F) \tag{502}$$

for the difference between the measured and conventionally predicted frequencies at the observers position. Equation (502) can also be written as

$$f_c' \sin^2\theta_t' - f_c \sin^2\theta_t = v_m/c\ (f_c \cos\psi_m - f_m F) \tag{503A}$$

$$f_c' \cos^2\theta_t' = f_c \cos^2\theta_t\ (1 + v_m/c\ F) \tag{503B}$$

When $\psi = \pi/2$, then[12]

$$C_\psi(\pi/2) = 0 \qquad \theta_{c\psi}(\pi/2) \neq 0 \qquad \theta_\psi(\pi/2) = 0 \tag{504}$$

$$\psi_m = \psi = \pi/2 \qquad F = 0 \tag{505}$$

and the conventional equation (491) gives $f_c' = f_c$ while the broken symmetry equation (497) gives $f_m' = f_m$ so that $f_c' - f_m' = f_c - f_m$ for this case as can be seen from equation (502). For this case also $\theta_t' = \theta_t$.

When $\psi = 0$ it follows that

$$C_\psi(0) = 1 \qquad \theta_{c\psi}(0) = 0 \qquad \theta_\psi(0) \neq 0 \tag{506}$$

$$F = \cos(\theta_f + \theta_v)/(\cos\,\theta_f \cos\,\theta_v) \tag{507}$$

$$= 1 - \tan\,\theta_f \tan\,\theta_v$$

$$= 1 + \tan\,\theta_t \tan\,\theta_v$$

$$\sim 1 + \theta_t\theta_v \sim 1 + \theta_t(\theta_x - \theta_t)$$

From equation (502) with $\psi = \psi_m = 0$ it follows that

$$f_c' - f_m' = (f_c - f_m)(1 + v_m/c) - v_m f_m/c \tan\,\theta_t \tan\,\theta_v \tag{508}$$

$$\sim (f_c - f_m)(1 + v_m/c) - v_m f_m/c\;\theta_t\theta_v$$

Measurements of $f_m' - f_c'$ and $f_m - f_c$ may possibly be used to determine θ_t , θ_x , θ_y , θ_z and θ_t' , θ_x' , θ_y' , θ_z' . For motion of the source away from the observer $\psi = \pi$, and the value of ψ_m to be used in equations (502) or (503) is $\psi_m = \pi \cos\,\theta_\psi(\pi)$.

7. CONCLUSIONS. The broken symmetry of space and time affects the elementary calculations of the measured physical quantities of electromagnetism such as charge and current densities and the electric and magnetic field vectors. The effects are manifested through the internal phase angles of the charge and current densities and the field vectors, which in turn are related to the internal phase angles of space and time coordinates. In the vicinity of the earth the internal phase angles of the spacetime coordinates and of the electromagnetic field vectors are determined primarily by gravity. Therefore gravitation must be considered to affect the basic calculations of electromagnetic theory. The gravity induced broken symmetry of spacetime should have measurable consequences in simple electromagnetic phenomena as the electric resistance of wires, the propagation of electromagnetic waves in matter, and the Hall effect. In some cases atomic and molecular structure effects may induce a broken symmetry in the space and time coordinates which is greater over macroscopic distances than the corresponding effect due to gravity. This is true for coherent states such as occur with superconductivity. In this case the induced coherent time state may be responsible for the high T_c superconductivity phenomenon in the planar copper oxides. The coherent time state elevates the normalized superconductivity energy gap by a factor $6/\pi \sim 1.91$ and may explain the experimentally observed enhanced normalized energy gaps associated with high temperature superconductors.

ACKNOWLEDGEMENT

The author wishes to thank Elizabeth K. Klein for typing this paper.

REFERENCES

1. Frampton, P. H., Gauge Field Theories, Benjamin/Cummings, Menlo Park, 1987.

2. Huang, K., Quarks, Leptons & Gauge Fields, World Scientific, Singapore, 1982.

3. Quigg, C., Gauge Theories of the Strong, Weak, and Electromagnetic Interactions, Benjamin/Cummings, Menlo Park, 1983.

4. Ludwig, W. and Falter, C., Symmetries in Physics, Springer-Verlag, New York, 1988.

5. Weinberg, S., Gravitation and Cosmology, John Wiley, New York, 1972.

6. Zee, A., Unity of Forces in the Universe, Vols. 1 & 2, World Scientific, Singapore, 1982.

7. Eddington, A. S., The Mathematical Theory of Relativity, Cambridge Univ. Press, Cambridge, 1952.

8. Møller, C., The Theory of Relativity, Oxford Univ. Press, Oxford, 1955.

9. Misner, C. W., Thorne, K. S., and Wheeler, J. A., Gravitation, Freeman, San Francisco, 1973.

10. Robertson, H. P. and Noonan, T. W., Relativity and Cosmology, Saunders, Philadelphia, 1968.

11. Weiss, R. A., Relativistic Thermodynamics, Exposition Press, New York, 1976.

12. Weiss, R. A., Gauge Theory of Thermodynamics, K & W Publications, Vicksburg, MS, 1989.

13. Morse, P. M. and Feshbach, H., Methods of Theoretical Physics, Parts 1 & 2, McGraw-Hill, New York, 1953.

14. Slater, J. C. and Frank, N. H., Introduction to Theoretical Physics, McGraw-Hill, New York, 1933.

15. Houston, W. V., Principles of Mathematical Physics, McGraw-Hill, New York, 1948.

16. Page, L., Introduction to Theoretical Physics, Van Nostrand, New York, 1952.

17. Joos, G., Theoretical Physics, Hafner, New York, 1950.

18. Jackson, J., Classical Electrodynamics, John Wiley, New York, 1975.

19. Panofsky, W. and Phillips, M., Classical Electricity and Magnetism, Addison-Wesley, Reading, MA, 1955.

20. Stratton, J., Electromagnetic Theory, McGraw-Hill, New York, 1941.

21. Smythe, W., Static and Dynamic Electricity, McGraw-Hill, New York, 1950.

22. Becker, R., Electromagnetic Fields and Interactions, Dover, New York, 1964.

23. Jeans, J., The Mathematical Theory of Electricity and Magnetism, Cambridge University Press, Cambridge, 1951.

24. Menzel, D., Mathematical Physics, Prentice-Hall, New York, 1953.

25. Whittaker, E., A History of the Theories of Aether and Electricity, Philosophical Library, New York, 1951.

26. Sommerfeld, A., Electrodynamics, Academic, New York, 1952.

27. Kong, J., Electromagnetic Wave Theory, John Wiley, 1986.

28. Mason, M. and Weaver, W., The Electromagnetic Field, Dover, New York, 1929.

29. Seymour, J., Electronic Devices and Components, John Wiley, New York, 1981.

30. Margaritondo, G., Huber, D. L. and Olson, C. G., "Photoemission Spectroscopy of the High-Temperature Superconductivity Gap", Science, Vol. 246, 10 November 1989, p. 770.

QUANTUM MECHANICS AND THE BROKEN SYMMETRY OF SPACE AND TIME

Richard A. Weiss
U. S. Army Engineer Waterways Experiment Station
Vicksburg, Mississippi 39180

ABSTRACT. The quantum mechanics of particles located in spacetime with broken internal symmetries is investigated. The forms of the momentum and angular momentum operators in asymmetric space and time are developed. Schrödinger's equation is written for both external and internal motion in asymmetric spacetime, and the eigenvalues and eigenfunctions for a free particle and a rotating object are determined. The problem of a particle in a box is considered for both an internal and an external space box. The energy eigenvalues and eigenfunctions for the one- and three-dimensional harmonic oscillator with broken internal symmetries are determined, and the concept of an internal space harmonic oscillator is introduced. The addition of angular momenta in a spacetime with broken azimuthal symmetry is examined and the existence of a corresponding gauge boson is investigated. In particular, the addition of spin and orbital angular momentum is considered for broken symmetry space and time, and the eigenvalues of the total angular momentum operator are calculated. The general problem of measurement in quantum mechanics is considered and the dichotomy of spacetime points in a continuum and the Heisenberg uncertainty principle is examined. Applications to high temperature superconductivity are suggested.

1. INTRODUCTION. The concepts of broken global and local symmetries have a long history in quantum mechanics.[1-3] Even in classical mechanics the concept of symmetry plays an important role; consider only the fact that the translational and rotational symmetry of space gives rise to the laws of conservation of momentum and angular momentum respectively, and translational symmetry in time yields the law of conservation of energy.[3-5] For instance the application of an external torque breaks the rotational symmetry of a system and the angular momentum becomes time dependent. In quantum mechanics the concept of broken symmetry becomes even more important when it is applied to global and gauge (local) symmetries.[1-3] For example, a broken global symmetry is associated with a massless scalar particle called the scalar Goldstone boson, while a broken gauge symmetry is associated with massive vector bosons and massive scalar bosons.[1,2] An example of the particles associated with a broken gauge symmetry are the massive W^+, W^-, Z° vector bosons and the neutral massive scalar Higgs boson H° that are associated with the standard electroweak theory.[1-3] At the macroscopic scale, a gauge theory of bulk matter has been developed on the basis of a relativistic trace equation whose form suggests the broken symmetries of the thermodynamic functions.[6,7]

Recently a new form of broken symmetry associated with spacetime coordinates has been proposed.[7] The broken symmetry of the spacetime coordinates is associated with the broken symmetry of the pressure field in matter and the vacuum.[7]

The manifestation of the broken symmetry of spacetime appears in the internal phase angles of the space and time coordinates. Euler's equation of motion relates the internal angle of the pressure to the internal phase angles of the coordinates.[7] The skewed nature of spacetime affects classical mechanics, hydrodynamics, the equilibrium of stars and planets, electromagnetism, atomic processes and the structure of atoms.[7]

The complex number cartesian space and time coordinates are written as[7]

$$\bar{\alpha} = \alpha e^{j\theta_\alpha} \tag{1}$$

$$\bar{t} = te^{j\theta_t} \tag{2}$$

where $\alpha = x, y, z$. Quantities which often occur in the calculations with broken symmetry spacetime are the angles $\beta_{\alpha\alpha}$ and β_{tt} which are defined by

$$\tan \beta_{\alpha\alpha} = \alpha \partial\theta_\alpha / \partial\alpha \tag{3}$$

$$\tan \beta_{tt} = t \partial\theta_t / \partial t \tag{4}$$

The measured values of the space and time coordinates are given by[7]

$$\alpha_m = \alpha \cos \theta_\alpha \tag{5}$$

$$t_m = t \cos \theta_t \tag{6}$$

and therefore

$$d\alpha_m/d\alpha = \cos \theta_\alpha - \alpha_m \tan \theta_\alpha \, d\theta_\alpha/d\alpha \tag{7}$$

$$dt_m/dt = \cos \theta_t - t_m \tan \theta_t \, d\theta_t/dt \tag{8}$$

The kinematics of particles in broken symmetry spacetime is treated in Reference 7. The definition of measured coordinates in equations (5) and (6) alleviates the dichotomy in quantum electrodynamics of having zero dimensional points in a spacetime continuum simultaneously with the validity of the Heisenberg uncertainty principle which implies infinite energy and momentum for these points. From equations (5) and (6) it is clear that many values of α and t can correspond to the measured values α_m and t_m respectively. A measured point in spacetime actually corresponds to an infinite set of possible values of α and t corresponding to the possible ambient conditions which determine the size of the internal phase angles of the coordinates. The apparent infinite momentum and energy values predicted by the uncertainty principle for points in spacetime do not occur for measured spacetime coordinates.

One of the peculiarities of space and time with broken internal symmetries is the possibility of internal motion of a particle which is externally at rest in spacetime.[7] In particular, if the magnitude of $\bar{\alpha}$ is fixed and t is taken to

be a scalar parameter then

$$d\bar{\alpha}/dt = j\bar{\alpha}d\theta_\alpha/dt \tag{9}$$

If the time is also treated as a complex number and if the magnitudes α and t are both fixed, then space and time rotate internally and

$$d\bar{\alpha}/d\bar{t} = \bar{\alpha}/\bar{t}d\theta_\alpha/d\theta_t = \alpha/td\theta_\alpha/d\theta_t e^{j(\theta_\alpha - \theta_t)} \tag{10}$$

where $\alpha = x, y, z$. With internal spin it is possible to have momentum and kinetic energy for an externally stationary particle (with α = constant and t = constant). For external motion θ_α and θ_t = constants while α and t are variables. However, the measured values of space and time coordinates given by equations (5) and (6) vary for both internal and external motion.

The angles of a spherical polar coordinate system are written as

$$\bar{\phi} = \phi e^{j\theta\phi} \qquad \bar{\psi} = \psi e^{j\theta\psi} \tag{11}$$

where $\bar{\phi}$ and $\bar{\psi}$ = complex number azimuthal angle and zenith angle respectively. The relevant trigonometric functions associated with $\bar{\phi}$ and $\bar{\psi}$ are

$$\sin \bar{\phi} = S_\phi e^{j\theta s\phi} \tag{12}$$

$$\cos \bar{\phi} = C_\phi e^{-j\theta c\phi} \tag{13}$$

$$\tan \bar{\phi} = S_\phi/C_\phi \, e^{j(\theta s\phi + \theta c\phi)} \tag{14}$$

where[7]

$$S_\phi = [\sin^2(\phi \cos \theta_\phi) + \sinh^2(\phi \sin \theta_\phi)]^{1/2} \tag{15}$$

$$C_\phi = [\cos^2(\phi \cos \theta_\phi) + \sinh^2(\phi \sin \theta_\phi)]^{1/2} \tag{16}$$

$$\tan \theta_{s\phi} = \cot(\phi \cos \theta_\phi)\tanh(\phi \sin \theta_\phi) \tag{17}$$

$$\tan \theta_{c\phi} = \tan(\phi \cos \theta_\phi)\tanh(\phi \sin \theta_\phi) \tag{18}$$

with similar expressions for $\bar{\psi}$. The measured angles are given by

$$\phi_m = \phi \cos \theta_\phi \qquad \psi_m = \psi \cos \theta_\psi \tag{19}$$

The measured sin, cos and tan functions are

$$(\sin \phi)_m = S_\phi \cos \theta_{s\phi} \tag{20}$$

$$(\cos \phi)_m = C_\phi \cos \theta_{c\phi} \tag{21}$$

$$(\tan \phi)_m = S_\phi / C_\phi \cos(\theta_{s\phi} + \theta_{c\phi}) \tag{22}$$

From equation (19) it follows that

$$\partial \phi_m / \partial \phi = \cos \theta_\phi - \phi_m \tan \theta_\phi \, \partial \theta_\phi / \partial \phi \tag{23}$$

$$\partial \psi_m / \partial \psi = \cos \theta_\psi - \psi_m \tan \theta_\psi \, \partial \theta_\psi / \partial \psi \tag{24}$$

In this way the measured values of angles and their trigonometric functions are calculated.

This paper investigates the effects of the skewed nature of space and time on some basic quantum systems. In particular Section 2 considers skewed single particle momentum, angular momentum and energy operators, Section 3 studies the Schrödinger equation for asymmetric spacetime, Section 4 develops the theory of a particle confined to a box in external and internal space, Section 5 calculates the eigenvalues and eigenfunctions for the harmonic oscillator with broken internal symmetry, and finally Section 6 investigates the addition of angular momentum in broken symmetry spacetime.

2. MOMENTUM, ANGULAR MOMENTUM AND ENERGY OPERATORS IN ASYMMETRIC SPACETIME.

This section develops the expressions for the measured values of the momentum, angular momentum and energy operators in space and time with broken internal symmetries.

A. Momentum and Energy Operators.

The complex number momentum operators for cartesian coordinates with broken internal symmetries are[7]

$$\bar{p}_\alpha = e^{j\theta p\alpha} \, p_\alpha = - i\hbar \partial / \partial \bar{\alpha} \tag{25}$$

where $\alpha = x, y, z$ and

$$p_\alpha = - i\hbar \cos \beta_{\alpha\alpha} \, \partial / \partial \alpha \tag{26}$$

$$\theta_{p\alpha} = - \theta_\alpha - \beta_{\alpha\alpha} \tag{27}$$

where $\beta_{\alpha\alpha}$ is given by equation (3). The energy operator is given by[7]

$$\bar{E} = i\hbar \partial / \partial \bar{t} \tag{28}$$

$$E = i\hbar \cos \beta_{tt} \, \partial / \partial t \tag{29}$$

$$\theta_E = - \theta_t - \beta_{tt} \tag{30}$$

where β_{tt} is given by equation (4). The imaginary number i and j satisfy

$$i^2 = -1 \qquad j^2 = -1 \qquad ij = ji \tag{31}$$

Therefore the Dirac, Klein-Gordon and Schrödinger equations must be complex number equations in internal space, and the energy eigenvalues and eigenfunctions must also be represented as complex numbers in internal space.[7] If the wave function to which the operators $\bar{p}_\alpha$ and $\bar{E}$ are applied is real then the measured momenta and energy operators are given by

$$p_{\alpha m} = \cos\theta_{p\alpha}\, p_\alpha = -i\hbar \cos\beta_{\alpha\alpha} \cos\theta_{p\alpha}\, \partial/\partial\alpha \tag{32}$$

$$E_m = \cos\theta_E\, E = i\hbar \cos\beta_{tt} \cos\theta_E\, \partial/\partial t \tag{33}$$

For zero internal phase angles the following results are obtained

$$p_\alpha = -i\hbar\partial/\partial\alpha \tag{34}$$

$$E = i\hbar\partial/\partial t \tag{35}$$

which are the standard operators of quantum mechanics.[8,9]

Generally the momentum and energy operators given by equations (25) and (28) respectively are applied to a complex number wave function exhibiting an internal phase angle as follows

$$\bar{\Psi} = \Psi e^{j\theta_\Psi} \tag{36}$$

whose measured value is given by

$$\Psi_m = \Psi \cos\theta_\Psi \tag{37}$$

The operator equations are then written as

$$\bar{p}_\alpha \bar{\Psi} = -i\hbar \cos\beta_{\alpha\alpha} \sec\beta_{\Psi\alpha}\, \partial\Psi/\partial\alpha\, e^{j\Phi_{\Psi\alpha}} \tag{38}$$

$$\bar{E}\bar{\Psi} = i\hbar \cos\beta_{tt} \sec\beta_{\Psi t}\, \partial\Psi/\partial t\, e^{j\Phi_{\Psi t}} \tag{39}$$

where

$$\tan\beta_{\Psi\alpha} = (\Psi\partial\theta_\Psi/\partial\alpha)/(\partial\Psi/\partial\alpha) \tag{40}$$

$$\tan\beta_{\Psi t} = (\Psi\partial\theta_\Psi/\partial t)/(\partial\Psi/\partial t) \tag{41}$$

$$\Phi_{\Psi\alpha} = \theta_\Psi + \beta_{\Psi\alpha} - \theta_\alpha - \beta_{\alpha\alpha} \tag{42}$$

$$\Phi_{\Psi t} = \theta_\Psi + \beta_{\Psi t} - \theta_t - \beta_{tt} \tag{43}$$

The real and imaginary parts of equations (38) and (39) are respectively

$$(\bar{p}_\alpha \bar{\Psi})_R = - i\hbar \cos \Phi_{\psi\alpha} \cos \beta_{\alpha\alpha} \sec \beta_{\psi\alpha} \, \partial\Psi/\partial\alpha \tag{44}$$

$$(\bar{p}_\alpha \bar{\Psi})_I = - i\hbar \sin \Phi_{\psi\alpha} \cos \beta_{\alpha\alpha} \sec \beta_{\psi\alpha} \, \partial\Psi/\partial\alpha \tag{45}$$

$$(\bar{E}\bar{\Psi})_R = i\hbar \cos \Phi_{\psi t} \cos \beta_{tt} \sec \beta_{\psi t} \, \partial\Psi/\partial t \tag{46}$$

$$(\bar{E}\bar{\Psi})_I = i\hbar \sin \Phi_{\psi t} \cos \beta_{tt} \sec \beta_{\psi t} \, \partial\Psi/\partial t \tag{47}$$

These expressions will be used to calculate the measured values of momentum and energy.

The measured values of the linear momenta and energy are defined by

$$p'_{\alpha m}\Psi = (\bar{p}_\alpha \bar{\Psi})_R = p_{\alpha m}\Psi_m \tag{48}$$

$$E'_m\Psi = (\bar{E}\bar{\Psi})_R = E_m\Psi_m \tag{49}$$

where

$$p'_{\alpha m} = - i\hbar \cos \Phi_{\psi\alpha} \cos \beta_{\alpha\alpha} \sec \beta_{\psi\alpha} \, \partial/\partial\alpha \tag{50}$$

$$E'_m = i\hbar \cos \Phi_{\psi t} \cos \beta_{tt} \sec \beta_{\psi t} \, \partial/\partial t \tag{51}$$

and where $p_{\alpha m}$ and E_m are the measured momentum and energy operators whose form will now be determined. This is done by using equation (37) which gives

$$\partial\Psi/\partial\alpha = A_\alpha \partial\Psi_m/\partial\alpha_m + B_\alpha \Psi_m \tag{52}$$

$$\partial\Psi/\partial t = A_t \partial\Psi_m/\partial t_m + B_t \Psi_m \tag{53}$$

where

$$A_\alpha = \sec \theta_\psi \, \partial\alpha_m/\partial\alpha \tag{54}$$

$$B_\alpha = \sec \theta_\psi \tan \theta_\psi \, \partial\theta_\psi/\partial\alpha \tag{55}$$

$$A_t = \sec \theta_\psi \, \partial t_m/\partial t \tag{56}$$

$$B_t = \sec \theta_\psi \tan \theta_\psi \, \partial\theta_\psi/\partial t \tag{57}$$

Combining equations (48) through (57) gives the measured momentum and energy operators respectively as

$$p_{\alpha m} = - i\hbar(C_\alpha \partial/\partial\alpha_m + D_\alpha) \tag{58}$$

$$E_m = i\hbar(C_t \partial/\partial t_m + D_t) \tag{59}$$

where

$$C_\alpha = \cos \Phi_{\psi\alpha} \cos \beta_{\alpha\alpha} \sec \beta_{\psi\alpha} \sec \theta_\psi \, \partial\alpha_m/\partial\alpha \tag{60}$$

$$D_\alpha = \cos \Phi_{\psi\alpha} \cos \beta_{\alpha\alpha} \sec \beta_{\psi\alpha} \sec \theta_\psi \tan \theta_\psi \, \partial\theta_\psi/\partial\alpha \tag{61}$$

$$C_t = \cos \Phi_{\psi t} \cos \beta_{tt} \sec \beta_{\psi t} \sec \theta_\psi \, \partial t_m/\partial t \tag{62}$$

$$D_t = \cos \Phi_{\psi t} \cos \beta_{tt} \sec \beta_{\psi t} \sec \theta_\psi \tan \theta_\psi \, \partial\theta_\psi/\partial t \tag{63}$$

where $\partial\alpha_m/\partial\alpha$ and $\partial t_m/\partial t$ are given by equations (7) and (8) respectively. Therefore the measured momentum and energy operators generally include the effects of the internal phase angle of the wave function. If the internal angle of the wave function is zero, $D_\alpha = 0$ and $D_t = 0$, and

$$p_{\alpha m} = - i\hbar \cos \theta_{p\alpha} \cos \beta_{\alpha\alpha} \, \partial\alpha_m/\partial\alpha \, \partial/\partial\alpha_m \tag{64}$$

$$E_m = i\hbar \cos \theta_E \cos \beta_{tt} \, \partial t_m/\partial t \, \partial/\partial t_m \tag{65}$$

which agrees with equations (32) and (33).

The conventional quantum mechanical momentum and energy operators are given by

$$p_{\alpha c} \Psi_m = - i\hbar \partial \Psi_m/\partial\alpha_m \tag{66}$$

$$E_c \Psi_m = i\hbar \partial \Psi_m/\partial t_m \tag{67}$$

where $p_{\alpha c}$ and E_c = conventional momentum and energy operators. Combining equations (58), (59), (66) and (67) gives

$$p_{\alpha c} - p_{\alpha m} = - i\hbar[(1 - C_\alpha)\partial/\partial\alpha_m - D_\alpha] \tag{68}$$

$$E_c - E_m = i\hbar[(1 - C_t)\partial/\partial t_m - D_t] \tag{69}$$

Differences between measured and conventionally calculated values of momentum and energy should be detectable (especially for particles in bulk matter) due to the broken symmetry of space and time. Finally, consider the momentum and energy operators for the case where space and time rotate internally for constant values of x, y, z, t . For this case equations (25) and (28) give respectively for $d\bar{\alpha} = j\bar{\alpha} d\theta_\alpha$ and $d\bar{t} = j\bar{t} d\theta_t$

$$\bar{p}_\alpha = ij\hbar/\bar{\alpha}\ \partial/\partial\theta_\alpha = ij\hbar/\alpha\ e^{-j\theta_\alpha}\ \partial/\partial\theta_\alpha \tag{70}$$

$$\bar{E} = -\ ij\hbar/\bar{t}\ \partial/\partial\theta_t = -\ ij\hbar/t\ e^{-j\theta_t}\ \partial/\partial\theta_t \tag{71}$$

These expressions will be used in Sections 4 and 5 where bounded motion in internally space is considered.

B. Angular Momentum Operators in Broken Symmetry Spacetime.

The generalization of the scalar forms of the angular momentum operators are given by[8,9]

$$\bar{L}_x = -\ i\hbar(\bar{y}\partial/\partial\bar{z} - \bar{z}\partial/\partial\bar{y}) \tag{72}$$

$$\bar{L}_y = -\ i\hbar(\bar{z}\partial/\partial\bar{x} - \bar{x}\partial/\partial\bar{z}) \tag{73}$$

$$\bar{L}_z = -\ i\hbar(\bar{x}\partial/\partial\bar{y} - \bar{y}\partial/\partial\bar{x}) \tag{74}$$

These broken symmetry angular momentum operators obey the standard commutation relations. Similarly, the complex number angular momentum operators can be written as the following generalization[8,9]

$$\bar{L}_x = i\hbar(\sin\bar{\phi}\ \partial/\partial\bar{\psi} + \cot\bar{\psi}\ \cos\bar{\phi}\ \partial/\partial\bar{\phi}) \tag{75}$$

$$\bar{L}_y = i\hbar(-\cos\bar{\phi}\ \partial/\partial\bar{\psi} + \cot\bar{\psi}\ \sin\bar{\phi}\ \partial/\partial\bar{\phi}) \tag{76}$$

$$\bar{L}_z = -\ i\hbar\partial/\partial\bar{\phi} \tag{77}$$

The eigenfunctions of $\bar{L}_z$ have been shown to be given by[7]

$$\bar{\Phi} = Ae^{i\bar{M}\bar{\phi}} \qquad \bar{M} = Me^{j\theta_M} \tag{78}$$

$$\bar{L}_z\bar{\Phi} = \hbar\bar{M}\bar{\Phi} \tag{79}$$

where for periodic external rotations with θ_ϕ = constant[7]

$$\bar{M} = \bar{M}_e = M_e e^{j\theta_{Me}} \tag{80A}$$

$$M_e = m\cos\theta_\phi \qquad \theta_{Me} = -\ \theta_\phi \tag{80B}$$

and for periodicity $\bar{M}_e\bar{\phi} = M_e\phi$ = real number, so that

$$\bar{\Phi} = \Phi = Ae^{iM_e\phi} \tag{81}$$

where for external motion θ_x , θ_y , θ_z , θ_r , θ_ϕ and θ_ψ = constants.

The generalization of the L^2 operator eigenvalue equation is given by

$$\bar{L}^2\bar{W} = \hbar^2\bar{\ell}(\bar{\ell} + 1)\bar{W} \tag{82}$$

where

$$\bar{L}^2 = -\hbar^2[1/\sin^2\bar{\psi}\ \partial^2/\partial\bar{\phi}^2 + 1/\sin\bar{\psi}\ \partial/\partial\bar{\psi}(\sin\bar{\psi}\ \partial/\partial\bar{\psi})] \tag{83}$$

where for external rotations in broken symmetry space[7]

$$\bar{\ell} = \bar{\ell}_e = \bar{M}'_e + \nu \qquad \bar{M}'_e = |m|\cos\theta_\phi\ e^{-j\theta\phi} \tag{84}$$

where the integer ν is given by $\nu = \ell - |m|$. The complete angular wave function is given by

$$\bar{\Psi} = \bar{\Phi}(\bar{\phi})\bar{W}(\bar{\psi}) \tag{85}$$

Combining equations (82) and (84) gives for external rotations in broken symmetry space

$$\bar{L}_e^2 = \hbar^2\ell(\ell + 1)\bar{g} \tag{86}$$

$$\bar{g} = [1 + |m|/\ell(\cos\theta_\phi\ e^{-j\theta\phi} - 1)][1 + |m|/(\ell + 1)(\cos\theta_\phi\ e^{-j\theta\phi} - 1)] \tag{87}$$

The technique for the generalization of the angular momentum rules to broken symmetry spacetime is now obvious. For instance, the raising and lowering operators L_+ and L_- are written for broken symmetry space as[8-11]

$$\bar{L}_\pm = \hbar[\bar{\ell}(\bar{\ell} + 1) - \bar{M}(\bar{M} \pm 1)] \tag{88}$$

which for external rotations in broken symmetry space becomes

$$\bar{L}_\pm^e = \hbar[\bar{\ell}_e(\bar{\ell}_e + 1) - \bar{M}_e(\bar{M}_e \pm 1)] \tag{89}$$

$$= \hbar[\ell(\ell + 1)\bar{g} - m(m + 1)\bar{s}]$$

where $\bar{g}$ is given in equation (87) and $\bar{s}$ is given by

$$\bar{s} = 1/(m + 1)\cos\theta_\phi\ e^{-j\theta\phi}(m\cos\theta_\phi\ e^{-j\theta\phi} + 1) \tag{90}$$

These equations reduce to the standard results when $\theta_\phi = 0$ because for this case $\bar{g} = 1$ and $\bar{s} = 1$.

The development of the measured angular momentum operators proceeds from equations (72) through (74). The definitions of the measured angular momentum operators are as follows

$$(\bar{L}_x\bar{\Psi})_R = L'_x\Psi = L_{xm}\Psi_m \tag{91}$$

$$(\bar{L}_y\bar{\Psi})_R = L'_y\Psi = L_{ym}\Psi_m \tag{92}$$

$$(\bar{L}_z\bar{\Psi})_R = L'_z\Psi = L_{zm}\Psi_m \tag{93}$$

where the measured angular momentum operators L_{xm}, L_{ym} amd L_{zm} are determined by the evaluation of the left hand sides of equations (91) through (93). Combining equations (1), (3), (72) through (74), and (91) through (93) gives in a fashion similar to that done for linear momentum

$$L'_x = -i\hbar(E'_{yz}y\partial/\partial z - E'_{zy}z\partial/\partial y) \tag{94}$$

$$L'_y = -i\hbar(E'_{zx}z\partial/\partial x - E'_{xz}x\partial/\partial z) \tag{95}$$

$$L'_z = -i\hbar(E'_{xy}x\partial/\partial y - E'_{yx}y\partial/\partial x) \tag{96}$$

where

$$E'_{yz} = \cos\Xi_{\Psi yz}\cos\beta_{zz}\sec\beta_{\Psi z} \tag{97}$$

$$E'_{zy} = \cos\Xi_{\Psi zy}\cos\beta_{yy}\sec\beta_{\Psi y} \tag{98}$$

$$E'_{zx} = \cos\Xi_{\Psi zx}\cos\beta_{xx}\sec\beta_{\Psi x} \tag{99}$$

$$E'_{xz} = \cos\Xi_{\Psi xz}\cos\beta_{zz}\sec\beta_{\Psi z} \tag{100}$$

$$E'_{xy} = \cos\Xi_{\Psi xy}\cos\beta_{yy}\sec\beta_{\Psi y} \tag{101}$$

$$E'_{yx} = \cos\Xi_{\Psi yx}\cos\beta_{xx}\sec\beta_{\Psi x} \tag{102}$$

and where

$$\Xi_{\Psi yz} = \theta_\Psi + \beta_{\Psi z} + \theta_y - \theta_z - \beta_{zz} \tag{103}$$

$$\Xi_{\Psi zy} = \theta_\Psi + \beta_{\Psi y} + \theta_z - \theta_y - \beta_{yy} \tag{104}$$

$$\Xi_{\Psi zx} = \theta_\Psi + \beta_{\Psi x} + \theta_z - \theta_x - \beta_{xx} \tag{105}$$

$$\Xi_{\Psi xz} = \theta_\Psi + \beta_{\Psi z} + \theta_x - \theta_z - \beta_{zz} \tag{106}$$

$$\Xi_{\Psi xy} = \theta_\Psi + \beta_{\Psi y} + \theta_x - \theta_y - \beta_{yy} \tag{107}$$

$$\Xi_{\Psi yx} = \theta_\Psi + \beta_{\Psi x} + \theta_y - \theta_x - \beta_{xx} \tag{108}$$

Combining equations (91) through (96) with equations (5) and (37) gives

$$L_{xm} = -i\hbar[y_m(E_{yz}\partial/\partial z_m + F_{yz}) - z_m(E_{zy}\partial/\partial y_m + F_{zy})] \tag{109}$$

$$L_{ym} = -i\hbar[z_m(E_{zx}\partial/\partial x_m + F_{zx}) - x_m(E_{xz}\partial/\partial z_m + F_{xy})] \tag{110}$$

$$L_{zm} = -i\hbar[x_m(E_{xy}\partial/\partial y_m + F_{xy}) - y_m(E_{yx}\partial/\partial x_m + F_{yx})] \tag{111}$$

where

$$E_{yz} = A_z E'_{yz} \sec\theta_y \qquad E_{zy} = A_y E'_{zy} \sec\theta_z \tag{112}$$

$$E_{zx} = A_x E'_{zx} \sec\theta_z \qquad E_{xz} = A_z E'_{xz} \sec\theta_x \tag{113}$$

$$E_{xy} = A_y E'_{xy} \sec\theta_x \qquad E_{yx} = A_x E'_{yx} \sec\theta_y \tag{114}$$

$$F_{yz} = B_z E'_{yz} \sec\theta_y \qquad F_{zy} = B_y E'_{zy} \sec\theta_z \tag{115}$$

$$F_{zx} = B_x E'_{zx} \sec\theta_z \qquad F_{xz} = B_z E'_{xz} \sec\theta_x \tag{116}$$

$$F_{xy} = B_y E'_{xy} \sec\theta_x \qquad F_{yx} = B_x E'_{yx} \sec\theta_y \tag{117}$$

where A_α and B_α are given by equations (54) and (55) respectively. If $\theta_\psi = 0$, equations (109) through (117) reduce to the real parts of the operator equations (72) through (74) without the internal phase angle of the wave function. The differences between the measured and the conventionally calculated angular momenta are given by

$$L_{xc} - L_{xm} = -i\hbar\{y_m[(1 - E_{yz})\partial/\partial z_m - F_{yz}] - z_m[(1 - E_{zy})\partial/\partial y_m - F_{zy}]\} \tag{118}$$

$$L_{yc} - L_{ym} = -i\hbar\{z_m[(1 - E_{zx})\partial/\partial x_m - F_{zx}] - x_m[(1 - E_{xz})\partial/\partial z_m - F_{xz}]\} \tag{119}$$

$$L_{zc} - L_{zm} = -i\hbar\{x_m[(1 - E_{xy})\partial/\partial y_m - F_{xy}] - y_m[(1 - E_{yx})\partial/\partial x_m - F_{yx}]\} \tag{120}$$

which may be experimentally detectable.

The same procedure applied to the spherical polar coordinate representation of the angular momenta given by equations (75) through (77) and equations (91) through (93) gives

$$L'_x = i\hbar(G'_x\partial/\partial\psi + H'_x\partial/\partial\phi) \tag{121}$$

$$L'_y = i\hbar(G'_y\partial/\partial\psi + H'_y\partial/\partial\phi) \tag{122}$$

$$L'_z = -i\hbar H'_z\partial/\partial\phi \tag{123}$$

$$G'_x = S_\phi \cos \Xi_{\Psi x\psi} \cos \beta_{\psi\psi} \sec \beta_{\Psi\psi} \tag{124}$$

$$G'_y = - C_\phi \cos \Xi_{\Psi y\psi} \cos \beta_{\psi\psi} \sec \beta_{\Psi\psi} \tag{125}$$

$$H'_x = C_\psi C_\phi / S_\psi \cos \Xi_{\Psi x\phi} \cos \beta_{\phi\phi} \sec \beta_{\Psi\phi} \tag{126}$$

$$H'_y = C_\psi S_\phi / S_\psi \cos \Xi_{\Psi y\phi} \cos \beta_{\phi\phi} \sec \beta_{\Psi\phi} \tag{127}$$

$$H'_z = \cos \Xi_{\Psi z\phi} \cos \beta_{\phi\phi} \sec \beta_{\Psi\phi} \tag{128}$$

where

$$\Xi_{\Psi x\psi} = \theta_\Psi + \beta_{\Psi\psi} + \theta_{s\phi} - \theta_\psi - \beta_{\psi\psi} \tag{129}$$

$$\Xi_{\Psi y\psi} = \theta_\Psi + \beta_{\Psi\psi} - \theta_{c\phi} - \theta_\psi - \beta_{\psi\psi} \tag{130}$$

$$\Xi_{\Psi x\phi} = \theta_\Psi + \beta_{\Psi\phi} - \theta_{c\psi} - \theta_{s\psi} - \theta_{c\phi} - \theta_\phi - \beta_{\phi\phi} \tag{131}$$

$$\Xi_{\Psi y\phi} = \theta_\Psi + \beta_{\Psi\phi} - \theta_{c\psi} - \theta_{s\psi} + \theta_{s\phi} - \theta_\phi - \beta_{\phi\phi} \tag{132}$$

$$\Xi_{\Psi z\phi} = \theta_\Psi + \beta_{\Psi\phi} - \theta_\phi - \beta_{\phi\phi} \tag{133}$$

and

$$\tan \beta_{\Psi\psi} = (\Psi \partial\theta_\Psi/\partial\psi)/(\partial\Psi/\partial\psi) \tag{134}$$

$$\tan \beta_{\Psi\phi} = (\Psi \partial\theta_\Psi/\partial\phi)/(\partial\Psi/\partial\phi) \tag{135}$$

The derivatives that appear in equations (121) through (123) are obtained from equation (37) to be

$$\partial\Psi/\partial\psi = A_\psi \partial\Psi_m/\partial\psi_m + B_\psi \Psi_m \tag{136}$$

$$\partial\Psi/\partial\phi = A_\phi \partial\Psi_m/\partial\phi_m + B_\phi \Psi_m \tag{137}$$

where

$$A_\psi = \sec \theta_\Psi \, \partial\psi_m/\partial\psi \tag{138}$$

$$B_\psi = \sec \theta_\Psi \tan \theta_\Psi \, \partial\theta_\Psi/\partial\psi \tag{139}$$

$$A_\phi = \sec \theta_\Psi \, \partial\phi_m/\partial\phi \tag{140}$$

$$B_\phi = \sec \theta_\Psi \tan \theta_\Psi \, \partial\theta_\Psi/\partial\phi \tag{141}$$

where $\partial\psi_m/\partial\psi$ and $\partial\phi_m/\partial\phi$ are given by equations (23) and (24). Combining equations (91) through (93) with equations (121) through (123) and equations (136) and (137) gives

$$L_{xm} = i\hbar(G_x\partial/\partial\psi_m + H_x\partial/\partial\phi_m + I_x) \tag{142}$$

$$L_{ym} = i\hbar(G_y\partial/\partial\psi_m + H_y\partial/\partial\phi_m + I_y) \tag{143}$$

$$L_{zm} = - i\hbar(H_z\partial/\partial\phi_m + I_z) \tag{144}$$

where

$$G_x = A_\psi G'_x \qquad G_y = A_\psi G'_y \tag{145}$$

$$H_x = A_\phi H'_x \qquad H_y = A_\phi H'_y \qquad H_z = A_\phi H'_z \tag{146}$$

$$I_x = B_\psi G'_x + B_\phi H'_x \qquad I_y = B_\psi G'_y + B_\phi H'_y \qquad I_z = B_\phi H'_z \tag{147}$$

The conventionally calculated angular momenta are given by[8-11]

$$L_{xc} = i\hbar(\sin\phi_m\,\partial/\partial\psi_m + \cot\psi_m\cos\phi_m\,\partial/\partial\phi_m) \tag{148}$$

$$L_{yc} = i\hbar(-\cos\phi_m\,\partial/\partial\psi_m + \cot\psi_m\sin\phi_m\,\partial/\partial\phi_m) \tag{149}$$

$$L_{zc} = - i\hbar\partial/\partial\phi_m \tag{150}$$

Therefore

$$L_{xc} - L_{xm} = i\hbar[(\sin\phi_m - G_x)\partial/\partial\psi_m + (\cot\psi_m\cos\phi_m - H_x)\partial/\partial\phi_m - I_x] \tag{151}$$

$$L_{yc} - L_{ym} = i\hbar[(-\cos\phi_m - G_y)\partial/\partial\psi_m + (\cot\psi_m\sin\phi_m - H_y)\partial/\partial\phi_m - I_y] \tag{152}$$

$$L_{zc} - L_{zm} = - i\hbar[(1 - H_z)\partial/\partial\phi_m - I_z] \tag{153}$$

For pure internal phase rotations of the coordinates, equations (72) through (74) become

$$\bar{L}_x = ij\hbar(\bar{y}/\bar{z}\,\partial/\partial\theta_z - \bar{z}/\bar{y}\,\partial/\partial\theta_y) \tag{154}$$

$$\bar{L}_y = ij\hbar(\bar{z}/\bar{x}\,\partial/\partial\theta_x - \bar{x}/\bar{z}\,\partial/\partial\theta_z) \tag{155}$$

$$\bar{L}_z = ij\hbar(\bar{x}/\bar{y}\,\partial/\partial\theta_y - \bar{y}/\bar{x}\,\partial/\partial\theta_x) \tag{156}$$

while equations (75) through (77) become

$$\bar{L}_x = -ij\hbar[(\sin\bar{\phi})/\bar{\psi}\ \partial/\partial\theta_\psi + (\cot\bar{\psi}\cos\bar{\phi})/\bar{\phi}\ \partial/\partial\theta_\phi] \tag{157}$$

$$\bar{L}_y = -ij\hbar[-(\cos\bar{\phi})/\bar{\psi}\ \partial/\partial\theta_\psi + (\cot\bar{\psi}\sin\bar{\phi})/\bar{\phi}\ \partial/\partial\theta_\phi] \tag{158}$$

$$\bar{L}_z = ij\hbar/\bar{\phi}\ \partial/\partial\theta_\phi \tag{159}$$

The calculation of L'_α and $L_{\alpha m}$ for these equations is elementary. From equations (91) through (93) and equations (154) through (156) it follows that

$$\begin{aligned} L'_x &= -i\hbar(E'_{iyz}\ y/z\ \partial/\partial\theta_z - E'_{izy}\ z/y\ \partial/\partial\theta_y) \\ L'_y &= -i\hbar(E'_{izx}\ z/x\ \partial/\partial\theta_x - E'_{ixz}\ x/z\ \partial/\partial\theta_z) \\ L'_z &= -i\hbar(E'_{ixy}\ x/y\ \partial/\partial\theta_y - E'_{iyx}\ y/x\ \partial/\partial\theta_x) \end{aligned} \tag{159A}$$

where

$$\begin{aligned} E'_{iyz} &= \sin\Xi^i_{\psi yz}\sec\beta_{\psi\theta z} & E'_{izy} &= \sin\Xi^i_{\psi zy}\sec\beta_{\psi\theta y} \\ E'_{izx} &= \sin\Xi^i_{\psi zx}\sec\beta_{\psi\theta x} & E'_{ixz} &= \sin\Xi^i_{\psi xz}\sec\beta_{\psi\theta z} \\ E'_{ixy} &= \sin\Xi^i_{\psi xy}\sec\beta_{\psi\theta y} & E'_{iyx} &= \sin\Xi^i_{\psi yx}\sec\beta_{\psi\theta x} \end{aligned} \tag{159B}$$

$$\begin{aligned} \tan\beta_{\psi\theta z} &= (\Psi\partial\theta_\psi/\partial\theta_z)/(\partial\Psi/\partial\theta_z) \\ \tan\beta_{\psi\theta y} &= (\Psi\partial\theta_\psi/\partial\theta_y)/(\partial\Psi/\partial\theta_y) \\ \tan\beta_{\psi\theta x} &= (\Psi\partial\theta_\psi/\partial\theta_x)/(\partial\Psi/\partial\theta_x) \end{aligned} \tag{159C}$$

$$\begin{aligned} \Xi^i_{\psi yz} &= \theta_\psi + \beta_{\psi\theta z} + \theta_y - \theta_z \\ \Xi^i_{\psi zy} &= \theta_\psi + \beta_{\psi\theta y} + \theta_z - \theta_y \\ \Xi^i_{\psi zx} &= \theta_\psi + \beta_{\psi\theta x} + \theta_z - \theta_x \\ \Xi^i_{\psi xz} &= \theta_\psi + \beta_{\psi\theta z} + \theta_x - \theta_z \\ \Xi^i_{\psi xy} &= \theta_\psi + \beta_{\psi\theta y} + \theta_x - \theta_y \\ \Xi^i_{\psi yx} &= \theta_\psi + \beta_{\psi\theta x} + \theta_y - \theta_x \end{aligned} \tag{159D}$$

The measured angular momenta are obtained from equations (91) through (93) to be

$$L_{xm} = -i\hbar[y/z(E_{iyz}\partial/\partial\theta_z + F_{iyz}) - z/y(E_{izy}\partial/\partial\theta_y + F_{izy})]$$

$$L_{ym} = -i\hbar[z/x(E_{izx}\partial/\partial\theta_x + F_{izx}) - x/z(E_{ixz}\partial/\partial\theta_z + F_{ixz})] \quad (159E)$$

$$L_{zm} = -i\hbar[x/y(E_{ixy}\partial/\partial\theta_y + F_{ixy}) - y/x(E_{iyx}\partial/\partial\theta_x + F_{iyx})]$$

where

$$E_{iyz} = E'_{iyz}\sec\theta_\psi \qquad E_{izy} = E'_{izy}\sec\theta_\psi$$

$$E_{izx} = E'_{izx}\sec\theta_\psi \qquad E_{ixz} = E'_{izx}\sec\theta_\psi \quad (159F)$$

$$E_{ixy} = E'_{ixy}\sec\theta_\psi \qquad E_{iyx} = E'_{iyx}\sec\theta_\psi$$

$$F_{iyz} = B_{\theta z}E'_{iyz} \qquad F_{izy} = B_{\theta y}E'_{izy}$$

$$F_{izx} = B_{\theta x}E'_{izx} \qquad F_{ixz} = B_{\theta z}E'_{ixz} \quad (159G)$$

$$F_{ixy} = B_{\theta y}E'_{ixy} \qquad F_{iyx} = B_{\theta x}E'_{iyx}$$

where

$$B_{\theta z} = \sec\theta_\psi \tan\theta_\psi \, \partial\theta_\psi/\partial\theta_z$$

$$B_{\theta y} = \sec\theta_\psi \tan\theta_\psi \, \partial\theta_\psi/\partial\theta_y \quad (159H)$$

$$B_{\theta z} = \sec\theta_\psi \tan\theta_\psi \, \partial\theta_\psi/\partial\theta_x$$

For the case when the wave function only rotates and $d\bar{\Psi} = j\bar{\Psi}d\theta_\psi$, then it follows from equations (91) through (93) and (154) through (156) that

$$L'_x = -i\hbar[y/z\cos\Xi^r_{\psi yz}\,\partial\theta_\psi/\partial\theta_z - z/y\cos\Xi^r_{\psi zy}\,\partial\theta_\psi/\partial\theta_y] \quad (160)$$

$$L'_y = -i\hbar[z/x\cos\Xi^r_{\psi zx}\,\partial\theta_\psi/\partial\theta_x - x/z\cos\Xi^r_{\psi xz}\,\partial\theta_\psi/\partial\theta_z] \quad (161)$$

$$L'_z = -i\hbar[x/y\cos\Xi^r_{\psi xy}\,\partial\theta_\psi/\partial\theta_y - y/x\cos\Xi^r_{\psi yx}\,\partial\theta_\psi/\partial\theta_x] \quad (162)$$

$$L_{xm} = L'_x\sec\theta_\psi \quad (163)$$

$$L_{ym} = L'_y\sec\theta_\psi \quad (164)$$

$$L_{zm} = L'_z\sec\theta_\psi \quad (165)$$

where

$$\Xi^{r}_{\psi yz} = \theta_{\psi} + \theta_{y} - \theta_{z} \qquad \Xi^{r}_{\psi zy} = \theta_{\psi} + \theta_{z} - \theta_{y} \tag{166}$$

$$\Xi^{r}_{\psi zx} = \theta_{\psi} + \theta_{z} - \theta_{x} \qquad \Xi^{r}_{\psi xz} = \theta_{\psi} + \theta_{x} - \theta_{y} \tag{167}$$

$$\Xi^{r}_{\psi xy} = \theta_{\psi} + \theta_{x} - \theta_{y} \qquad \Xi^{r}_{\psi yx} = \theta_{\psi} + \theta_{y} - \theta_{x} \tag{168}$$

The measured angular momenta for this case are given by equations (160) through (162).

From equations (91) through (93) and equations (157) through (159) the calculation of L'_{α} and $L_{\alpha m}$ for the internal phase rotations of spherical polar coordinates and for a wave function whose magnitude and internal phase angle are variable is done as follows

$$\begin{aligned} L'_{x} &= i\hbar(G'_{ix}\ \partial/\partial\theta_{\psi} + H'_{ix}\ \partial/\partial\theta_{\phi}) \\ L'_{y} &= i\hbar(G'_{iy}\ \partial/\partial\theta_{\psi} + H'_{iy}\ \partial/\partial\theta_{\phi}) \\ L'_{z} &= -\ i\hbar H'_{iz}\ \partial/\partial\theta_{\phi} \end{aligned} \tag{168A}$$

where

$$\begin{aligned} G'_{ix} &= S_{\phi}/\psi \sec \beta_{\psi\theta\psi} \sin \Xi^{i}_{\psi x\psi} \\ G'_{iy} &= -\ C_{\phi}/\psi \sec \beta_{\psi\theta\psi} \sin \Xi^{i}_{\psi y\psi} \end{aligned} \tag{168B}$$

$$\begin{aligned} H'_{ix} &= C_{\psi}C_{\phi}/(S_{\psi}\phi) \sec \beta_{\psi\theta\phi} \sin \Xi^{i}_{\psi x\phi} \\ H'_{iy} &= C_{\psi}S_{\phi}/(S_{\psi}\phi) \sec \beta_{\psi\theta\phi} \sin \Xi^{i}_{\psi y\phi} \\ H'_{iz} &= 1/\phi \sec \beta_{\psi\theta\phi} \sin \Xi^{i}_{\psi z\phi} \end{aligned} \tag{168C}$$

where

$$\tan \beta_{\psi\theta\psi} = (\Psi\partial\theta_{\psi}/\partial\theta_{\psi})/(\partial\Psi/\partial\theta_{\psi}) \tag{168D}$$

$$\tan \beta_{\psi\theta\phi} = (\Psi\partial\theta_{\psi}/\partial\theta_{\phi})/(\partial\Psi/\partial\theta_{\phi}) \tag{168E}$$

and

$$\overset{i}{\Psi}_{x\psi} = \theta_\psi + \beta_{\psi\theta\psi} + \theta_{s\phi} - \theta_\psi$$

$$\overset{i}{\Psi}_{y\psi} = \theta_\psi + \beta_{\psi\theta\psi} - \theta_{c\phi} - \theta_\psi$$

$$\overset{i}{\Psi}_{x\phi} = \theta_\psi + \beta_{\psi\theta\phi} - \theta_{c\psi} - \theta_{s\psi} - \theta_{c\phi} - \theta_\phi \qquad (168F)$$

$$\overset{j}{\Psi}_{y\phi} = \theta_\psi + \beta_{\psi\theta\phi} - \theta_{c\psi} - \theta_{s\psi} + \theta_{s\phi} - \theta_\phi$$

$$\overset{i}{\Psi}_{z\phi} = \theta_\psi + \beta_{\psi\theta\phi} - \theta_\phi$$

The measured angular momenta can be obtained from equations (91) through (93) so that for pure internal rotations of spherical polar coordinates

$$L_{xm} = i\hbar(E_{ix\psi}\partial/\partial\theta_\psi + E_{ix\phi}\partial/\partial\theta_\phi + F_{ix\psi} + F_{ix\phi})$$

$$L_{ym} = i\hbar(E_{iy\psi}\partial/\partial\theta_\psi + E_{iy\phi}\partial/\partial\theta_\phi + F_{iy\psi} + F_{iy\phi}) \qquad (168G)$$

$$L_{zm} = -i\hbar(E_{iz\phi}\partial/\partial\theta_\phi + F_{iz\phi})$$

where

$$E_{ix\psi} = G'_{ix}\sec\theta_\psi \qquad E_{ix\phi} = H'_{ix}\sec\theta_\psi$$

$$E_{iy\psi} = G'_{iy}\sec\theta_\psi \qquad E_{iy\phi} = H'_{iy}\sec\theta_\psi \qquad (168H)$$

$$E_{iz\psi} = 0 \qquad E_{iz\phi} = H'_{iz}\sec\theta_\psi$$

$$F_{ix\psi} = B_{\theta\psi}G'_{ix} \qquad F_{ix\phi} = B_{\theta\phi}G'_{ix}$$

$$F_{iy\psi} = B_{\theta\psi}G'_{iy} \qquad F_{iy\phi} = B_{\theta\phi}G'_{iy} \qquad (168I)$$

$$F_{iz\psi} = 0 \qquad F_{iz\phi} = B_{\theta\phi}G'_{iz}$$

where

$$B_{\theta\psi} = \sec\theta_\psi \tan\theta_\psi\, \partial\theta_\psi/\partial\theta_\psi$$

$$B_{\theta\phi} = \sec\theta_\psi \tan\theta_\psi\, \partial\theta_\psi/\partial\theta_\phi \qquad (168J)$$

The case where the wave function only rotates in internal space will now be considered.

From equations (91) through (93) and equations (157) through (159) it follows that for spherical polar coordinates and $d\bar{\Psi} = j\bar{\Psi}d\theta_\psi$

$$L'_x = i\hbar(G'_{xr}\partial\theta_\psi/\partial\theta_\psi + H'_{xr}\partial\theta_\psi/\partial\theta_\phi) \tag{169}$$

$$L'_y = i\hbar(G'_{yr}\partial\theta_\psi/\partial\theta_\psi + H'_{yr}\partial\theta_\psi/\partial\theta_\phi) \tag{170}$$

$$L'_z = -i\hbar H'_{zr}\partial\theta_\psi/\partial\theta_\phi \tag{171}$$

where

$$G'_{xr} = S_\phi/\psi \cos \Xi^r_{\Psi x\psi} \qquad H'_{xr} = C_\psi C_\phi/(S_\psi\phi) \cos \Xi^r_{\Psi x\phi} \tag{172}$$

$$G'_{yr} = -C_\phi/\psi \cos \Xi^r_{\Psi y\psi} \qquad H'_{yr} = C_\psi S_\phi/(S_\psi\phi) \cos \Xi^r_{\Psi y\phi} \tag{173}$$

$$G'_{zr} = 0 \qquad H'_{zr} = 1/\phi \cos \Xi^r_{\Psi z\phi} \tag{174}$$

$$\Xi^r_{\Psi x\psi} = \theta_\Psi + \theta_{s\phi} - \theta_\psi \tag{175}$$

$$\Xi^r_{\Psi y\psi} = \theta_\Psi - \theta_{c\phi} - \theta_\psi \tag{176}$$

$$\Xi^r_{\Psi x\phi} = \theta_\Psi - \theta_{c\psi} - \theta_{s\psi} - \theta_{c\phi} - \theta_\phi \tag{177}$$

$$\Xi^r_{\Psi y\phi} = \theta_\Psi - \theta_{c\psi} - \theta_{s\psi} + \theta_{s\phi} - \theta_\phi \tag{178}$$

$$\Xi^r_{\Psi z\phi} = \theta_\Psi - \theta_\phi \tag{179}$$

The values of $L_{\alpha m}$ are then obtained from equations (163) through (165).

3. SCHRÖDINGER EQUATION FOR ASYMMETRIC SPACETIME. This section considers Schrödinger's equation for two extremes of the variation of coordinates: a) external space and time variation of coordinates with constant internal phase angle, and b) internal variation of spacetime coordinates with constant coordinate magnitudes and varying internal phase angles. The single particle energy is calculated for both these cases. The general case where both the magnitude and the internal phase angle vary has already been treated for Schrödinger's equation.[7] In cartesian coordinates Schrödinger's time independent equation for coordinates with broken internal symmetry is written as

$$-\hbar^2/(2\mu)(\partial^2/\partial\bar{x}^2 + \partial^2/\partial\bar{y}^2 + \partial^2/\partial\bar{z}^2)\bar{\Psi} + \bar{V}\bar{\Psi} = \bar{E}\bar{\Psi} \tag{180}$$

where $\bar{V} = \bar{V}(\bar{x},\bar{y},\bar{z})$ = complex number potential, and $\bar{E}$ = complex number energy. The corresponding time dependent Schrödinger equation is given by[7]

$$- \hbar^2/(2\mu)\sum_{\alpha}\partial^2\bar{\Psi}/\partial\bar{\alpha}^2 + \bar{V}\bar{\Psi} = i\hbar\partial\bar{\Psi}/\partial\bar{t} \tag{181}$$

where $\alpha = x, y, z$. The generalization of the standard scalar time independent Schrödinger equation in spherical polar coordinates to the case of broken symmetry space and time is given by[7]

$$\frac{\partial^2\bar{\Psi}}{\partial\bar{r}^2} + \frac{2}{\bar{r}}\frac{\partial\bar{\Psi}}{\partial\bar{r}} + \frac{1}{\bar{r}^2\sin\bar{\psi}}\frac{\partial}{\partial\bar{\psi}}(\sin\bar{\psi}\frac{\partial\bar{\Psi}}{\partial\bar{\psi}}) + \frac{1}{\bar{r}^2\sin^2\bar{\psi}}\frac{\partial^2\bar{\Psi}}{\partial\bar{\phi}^2} + \frac{8\pi^2\mu}{h^2}[\bar{E} - \bar{V}(\bar{r})]\bar{\Psi} = 0 \tag{182}$$

The wave function in equation (182) can be separated as follows

$$\bar{\Psi} = \bar{R}(\bar{r})\bar{W}(\bar{\psi})\bar{\Phi}(\bar{\phi}) \tag{183}$$

which gives the following complex number generalization of the standard scalar equations[7]

$$d^2\bar{\Phi}/d\bar{\phi}^2 + \bar{M}^2\bar{\Phi} = 0 \tag{184}$$

$$1/\sin\bar{\psi}\ d/d\bar{\psi}(\sin\bar{\psi}\ d\bar{W}/d\bar{\psi}) + (\bar{\beta} - \bar{M}^2/\sin^2\bar{\psi})\bar{W} = 0 \tag{185}$$

$$\bar{r}^2\, d^2\bar{R}/d\bar{r}^2 + 2\bar{r}\, d\bar{R}/d\bar{r} + (\bar{k}^2\bar{r}^2 - \bar{\beta})\bar{R} = 0 \tag{186}$$

where

$$\bar{k}^2 = 2\mu/\hbar^2(\bar{E} - \bar{V}) \tag{187A}$$

$$\bar{M} = Me^{j\theta_M} \tag{187B}$$

$$\bar{\beta} = \beta e^{j\theta_\beta} \tag{187C}$$

where $\bar{M}$ and $\bar{\beta}$ are complex number separation constants. The Schrödinger equations (180) through (186) will now be considered for the special cases of external and internal coordinate variation. The combined case of external and internal motion has already appeared in the literature.[7]

A. External Coordinate Variation

For external coordinate variation in cartesian coordinates with broken internal symmetry the internal phase angles θ_x , θ_y and θ_z are constants that depend only on the ambient pressure and energy density. Then

$$d\bar{x} = dxe^{j\theta_x} \qquad d\bar{y} = dye^{j\theta_y} \qquad d\bar{z} = dze^{j\theta_z} \tag{188}$$

where the constant internal phase angles are written as

$$\theta_x = \theta_x(P,E) \qquad \theta_y = \theta_y(P,E) \qquad \theta_z = \theta_z(P,E) \tag{189}$$

Schrödinger's equation (180) becomes

$$-\hbar^2/(2\mu)[e^{-2j\theta x}\,\partial^2\bar{\Psi}_e/\partial x^2 + e^{-2j\theta y}\,\partial^2\bar{\Psi}_e/\partial y^2 + e^{-2j\theta z}\,\partial^2\bar{\Psi}_e/\partial z^2] + \bar{V}\bar{\Psi}_e = \bar{E}\bar{\Psi}_e \quad (190)$$

For the free particle with external motion, the wave function is given by

$$\bar{\Psi}_e = e^{i(\bar{k}_{ex}\bar{x} + \bar{k}_{ey}\bar{y} + \bar{k}_{ez}\bar{z})} \quad (191)$$

where the complex number wave numbers are given by

$$\bar{k}_{ex} = k_{ex}e^{j\theta kex} \qquad \bar{k}_{ey} = k_{ey}e^{j\theta key} \qquad \bar{k}_{ez} = k_{ez}e^{j\theta kez} \quad (192)$$

and where because of the periodicity condition in measured coordinates[7]

$$\begin{aligned} \bar{k}_{ex}\bar{x} &= k_{ex}x & \theta_{kex} &= -\theta_x \\ \bar{k}_{ey}\bar{y} &= k_{ey}y & \theta_{key} &= -\theta_y \\ \bar{k}_{ez}\bar{z} &= k_{ez}z & \theta_{kez} &= -\theta_z \end{aligned} \quad (193)$$

and therefore $\bar{\Psi}_e = \Psi_e$ for periodicity. Equation (193) determines the internal phase angles of the wave numbers for periodic waves in broken symmetry space. The spatial wavelengths of the waves are given by

$$\bar{\lambda}_{ex} = 2\pi/\bar{k}_{ex} \qquad \bar{\lambda}_{ey} = 2\pi/\bar{k}_{ey} \qquad \bar{\lambda}_{ez} = 2\pi/\bar{k}_{ez} \quad (194)$$

where

$$\bar{\lambda}_{ex} = \lambda_{ex}e^{j\theta\lambda ex} \qquad \bar{\lambda}_{ey} = \lambda_{ey}e^{j\theta\lambda ey} \qquad \bar{\lambda}_{ez} = \lambda_{ez}e^{j\theta\lambda ez} \quad (195)$$

Then it follows that

$$k_{ex} = 2\pi/\lambda_{ex} \qquad k_{ey} = 2\pi/\lambda_{ey} \qquad k_{ez} = 2\pi/\lambda_{ez} \quad (196)$$

$$\theta_{kex} = -\theta_{\lambda ex} = -\theta_x \quad (197)$$

$$\theta_{key} = -\theta_{\lambda ey} = -\theta_y \quad (198)$$

$$\theta_{kez} = -\theta_{\lambda ez} = -\theta_z \quad (199)$$

For periodic waves the wave numbers and wavelengths must adjust themselves to the local spacetime conditions according to equations (197) through (199).

The measured wavelengths are given by

$$\lambda_{exm} = \lambda_{ex} \cos \theta_{\lambda ex} = \lambda_{ex} \cos \theta_x \tag{200}$$

$$\lambda_{eym} = \lambda_{ey} \cos \theta_{\lambda ey} = \lambda_{ey} \cos \theta_y \tag{201}$$

$$\lambda_{ezm} = \lambda_{ez} \cos \theta_{\lambda ez} = \lambda_{ez} \cos \theta_z \tag{202}$$

while the measured wave numbers are

$$k_{exm} = k_{ex} \cos \theta_{kex} = k_{ex} \cos \theta_x \tag{203}$$

$$k_{eym} = k_{ey} \cos \theta_{key} = k_{ey} \cos \theta_y \tag{204}$$

$$k_{ezm} = k_{ez} \cos \theta_{kez} = k_{ez} \cos \theta_z \tag{205}$$

Combining equations (196) through (205) gives

$$k_{exm} = 2\pi/\lambda_{exm} \cos^2 \theta_x \tag{206}$$

$$k_{eym} = 2\pi/\lambda_{eym} \cos^2 \theta_y \tag{207}$$

$$k_{ezm} = 2\pi/\lambda_{ezm} \cos^2 \theta_z \tag{208}$$

Note that $k_{exm} \neq 2\pi/\lambda_{exm}$ and so on. Finally, for insertion into the wave function in equation (191) one has

$$\bar{k}_{ex}\bar{x} = k_{ex}x = k_{exm}x_m/\cos^2 \theta_x = 2\pi x_m/\lambda_{exm} \tag{209}$$

$$\bar{k}_{ey}\bar{y} = k_{ey}y = k_{eym}y_m/\cos^2 \theta_y = 2\pi y_m/\lambda_{eym} \tag{210}$$

$$\bar{k}_{ez}\bar{z} = k_{ez}z = k_{ezm}z_m/\cos^2 \theta_z = 2\pi z_m/\lambda_{ezm} \tag{211}$$

Consider the case of free particles moving in external space that exhibits broken internal symmetry. The single particle energy can be obtained from equations (190) through (199) to be

$$\bar{E}_e = \hbar^2/(2\mu)(k_{ex}^2 e^{-2j\theta_x} + k_{ey}^2 e^{-2j\theta_y} + k_{ez}^2 e^{-2j\theta_z}) \tag{212}$$

$$= \hbar^2 \bar{k}_e^2/(2\mu)$$

where

$$\bar{k}_e^2 = \bar{k}_{ex}^2 + \bar{k}_{ey}^2 + \bar{k}_{ez}^2 \tag{213}$$

The measured free particle energy is given by the real part of equation (212) as

$$E_{em} = \hbar^2 k_e^2/(2\mu)\ \cos(2\theta_{ke}) \tag{214}$$

where k_e and θ_{ke} are obtained from equation (213) and equations (197) through (199) as

$$k_e^2 \cos(2\theta_{ke}) = k_{ex}^2 \cos(2\theta_x) + k_{ey}^2 \cos(2\theta_y) + k_{ez}^2 \cos(2\theta_z) \tag{215}$$

$$k_e^2 \sin(2\theta_{ke}) = k_{ex}^2 \sin(2\theta_x) + k_{ey}^2 \sin(2\theta_y) + k_{ez}^2 \sin(2\theta_z) \tag{216}$$

Therefore the measured energy is

$$E_{em} = \hbar^2/(2\mu)[k_{ex}^2 \cos(2\theta_x) + k_{ey}^2 \cos(2\theta_y) + k_{ez}^2 \cos(2\theta_z)] \tag{217}$$

This equation will be used in Section 4 to calculate the energy of a particle localized in an external space box.

For external motion in spherical coordinates with broken internal symmetry and θ_ϕ , θ_ψ and θ_r constant, equations (184) through (186) become[7]

$$d^2\bar{\Phi}_e/d\bar{\phi}^2 + \bar{M}_e^2\bar{\Phi}_e = 0 \tag{218}$$

$$1/\sin\ \bar{\psi}\ d/d\bar{\psi}(\sin\ \bar{\psi}\ d\bar{W}_e/d\bar{\psi}) + (\bar{\beta}_e - \bar{M}_e^2/\sin^2\ \bar{\psi})\bar{W}_e = 0 \tag{219}$$

$$\bar{r}^2 d^2\bar{R}_e/d\bar{r}^2 + 2\bar{r}d\bar{R}_e/d\bar{r} + (\bar{k}^2\bar{r}^2 - \bar{\beta}_e)\bar{R}_e = 0 \tag{220}$$

where[7]

$$\bar{M}_e = m \cos\ \theta_\phi\ e^{-j\theta\phi} \qquad M_e = m \cos\ \theta_\phi \tag{221}$$

$$\bar{M}_e' = |m| \cos\ \theta_\phi\ e^{-j\theta\phi} \qquad M_e' = |m| \cos\ \theta_\phi \tag{222}$$

$$\bar{\beta}_e = \bar{\ell}_e(\bar{\ell}_e + 1) \tag{223}$$

$$\bar{\ell}_e = \bar{M}_e' + \ell - |m| \tag{224}$$

where m = ordinary magnetic quantum number = 0 , ±1 , ±2 , ···. For periodic rotations $\bar{M}_e\bar{\phi} = M_e\phi$ = real number, so that[7]

$$d^2\Phi_e/d\phi^2 + M_e^2\Phi_e = 0 \tag{225}$$

$$\Phi_e = e^{\pm iM_e\phi} = e^{\pm im\phi \cos\ \theta_\phi} = e^{\pm im\phi_m} \tag{226}$$

where $\phi_m = \phi \cos\ \theta_\phi$. Thus $\bar{\Phi}_e = \Phi_e$ = real number in internal space (but obviously a complex number in external space).

Consider now the calculation of the quantized energy values for a rigid rotator in broken symmetry spacetime. For rotations about the z axis Schrödinger's equation is obtained from equation (71) to be

$$-\hbar^2/(2\bar{I})\partial^2\bar{\Phi}/\partial\bar{\phi}^2 = \bar{E}_z\bar{\Phi} \tag{227}$$

whose solution is

$$\bar{\Phi} = Ae^{i\bar{M}\bar{\phi}} \tag{228}$$

which is also the eigenfunction for $\bar{L}_z$ as shown in equations (78) and (79). Combining equations (227) and (228) gives the kinetic energy of the rotator as

$$\bar{E}_z = \hbar^2\bar{M}^2/(2\bar{I}) \tag{229}$$

where $\bar{I}$ = complex number moment of inertia about the z axis. The measured energy is given by

$$E_{zm} = \hbar^2M^2/(2I)\cos(2\theta_M - \theta_I) \tag{230}$$

For periodic external rotations with θ_ϕ = constant it follows from equations (80) and (230) that

$$E^e_{zm} = \hbar^2M_e^2/(2I)\cos(2\theta_\phi + \theta_I) \tag{231}$$

$$= \hbar^2m^2/(2I)\cos^2\theta_\phi\cos(2\theta_\phi + \theta_I)$$

Equation (231) can be written as

$$E^e_{zm} = \hbar^2m^2/(2I_{eff}) \tag{232}$$

where the effective moment of inertia is

$$I_{eff} = I\sec(2\theta_\phi + \theta_I)\sec^2\theta_\phi \tag{233}$$

The complex moment of inertia is written as

$$\bar{I} = \mu\bar{r}^2 \qquad I = \mu r^2 \qquad \theta_I = 2\theta_r \tag{234}$$

$$I_m = I\cos\theta_I = I\cos(2\theta_r) = I_c(1 - \tan^2\theta_r) \tag{235}$$

where I_c = conventionally calculated moment of inertia given by

$$I_c = \mu r_m^2 = I\cos^2\theta_r \tag{236}$$

Combining equations (233) through (236) gives

$$I_{eff} = I_c \sec[2(\theta_\phi + \theta_r)] \sec^2 \theta_\phi \sec^2 \theta_r \tag{237}$$

Now consider rigid rotations in three dimensional space with broken symmetries. In this case the rotational energy of a body is given by[7]

$$\bar{E} = \hbar^2/(2\bar{I})\bar{\ell}(\bar{\ell} + 1) \tag{238}$$

where for external motion $\bar{\ell} = \bar{\ell}_e$ where $\bar{\ell}_e$ is given by equation (224). Therefore for external motion

$$\bar{E}^e = \hbar^2/(2\bar{I})\bar{\ell}_e(\bar{\ell}_e + 1) \tag{239}$$

$$= \hbar^2/(2\bar{I})(\bar{M}'_e + \nu)(\bar{M}'_e + \nu + 1)$$

where ν is an integer given by $\nu = \ell - |m|$. Combining equations (222), (224) and (239) gives the measured energy eigenvalues as

$$E^e_m = \hbar^2/(2I)[|m|^2 L_2 + |m|(2\nu + 1)L_1 + \nu(\nu + 1)L_o] \tag{240}$$

where

$$L_2 = \cos^2 \theta_\phi \cos(2\theta_\phi + \theta_I) \tag{241}$$

$$L_1 = \cos \theta_\phi \cos(\theta_\phi + \theta_I) \tag{242}$$

$$L_o = \cos \theta_I \tag{243}$$

Equivalently, it follows from equation (239) that

$$E^e_m = \hbar^2/(2I)[\ell_e^2 \cos(2\theta_{\ell e} - \theta_I) + \ell_e \cos(\theta_{\ell e} - \theta_I)] \tag{244}$$

where[7]

$$\ell_e = \ell[1 - |m|/\ell(2 - |m|/\ell)\sin^2 \theta_\phi]^{1/2} \tag{245}$$

$$\tan \theta_{\ell e} = -(|m| \sin \theta_\phi \cos \theta_\phi)/(\ell - |m| \sin^2 \theta_\phi) \tag{246}$$

B. Internal Coordinate Motion

The case where the cartesian and spherical coordinate magnitudes (x , y , z and r , ψ , ϕ respectively) are held fixed and the system is moving in internal coordinate space with θ_x , θ_y , θ_z and θ_r , θ_ψ , θ_ϕ as variables is now considered. Also t = constant and θ_t is a variable.

First the case of cartesian coordinates is treated. From equation (70) it follows that for α = constant

$$\bar{p}_\alpha^2 = -\hbar^2 \partial^2/\partial\bar{\alpha}^2 \tag{247}$$

$$= \hbar^2/\bar{\alpha}^2(\partial^2/\partial\theta_\alpha^2 - j\partial/\partial\theta_\alpha) \quad . \tag{248}$$

where $\alpha = x, y, z$. Combining equations (180), (247) and (248) yields Schrödinger's equation for internal cartesian coordinate motion

$$-\hbar^2/(2\mu)\sum_\alpha \partial^2\bar{\Psi}_i/\partial\bar{\alpha}^2 + \bar{V}_i\bar{\Psi}_i = \bar{E}_i\bar{\Psi}_i \tag{249}$$

$$\hbar^2/(2\mu)\sum_\alpha 1/\bar{\alpha}^2(\partial^2\bar{\Psi}_i/\partial\theta_\alpha^2 - j\partial\bar{\Psi}_i/\partial\theta_\alpha) + \bar{V}_i\bar{\Psi}_i = \bar{E}_i\bar{\Psi}_i \tag{250}$$

where the subscript "i" refers to internal motion. The time dependent Schrödinger equation is obtained from equations (28), (71) and (250) by writing

$$\bar{E}_i\bar{\Psi}_i = i\hbar\partial\bar{\Psi}_i/\partial\bar{t} \tag{251}$$

$$= -ij\hbar/\bar{t}\ \partial\bar{\Psi}_i/\partial\theta_t \tag{252}$$

$$= -ij\hbar/t\ e^{-j\theta_t}\partial\bar{\Psi}_i/\partial\theta_t \tag{253}$$

remembering that $d\bar{t} = j\bar{t}d\theta_t$.

Consider a free particle located at a fixed position in external space x, y, z and moving in internal space with θ_x , θ_y , θ_z as variables. The wave function is written as

$$\bar{\Psi}_i(\theta_x,\theta_y,\theta_z) = e^{i(\bar{k}_{ix}\bar{x} + \bar{k}_{iy}\bar{y} + \bar{k}_{iz}\bar{z})} \tag{254}$$

$$= \exp\, i[\bar{k}_{ix}x\, \exp(j\theta_x) + \bar{k}_{iy}y\, \exp(j\theta_y) + \bar{k}_{iz}z\, \exp(j\theta_z)]$$

The requirement that equation (254) be a solution of Schrödinger's equation for a free particle gives the single particle energy for internal motion to be

$$\bar{E}_i = \hbar^2/(2\mu)(\bar{k}_{ix}^2 + \bar{k}_{iy}^2 + \bar{k}_{iz}^2) \tag{255}$$

where

$$\bar{k}_{ix} = k_{ix}e^{j\theta kix} \qquad \bar{k}_{iy} = k_{iy}e^{j\theta kiy} \qquad \bar{k}_{iz} = k_{iz}e^{j\theta kiz} \tag{256}$$

which are complex number constants. The measured single particle energy is given by

$$E_{im} = \hbar^2/(2\mu)[k_{ix}^2\cos(2\theta_{kix}) + k_{iy}^2\cos(2\theta_{kiy}) + k_{iz}^2\cos(2\theta_{kiz})] \tag{257}$$

In Section 4 this equation will be used to calculate the energy of a particle trapped in an internal space box. The phase in equation (254) is not a real

number because there is no periodicity in measured real space as x , y and z are fixed. Therefore

$$\bar{k}_{ix}\bar{x} = k_{ix}xe^{j(\theta_{kix} + \theta_x)} \tag{258}$$

$$\bar{k}_{iy}\bar{y} = k_{iy}ye^{j(\theta_{kiy} + \theta_y)} \tag{259}$$

$$\bar{k}_{iz}\bar{z} = k_{iz}ze^{j(\theta_{kiz} + \theta_z)} \tag{260}$$

Equation (254) also follows from equation (25) which becomes

$$- i\partial\bar{\Psi}_i/\partial\bar{x} = \bar{k}_{ix}\bar{\Psi}_i \tag{261}$$

$$- i\partial\bar{\Psi}_i/\partial\bar{y} = \bar{k}_{iy}\bar{\Psi}_i \tag{262}$$

$$- i\partial\bar{\Psi}_i/\partial\bar{z} = \bar{k}_{iz}\bar{\Psi}_i \tag{263}$$

For constant $\bar{E}_i$, equation (251) gives

$$\bar{\Psi}_i(\theta_t) = \exp(- i\bar{E}_i\bar{t}/\hbar) = \exp[- i\bar{E}_i t/\hbar \exp(j\theta_t)] \tag{264}$$

with t = constant and $\bar{t}$ given by equation (2). Therefore for x , y , z and t = constants it is easier (for the free particle case) to solve the Schrödinger equations (249) and (251) in terms of $\bar{x}$, $\bar{y}$, $\bar{z}$ and $\bar{t}$ and then use equations (1) and (2) to express the results in terms θ_α and θ_t , rather than to directly solve equations (250) and (253). On the other hand, if the potential $\bar{V}_i$ is explicitly a function of the internal phase angles $\bar{V}_i = \bar{V}_i(\theta_x,\theta_y,\theta_z)$ with x , y and z = constants, then it may be more convenient to solve equations (250) and (253) directly. Yet even in this case it is possible to use the conventional form of Schrodinger's equation (249) by using equation (1) to rewrite the potential as

$$\bar{V}_i(\theta_x,\theta_y,\theta_z) = \bar{V}_i[- j\ \ell n(\bar{x}/x), - j\ \ell n(\bar{y}/y), - j\ \ell n(\bar{z}/z)] \tag{265}$$

$$\equiv \bar{W}_i(\bar{x},\bar{y},\bar{z})$$

and equation (249) becomes

$$- \hbar^2/(2\mu)\sum_{\alpha}\partial^2\bar{\Psi}_i/\partial\bar{\alpha}^2 + \bar{W}_i\bar{\Psi}_i = \bar{E}_i\bar{\Psi}_i \tag{266}$$

Equations (250) and (266) are equivalent.

The situation of internal phase rotations in spherical polar coordinates with r , ψ and ϕ fixed is now considered. For this case Schrödinger's equation (182) becomes

$$- 1/\bar{r}^2(\partial^2\bar{\Psi}_i/\partial\theta_r^2 + j\partial\bar{\Psi}_i/\partial\theta_r) - 1/(\bar{r}^2\bar{\psi}\sin\bar{\psi})\partial/\partial\theta_\psi(1/\bar{\psi}\sin\bar{\psi}\,\partial\bar{\Psi}_i/\partial\theta_\psi) \quad (267)$$

$$- 1/(\bar{r}^2\bar{\phi}^2\sin^2\bar{\psi})(\partial^2\bar{\Psi}_i/\partial\theta_\phi^2 - j\partial\bar{\Psi}_i/\partial\theta_\phi) + 8\pi^2\mu/h^2(\bar{E}_i - \bar{V}_i)\bar{\Psi}_i = 0$$

Equation (267) can be separated into three independent equations by writing

$$\bar{\Psi}_i = \bar{R}_i(\theta_r)\bar{W}_i(\theta_\psi)\bar{\Phi}_i(\theta_\phi) \quad (268)$$

which gives the following equations

$$- 1/\bar{\phi}^2(d^2\bar{\Phi}_i/d\theta_\phi^2 - jd\bar{\Phi}_i/d\theta_\phi) + \bar{M}_i^2\bar{\Phi}_i = 0 \quad (269)$$

$$- 1/(\bar{\psi}\sin\bar{\psi})d/d\theta_\psi(1/\bar{\psi}\sin\bar{\psi}\,d\bar{W}_i/d\theta_\psi) + (\bar{\beta}_i - \bar{M}_i^2/\sin^2\bar{\psi})\bar{W}_i = 0 \quad (270)$$

$$- d^2\bar{R}_i/d\theta_r^2 - jd\bar{R}_i/d\theta_r + (\bar{k}_i^2\bar{r}^2 - \bar{\beta}_i)\bar{R}_i = 0 \quad (271)$$

where

$$\bar{k}_i^2 = 8\pi^2\mu/h^2[\bar{E}_i - \bar{V}_i(\theta_r,\theta_\psi,\theta_\phi,r,\psi,\phi)] \quad (272)$$

and where

$$\bar{M}_i = M_i e^{j\theta_{Mi}} \quad (273)$$

is the complex magnetic quantum number for internal azimuthal angle rotations, and possibly $\bar{M}_i = \bar{M}_i(\phi)$ because ϕ = constant. The solution of equations (184) through (186) are relatively easy for external motion when θ_ϕ = constant and $\bar{M}$ and $\bar{\beta}$ are constants given in equations (218) through (224).[7] If θ_ϕ is not constant but instead, at the other extreme, θ_ϕ is variable and ϕ = constant then the separation constant for internal motion $\bar{M}_i$ must be independent of θ_ϕ so that certainly $\bar{M}_i \neq \bar{M}_e$. Therefore equations (218) through (220) are valid for θ_r , θ_ψ and θ_ϕ = constants while equations (269)through (271) are valid for r , ψ , ϕ = constants.

The solution to equation (269) can be obtained from a solution of equation (184) with ϕ = constant. This solution is

$$\bar{\Phi}_i(\theta_\phi) = \exp(\pm i\bar{M}_i\bar{\phi}) \quad (274)$$

$$= \exp[\pm iM_i\phi\,\exp j(\theta_{Mi} + \theta_\phi)]$$

Note that $\theta_{Mi} + \theta_\phi \neq 0$ for internal coordinate motion whereas $\theta_{Me} + \theta_\phi = 0$ for external periodic motion. Similarly, the solution of equation (270) can be obtained from the solution of equation (185) with ψ = constant as follows[7]

$$\bar{W}_i(\theta_\psi) = \bar{P}_{\bar{L}_i}^{\bar{M}_i}[\cos(\psi e^{j\theta_\psi})] \quad (275)$$

with

$$\bar{\beta} = \bar{\beta}_i = \bar{L}_i(\bar{L}_i + 1) \qquad \bar{L}_i = \bar{M}_i + \delta \tag{276}$$

where $\delta = 0,1,2,3,\cdots$. Because ϕ = constant, $\bar{M}_i$ cannot be related to an integer as was the case for the external complex magnetic quantum number $\bar{M}_e$ given in equation (80). The solution of the radial equation (271) depends on the form of the potential $\bar{V}_i$.

The Schrödinger equation for the internal space rigid body rotator is now examined. For rotations about the z axis, but with ϕ = constant and $d\bar{\phi} = j\bar{\phi}d\theta_\phi$, it follows from equation (227) that

$$\hbar^2/(2\bar{I}\bar{\phi}^2)(d^2\bar{\Phi}_i/d\theta_\phi^2 - jd\bar{\Phi}_i/d\theta_\phi) + \bar{V}_i(\theta_\phi)\bar{\Phi}_i = \bar{E}_{zi}\bar{\Phi}_i \tag{277}$$

Equation (277) can also be obtained directly from equation (159). The solution to equation (277) for $\bar{V}_i = 0$ is given by equation (274), and for this case a comparison of equations (269) and (277) gives the energy as

$$\bar{E}_{zi} = \hbar^2\bar{M}_i^2/(2\bar{I}) \tag{278}$$

for free internal rotations. Also from equation (159)

$$\bar{L}_z\bar{\Phi}_i = \hbar\bar{M}_i\bar{\Phi}_i \tag{279}$$

$$\bar{E}_{zi}\bar{\Phi}_i = \bar{L}_z^2/(2\bar{I})\bar{\Phi}_i \tag{280}$$

The energy in equation (278) can be obtained directly by operating with the left hand side of equation (277) with $\bar{V}_i = 0$ on the solution given in equation (274). The measured energy is given by

$$E_{zim} = \hbar^2 M_i^2/(2I)\cos(2\theta_{Mi} - \theta_I) \tag{281}$$

For internal rotation in three dimensions it follows from equations (83), (269) and (270) that for the rigid rotator

$$\bar{L}_i^2 = \hbar^2[1/(\bar{\phi}^2\sin^2\bar{\psi})(\partial^2/\partial\theta_\phi^2 - j\partial/\partial\theta_\phi) + 1/(\bar{\psi}\sin\bar{\psi})\partial/\partial\theta_\psi(1/\bar{\psi}\sin\bar{\psi}\,\partial/\partial\theta_\psi)] \tag{282}$$

$$= \hbar^2[\bar{M}_i^2/\sin^2\bar{\psi} + 1/(\bar{\psi}\sin\bar{\psi})\partial/\partial\theta_\psi(1/\bar{\psi}\sin\bar{\psi}\,\partial/\partial\theta_\psi)]$$

$$= \hbar^2\bar{\beta}_i = \hbar^2\bar{L}_i(\bar{L}_i + 1)$$

so that

$$\bar{E}_i = \hbar^2/(2\bar{I})\bar{L}_i(\bar{L}_i + 1) \tag{283}$$

where $\bar{\mathcal{L}}_i$ is given by equation (276). The measured energy is obtained from equations (276) and (283) as

$$E_{im} = \hbar^2/(2I)[K_2 M_i^2 + (2\delta + 1)K_1 M_i + \delta(\delta + 1)K_o] \tag{284}$$

where

$$K_2 = \cos(2\theta_{Mi} - \theta_I) \tag{285}$$

$$K_1 = \cos(\theta_{Mi} - \theta_I) \tag{286}$$

$$K_o = \cos\theta_I \tag{287}$$

Apparently M_i is a continuous real number that can assume any value. Thus the spectrum of the internal space rotator is continuous.

4. PARTICLE IN BOX FOR ASYMMETRIC SPACETIME. From the previous section it is clear that particles can have internal as well as external motions. This section considers the motion of a particle in a box first for the case of motion in external space with broken internal symmetries, and secondly for motion in a box located in internal space.

A. Particle in External Space Box with Broken Symmetry.

Consider a particle in a box of complex number space dimensions $\bar{a}$, $\bar{b}$ and $\bar{c}$. The measured dimensions of the box are

$$a_m = a\cos\theta_a \qquad b_m = b\cos\theta_b \qquad c_m = c\cos\theta_c \tag{288}$$

and it is assumed that the internal phase angles of the coordinates are constants and $\theta_x = \theta_a$, $\theta_y = \theta_b$, $\theta_z = \theta_c$. The wave function obtained from equation (191) will be written as

$$\Psi_e = A\sin(k_{ex}x)\sin(k_{ey}y)\sin(k_{ez}z) \tag{289}$$

If $\Psi_e = 0$ at the boundaries $x = a$, $y = b$ and $z = c$, it follows that

$$k_{ex}a = n\pi \qquad k_{ey}b = m\pi \qquad k_{ez}c = \ell\pi \tag{290}$$

and therefore equations (196), (209) through (211) and (290) give

$$\lambda_x = 2a/n \qquad \lambda_{xm} = 2a_m/n$$

$$\lambda_y = 2b/m \qquad \lambda_{ym} = 2b_m/m \tag{291}$$

$$\lambda_z = 2c/\ell \qquad \lambda_{zm} = 2c_m/\ell$$

Combining equations (217), (288) and (290) gives the measured energy for a particle moving externally in a broken symmetry space box as

$$E_{em} = \hbar^2\pi^2/(2\mu)[n^2/a^2\cos(2\theta_a) + m^2/b^2\cos(2\theta_b) + \ell^2/c^2\cos(2\theta_c)] \quad (292)$$

$$= \hbar^2\pi^2/(2\mu)(I_a n^2/a_m^2 + I_b m^2/b_m^2 + I_c \ell^2/c_m^2)$$

where

$$I_a = \cos^2\theta_a \cos(2\theta_a)$$

$$I_b = \cos^2\theta_b \cos(2\theta_b) \quad (293)$$

$$I_c = \cos^2\theta_c \cos(2\theta_c)$$

The conventionally calculated energy for a particle in a box is given by

$$E_{ec} = \hbar^2\pi^2/(2\mu)(n^2/a_m^2 + m^2/b_m^2 + \ell^2/c_m^2) \quad (294)$$

and therefore

$$E_{ec} - E_{em} = \hbar^2\pi^2/(2\mu)[(1 - I_a)n^2/a_m^2 + (1 - I_b)m^2/b_m^2 + (1 - I_c)\ell^2/c_m^2] \quad (295A)$$

$$\sim 3\hbar^2\pi^2/(2\mu)(n^2\theta_a^2/a_m^2 + m^2\theta_b^2/b_m^2 + \ell^2\theta_c^2/c_m^2)$$

For $\theta_a \sim \theta_b \sim \theta_c = \theta_\alpha$ it follows from equations (294) and (295A) that

$$(E_{ec} - E_{em})/E_{ec} \sim 3\theta_\alpha^2 \quad (295B)$$

The difference between the predicted and measured energies may be detectable and may lead to an estimate of the value of θ_α^2 .

Consider a rigid rotator oscillating between two angular positions $\phi = 0$ and $\phi = \phi_a$. Then the wave function is given by a form similar to equation (78) for broken symmetry space

$$\bar{\Phi} = Ae^{i\bar{M}_o\bar{\phi}} \quad (296)$$

which must be a solution of Schrödinger's equation for a body rotating about the z axis

$$-\hbar^2/(2\bar{I})\partial^2\bar{\Phi}/\partial\bar{\phi}^2 = \bar{E}_z^e\bar{\Phi} \quad (297)$$

which gives the energy as

$$\bar{E}_z^e = \hbar^2\bar{M}_o^2/(2\bar{I}) \quad (298)$$

whose real part gives the measured energy as

$$E^{e}_{zm} = \hbar^2 M_o^2/(2I)\ \cos(2\theta_{Mo} - \theta_I) \tag{299}$$

where the constant $\bar{M}_o$ for the external rotational oscillator is written as

$$\bar{M}_o = M_o e^{j\theta Mo} \tag{300}$$

and the values of M_o and θ_{Mo} are to be determined from boundary and periodicity conditions. The periodicity conditions give[7]

$$\theta_{Mo} = -\theta_\phi \qquad \bar{M}_o\bar{\phi} = M_o\phi \tag{301}$$

In order to satisfy the boundary conditions the wave function derived from equation (296) for the rotational oscillator must be of the form

$$\Phi = A\ \sin(M_o\phi) \tag{302}$$

where M_o is to be determined by the boundary condition at $\phi = \phi_a$. The condition that $\Phi = 0$ at $\phi = \phi_a$ gives

$$M_o\phi_a = n\pi \tag{303}$$

or

$$M_o = n\pi/\phi_a \tag{304}$$

Therefore the measured energy is obtained from equations (299), (301) and (304) to be

$$E^{e}_{zm} = \hbar^2\pi^2/(2I)(n/\phi_a)^2\ \cos(2\theta_\phi + \theta_I) \tag{305}$$

F. Particle in Internal Space Box

Consider a box in internal coordinate space with variables θ_x , θ_y and θ_z which are bounded by

$$\begin{aligned} \theta_a < \theta_x < \theta_b \\ \theta_c < \theta_y < \theta_d \\ \theta_e < \theta_z < \theta_f \end{aligned} \tag{306}$$

corresponding to the external coordinates x , y , z which are constants. For the wave function take the real values of equation (254) as follows

$$\Psi_i = \Psi_{ix}\Psi_{iy}\Psi_{iz} \tag{307}$$

where

$$\Psi_{ix} = \sin[k_{ix}x \sin(\theta_{kix} + \theta_x)]$$

$$\Psi_{iy} = \sin[k_{iy}y \sin(\theta_{kiy} + \theta_y)] \tag{308}$$

$$\Psi_{iz} = \sin[k_{iz}z \sin(\theta_{kiz} + \theta_z)]$$

At the boundaries of the internal space box the wave function must be zero. The conditions for the vanishing of the wave function at the boundaries are

$$k_{ix}x \sin(\theta_{kix} + \theta_b) = n\pi \tag{309}$$

$$k_{ix}x \sin(\theta_{kix} + \theta_a) = 0 \tag{310}$$

$$k_{iy}y \sin(\theta_{kiy} + \theta_d) = m\pi \tag{311}$$

$$k_{iy}y \sin(\theta_{kiy} + \theta_c) = 0 \tag{312}$$

$$k_{iz}z \sin(\theta_{kiz} + \theta_f) = \ell\pi \tag{313}$$

$$k_{iz}z \sin(\theta_{kiz} + \theta_e) = 0 \tag{314}$$

where n , m and ℓ are integers. From equations (310), (312) and (314) it follows that

$$\theta_{kix} = -\theta_a \qquad \theta_{kiy} = -\theta_c \qquad \theta_{kiz} = -\theta_e \tag{315}$$

Combining equations (309), (311), (313) and (315) gives

$$k_{ix} = n\pi/[x \sin(\theta_b - \theta_a)]$$

$$k_{iy} = m\pi/[y \sin(\theta_d - \theta_c)] \tag{316}$$

$$k_{iz} = \ell\pi/[z \sin(\theta_f - \theta_e)]$$

where n , m and $\ell = 0,1,2,3,4,\cdots$. Placing the results of equations (315) and (316) into the expression for the measured energy eigenvalues given by equation (257) yields for a particle in an internal space box

$$E_{im} = \hbar^2\pi^2/(2\mu)(J_x'n^2/x^2 + J_y'm^2/y^2 + J_z'\ell^2/z^2) \tag{317}$$

where

$$J_x' = \cos(2\theta_a)/\sin^2(\theta_b - \theta_a)$$

$$J_y' = \cos(2\theta_c)/\sin^2(\theta_d - \theta_c) \tag{318}$$

$$J_z' = \cos(2\theta_e)/\sin^2(\theta_f - \theta_e)$$

During the internal motion of the particle the measured position of the particle also changes because $\alpha_m = \alpha \cos\theta_\alpha$ with α = constant and θ_α being variables.

The values of x , y and z must be measured relative to some fixed center of force, such as the nucleus of an atom, when the internal oscillations of an electron are being considered. This determines $E_{im}(x,y,z)$. The energy of physical interest is then

$$U_m = \int E_{im} n \, dV \tag{319}$$

where n = particle number density. The quantity U_m gives the measured energy associated with the internal oscillations of the coordinates. The integral in equation (319) may be logarithmically divergent, and renormalization techniques will have to be used for the calculation of the internal energy.

Finally, consider the angular internal rotational oscillations corresponding to a fixed angular magnitude ϕ of a body. The wave function for this internal space rigid rotator is given by equation (274). Therefore considering only a real portion of the wave function gives

$$\Phi_i(\theta_\phi) = \sin[M_i\phi \sin(\theta_{Mi} + \theta_\phi)] \tag{320}$$

If the rotator is restricted to an angular sector in internal space given by $\theta_a < \theta_\phi < \theta_b$ then

$$M_i\phi \sin(\theta_{Mi} + \theta_a) = 0 \tag{321}$$

$$M_i\phi \sin(\theta_{Mi} + \theta_b) = n\pi \tag{322}$$

where $n = 0,1,2,3,\cdots$. Equation (321) gives

$$\theta_{Mi} = -\theta_a \tag{323}$$

while equation (322) gives

$$M_i = n\pi/[\phi \sin(\theta_b - \theta_a)] \tag{324}$$

Substituting equations (323) and (324) into equation (281) gives the measured angular oscillation energy eigenvalues as

$$E_{im} = \hbar^2\pi^2/(2I)(n/\phi)^2 \cos(2\theta_a + \theta_I)/\sin^2(\theta_b - \theta_a) \tag{325}$$

During the internal angular oscillations, the measured angle will oscillate according to $\phi_m = \phi \cos \theta_\phi$ where ϕ = constant.

5. LINEAR HARMONIC OSCILLATOR. This section studies the simple harmonic oscillator in broken symmetry spacetime. The study is separated into four parts, a treatment of the one- and three-dimensional harmonic oscillators in external spacetime with broken internal symmetries and, a discussion of the one- and three-dimensional harmonic oscillators in internal space.

A. One-Dimensional Harmonic Oscillator in External Space with Broken Internal Symmetries.

The standard Schrödinger equation for the one-dimensional simple harmonic oscillator is given by[8]

$$d^2\psi/dx^2 + (\lambda - \alpha^2x^2)\psi = 0 \tag{326}$$

where

$$\lambda = 8\pi^2\mu E/h^2 \tag{327}$$

$$\alpha = 4\pi^2\mu\nu/h \tag{328}$$

After a number of transformations, the solution comes down to a power series whose coefficients are related by[8]

$$a_{\sigma+2}/a_\sigma = -(\lambda/\alpha - 2\sigma - 1)/[(\sigma + 1)(\sigma + 2)] \tag{329}$$

where σ = integer. Only if the series breaks off at a value of $\sigma = k$ can a finite polynomial solution be found. Therefore the energy eigenvalues are

$$E_k = (k + 1/2)h\nu \tag{330}$$

where $k = 0,1,2,3,\cdots$. The eigenfunctions are[8]

$$\psi_k = N_k e^{-\xi^2/2} H_k(\xi) \tag{331}$$

$$\xi = x\sqrt{\alpha} \tag{332}$$

where $H_k(\xi)$ = Hermite polynomial of degree k.

The generalization of equation (326) to broken symmetry space and time with θ_x and θ_t as constants is given by

$$d^2\bar{\Psi}_e/d\bar{x}^2 + (\bar{\lambda}_e - \bar{\alpha}_e^2\bar{x}^2)\bar{\Psi}_e = 0 \tag{333}$$

where the "e" refers to external motion with x variable and θ_x = constant, and where

$$\bar{\lambda}_e = 8\pi^2\mu\bar{E}_e/h^2 \tag{334}$$

$$\bar{\alpha}_e = 4\pi^2\mu\bar{\nu}_e/h \tag{335}$$

$$d\bar{x} = dx\, e^{j\theta x} \tag{336}$$

Performing the same procedure as was done for the scalar case gives the following set of complex number coefficients for a power series

$$\bar{a}_{\sigma+2}/\bar{a}_\sigma = -(\bar{\lambda}/\bar{\alpha} - 2\sigma - 1)/[(\sigma + 1)(\sigma + 2)] \tag{337}$$

where again σ = integer. If the break off occurs at $\bar{\xi}_e^k$ where

$$\bar{\xi}_e = \bar{x}\sqrt{\bar{\alpha}_e} \tag{338}$$

and k = integer. Then it follows that for $\bar{E}_k^e = E_k^e e^{j\theta_{Ek}^e}$

$$\bar{E}_k^e = (k + 1/2)h\bar{\nu}_e \qquad E_k^e = (k + 1/2)h\nu_e \qquad \theta_{Ek}^e = \theta_{\nu e} \tag{339}$$

where k = 0,1,2,3,$\cdots$. The wave functions are

$$\bar{\Psi}_{ek} = \bar{N}_{ek}e^{-\bar{\xi}_e^2/2}\bar{H}_k(\bar{\xi}_e) \tag{340}$$

where

$$(\bar{\alpha}_e)^{1/2} = 2\pi(\mu/h)^{1/2}\,\nu_e^{1/2}\,e^{j\theta_{\nu e}/2} \tag{341}$$

$$\bar{\xi}_e = 2\pi(\mu/h)^{1/2}\,x\nu_e^{1/2}\,e^{j(\theta_{\nu e}/2 + \theta_x)} \tag{342}$$

The measured energy eigenvalues are given by

$$E_{km}^e = E_k^e \cos\theta_{Ek}^e$$

$$= (k + 1/2)h\nu_e \cos\theta_{\nu e} = (k + 1/2)h\nu_{em} \tag{343}$$

Thus the standard result is valid if measured energies and frequencies are used as in equation (343). The internal phase angle of the frequency is related to the fundamental asymmetry of spacetime because $\theta_{\nu e} = -\theta_t$. Therefore for the one-dimensional linear harmonic oscillator, the effects of broken spacetime symmetry occur only through a complex number frequency and not

through a complex quantum number. It will be shown subsequently that for the three-dimensional linear harmonic oscillator a complex quantum number must be introduced in addition to a complex number frequency.

B. Three-Dimensional Linear Harmonic Oscillator in Broken Symmetry External Space.

The radial component of the standard Schrödinger equation for the isotropic three-dimensional simple harmonic oscillator is given by[12,13]

$$d^2u/dr^2 - [\ell(\ell + 1)/r^2 - 2\mu/\hbar^2(E - 1/2\mu\omega^2r^2)]u = 0 \tag{344}$$

where the total wave function is written as

$$\Psi = 1/ru(r)W(\psi)\Phi(\phi) \tag{345}$$

After a sequence of standard transformations the solution can be written as an infinite series whose coefficients are related by[12,13]

$$c_{k+1}/c_k = - [E/(h\nu) - \ell - 2k - 3/2]/[(k + 1)(2\ell + 3k + 3)] \tag{346}$$

The condition for a break off solution is[12,13]

$$E = (2k + \ell + 3/2)h\nu \tag{347}$$

$$= (n + 1/2)h\nu$$

$$= (2n' + \ell - 1/2)h\nu$$

where some standard notations in common use are

$$n = 2k + \ell + 1 \tag{348}$$

$$n' = k + 1 \tag{349}$$

and $k = 0,1,2,3,\cdots$, $n = 1,2,3,\cdots$, $n' = 1,2,3,\cdots$, and $\ell = 0,1,2,3,\cdots$. Note that n = principal quantum number.

The generalization to the case of the three-dimensional linear harmonic oscillator in broken symmetry spacetime with θ_r, θ_ψ and θ_ϕ as constants is given by

$$d^2\bar{u}_e/d\bar{r}^2 - [\bar{\ell}_e(\bar{\ell}_e + 1)/\bar{r}^2 - 2\mu/\hbar^2(\bar{E}_e - 1/2\mu\bar{\omega}_e^2\bar{r}^2)]\bar{u}_e = 0 \tag{350}$$

where

$$d\bar{r} = dre^{j\theta_r} \tag{351}$$

and where $\bar{\ell}_e$ = complex angular momentum quantum number. A break off solution to the broken symmetry zenith angle equation does not require $\bar{\ell}_e$ to be an integer,

but only that[7]

$$\bar{\mathcal{L}}_e - \bar{M}'_e = \ell - |m| = \text{integer} \tag{352}$$

where[7]

$$\bar{M}'_e = |m| \cos \theta_\phi \, e^{-j\theta\phi} \tag{353}$$

For this case it is easy to see that the break off solution for equation (350) is given by the following generalization to equation (346)

$$\bar{c}_{k+1}/\bar{c}_k = -[\bar{E}_e/(h\bar{\nu}_e) - \bar{\mathcal{L}}_e - 2k - 3/2]/[(k+1)(2\bar{\mathcal{L}}_e + 3k + 3)] \tag{354}$$

so that the complex number energy can be written in the following equivalent forms

$$\begin{aligned} \bar{E}_e &= (\bar{\eta}_e + 1/2)h\bar{\nu}_e \\ &= (\bar{\mathcal{L}}_e + 2k + 3/2)h\bar{\nu}_e \\ &= (\bar{\mathcal{L}}_e + 2n' - 1/2)h\bar{\nu}_e \\ &= (\bar{M}'_e + n - |m| + 1/2)h\bar{\nu}_e \\ &= (\bar{M}'_e + \ell - |m| + 2k + 3/2)h\bar{\nu}_e \end{aligned} \tag{355}$$

where

$$\begin{aligned} \bar{\eta}_e = \eta_e e^{j\theta_{\eta e}} &= 2k + \bar{\mathcal{L}}_e + 1 \\ &= 2k + \bar{M}'_e + \ell - |m| + 1 \\ &= \bar{M}'_e + n - |m| \end{aligned} \tag{356}$$

For $\theta_\phi = 0$ equation (356) reduces to equation (348). From equations (352), (353) and (356) it follows that[7]

$$\eta_e \cos \theta_{\eta e} = n - |m| \sin^2 \theta_\phi \tag{357}$$

$$\eta_e \sin \theta_{\eta e} = -|m| \cos \theta_\phi \sin \theta_\phi \tag{358}$$

$$\mathcal{L}_e \cos \theta_{\mathcal{L}e} = \ell - |m| \sin^2 \theta_\phi \tag{359}$$

$$\mathcal{L}_e \sin \theta_{\mathcal{L}e} = -|m| \cos \theta_\phi \sin \theta_\phi \tag{360}$$

so that[7]

$$\eta_e = n[1 - |m|/n(2 - |m|/n)\sin^2 \theta_\phi]^{1/2} \tag{361}$$

$$\tan \theta_{\eta e} = - (|m| \cos \theta_\phi \sin \theta_\phi)/(n - |m|\sin^2 \theta_\phi) \tag{362}$$

$$\mathcal{L}_e = \ell[1 - |m|/\ell(2 - |m|/\ell)\sin^2 \theta_\phi]^{1/2} \tag{363}$$

$$\tan \theta_{\mathcal{L}e} = - (|m| \cos \theta_\phi \sin \theta_\phi)/(\ell - |m| \sin^2 \theta_\phi) \tag{364}$$

These quantities are required to calculate the measured values of the energy eigenvalues.

The measured energy eigenvalues are obtained from equation (355) to be

$$E_{em}/(h\nu_e) = \eta_e \cos(\theta_{\eta e} + \theta_{\nu e}) + 1/2 \cos \theta_{\nu e} \tag{365}$$

$$= \mathcal{L}_e \cos(\theta_{\mathcal{L}e} + \theta_{\nu e}) + (2k + 3/2)\cos \theta_{\nu e}$$

$$= \mathcal{L}_e \cos(\theta_{\mathcal{L}e} + \theta_{\nu e}) + (2n' - 1/2)\cos \theta_{\nu e}$$

$$= |m| \cos \theta_\phi \cos(\theta_{\nu e} - \theta_\phi) + (n - |m| + 1/2)\cos \theta_{\nu e}$$

$$= |m| \cos \theta_\phi \cos(\theta_{\nu e} - \theta_\phi) + (\ell - |m| + 2k + 3/2)\cos \theta_{\nu e}$$

The imaginary part E_{eI} of the energy in equation (355) can be obtained as

$$E_{eI}/(h\nu_e) = \eta_e \sin(\theta_{\eta e} + \theta_{\nu e}) + 1/2 \sin \theta_{\nu e} \tag{366}$$

$$= \mathcal{L}_e \sin(\theta_{\mathcal{L}e} + \theta_{\nu e}) + (2k + 3/2)\sin \theta_{\nu e}$$

$$= \mathcal{L}_e \sin(\theta_{\mathcal{L}e} + \theta_{\nu e}) + (2n' - 1/2)\sin \theta_{\nu e}$$

$$= |m| \cos \theta_\phi \sin(\theta_{\nu e} - \theta_\phi) + (n - |m| + 1/2)\sin \theta_{\nu e}$$

$$= |m| \cos \theta_\phi \sin(\theta_{\nu e} - \theta_\phi) + (\ell - |m| + 2k + 3/2)\sin \theta_{\nu e}$$

Then

$$E_e = (E_{eR}^2 + E_{eI}^2)^{1/2} \qquad \tan \theta_{Ee} = E_{eI}/E_{eR} \tag{367}$$

If $\theta_{\nu e} = - \theta_t$ is negligible in equation (365) these expressions become

$$E_{em}/(h\nu_e) \sim \eta_e \cos\theta_{\eta e} + 1/2 \tag{368}$$

$$= \mathcal{L}_e \cos\theta_{\mathcal{L}e} + 2k + 3/2$$

$$= \mathcal{L}_e \cos\theta_{\mathcal{L}e} + 2n' - 1/2$$

$$= n + 1/2 - |m| \sin^2\theta_\phi$$

$$= \ell + 2k + 3/2 - |m| \sin^2\theta_\phi$$

$$E_{eI}/(h\nu_e) \sim \eta_e \sin\theta_{\eta e} \tag{369}$$

$$= \mathcal{L}_e \sin\theta_{\mathcal{L}e}$$

$$= - |m| \cos\theta_\phi \sin\theta_\phi$$

and

$$\tan\theta_{Ee} \sim (\eta_e \sin\theta_{\eta e})/(\eta_e \cos\theta_{\eta e} + 1/2) \tag{370}$$

$$= - (|m| \cos\theta_\phi \sin\theta_\phi)/(n + 1/2 - |m| \sin^2\theta_\phi)$$

$$\theta_{Ee} \sim - |m|\theta_\phi/(n + 1/2)$$

C. One-Dimensional Linear Harmonic Oscillator in Internal Space.

Consider the internal harmonic motion at a point along the x axis. For this case the magnitude x = constant and the potential energy is written in terms of the variable internal phase angle θ_x as

$$\bar{V}_i(\theta_x) = 1/2\bar{K}_{ix}\theta_x^2 \tag{371}$$

The Schrödinger equation (250) for this case is written as

$$\hbar^2/(2\mu x^2)e^{-2j\theta_x}(\partial^2\bar{\Psi}_i/\partial\theta_x^2 - j\partial\bar{\Psi}_i/\partial\theta_x) + 1/2\bar{K}_{ix}\theta_x^2\,\bar{\Psi}_i = \bar{E}_i\bar{\Psi}_i \tag{372}$$

Equivalently Schrödinger's equation can be written in the form of equation (266) as

$$- \hbar^2/(2\mu)\partial^2\bar{\Psi}_i/\partial\bar{x}^2 + \bar{W}_i\bar{\Psi}_i = \bar{E}_i\bar{\Psi}_i \tag{373}$$

$$\bar{W}_i(\bar{x}) = - 1/2\ \bar{K}_{ix}\ \ell n^2(\bar{x}/x) \tag{374}$$

where because x = constant

$$d\bar{x} = j\bar{x}d\theta_x \tag{375}$$

Equation (373) can be written as

$$\partial^2\bar{\Psi}_i/\partial\bar{x}^2 + [\bar{\lambda}_i + \bar{\alpha}_i^2\, \ell n^2(\bar{x}/x)]\bar{\Psi}_i = 0 \tag{376}$$

where

$$\bar{\lambda}_i = 2\mu\bar{E}_i/\hbar^2 \tag{377}$$

$$\bar{K}_{ix} = 4\pi^2\mu\bar{\nu}_i^2 = \mu\bar{\omega}_i^2 \tag{378}$$

$$\bar{\alpha}_i^2 = \mu\bar{K}_{ix}/\hbar^2 = 16\pi^4\mu^2\bar{\nu}_i^2/h^2 \tag{379}$$

$$\bar{\alpha}_i = \mu\bar{\omega}_i/\hbar$$

D. Three-Dimensional Linear Harmonic Oscillator in Internal Space.

The potential energy of a three-dimensional internal space harmonic oscillator with x , y , z = constants is written as

$$\bar{V}_i = 1/2(\bar{K}_{ix}\theta_x^2 + \bar{K}_{iy}\theta_y^2 + \bar{K}_{iz}\theta_z^2) \tag{381}$$

For central symmetry in internal space this simplifies to

$$\bar{V}_i(\theta_r) = 1/2\bar{K}_{ir}\theta_r^2 = -\,1/2\;\bar{K}_{ir}\;\ell n^2(\bar{r}/r) = \bar{W}_i(\bar{r}) \tag{382}$$

where r = constant. The radial Schrödinger equation becomes

$$d^2\bar{u}_i/d\bar{r}^2 - \{\bar{\ell}_i(\bar{\ell}_i + 1)/\bar{r}^2 - 2\mu/\hbar^2[\bar{E}_i + 1/2\mu\bar{\omega}_i^2\, \ell n^2(\bar{r}/r)]\}\bar{u}_i = 0 \tag{383}$$

where

$$d\bar{r} = j\bar{r}d\theta_r \tag{384}$$

$$\bar{\omega}_i^2 = \bar{K}_{ir}/\mu \tag{385}$$

and where

$$\bar{\ell}_i = \bar{M}_i + \delta \tag{386}$$

where $\delta = 0,1,2,3,\cdots$. Equation (383) can be rewritten as

$$d^2\bar{u}_i/d\bar{r}^2 + [\bar{\lambda}_i + \bar{\alpha}_i^2\, \ell n^2(\bar{r}/r) - \bar{\ell}_i(\bar{\ell}_i + 1)/\bar{r}^2]\bar{u}_i = 0 \tag{387}$$

where $\bar{\lambda}_i$ and $\bar{\alpha}_i$ are given by equations (377), (380) and (385). Making the substitution $\bar{u}_i = \bar{r}\bar{R}_i$ in equation (387) gives

$$d^2\bar{R}_i/d\bar{r}^2 + 2/\bar{r}\ d\bar{R}_i/d\bar{r} + [\bar{\lambda}_i + \bar{\alpha}_i^2\ \ell n^2(\bar{r}/r) - \bar{\ell}_i(\bar{\ell}_i + 1)/\bar{r}^2]\bar{R}_i = 0 \qquad (388)$$

Equations equivalent to (387) and (388) can be written using equation (384) as follows

$$- d^2\bar{u}_i/d\theta_r^2 + jd\bar{u}_i/d\theta_r + [(\bar{\lambda}_i - \bar{\alpha}_i^2\theta_r^2)\bar{r}^2 - \bar{\ell}_i(\bar{\ell}_i + 1)]\bar{u}_i = 0 \qquad (389)$$

$$- d^2\bar{R}_i/d\theta_r^2 - jd\bar{R}_i/d\theta_r + [(\bar{\lambda}_i - \bar{\alpha}_i^2\theta_r^2)\bar{r}^2 - \bar{\ell}_i(\bar{\ell}_i + 1)]\bar{R}_i = 0 \qquad (390)$$

The author has not solved equations (387) through (390).

6. ADDITION OF ANGULAR MOMENTUM. This section studies the addition of angular momentum in space and time with broken internal symmetries. It will be shown that the standard addition law for angular momenta must be modified for the case where the magnetic quantum number exhibits a broken internal symmetry.

A. Complex External Magnetic Quantum Numbers.

Consider two complex magnetic quantum numbers that describe the motion of two particles in external space with constant broken internal symmetries[7]

$$\bar{M}_1 = M_1 e^{j\theta_{M1}} = m_1 \cos\theta_{M1}\ e^{j\theta_{M1}} = m_1 \cos\theta_{\phi 1}\ e^{-j\theta_{\phi 1}} \qquad (391)$$

$$\bar{M}_2 = M_2 e^{j\theta_{M2}} = m_2 \cos\theta_{M2}\ e^{j\theta_{M2}} = m_2 \cos\theta_{\phi 2}\ e^{-j\theta_{\phi 2}} \qquad (392)$$

where equation (80) was used for external motion with $\theta_{Mi} = -\theta_{\phi i}$ = constants and where m_1 and m_2 are integers $0,\pm1,\pm2,\pm3,\cdots$. The ordinary rule of addition of magnetic quantum numbers would give[10,11]

$$\bar{M} = \bar{M}_1 + \bar{M}_2 \qquad (393)$$

where $\bar{M}$, being a magnetic quantum number, is itself written as

$$\bar{M} = Me^{j\theta_M} = m \cos\theta_M\ e^{j\theta_M} = m \cos\theta_\phi\ e^{-j\theta_\phi} \qquad (394)$$

where m should be given by

$$m = m_1 + m_2 = \text{integer} \qquad (395)$$

and θ_ϕ = constant. The real and imaginary parts of equation (393) are

$$m \cos^2\theta_M = m_1 \cos^2\theta_{M1} + m_2 \cos^2\theta_{M2} \qquad (396)$$

$$m \sin\theta_M \cos\theta_M = m_1 \sin\theta_{M1} \cos\theta_{M1} + m_2 \sin\theta_{M2} \cos\theta_{M2} \qquad (397)$$

Equations (396) and (397) determine m and θ_M in terms of m_1 , θ_{M1} and m_2 , θ_{M2} . In general the calculated value of m will not be an integer $m = m_1 + m_2$ except for the special case of $\theta_M = \theta_{M1} = \theta_{M2}$. Therefore equations (396) and (397) do not yield integer values for m , and therefore equation (393) in general does not agree with equation (395). This suggests that in broken symmetry space-time there exists an internal space scalar particle such that the addition law for complex magnetic quantum numbers must be written as

$$\bar{M} + A = \bar{M}_1 + \bar{M}_2 \tag{398}$$

instead of equation (393). Equation (398) then represents two equations for the two unknowns A and θ_M in the following manner

$$m \cos^2 \theta_M + A = m_1 \cos^2 \theta_{M1} + m_2 \cos^2 \theta_{M2} \tag{399}$$

$$m \sin \theta_M \cos \theta_M = m_1 \sin \theta_{M1} \cos \theta_{M1} + m_2 \sin \theta_{M2} \cos \theta_{M2} \tag{400}$$

where the integer m is given by equation (395).

The solution of equations (399) and (400) is easily obtained to be

$$A = - m/2 \left[1 + (1 - 4f^2)^{1/2}\right] + m_1 \cos^2 \theta_{M1} + m_2 \cos^2 \theta_{M2} \tag{401}$$

$$\cos^2 \theta_M = 1/2 \left[1 + (1 - 4f^2)^{1/2}\right] \tag{402}$$

where

$$f = 1/m \left(m_1 \sin \theta_{M1} \cos \theta_{M1} + m_2 \sin \theta_{M2} \cos \theta_{M2}\right) \tag{403}$$

If $m = 0$, then $A = 0$. Equation (401) can be written as

$$A = - m \cos^2 \theta_M + m_1 \cos^2 \theta_{M1} + m_2 \cos^2 \theta_{M2} \tag{404}$$

If $|m_1| = |m_2|$ then $m = 0$ or $2m_1$, and $A = 0$ and $\theta_M = \theta_{M1} = \theta_{M2}$ for this special case. Thus only when $|m_1| \neq |m_2|$ will the A particle carry angular momentum. Therefore in general equation (393) is not valid except for the special case $|m_1| = |m_2|$. If $\theta_M = \theta_{M1} = \theta_{M2}$, then $A = 0$ for all values of m_1 and m_2 . Finally, if $\theta_{M1} = 0$ and $\theta_{M2} = 0$ corresponding to an internally symmetrical system then $\theta_M = 0$ and $A = 0$.

The additional internal scalar particle that is required for the conservation of quantized angular momentum in systems with broken azimuthal symmetry is called the muthon.[7] This particle may be important in the explanation of high temperature superconductivity in the planar copper oxides because it may mediate the attractive forces between Cooper pairs of electrons or holes. It may be responsible for the coherent time state that is associated with high temperature superconductivity.

B. Addition of External Angular Momenta

The standard result for the addition of angular momentum operators is that they add vectorially as follows[9]

$$\vec{J} = \vec{J}_1 + \vec{J}_2 \tag{405}$$

whose z - component equation is

$$J_z = J_{1z} + J_{2z} \tag{406}$$

These operators have the following eigenvalues[9]

$$\vec{J}^2 = j(j+1)\hbar \tag{407}$$

$$J_z = m_J \hbar \tag{408}$$

where $j = 0,1,2,3,\cdots$, and $m_J = j, j-1, j-2, \cdots, 0, -1, -2, \cdots, -j$.

The corresponding equations for a broken symmetry spacetime are

$$\bar{\vec{J}} + \vec{A} = \bar{\vec{J}}_1 + \bar{\vec{J}}_2 \tag{409}$$

$$\bar{J}_z + A_z = \bar{J}_{1z} + \bar{J}_{2z} \tag{410}$$

where $\vec{A}$ and A_z do not have internal phases. The following eigenvalues are associated with external momenta in broken symmetry space and time

$$\bar{\vec{J}}^2 = \bar{J}(\bar{J}+1) \tag{411}$$

$$\bar{J}_z = \bar{M}_J = m_J \cos\theta_{MJ}\, e^{j\theta_{MJ}} \tag{412}$$

$$\bar{J}_{1z} = \bar{M}_{J1} = m_{J1} \cos\theta_{MJ1}\, e^{j\theta_{MJ1}} \tag{413}$$

$$\bar{J}_{2z} = \bar{M}_{J2} = m_{J2} \cos\theta_{MJ2}\, e^{j\theta_{MJ2}} \tag{414}$$

where for external motion θ_{MJ1} , θ_{MJ2} and θ_{MJ} are constants, and[7]

$$\bar{J} = \bar{M}'_J + j - |m_J| \qquad \bar{M}'_J = |m_J| \cos\theta_{MJ}\, e^{j\theta_{MJ}} \tag{415}$$

and $\bar{M}_J$ is related to $\bar{M}_{J1}$ and $\bar{M}_{J2}$ by placing equations (412) through (414) into equation (410) which gives a result essentially equivalent to equation (398), and where as before $m_J = m_{J1} + m_{J2}$. Equation (410) determines A_z and θ_{MJ} .

For the case of the coupling of spin and angular momentum, the following are standard results[9-11]

$$\vec{J} = \vec{L} + \vec{S} \tag{416}$$

$$J_z = L_z + S_z \tag{417}$$

$$\vec{L}^2 = \ell(\ell + 1) \tag{418}$$

$$\vec{S}^2 = s(s + 1) = 3/4 \tag{419}$$

$$L_z = m_L \tag{420}$$

$$S_z = m_S \tag{421}$$

$$J_z = m_J = m_L + m_S \tag{422}$$

$$\vec{J}^2 = j(j + 1) \tag{423}$$

$$j = |m_J| + \nu \tag{424}$$

$$\ell = |m_L| + \nu' \tag{425}$$

$$s = |m_S| = 1/2 \tag{426}$$

where $\ell = 0,1,2,3,\cdots$; $m_L = 0,\pm1,\pm2,\cdots,\pm\ell$; $m_S = \pm1/2$; $m_J = \pm1/2, \pm3/2, \pm5/2, \cdots$; $j = 1/2, 3/2, 5/2, \cdots$; and ν and $\nu' = 0,1,2,3,\cdots$.

The generalization of these equations to the case of broken internal symmetry for external motion is as follows

$$\vec{\bar{J}} + \vec{A} = \vec{\bar{L}} + \vec{\bar{S}} \tag{416A}$$

$$\bar{J}_z + A_z = \bar{L}_z + \bar{S}_z \tag{417A}$$

$$\vec{\bar{L}}^2 = \bar{\ell}(\bar{\ell} + 1) \tag{418A}$$

$$\vec{\bar{S}}^2 = \bar{s}(\bar{s} + 1) \tag{419A}$$

$$\bar{L}_z = \bar{M}_L = m_L \cos\theta_{ML}\, e^{j\theta_{ML}} \tag{420A}$$

$$\bar{S}_z = \bar{M}_S = m_S \cos\theta_{MS}\, e^{j\theta_{MS}} \tag{421A}$$

$$\bar{J}_z = \bar{M}_J = m_J \cos\theta_{MJ}\, e^{j\theta_{MJ}} \qquad m_J = m_L + m_S \tag{422A}$$

$$\vec{\bar{J}}^2 = \bar{J}(\bar{J} + 1) \tag{423A}$$

$$\bar{J} = \bar{M}_J' + j - |m_J| \tag{424A}$$

$$\bar{\mathcal{L}} = \bar{M}_L' + \ell - |m_L| \tag{425A}$$

$$\bar{\mathcal{S}} = \bar{M}_S' \tag{426A}$$

where

$$\bar{M}_J' = |m_J| \cos \theta_{MJ} \, e^{j\theta_{MJ}} \qquad \bar{M}_J = m_J \cos \theta_{MJ} \, e^{j\theta_{MJ}} \tag{427}$$

$$\bar{M}_L' = |m_L| \cos \theta_{ML} \, e^{j\theta_{ML}} \qquad \bar{M}_L = m_L \cos \theta_{ML} \, e^{j\theta_{ML}} \tag{428}$$

$$\bar{M}_S' = |m_S| \cos \theta_{MS} \, e^{j\theta_{MS}} \qquad \bar{M}_S = m_S \cos \theta_{MS} \, e^{j\theta_{MS}} \tag{429}$$

and where $m_S = \pm 1/2$ and $|m_S| = 1/2$ in equation (429). Note that $\vec{A}$ is a scalar in internal space. The unknown constants A_z and θ_{MJ} are then calculated from equations (401) through (403) in terms of m_L , m_S , θ_{ML} and θ_{MS} using the following value of the function f

$$f = 1/m_J \, (m_L \sin \theta_{ML} \cos \theta_{ML} + m_S \sin \theta_{MS} \cos \theta_{MS}) \tag{430}$$

For the case of spin and orbital angular momentum $m_J \neq 0$ because spin is half-integer and $m_J = m_L + m_S \neq 0$. For this case $A_z \neq 0$ always. From equation (417A) it follows that

$$m_J \cos \theta_{MJ} \, e^{j\theta_{MJ}} + A_z = m_L \cos \theta_{ML} \, e^{j\theta_{ML}} + m_S \cos \theta_{MS} \, e^{j\theta_{MS}} \tag{431}$$

whose real and imaginary parts are given by

$$m_J \cos^2 \theta_{MJ} + A_z = m_L \cos^2 \theta_{ML} + m_S \cos^2 \theta_{MS} \tag{432}$$

$$m_J \sin \theta_{MJ} \cos \theta_{MJ} = m_L \sin \theta_{ML} \cos \theta_{ML} + m_S \sin \theta_{MS} \cos \theta_{MS} \tag{433}$$

This gives A_z and θ_{MJ} as in equations (401) and (402). Note that equation (422) $m_J = m_L + m_S$ is a universal law that holds whether or not spacetime has broken internal symmetries and must be used in conjunction with equations (430), (432) and (433) to determine A_z and θ_{MJ} .

By writing the complex quantum numbers $\bar{J}$, $\bar{\mathcal{L}}$ and $\bar{\mathcal{S}}$ as

$$\bar{J} = Je^{j\theta J} \qquad \bar{\mathcal{L}} = \mathcal{L}e^{j\theta \mathcal{L}} \qquad \bar{\mathcal{S}} = \mathcal{S}e^{j\theta \mathcal{S}} \tag{434}$$

equations (424A), (425A) and (426A) can be used to obtain expressions for J , θ_J , $\mathcal{L}$, $\theta_{\mathcal{L}}$ and $\mathcal{S}$, $\theta_{\mathcal{S}}$. From equation (424A) it follows that

$$J \cos \theta_J = |m_J| \cos^2 \theta_{MJ} + j - |m_J| \tag{435}$$

$$J \sin \theta_J = |m_J| \sin \theta_{MJ} \cos \theta_{MJ} \tag{436}$$

which gives

$$\tan \theta_J = (|m_J| \sin \theta_{MJ} \cos \theta_{MJ})/(j - |m_J| \sin^2 \theta_{MJ}) \tag{437}$$

$$\theta_J \sim |m_J| \theta_{MJ}/j \tag{438}$$

$$J = j[1 - |m_J|/j(2 - |m_J|/j)\sin^2 \theta_{MJ}]^{1/2}$$

Equation (425A) gives

$$\mathfrak{L} \cos \theta_{\mathfrak{L}} = |m_L| \cos^2 \theta_{ML} + \ell - |m_L| \tag{440}$$

$$\mathfrak{L} \sin \theta_{\mathfrak{L}} = |m_L| \sin \theta_{ML} \cos \theta_{ML} \tag{441}$$

which gives[7]

$$\tan \theta_{\mathfrak{L}} = (|m_L| \sin \theta_{ML} \cos \theta_{ML})/(\ell - |m_L| \sin^2 \theta_{ML}) \tag{442}$$

$$\theta_{\mathfrak{L}} \sim |m_L| \theta_{ML}/\ell \tag{443}$$

$$\mathfrak{L} = \ell[1 - |m_L|/\ell(2 - |m_L|/\ell)\sin^2 \theta_{ML}]^{1/2} \tag{444}$$

Finally for the spin, equations (426A) and (429) give

$$\mathfrak{S} = |m_S| \cos \theta_{MS} = 1/2 \cos \theta_{MS} \tag{445}$$

$$\theta_{\mathfrak{S}} = \theta_{MS} \tag{446}$$

Because both spin and orbital angular momentum are associated with rotations it follows that for external motion

$$\theta_{ML} = -\theta_{\phi L} \qquad \theta_{MS} = -\theta_{\phi S} \qquad \theta_{MJ} = -\theta_{\phi J} \tag{447}$$

where $\theta_{\phi L}$, $\theta_{\phi S}$ and $\theta_{\phi J}$ are constants.

A theory of the addition of internal spin and internal orbital angular momentum can be developed except that for this case the basic representation in equation (394) is not valid when θ_M and θ_ϕ are variables for a fixed value of m.

7. CONCLUSION. The broken symmetry of spacetime requires that the momentum, angular momentum and energy operators are complex numbers in internal space. These operators can be expressed in terms of the measured space and time coordinates by noting that the measured coordinates are the real parts of the complex number coordinates. The energy eigenvalues and eigenfunctions of the 3-D linear harmonic oscillator are obtained for broken symmetry space and time and it is found that the principal and magnetic quantum numbers are complex numbers whose difference must be an integer. The momentum operator, energy operator and Schrödinger's equation can be developed for the case of pure internal phase rotations with constant coordinate magnitude, and the equations for an internal space harmonic oscillator and for a particle in an internal space box can be developed. A muthon particle must exist to account for the conservation of angular momentum in space and time with a broken azimuthal symmetry. This particle may play a role in the explanation of high temperature superconductivity in the planar copper oxides where broken azimuthal symmetry may be associated with a coherent time field and a raised level of the superconductivity energy gap.

ACKNOWLEDGEMENT

The author would like to thank Elizabeth K. Klein for typing this paper.

REFERENCES

1. Aitchison, I. and Hey, A., Gauge Theories in Particle Physics, Adam Hilger, Bristol, 1982.

2. Cheng, T. and Li, L. Gauge Theory of Elementary Particle Physics, Oxford Univ. Press, New York, 1984.

3. Okun, L. B., Particle Physics, Harwood Academic, New York, 1985.

4. Sakurai, J. J., Invariance Principles and Elementary Particles, Princeton University Press, Princeton, 1964.

5. Nishijima, K., Fundamental Particles, Benjamin, New York, 1964.

6. Weiss, R. A., Relativistic Thermodynamics, Exposition Press, New York, 1976.

7. Weiss, R. A., Gauge Theory of Thermodynamics, K&W Publications, Vicksburg, MS, 1989.

8. Pauling, L. and Wilson, E. B., Introduction to Quantum Mechanics, McGraw-Hill, New York, 1935.

9. Schiff, L. I., Quantum Mechanics, McGraw-Hill, New York, 1955.

10. Edmonds, A. R., Angular Momentum in Quantum Mechanics, Princeton Univ. Press, Princeton, 1957.

11. Condon, E. U. and Schortley, G. H., The Theory of Atomic Spectra, Cambridge Univ. Press, New York, 1963.

12. Eder, G., Nuclear Forces, MIT Press, Cambridge, MA, 1968.

13. Morse, P. M. and Feshbach, H., Methods of Theoretical Physics, Part 2, McGraw-Hill, New York, 1953.

GAUGE THEORY OF TIME

Richard A. Weiss
U. S. Army Engineer Waterways Experiment Station
Vicksburg, Mississippi 39180

ABSTRACT. A gauge invariant differential equation is developed that relates time to the pressure and energy density of a thermodynamic system. This renormalization group equation for time is developed as a relativistic trace equation that involves the same gauge parameters that are used in the relativistic trace equation for the internal energy. In conjunction with the time equation and the internal energy equation, two equations giving the dependence of the dimensions of space and time on the local energy density and pressure are developed. Four simultaneous equations must therefore be solved to determine the reaction rates of processes that occur in real media. The results of these calculations indicate that physical processes occurring in a pressure and energy density field associated with an ambient background material will run more slowly than the reaction rates predicted by conventional calculations. This is due to the non-ideal character of the state equation of the background substance; for ideal state equations the renormalized and conventionally calculated reaction rates are equal. The scale and gauge invariance requirement of the equation of time gives a set of equations that describe the time evolution of an interacting thermodynamic system in terms of the fundamental gauge parameters of the equation of time. The effects of mechanical or electromagnetic radiation on the equation of time is treated by a perturbation technique. Applications to solids, quantum liquids and real gases are considered. A theory of coherent time states in matter is developed and applied to the development of a theory of high temperature superconductivity in the planar copper oxides. This theory predicts that the value of the normalized superconductivity energy gap $2\Delta/kT_c$ should be $6/\pi$ times greater than the value of the normalized energy gap predicted by the BCS theory for conventional superconductors.

1. INTRODUCTION. The nature of time is still a mystery today although theories about time date back to the ancient civilizations.[1-27] The Babylonians, Greeks and Romans believed in cyclic time and the theme of recurrence appears repeatedly in their literature.[9-11] Aristotle and Plato both believed in cyclic time. With the advent of Judaism and Christianity came the idea that events occurred in linear time.[9,11] Only with the development of the mechanical clock did time acquire an abstract nature of mathematically measured sequences, and time as an abstract parameter was introduced by Galileo in his studies of the motion of bodies.[9,11] Newton and Leibniz introduced the concept of instantaneous velocity, and it was Newton who introduced the idea of absolute time that flows uniformly without any relation to external things.[11] However, it was Einstein who made the important discovery that time is in fact local and dependent on the relative velocity of observers.[2-14,17,18,22-27] It was also Einstein who demonstrated that time slows down in the presence of a gravitational field by predicting the gravitational red shift of spectral lines in light from stars like the sun.[5,17] Time also plays an

important role in the rhythms and processes of biological systems.[5,17] In thermodynamics time is closely related to the second law of thermodynamics for irreversible processes where the concept of time's arrow appears.[1-4,7,8,22,24] Cosmology is intimately related to the concepts of time, and it has been suggested that the expansion of the universe is the prime cause of the arrow of time in thermodynamic systems.[7,11,16-18] In quantum gravity theory the concept of time itself becomes nebulous and may not be definable in the early universe.[28] Time and thermodynamics are closely intertwined. A deficiency in the present day concepts of time is the lack of a theory that describes the effects of ambient bulk matter and energy on time, i.e., a theory that describes the effects of pressure and temperature on time in gases, liquids and solids.

This paper develops a thermodynamic gauge theory of time that is based on a previously developed relativistic gauge theory of thermodynamics.[29] This gauge theory of bulk matter and energy is formulated by means of a relativistic trace equation which can be written as follows[29]

$$\mathcal{E} + \beta_{\mathcal{E}} - 3\beta_P = \mathcal{E}^a + \beta^a_{\mathcal{E}} \tag{1}$$

where the gauge functions $\beta_{\mathcal{E}}$, β_P and $\beta^a_{\mathcal{E}}$ are given by

$$\beta_{\mathcal{E}} = T/V(dU/dT)_{PV} \equiv T/V\ C_{PV} \tag{2}$$

$$\beta_P = d/dV(PV)_U \equiv P - K_U \tag{3}$$

$$\beta^a_{\mathcal{E}} = T/V(dU^a/dT)_{P^aV} \tag{4}$$

and where $\mathcal{E}$, U and P = renormalized energy density, internal energy and pressure respectively; $\mathcal{E}^a$, U^a and P^a = unrenormalized energy density, internal energy and pressure respectively; T and V = absolute temperature and volume respectively, and the heat capacity at constant PV is given by

$$C_{PV} = (dU/dT)_{PV} \tag{5}$$

and where the bulk modulus at constant U is given by

$$K_U = - V(dP/dV)_U = n(dP/dn)_U \tag{6}$$

where n = 1/V . Equation (1) requires the renormalized state equation to be softer at high densities than the state equation predicted by the unrenormalized calculation.[29] Figure 1 shows a comparison of the average energy per particle for the renormalized and conventional neutron star matter state equations.

A generalization of equation (1) to thermodynamic systems with broken internal symmetry has been proposed for partially coherent matter in the form[30]

$$\bar{\mathcal{E}} + \bar{\beta}_{\mathcal{E}} - 3\bar{\beta}_P = \mathcal{E}^a + \beta^a_{\mathcal{E}} \tag{7}$$

where the complex number gauge functions $\bar{\beta}_E$ and $\bar{\beta}_P$ are given by[30]

$$\bar{\beta}_E = T/V(d\bar{U}/dT)_{\bar{P}V} \qquad \bar{\beta}_P = d/dV(\bar{P}V)_{\bar{U}} \tag{8}$$

and $\bar{U}$ and $\bar{P}$ = complex number internal energy and pressure respectively. The vacuum equations corresponding to equations (1) and (7) are respectively[30]

$$E^v + \beta_E^v - 3\beta_P^v = 0 \tag{9}$$

$$\bar{E}^v + \bar{\beta}_E^v - 3\bar{\beta}_P^v = 0 \tag{10}$$

Equation (9) determines E^v and P^v while equation (10) determines E^v , θ_{Ev} and P^v , θ_{Pv} . Thus E^v and P^v have finite values for the vacuum.[30] Equations (1), (7), (9) and (10) also have zero-temperature forms which are written as follows

$$E_o - 3\beta_P^o = E_o^a \qquad \bar{E}_o - 3\bar{\beta}_P^o = E_o^a \tag{11}$$

$$E_o^v - 3\beta_P^{ov} = 0 \qquad \bar{E}_o^v - 3\bar{\beta}_P^{ov} = 0 \tag{12}$$

where[29,30]

$$\beta_P^o = (1 + \gamma_o)P_o - K_o \qquad \bar{\beta}_P^o = (1 + \bar{\gamma}_o)\bar{P}_o - \bar{K}_o \tag{13}$$

$$\beta_P^{ov} = (1 + \gamma_o^v)P_o^v - K_o^v \qquad \bar{\beta}_P^{ov} = (1 + \bar{\gamma}_o^v)\bar{P}_o^v - \bar{K}_o^v \tag{14}$$

where γ_o , K_o and $\bar{\gamma}_o$, $\bar{K}_o$ = scalar and complex number values of the zero-temperature Grüneisen parameter and bulk modulus of matter respectively, and γ_o^v , K_o^v and $\bar{\gamma}_o^v$, $\bar{K}_o^v$ = corresponding values of these quantities for the vacuum. The zero-temperature values of the bulk modulus are given by

$$\begin{aligned} K_o &= ndP_o/dn & \bar{K}_o &= nd\bar{P}_o/dn \\ K_o^v &= ndP_o^v/dn & \bar{K}_o^v &= nd\bar{P}_o^v/dn \end{aligned} \tag{15}$$

For T = 0 the value of K_U in equation (6) is[29]

$$K_U^o = K_o - \gamma_o P_o \qquad \bar{K}_U^o = \bar{K}_o - \bar{\gamma}_o \bar{P}_o \tag{16}$$

The zero-temperature Grüneisen parameter γ_o is the gauge parameter for the T = 0 theory.[29,30] For systems with PV = αU or $\bar{P}V = \alpha\bar{U}$ where α = constant it follows that

$$\begin{aligned} &\beta_E = 0 \quad \beta_P = 0 \quad E = E^a \\ &\bar{\beta}_E = 0 \quad \bar{\beta}_P = 0 \quad \bar{E} = E^a = \text{real number} \end{aligned} \tag{17}$$

Equation (17) describes the ideal classical gas and the ideal nonrelativistic and ultrarelativistic Fermi gases. It is plausible that the gauge theory of time that will be developed in this paper must incorporate the functions β_E and β_P as coefficients in the fundamental equation of time in a manner similar to the way they are used in the energy trace equation (1).

It has been suggested that the complex number trace equation (7) with its complex number internal energy and pressure implies that the coordinates of space and time must also exhibit broken internal symmetries and be represented as complex numbers.[30] The internal energy, pressure and spacetime coordinates can then be written as[30]

$$\bar{U} = Ue^{j\theta U} \qquad \bar{P} = Pe^{j\theta P} \tag{18}$$

$$\bar{x} = xe^{j\theta x} \qquad \bar{y} = ye^{j\theta y} \tag{19}$$

$$\bar{z} = ze^{j\theta z} \qquad \bar{t} = te^{j\theta t} \tag{20}$$

where U , θ_U , P , θ_P , x , θ_x , y , θ_y , z , θ_z and t , θ_t = magnitudes and internal phase angles of the internal energy, pressure, spatial coordinates and time respectively.[30] Time varying processes are described in this paper as being incoherent (or linear) if $dt >> td\theta_t$ and coherent (or circular) if $dt << td\theta_t$. In the general case $dt \sim td\theta_t$ and time both stretches and rotates with changes in the energy density and pressure of matter.

Equation (1) with its gauge functions β_E and β_P was developed in part to account for the need to have a softer equation of state of nuclear matter and the neutron gas.[29] This paper will use the gauge functions β_E and β_P to develop renormalization group equations for coherent and incoherent time whose solutions determine the reduced rate of processes that occur within the energy density and pressure fields of real (non-ideal) thermodynamic systems. Time is locally dependent on the energy density and pressure of the ambient matter, $t = t(E,P)$ or $\bar{t} = \bar{t}(E,P)$, and is determined by relativistic renormalization group equations. These calculations do not change the rate of processes that occur within an ideal ambient thermodynamic system. A theory of coherent (or circular) time is developed and applied to high temperature superconductivity in the planar copper oxides.

This paper is organized as follows: Section 2 develops a gauge theory of scalar and complex number time and presents a basic partial differential equation for time in bulk matter and the vacuum, Section 3 treats the differential equations that determine the dimensions of space and time in ambient bulk matter and energy, Section 4 considers scale and gauge invariance of the equation of time and its connection with nonequilibrium thermodynamics, Section 5 studies the time equation for solids, quantum liquids and real gases, Section 6 considers the effects of vibrations on the determination of time in matter, and finally Section 7 investigates coherent time states in matter and their application to high temperature superconductivity.

2. GAUGE THEORY OF TIME. This section develops a relativistic partial differential equation of time that describes the dependence of time on the local energy density and pressure of matter and energy. Both scalar and complex number equations of time are developed which can be expressed in terms of the temperature and particle number density of matter.

A. Scalar Equation of Time.

The chosen form of the equation of time is based on five conditions: a) that it be linear in time, b) it must contain the Minkowski spacetime signature, c) it must involve the gauge functions β_E and β_P so that the time equation affects real thermodynamic systems and produces a null effect for ideal systems in a manner similar to that of the energy trace equation (1), d) that the effect in real gases occurs in the third and higher virial time coefficients in order to agree with the effects of the energy trace equation (1) on the real gases, and e) that the time equation be scale and gauge invariant. These conditions and an inspection of the form of the energy trace equation (1) suggest that the proper equation for incoherent (scalar) time is

$$t - \beta_E \, \partial t/\partial E + 3\beta_P \, \partial t/\partial P = t^a - \beta_E^a \, \partial t^a/\partial E^a \tag{21}$$

where t = renormalized time, and t^a = unrenormalized time. Equation (21) can be written in operator form as

$$\mathcal{L} t = t^a - \beta_E^a \, \partial t^a/\partial E^a \tag{22}$$

where the time operator $\mathcal{L}$ is given by

$$\mathcal{L} = 1 - \beta_E \, \partial/\partial E + 3\beta_P \, \partial/\partial P \tag{23}$$

Equation (21) is a renormalization group equation of time that treats time as a function of the local energy density and pressure, $t = t(E,P)$. The zero-temperature version of the time equation (21) is written as

$$t_o + 3\beta_P^o \, dt_o/dP_o = t_o^a \tag{24}$$

where t_o and t_o^a = zero-temperature values of the renormalized and unrenormalized time, P_o = zero-temperature value of the pressure, and β_P^o is given by equation (13). From equation (21) it follows that

$$t > t^a \qquad \beta_P < 0 \qquad \text{high density and temperature} \tag{25}$$

$$t < t^a \qquad \beta_P > 0 \qquad \text{low density and temperature} \tag{26}$$

while equation (24) gives

$$t_o > t_o^a \qquad \beta_P^o < 0 \qquad \text{high density} \tag{27}$$

$$t_o < t_o^a \qquad \beta_P^o > 0 \qquad \text{low density} \tag{28}$$

Figure 2 shows the relative behaviour of t and t^a in terms of density. For ideal systems $\beta_E = 0$, $\beta_P = 0$ and $\beta_E^a = 0$ so that equation (21) gives $t = t^a$ and equation (24) gives $t_o = t_o^a$ for an ideal T = 0 system.

According to equation (21) the renormalized time depends on the local enerry density and pressure of matter and energy. At high densities and temperatures of interacting matter and energy the renormalized time slows down (the time interval between two events is larger) relative to the unrenormalized time evaluated at the same energy density and pressure. Accordingly, the rates of processes that occur at high pressures and temperatures are expected to be slower than is conventionally predicted. Equation (21) gives the renormalized time in the presence of ambient matter and energy. For the vacuum equation (21) becomes

$$t^v - \beta_E^v \, \partial t^v/\partial E^v + 3\beta_P^v \, \partial t^v/\partial P^v = t^{va} \tag{29}$$

where the renormalized vacuum energy density E^v and pressure P^v are obtained from equation (9) as the homogeneous solution to the energy trace equation. Because t^{va} = constant, equation (29) can be rewritten as

$$t^{v'} - \beta_E^v \, \partial t^{v'}/\partial E^v + 3\beta_P^v \, \partial t^{v'}/\partial P^v = 0 \tag{30}$$

by a change of variable $t^{v'} = t^v - t^{va}$. Equations (9) and (30) show that t^v , E^v and P^v have nonzero values for the vacuum. The general solution of equation (21) can be written as

$$t = t_p(E,P,t^a) + t^{v'}(E^v,P^v) \tag{31}$$

where t_p = particular solution of equation (21). The energy trace equation (1) can be used to determine E(n,T) and P(n,T), and if $t^a = t^a(n,T)$ and $t^v = t^v(n,T)$ *it follows that*

$$t = t_p(n,T) + t^{v'}(n,T) \tag{32}$$

where n = particle number density of ambient matter. The constant t^{va} = characteristic time for a process occurring when the ambient medium is the vacuum.

B. Complex Number Time Equation.

It has been suggested that time is associated with an internal phase and must be written as in equation (20). Therefore the simplest generalization of the scalar time equation (21) is the following complex number equation for partially coherent time

$$\bar{t} - \beta_E \, \partial\bar{t}/\partial E + 3\beta_P \, \partial\bar{t}/\partial P = t^a - \beta_E^a \, \partial t^a/\partial E^a \tag{33}$$

The solution of equation (33) gives t and θ_t in terms of energy density and pressure. The complex number equation of time for the vacuum is obtained from equation (33) with t^{av} = constant to be

$$\bar{t}^{v\prime} - \beta_E^v \, \partial\bar{t}^{v\prime}/\partial E^v + 3\beta_P^v \, \partial\bar{t}^{v\prime}/\partial P^v = 0 \tag{34}$$

where the constant value of t^{va} is introduced as $\bar{t}^{v\prime} = \bar{t}^{v} - t^{va}$. The solution to equation (34) gives t^v and θ_t^v . The general solution of equation (33) can be written as the sum of a particular and a homogeneous (vacuum) solution as follows

$$\bar{t} = \bar{t}_p(n,T) + \bar{t}^{v\prime}(n,T) \tag{35}$$

A more complicated generalization of equation (21) would incorporate complex number energy density and pressure as follows

$$\bar{t} - \bar{\beta}_E \, \partial\bar{t}/\partial\bar{E} + 3\bar{\beta}_P \, \partial\bar{t}/\partial\bar{P} = t^a - \beta_E^a \, \partial t^a/\partial E^a \tag{36}$$

Correspondingly, a more complicated vacuum time equation is obtained from equation (36) to be

$$\bar{t}^v - \bar{\beta}_E^v \, \partial\bar{t}^{v\prime}/\partial\bar{E}^v + 3\bar{\beta}_P^v \, \partial\bar{t}^{v\prime}/\partial\bar{P}^v = 0 \tag{37}$$

Neither equation (36) or (37) is considered in this paper. The complex number time equation (33) will be used in Section 7 to introduce the concept of coherent time states in matter and to describe high temperature superconductivity in the planar copper oxides.

C. State Equation for Time and Alternate Form of Time Equation.

In this subsection the dependence of time on density and temperature is considered and the equation of time is rewritten in terms of these parameters. For instance, the chain rule for derivatives gives

$$\partial t/\partial n = \partial t/\partial E \, \partial E/\partial n + \partial t/\partial P \, \partial P/\partial n \tag{38}$$

$$\partial t/\partial T = \partial t/\partial E \, \partial E/\partial T + \partial t/\partial P \, \partial P/\partial T \tag{39}$$

where n = particle number density and T = absolute temperature. In fact E and P are more fundamental quantities than n and T , but n and T are conventionally used to write the state equations for matter. From equations (38) and (39) it follows that

$$\partial t/\partial P = e\partial t/\partial n - f\partial t/\partial T \tag{40}$$

$$\partial t/\partial E = h\partial t/\partial T - g\partial t/\partial n \tag{41}$$

where

$$e = 1/D_e \, \partial E/\partial T \qquad f = 1/D_e \, \partial E/\partial n \tag{42}$$

$$h = 1/D_e \, \partial P/\partial n \qquad g = 1/D_e \, \partial P/\partial T \tag{43}$$

$$D_e = \partial P/\partial n \, \partial E/\partial T - \partial P/\partial T \, \partial E/\partial n \tag{44}$$

Combining equations (40) and (41) with the equations of time (21) and (33) gives the following alternative forms for the equations of time

$$t - q_1 \, \partial t/\partial T + s_1 \, \partial t/\partial n = t^a - q^a \, \partial t^a/\partial T + s^a \, \partial t^a/\partial n \tag{45A}$$

$$\bar{t} - q_1 \, \partial \bar{t}/\partial T + s_1 \, \partial \bar{t}/\partial n = t^a - q^a \, \partial t^a/\partial T + s^a \, \partial t^a/\partial n \tag{45B}$$

where

$$q_1 = h\beta_E + 3f\beta_P \tag{46}$$

$$s_1 = g\beta_E + 3e\beta_P \tag{47}$$

$$q^a = h^a \beta_E^a \tag{48}$$

$$s^a = g^a \beta_E^a \tag{49}$$

where h^a and g^a are calculated in a manner identical to that in equations (43) and (44) except that the superscript "a" is added to $\bar{E}$ and P . Solutions to equations (45A) and (45B) are useful for astrophysical and geophysical problems because having a state equation for time of the form $t = t(n,T)$ and $\theta_t = \theta_t(n,T)$ leads to a calculation of the rate of change of time with radial distance as follows

$$\partial t/\partial r = \partial t/\partial T \, \partial T/\partial r + \partial t/\partial n \, \partial n/\partial r \tag{50A}$$

$$\partial \theta_t/\partial r = \partial \theta_t/\partial T \, \partial T/\partial r + \partial \theta_t/\partial n \, \partial n/\partial r \tag{50B}$$

where $\partial T/\partial r$ and $\partial n/\partial r$ are calculated from stellar equilibrium equations. Equation (45B) is used in Section 7 to describe coherent time states and high temperature superconductivity.

D. Formal Solution for the Zero-Temperature Scalar Time Equation.

The solutions of the time equations (21) and (33) or (45A) and (45B) are generally difficult to obtain. However, the solution to the zero-temperature time equation (24) can be obtained formally because of its simple form. Standard methods show that equation (24) has the following solution[31]

$$t_o = C_o e^{\phi(n)} + 1/3 e^{\phi(n)} \int^n G_o'/H_o' \, dn'/n' \tag{51A}$$

where

$$G_o' = e^{-\phi(n')} t_o^a(n') \qquad H_o' = \beta_p^o(n')/K_o(n') \tag{51B}$$

$$\phi(n) = -1/3\int 1/H_o \, dn/n \qquad H_o = \beta_p^o(n)/K_o(n) \tag{52}$$

where the zero-temperature bulk modulus K_o is given by equation (15) and where from equation (13)

$$H_o = \beta_p^o(n)/K_o(n) = (1 + \gamma_o)P_o/K_o - 1 \tag{53}$$

The zero-temperature vacuum solution is

$$t_o^v = C_o e^{\phi(n)} + t_o^{va} \tag{54}$$

Finally, because of the appearance of the renormalized quantities P_o, K_o and γ_o in equation (51) it is clear that a prerequisite for the solution of the zero-temperature time equation (24) is the solution of the zero-temperature energy trace equation (11).

E. Renormalized Process Rates.

The simplest calculation of renormalized process rates is to assume t = characteristic renormalized time for a process, and t^a = characteristic unrenormalized time for a process. Then the corresponding rates are given by

$$R = 1/t \qquad R^a = 1/t^a \tag{55}$$

where R = renormalized process rate, and R^a = rate given by conventional calculation. Placing equation (55) into equation (11) gives

$$1/R^2(R + \beta_E \, \partial R/\partial E - 3\beta_p \, \partial R/\partial P) = (1/R^a)^2(R^a + \beta_E^a \, \partial R^a/\partial E^a) \tag{56}$$

Equation (56) gives the renormalized reaction rate of a process in terms of the conventionally calculated reaction rate. The reaction rate solution of equation (56) is of the form $R = R(E,P)$. For an ideal state equation of the ambient matter in which the reaction is occurring $\beta_E = 0$ and $\beta_p = 0$ so that for this case $R = R^a$. For T = 0 the reaction rate equation (56) becomes

$$1/R_o^2(R_o - 3\beta_p^o \, dR_o/dP_o) = 1/R_o^a \tag{57}$$

From equation (56) it follows that

$$R < R^a \left.\begin{array}{l} \\ \\ \end{array}\right\} \begin{array}{l} \text{high density interacting Fermi Gases} \\ \\ \text{high temperature real gases} \end{array} \tag{58}$$

$$R > R^a \left.\begin{array}{l} \\ \\ \end{array}\right\} \begin{array}{l} \text{low density Fermi gases} \\ \\ \text{low temperature real gases} \end{array} \qquad (59)$$

Equation (57) gives

$$R_o < R_o^a \quad \text{high density } T = 0 \text{ interacting Fermi gas} \qquad (60)$$

$$R_o > R_o^a \quad \text{low density } T = 0 \text{ interacting Fermi gas} \qquad (61)$$

For ideal systems $R = R^a$ and $R_o = R_o^a$. An analysis similar to that given in equations (38) through (44) gives

$$\partial R/\partial P = e\partial R/\partial n - f\partial R/\partial T \qquad (62)$$

$$\partial R/\partial E = h\partial R/\partial T - g\partial R/\partial n \qquad (63)$$

so that the reaction rate equation (56) can be written as

$$1/R^2(R + q_1\ \partial R/\partial T - s_1\ \partial R/\partial n) = (1/R^a)^2(R^a + q^a\ \partial R^a/\partial T - s^a\ \partial R^a/\partial n) \qquad (64)$$

where q_1 , s_1 , q^a and s^a are given by equations (46) through (49) respectively. The solution of equation (64) gives the renormalized reaction rate as $R = R(n,T)$. Finally, it should be pointed out that the definition of reaction rate given by equation (55) is a simplified version of the more general definition

$$R = dN/dt \qquad R^a = dN/dt^a \qquad (65)$$

where N = species particle number. From equation (65) it follows that the renormalized incoherent reaction rate is

$$R = R^a/(dt/dt^a) \qquad (66)$$

where t is given by the solution of the time equations (21) or (45). Equation (64) is useful when only a characteristic time is known for a process from which the rates can be estimated by an inverse time as in equation (55). Equation (66) is useful when a formal reaction rate function R^a given in equation (65) is known for a process. Figure 3 shows the relative behaviour of R and R^a in terms of particle number density. The complex number rate equations analogous to equations (56) and (64) are obtained from the replacement $R \to \bar{R}$. For this case equation (66) becomes

$$\bar{R} = R^a/(d\bar{t}/dt^a) \qquad (66A)$$

For completely coherent time variation $d\bar{t} = j\bar{t}d\theta_t$ and

$$\bar{R} = R^a/(j\bar{t}d\theta_t/dt^a) \qquad (66B)$$

a result which will be examined in Section 7.

3. DIMENSIONS OF SPACE AND TIME. This section considers the possibility of the dependence of the basic dimensions of space and time on the energy density and pressure of the ambient matter and energy. It is not apriori clear that the dimensions of space and time are independent of the local pressure and energy density. This section develops a set of four coupled partial differential equations for the energy density, time, space dimension, and time dimension. At low energy density and pressure common experience suggests that

$$D_s \sim 3 \qquad D_t \sim 1 \tag{67}$$

where D_s = renormalized space dimension and D_t = renormalized time dimension. Equation (67) may not be valid for high densities and temperatures. In this paper it will be assumed that $D_s = D_s(n,T)$ and $D_t = D_t(n,T)$ are unknown functions of particle number density and temperature that need to be determined along with $E(n,T)$ and $t(n,T)$ by a set of four coupled renormalization group equations that are similar in form to the energy trace equation (1) and the equation of time (21). It is sometimes convenient to write

$$D_s = 3 + d_s \qquad D_t = 1 + d_t \tag{68}$$

$$D_s^a = 3 + d_s^a \qquad D_t^a = 1 + d_t^a \tag{69}$$

where D_s^a and D_t^a = unrenormalized space and time dimensions respectively. Space and time are fractal when $d_s < 0$ and $d_t < 0$.[30]

For reasons similar to those used for the development of equations (1) and (21) it is suggested that the four coupled energy density, time, time dimension, and space dimension equations are

$$E + D_t\beta_E - D_s\beta_P = E^a + D_t^a\beta_E^a \tag{70}$$

$$t - D_t\beta_E\, \partial t/\partial E + D_s\beta_P\, \partial t/\partial P = t^a - D_t^a\beta_E^a\, \partial t^a/\partial E^a \tag{71}$$

$$D_t + D_t\beta_E\, \partial D_t/\partial E - D_s\beta_P\, \partial D_t/\partial P = D_t^a + D_t^a\beta_E^a\, \partial D_t^a/\partial E^a \tag{72}$$

$$D_s + D_t\beta_E\, \partial D_s/\partial E - D_s\beta_P\, \partial D_s/\partial P = D_s^a + D_t^a\beta_E^a\, \partial D_s^a/\partial E^a \tag{73}$$

The four unknown quantities that are determined by these equations are E , t , D_t and D_s . For low density matter $D_t = D_t^a = 1$ and $D_s = D_s^a = 3$ and in this limit the left hand sides of equations (70) through (73) have the Minkowski signature $(1 , -1 , -1 , -1)$. Using equations (38) through (44) allows equations (70) through (73) to be written as

$$E + D_t\beta_E - D_s\beta_P = E^a + D_t^a\beta_E^a \tag{74}$$

$$t - q_2\, \partial t/\partial T + s_2\, \partial t/\partial n = t^a - q_D^a\, \partial t^a/\partial T + s_D^a\, \partial t^a/\partial n \tag{75}$$

$$D_t + q_2\,\partial D_t/\partial T - s_2\,\partial D_t/\partial n = D_t^a + q_D^a\,\partial D_t^a/\partial T - s_D^a\,\partial D_t^a/\partial n \tag{76}$$

$$D_s + q_2\,\partial D_s/\partial T - s_2\,\partial D_s/\partial n = D_s^a + q_D^a\,\partial D_s^a/\partial T - s_D^a\,\partial D_s^a/\partial n \tag{77}$$

where

$$q_2 = hD_t\beta_E + fD_s\beta_P = h(1 + d_t)\beta_E + f(3 + d_s)\beta_P \qquad q_D^a = D_t^a q^a \tag{78}$$

$$s_2 = gD_t\beta_E + eD_s\beta_P = g(1 + d_t)\beta_E + e(3 + d_s)\beta_P \qquad s_D^a = D_t^a s^a \tag{79}$$

The rate equations (56) and (64) with space and time dimensions D_s and D_t respectively become

$$1/R^2(R + D_t\beta_E\,\partial R/\partial E - D_s\beta_P\,\partial R/\partial P) = (1/R^a)^2(R^a + D_t^a\beta_E^a\,\partial R^a/\partial E^a) \tag{80}$$

$$1/R^2(R + q_2\,\partial R/\partial T - s_2\,\partial R/\partial n) = (1/R^a)^2(R^a + q_D^a\,\partial R^a/\partial T - s_D^a\,\partial R^a/\partial n) \tag{81}$$

Equations (80) and (81) can replace equations (71) and (75) respectively.

For systems with $PV = \alpha U$, such as the ideal classical gas, ideal nonrelativistic Fermi gas and the ideal ultrarelativistic Fermi gas, $\beta_E = \beta_E^a = 0$ and $\beta_P = \beta_P^a = 0$ and it follows that

$$E = E^a \qquad t = t^a \qquad R = R^a \tag{82}$$

$$D_t = D_t^a = 1 \qquad D_s = D_s^a = 3 \tag{83}$$

$$d_t = d_t^a = 0 \qquad d_s = d_s^a = 0 \tag{84}$$

The choice $D_t^a \sim 1$ and $D_s^a \sim 3$ for low density and pressure systems is due to common experience. From equations (70) through (73) it follows that

$E = E^a$ $\quad t = t^a$ $\quad R = R^a$ $\quad D_t = D_t^a = 1$ $\quad D_s = D_s^a = 3$ $\quad$ zero density (85)

$E > E^a$ $\quad t < t^a$ $\quad R > R^a$ $\quad D_t > D_t^a$ $\quad D_s > D_s^a$ $\quad$ low density (86)

$E < E^a$ $\quad t > t^a$ $\quad R < R^a$ $\quad D_t < D_t^a$ $\quad D_s < D_s^a$ $\quad$ high density (87)

$E = E^a$ $\quad t = t^a$ $\quad R = R^a$ $\quad D_t = D_t^a$ $\quad D_s = D_s^a$ $\quad$ infinite density (asymptotic freedom) (88)

Figures 4 and 5 show the relative variation of D_t and D_t^a and D_s and D_s^a respectively. These figures assume a compactification of dimensions to the values $D_s^a = 3$ and $D_t^a = 1$ in the zero density limit. At high densities the renormalized dimensions satisfy $D_t < D_t^a$ and $D_s < D_s^a$ so that even at high densities the

renormalized dimensions D_s and D_t may not be too different from the 3 + 1 Minkowski spacetime.

The vacuum is a real system with the properties

$$\mathcal{E}^{va} = 0 \qquad t^{va} = \text{constant} \tag{89}$$

$$D_s^{va} = 3 \qquad D_t^{va} = 1 \qquad d_s^{va} = 0 \qquad d_t^{va} = 0 \tag{90}$$

For the vacuum equations (70) through (73) become

$$\mathcal{E}^v + D_t^v \beta_{\mathcal{E}}^v - D_s^v \beta_P^v = 0 \tag{91}$$

$$t^v - D_t^v \beta_{\mathcal{E}}^v \, \partial t^v / \partial \mathcal{E}^v + D_s^v \beta_P^v \, \partial t^v / \partial P^v = t^{va} \tag{92}$$

$$D_t^v + D_t^v \beta_{\mathcal{E}}^v \, \partial D_t^v / \partial \mathcal{E}^v - D_s^v \beta_P^v \, \partial D_t^v / \partial P^v = D_t^{va} \tag{93}$$

$$D_s^v + D_t^v \beta_{\mathcal{E}}^v \, \partial D_s^v / \partial \mathcal{E}^v - D_s^v \beta_P^v \, \partial D_s^v / \partial P^v = D_s^{va} \tag{94}$$

The vacuum is a real system that exists at finite values of density and temperature. In this paper the renormalized vacuum is simply the set of solutions to equations (70) through (73) with $\mathcal{E}^a = 0$ in equation (70). The corresponding vacuum equations for equations (74) through (77) are

$$\mathcal{E}^v + D_t^v \beta_{\mathcal{E}}^v - D_s^v \beta_P^v = 0 \tag{91A}$$

$$t^v - q_2^v \, \partial t^v / \partial T + s_2^v \, \partial t^v / \partial n = t^{va} \tag{92A}$$

$$D_t^v + q_2^v \, \partial D_t^v / \partial T - s_2^v \, \partial D_t^v / \partial n = 1 \tag{93A}$$

$$D_s^v + q_2^v \, \partial D_s^v / \partial T - s_2^v \, \partial D_s^v / \partial n = 3 \tag{94A}$$

which allow the calculation of renormalized vacuum parameters. Equations (93), (94), (93A) and (94A) can be made homogeneous by writing

$$D_t^v = 1 + d_t^v \qquad D_s^v = 3 + d_s^v \tag{95}$$

It should be noted that $D_t^v \neq D_t^{va} = 1$ and $D_s^v \neq D_s^{va} = 3$. Finally, the vacuum time equations (92) and (92A) can be made homogeneous by writing $t^{v'} = t^v - t^{va}$ as was done for equation (30). The constant t^{va} = characteristic time for a reaction occurring when the ambient medium is the vacuum. The time t^{va} depends only on physical constants and the kinematic parameters of the reaction. The renormalized vacuum values $\mathcal{E}^v$, t^v , D_t^v and D_s^v depend on density and temperature. The homogeneous forms of the vacuum equations (91) through (94) are

$$E^v + (1 + d_t^v)\beta_E^v - (3 + d_s^v)\beta_P^v = 0 \tag{91B}$$

$$t^{v'} - (1 + d_t^v)\beta_E^v \, \partial t^{v'}/\partial E^v + (3 + d_s^v)\beta_P^v \, \partial t^{v'}/\partial P^v = 0 \tag{92B}$$

$$d_t^v + (1 + d_t^v)\beta_E^v \, \partial d_t^v/\partial E^v - (3 + d_s^v)\beta_P^v \, \partial d_t^v/\partial P^v = 0 \tag{93B}$$

$$d_s^v + (1 + d_t^v)\beta_E^v \, \partial d_s^v/\partial E^v - (3 + d_s^v)\beta_P^v \, \partial d_s^v/\partial P^v = 0 \tag{94B}$$

while the homogeneous forms of the vacuum equations (91A) through (94A) are

$$E^v + (1 + d_t^v)\beta_E^v - (3 + d_s^v)\beta_P^v = 0 \tag{91C}$$

$$t^{v'} - q_2^v \, \partial t^{v'}/\partial T + s_2^v \, \partial t^{v'}/\partial n = 0 \tag{92C}$$

$$d_t^v + q_2^v \, \partial d_t^v/\partial T - s_2^v \, \partial d_t^v/\partial n = 0 \tag{93C}$$

$$d_s^v + q_2^v \, \partial d_s^v/\partial T - s_2^v \, \partial d_s^v/\partial n = 0 \tag{94C}$$

where from equations (46), (47), (78) and (79) it follows that

$$q_2^v = hD_t^v\beta_E^v + fD_s^v\beta_P^v = h(1 + d_t^v)\beta_E^v + f(3 + d_s^v)\beta_P^v \tag{95A}$$

$$s_2^v = gD_t^v\beta_E^v + eD_s^v\beta_P^v = g(1 + d_t^v)\beta_E^v + e(3 + d_s^v)\beta_P^v \tag{95B}$$

The zero-temperature forms of equations (70) through (73) are

$$E^o - D_s^o\beta_P^o = E_o^a \tag{96}$$

$$t^o + D_s^o\beta_P^o \, \partial t^o/\partial P^o = t_o^a \tag{97}$$

$$D_t^o - D_s^o\beta_P^o \, \partial D_t^o/\partial P^o = D_{to}^a \tag{98}$$

$$D_s^o - D_s^o\beta_P^o \, \partial D_s^o/\partial P^o = D_{so}^a \tag{99}$$

where sometimes it is convenient to use the following notation

$$D_t^o = 1 + d_t^o \qquad D_s^o = 3 + d_s^o \tag{100}$$

$$D_{to}^a = 1 + d_{to}^a \qquad D_{so}^a = 3 + d_{so}^a \tag{101}$$

Equations (96) through (99) are four equations for determining the renormalized values E^o , t^o , D_t^o and D_s^o for zero-temperature solids and quantum liquids. As described in Reference 29 an additional equation is required to determine γ_o that occurs in equation (96). From equations (96) through (99) it follows that E^o , t^o , D_t^o and D_s^o have the same variation with density as is shown in equations (85) through (87) and in Figures 1, 2, 4 and 5 respectively. The solution of equations (96) through (99) are generally difficult because of their coupled nature. An approximate solution for equations (98) and (99) can be found by first taking $D_s^o = 3$ in equation (96) and solving for E^o and P^o (and γ^o , with an additional differential equation coming from a power series expansion in the temperature as in Reference 29). Then the solutions to equations (98) and (99) can be written approximately as[31]

$$D_t^o \sim E^o e^{-\phi(n)} - 1/3\ e^{-\phi(n)} \int^n G_{Dt}^{o'}/H_o'\ dn'/n' \tag{102A}$$

$$D_s \sim F^o e^{-\phi(n)} - 1/3\ e^{-\phi(n)} \int^n G_{Ds}^{o'}/H_o'\ dn'/n' \tag{102B}$$

where

$$G_{Dt}^{o'} = e^{\phi(n')} D_{to}^a(n') \qquad G_{Ds}^{o'} = e^{\phi(n')} D_{so}^a(n') \tag{103}$$

where $\phi(n)$ and H_o' are given in equations (52) and (51B) respectively. The zero-temperature vacuum equations are obtained from equations (96) through(99) to be

$$E^{ov} - D_s^{ov}\beta_P^{ov} = 0 \tag{104}$$

$$t^{ov} + D_s^{ov}\beta_P^{ov}\ \partial t^{ov}/\partial P^{ov} = t_o^{va} \tag{105}$$

$$D_t^{ov} - D_s^{ov}\beta_P^{ov}\ \partial D_t^{ov}/\partial P^{ov} = 1 \tag{106}$$

$$D_s^{ov} - D_s^{ov}\beta_P^{ov}\ \partial D_s^{ov}/\partial P^{ov} = 3 \tag{107}$$

where t_o^{va} = constant = characteristic time for a process to occur in the vacuum at zero temperature. Equations (104) through (107) can be made into homogeneous equations by writing $t^{ov'} = t^{ov} - t_o^{va}$ and

$$D_t^{ov} = 1 + d_t^{ov} \qquad D_{to}^{va} = 1 \qquad d_{to}^{va} = 0 \tag{108}$$

$$D_s^{ov} = 3 + d_s^{ov} \qquad D_{so}^{va} = 3 \qquad d_{so}^{va} = 0 \tag{109}$$

and therefore from equations (106) through (109)

$$d_t^{ov} \sim E^o e^{-\phi(n)} \qquad d_s^{ov} \sim F^o e^{-\phi(n)} \tag{110}$$

Note that the renormalized zero-temperature vacuum does not have the 3 + 1

geometry. Equations (104) through (107) are four equations for E^{ov} , t^{ov} , D_t^{ov} and D_s^{ov} for the renormalized zero-temperature vacuum.

4. SCALE AND GAUGE INVARIANCE AND NONEQUILIBRIUM THERMODYNAMICS. This section develops simple expressions for the time evolution of an interacting thermodynamic system by considering the scale and gauge invariance of the left hand sides of the time equations (21) and (33). Suppose that for fixed energy density and pressure a scale and gauge transformation of time is made of the form

$$t' = te^{-\phi} = t(1 - \phi + \cdots) \tag{111}$$

$$\bar{t}' = \bar{t}e^{-j\phi} = \bar{t}(1 - j\phi + \cdots) \tag{112}$$

in the left hand sides of the time equations (21) and (33) respectively by changing some physical parameters such as interparticle interaction parameters. Using equation (111) and the assumed scale invariance of the left hand side of equation (21) gives

$$(1 - \beta_E' \partial/\partial E + 3\beta_P' \partial/\partial P)te^{-\phi} = e^{-\phi}(1 - \beta_E \partial/\partial E + 3\beta_P \partial/\partial P)t \tag{113}$$

Some elementary calculations show that equation (113) is equivalent to

$$\beta_E'(-\partial t/\partial E + t\partial\phi/\partial E) + 3\beta_P'(\partial t/\partial P - t\partial\phi/\partial P) = -\beta_E \partial t/\partial E + 3\beta_P \partial t/\partial P \tag{114}$$

Assuming separability, equation (114) can be written as

$$\beta_E' = \beta_E/[1 - (t\partial\phi/\partial E)/(\partial t/\partial E)] \sim \beta_E[1 + (t\partial\phi/\partial E)/(\partial t/\partial E)] \tag{115}$$

$$\beta_P' = \beta_P/[1 - (t\partial\phi/\partial P)/(\partial t/\partial P)] \sim \beta_P[1 + (t\partial\phi/\partial P)/(\partial t/\partial P)] \tag{116}$$

which relate the transformed gauge parameters to their original values. The approximations in equations (115) and (116) are valid for short time scales.

A power series expansion of β_E' and β_P' gives

$$\beta_E' = \beta_E + d\beta_E/dt(t' - t) + \cdots \tag{117}$$

$$\beta_P' = \beta_P + d\beta_P/dt(t' - t) + \cdots \tag{118}$$

Using equation (111) in equations (117) and (118) gives

$$\beta_E' = \beta_E - (d\beta_E/dt)\phi t + \cdots \tag{119}$$

$$\beta_P' = \beta_P - (d\beta_P/dt)\phi t + \cdots \tag{120}$$

Combining equations (119) and (120) with equations (115) and (116) respectively gives the following time evolution equations for the gauge parameters

$$(d\beta_E/dt)_- = -[\beta_E/\phi\ (\partial\phi/\partial E)/(\partial t/\partial E)]/[1 - t(\partial\phi/\partial E)/(\partial t/\partial E)] \tag{121}$$

$$(d\beta_P/dt)_- = -[\beta_P/\phi\ (\partial\phi/\partial P)/(\partial t/\partial P)]/[1 - t(\partial\phi/\partial P)/(\partial t/\partial P)] \tag{122}$$

where to obtain equations (121) and (122) a negative exponent sign was chosen in equation (111). If a positive sign had been chosen in equation (111) the time evolution equations would be

$$(d\beta_E/dt)_+ = -[\beta_E/\phi\ (\partial\phi/\partial E)/(\partial t/\partial E)]/[1 + t(\partial\phi/\partial F)/(\partial t/\partial E)] \tag{123}$$

$$(d\beta_P/dt)_+ = -[\beta_P/\phi\ (\partial\phi/\partial P)/(\partial t/\partial P)]/[1 + t(\partial\phi/\partial P)/(\partial t/\partial P)] \tag{124}$$

where the gauge parameters β_E and β_P are given in equations (2) and (3) respectively.

Because the time evolution equations must be independent of the choice of the sign of the exponent in equation (111) the rate of change of the gauge parameters must be written as

$$d\beta_E/dt = 1/2[(d\beta_E/dt)_- + (d\beta_E/dt)_+] \tag{125}$$

$$d\beta_P/dt = 1/2[(d\beta_P/dt)_- + (d\beta_P/dt)_+] \tag{126}$$

Combining equations (121) through (126) gives

$$d\beta_E/dt = -(\beta_E\psi_E/\phi)/(1 - t^2\psi_E^2) \tag{127}$$

$$d\beta_P/dt = -(\beta_P\psi_P/\phi)/(1 - t^2\psi_P^2) \tag{128}$$

where

$$\psi_E = (\partial\phi/\partial E)/(\partial t/\partial E) \tag{129}$$

$$\psi_P = (\partial\phi/\partial P)/(\partial t/\partial P) \tag{130}$$

Because ϕ appears symmetrically in equations (127) and (128) a positive or negative choice for the sign of ϕ does not alter the result. A similar analysis using the gauge transformation given in equation (112) gives the result

$$d\beta_E/d\bar{t} = -(\beta_E\bar{\psi}_E/\phi)/(1 + \bar{t}^2\bar{\psi}_E^2) \tag{131}$$

$$d\beta_P/d\bar{t} = -(\beta_P\bar{\psi}_P/\phi)/(1 + \bar{t}^2\bar{\psi}_P^2) \tag{132}$$

where

$$\bar{\psi}_E = (\partial\phi/\partial E)/(\partial\bar{t}/\partial E) \tag{133}$$

$$\bar{\psi}_P = (\partial\phi/\partial P)/(\partial\bar{t}/\partial P) \tag{134}$$

Again, because ϕ appears symmetrically in equations (131) and (132) it follows that these equations are independent of the sign in the exponent of equation (112). For coherent time variation with $d\bar{t} = j\bar{t}d\theta_t$ it follows from equations (131) and (132) that

$$1/t\ d\beta_E/d\theta_t = -(\beta_E\psi'_E/\phi)/(1 - \psi'^2_E) \tag{135}$$

$$1/t\ d\beta_P/d\theta_t = -(\beta_P\psi'_P/\phi)/(1 - \psi'^2_P) \tag{136}$$

where

$$\psi'_E = (\partial\phi/\partial E)/(\partial\theta_t/\partial E) \tag{137}$$

$$\psi'_P = (\partial\phi/\partial P)/(\partial\theta_t/\partial P) \tag{138}$$

A similar analysis can be done for equations (45A) and (45B) to obtain expressions for dq_1/dt and ds_1/dt , however a simpler way is to combine equations (46) and (47) with equations (127) and (128)

Equations (127) and (128) become more simple for short time scales

$$d\beta_E/dt = -\beta_E\psi_E/\phi \tag{139}$$

$$d\beta_P/dt = -\beta_P\psi_P/\phi \tag{140}$$

By writing ϕ as a product $\phi = f(P)g(E)$, it is easy to show that the solution to equations (139) and (140) is

$$\phi = c(\beta_E\beta_P)^{-1} = c[T/V\ C_{PV}(P - K_U)]^{-1} \tag{141}$$

where c = constant. The value of c can be chosen by evaluating equation (141) at a specific thermodynamic state with $\phi = \phi_1$, $T = T_1$ and $n = n_1$. Equations (127) and (128) describe the time evolution of an interacting thermodynamic system in terms of the gauge parameters β_P and β_E and in terms of a potential function $\phi(E,P)$. Thus the gauge parameters β_P and β_E are the basic physical quantities of nonequilibrium thermodynamics. Equation (128) has a proper T = 0 limit

$$d\beta^o_P/dt_o = -(\beta^o_P\psi^o_P/\phi^o)/(1 - t^2_o\psi^{o2}_P) \tag{142}$$

where β_P^o is given by equation (13) and ψ_P^o is obtained from equation (130) to be

$$\psi_P^o = (\partial\phi^o/\partial P^o)/(\partial t_o/\partial P^o) \tag{143}$$

The basic time evolution equations for interacting thermodynamic systems can be obtained from the scale and gauge invariance of the fundamental equations of time (21) and (33) respectively.

5. THERMODYNAMIC PARAMETERS FOR SOLIDS, QUANTUM LIQUIDS AND REAL GASES. This section evaluates the thermodynamic parameters e , f , g , h , β_E and β_P that appear in the equations of time (45A) and (45B) and the reaction rate equations (64) and (66).

A. Solids and Quantum Liquids.

The state equation of a solid or Fermi liquid in which a nuclear, atomic or molecular reaction is occurring is assumed to have the following simple closed form[29]

$$E = E_o + E_j T^j \tag{144}$$

$$P = P_o + P_j T^j \tag{145}$$

where E and P = renormalized energy density and pressure respectively, E_o and P_o = corresponding zero-temperature values of the energy density and pressure, and E_j and P_j = thermal coefficients for energy density and pressure respectively, and where[29]

j = 1	high temperature solid
j = 2	low temperature Fermi gas
j = 5/2	low temperature molecular Bose gas
j = 4	low temperature solid

Combining equations (44), (144) and (145) gives

$$D_e = jT^{j-1}(\ell_j + m_j T^j) \tag{146}$$

$$1/D_e = (j\ell_j T^{j-1})^{-1}(1 - m_j/\ell_j T^j + \cdots) \tag{147}$$

where

$$\ell_j = E_j\, \partial P_o/\partial n - P_j\, \partial E_o/\partial n \tag{148}$$

$$m_j = E_j\, \partial P_j/\partial n - P_j\, \partial E_j/\partial n \tag{149}$$

Then equations (42) and (43) are used to calculate the coefficients e , f , g , and h as follows

$$e = E_j/(\ell_j + m_j T^j) \tag{150}$$

$$f = (\partial E_o/\partial n + \partial E_j/\partial n\ T^j)/D_e \tag{151}$$

$$g = P_j/(\ell_j + m_j T^j) \tag{152}$$

$$h = (\partial P_o/\partial n + \partial P_j/\partial n\ T^j)/D_e \tag{153}$$

For small T equations (150) through (153) can be expanded in a power series in T . These equations determine e , f , g and h for state equations of the form given in equations (144) and (145). The gauge functions are required for the solution of the time and rate equations (45) and (64) or (66) respectively. The functions β_E and β_P are defined in equations (2) and (3) which combined with the state equations (144) and (145) give[29,30]

$$\beta_E = j[1 + \gamma_o P_o/(P_o - K_o)]E_j T^j \tag{154}$$

$$\beta_P = (1 + \gamma_o)P_o - K_o + E_j\ V d\gamma_o/dV\ T^j \tag{155}$$

For the conventional state equation of the form

$$E^a = E^a_o + E^a_j T^j \tag{156}$$

$$P^a = P^a_o + P^a_j T^j \tag{157}$$

it follows that

$$D^a_e = jT^{j-1}(\ell^a_j + m^a_j T^j) \tag{158}$$

$$\ell^a_j = E^a_j\ \partial P^a_o/\partial n - P^a_j\ \partial E^a_o/\partial n \tag{159}$$

$$m^a_j = E^a_j\ \partial P^a_j/\partial n - P^a_j\ \partial E^a_j/\partial n \tag{160}$$

$$g^a = P^a_j/(\ell^a_j + m^a_j T^j) \tag{161}$$

$$h^a = (\partial P^a_o/\partial n + \partial P^a_j/\partial n\ T^j)/D^a_e \tag{162}$$

$$\beta^a_E = j[1 + \gamma^a_o P^a_o/(P^a_o - K^a_o)]E^a_j T^j \tag{163}$$

which are the quantities required to evaluate the right hand sides of the time equation (45) and the rate equation (64). Before equations (45) and (64) can be solved, the energy trace equation (1) must be solved to determine the renormalized functions E_o , P_o , E_j , P_j , β_E and β_P in terms of the correspond-

ing unrenormalized quantities $\mathcal{E}_o^a$, P_o^a , $\mathcal{E}_j^a$, P_j^a , $\beta_{\mathcal{E}}^a$ and β_P^a . Finally, the unrenormalized reaction rate R^a that appears in equations (64) and (66) for processes that occur within solids or quantum liquids may be of the form

$$R^a = R_o^a + R_T T^\sigma \tag{164}$$

where σ = constant. Equation (164) allows processes to occur at T = 0 due to pressure.

B. Real Gases.

For real gases the renormalized state equations are given by a solution of the relativistic energy trace equation (1) as[29]

$$P = nR_G T(1 + Bn + Cn^2 + Dn^3 + \cdots) \tag{165}$$

$$\mathcal{E} = nR_G T(3/2 - TdB/dT\ n - 1/2TdC/dT\ n^2 - 1/3TdD/dT\ n^3 - \cdots) \tag{166}$$

where R_G = gas constant. The corresponding unrenormalized state equations are written as[29]

$$P^a = nR_G T(1 + B^a n + C^a n^2 + D^a n^3 + \cdots) \tag{167}$$

$$\mathcal{E}^a = nR_G T(3/2 - TdB^a/dT\ n - 1/2TdC^a/dT\ n^2 - 1/3TdD^a/dT\ n^3 - \cdots) \tag{168}$$

where Reference 29 gives the connection between the renormalized and unrenormalized virial coefficients by solving equation (1) out to third order. Combining equations (44), (165) and (166) gives after some algebra

$$D_e = n^2 R_G^2 T(\alpha_o + \alpha_1 n + \alpha_2 n^2 + \cdots) \tag{169}$$

where

$$\alpha_o = 3/2B - T^2 d^2B/dT^2 - 3/2TdB/dT \tag{170}$$

$$\alpha_1 = 3C - 1/2d/dT(T^2 dC/dT) - 2Bd/dT(T^2 dB/dT) + 2TdB/dT\ d/dT(TB) \tag{171}$$

$$\alpha_2 = 9/2D - 5/6TdD/dT - 1/3T^2 d^2D/dT^2 - 4CTdB/dT - 3CT^2 d^2B/dT^2 \tag{172}$$

$$- 1/2BTdC/dT + 7/2T^2 dB/dT\ dC/dT - BT^2 d^2C/dT^2$$

Note that D_e given by equation (169) begins with a second order term n^2 whereas the pressure and energy density begin with linear terms in n as shown in equations (165) and (166).

The values of e , f , g and h for the real gases are then calculated using equations (42) through (44) as follows

$$e = nR_G/D_e[3/2 - d/dT(T^2dB/dT)n - 1/2d/dT(T^2dC/dT)n^2 - \cdots] \quad (173)$$

$$= 1/(nR_GT\alpha_o)(3/2 - e_1n + e_2n^2 - \cdots)$$

where

$$e_1 = 3/2\alpha_1/\alpha_o + d/dT(T^2dB/dT) \quad (174)$$

$$e_2 = 3/2[(\alpha_1/\alpha_o)^2 - \alpha_2/\alpha_o] + \alpha_1/\alpha_o\ d/dT(T^2dB/dT) - 1/2d/dT(T^2dC/dT) \quad (175)$$

$$f = R_GT/D_e(3/2 - 2TdB/dT\ n - 3/2TdC/dT\ n^2 - \cdots) \quad (176)$$

$$= 1/(n^2R_G\alpha_o)(3/2 - f_1n + f_2n^2 - \cdots) \quad (177)$$

where

$$f_1 = 3/2\alpha_1/\alpha_o + 2TdB/dT \quad (178)$$

$$f_2 = 3/2[(\alpha_1/\alpha_o)^2 - \alpha_2/\alpha_o] + 2\alpha_1/\alpha_o\ TdB/dT - 3/2TdC/dT \quad (179)$$

$$g = nR_G/D_e[1 + d/dT(TB)n + d/dT(TC)n^2 + \cdots) \quad (180)$$

$$= 1/(nR_GT\alpha_o)(1 + g_1n + g_2n^2 + \cdots)$$

where

$$g_1 = d/dT(TB) - \alpha_1/\alpha_o \quad (181)$$

$$g_2 = (\alpha_1/\alpha_o)^2 - \alpha_2/\alpha_o + d/dT(TC) - \alpha_1/\alpha_o\ d/dT(TB) \quad (182)$$

$$h = R_GT/D_e(1 + 2Bn + 3Cn^2 + 4Dn^3 + \cdots) \quad (183)$$

$$= 1/(n^2R_G\alpha_o)(1 + h_1n + h_2n^2 + \cdots)$$

where

$$h_1 = 2B - \alpha_1/\alpha_o \quad (184)$$

$$h_2 = 3C - 2B\alpha_1/\alpha_o + (\alpha_1/\alpha_o)^2 - \alpha_2/\alpha_o \quad (185)$$

The corresponding unrenormalized coefficients g^a and h^a that appear in the right hand sides of equations (45) and (64) are obtained in a similar way with the substitutions $B \to B^a$, $C \to C^a$ and $D \to D^a$ in equations (180) through (185) and in equations (170) through (172).

The values of β_E and β_p that appear in the time equation (45) and the rate equations (64) and (66) have already been evaluated for the real gases and are given by[29]

$$\beta_p = [2/3(\beta - Td\beta/dT) - \beta]n^2 + \cdots \tag{186}$$

$$= - R_G T(B + 2/3TdB/dT)n^2 + \cdots$$

$$\beta_E = nT(Y - nZ) \tag{187}$$

where

$$Y = R_G[3/2 - 1/\beta(\beta - Td\beta/dT)] \tag{188}$$

$$= R_G(3/2 + T/B\ dB/dT)$$

$$Z = Td^2\beta/dT^2 - 2R_G\Gamma/\beta^2(\beta - Td\beta/dT) + 1/\beta\ d\beta/dT(\beta - Td\beta/dT) + R_G/\beta(\Gamma - Td\Gamma/dT) \tag{189}$$

$$= R_G[2TdB/dT + T^2d^2B/dT^2 + 2C/B^2TdB/dT - (1 + T/B\ dB/dT)TdB/dT - T/B\ dC/dT]$$

and where the two forms of the second and third virial coefficients are related by[29]

$$\beta = R_G TB \qquad \Gamma = R_G TC \tag{190}$$

The values of β_p^a and β_E^a are obtained by analogous formulas with the superscript "a" added. The connection between B^a , C^a and B , C is given in Reference 29. The reaction rate of processes that occur in real gases can be written in the form

$$R^a = g^a(n)T^\kappa \tag{191}$$

where κ = constant. Thus all of the quantities necessary to solve the reaction rate equations (64) or (66) have been determined for processes occurring in the real gases.

6. INCOHERENT TIME IN A RADIATION FIELD. This section considers the effects of radiation (mechanical or electromagnetic) on the local time in matter or energy. The effects can be calculated by considering the presence of radiation as producing a perturbation in the time calculated by equation (45) as follows

$$t + t_r - (q_l + q_{lr})\partial/\partial T(t + t_r) + (s_l + s_{lr})\partial/\partial n(t + t_r) \tag{192}$$

$$= t^a + t_r^a - (q^a + q_r^a)\partial/\partial T(t^a + t_r^a) + (s^a + s_r^a)\partial/\partial n(t^a + t_r^a)$$

Subtracting equation (45) from equation (192) and retaining only first order terms gives the renormalization group equation for time in a radiation field in matter or energy as

$$t_r - q_l\ \partial t_r/\partial T + s_l\ \partial t_r/\partial n - q_{lr}\ \partial t/\partial T + s_{lr}\ \partial t/\partial n = t_r^a + H_r^a \tag{193}$$

where

$$q_{lr} = h_r\beta_E + h\beta_{Er} + 3f_r\beta_P + 3f\beta_{Pr} \tag{194}$$

$$s_{lr} = g_r\beta_E + g\beta_{Er} + 3e_r\beta_P + 3e\beta_{Pr} \tag{195}$$

$$H_r^a = -\beta_E^a h^a\ \partial t_r^a/\partial T + \beta_E^a g^a\ \partial t_r^a/\partial n - q_r^a\ \partial t^a/\partial T + s_r^a\ \partial t^a/\partial n \tag{196}$$

and where

$$h_r = (\partial P_r/\partial n - hZ_r)/D_e \tag{197}$$

$$f_r = (\partial E_r/\partial n - fZ_r)/D_e \tag{198}$$

$$g_r = (\partial P_r/\partial T - gZ_r)/D_e \tag{199}$$

$$e_r = (\partial E_r/\partial T - eZ_r)/D_e \tag{200}$$

$$Z_r = \partial P_r/\partial n\ \partial E/\partial T + \partial E_r/\partial T\ \partial P/\partial n - \partial P_r/\partial T\ \partial E/\partial n - \partial E_r/\partial n\ \partial P/\partial T \tag{201}$$

$$q_r^a = h_r^a\beta_E^a + h^a\beta_{Er}^a \tag{202}$$

$$s_r^a = g_r^a\beta_E^a + g^a\beta_{Er}^a \tag{203}$$

where D_e is given by equation (44) and where

$$\beta_{Er} = T/V\{[d/dT(U + U_r)]_{(P+P_r)V} - [dU/dT]_{PV}\} \tag{204}$$

$$\beta_{Pr} = d/dV[(P + P_r)V]_{U+U_r} - d/dV(PV)_U \tag{205}$$

The value of β_{Er}^a is obtained from equation (204) with the superscript "a" added to U and U_r. Finally, combining equations (194), (195) and (197) through (203) gives

$$q_{lr} = q_l(f\partial P_r/\partial T - h\partial E_r/\partial T) - s_l(f\partial P_r/\partial n - h\partial E_r/\partial n) + h\beta_{Er} + 3f\beta_{Pr} \quad (206)$$

$$s_{lr} = q_l(e\partial P_r/\partial T - g\partial E_r/\partial T) - s_l(e\partial P_r/\partial n - g\partial E_r/\partial n) + g\beta_{Er} + 3e\beta_{Pr} \quad (207)$$

All the necessary quantities for solving the radiation time equation (193) have now been determined.

7. COHERENT TIME AND HIGH TEMPERATURE SUPERCONDUCTIVITY. This section develops a relativistic equation for coherent time and considers applications to the theory of high temperature superconductivity. First the complex number time equation is written in its general form where both the magnitude and the internal phase angle of time vary with temperature and pressure. Secondly the case of slow processes is considered where time varies linearly and the internal phase angle can be set to zero and the scalar incoherent time equation is regained. Thirdly the case of ultrafast time changes is considered in which time coherently rotates in internal space. Finally, applications to high temperature superconductivity are considered.[32-38]

A. General Form of the Complex Number Time Equation.

The complex number time equations (33) and (45A) can be written as

$$\mathcal{L}\bar{t} = \bar{t}(1 + L_{inc} + jL_{coh}) = t^a - \beta_E^a \, \partial t^a/\partial E^a \quad (208)$$

where $\mathcal{L}$ = time operator given by equation (23), and where L_{inc} and L_{coh} = incoherent and coherent time components respectively which are given by

$$L_{inc} = 1/t(-\beta_E \, \partial t/\partial E + 3\beta_p \, \partial t/\partial P) \quad (209)$$

$$= 1/t(-q_l \, \partial t/\partial T + s_l \, \partial t/\partial n) \quad (210)$$

$$L_{coh} = -\beta_E \, \partial\theta_t/\partial E + 3\beta_P \, \partial\theta_t/\partial P \quad (211)$$

$$= -q_l \, \partial\theta_t/\partial T + s_l \, \partial\theta_t/\partial n \quad (212)$$

where in order to obtain the coherent components in equations (211) and (212) the following coherent time rotation is used

$$d\bar{t} = j\bar{t}d\theta_t = jte^{j\theta_t} d\theta_t \quad (213)$$

The real and imaginary components of equation (208) are obtained using equations (209) and (211) to be

$$t \cos\theta_t - \beta_E \sec\beta_{tE} \cos(\theta_t + \beta_{tE}) \, \partial t/\partial E \quad (214)$$

$$+ 3\beta_p \sec\beta_{tP} \cos(\theta_t + \beta_{tP}) \, \partial t/\partial P = t^a - \beta_E^a \, \partial t^a/\partial E^a$$

$$t \sin \theta_t - \beta_E \sec \beta_{tE} \sin(\theta_t + \beta_{tE})\ \partial t/\partial E \tag{215}$$

$$+ 3\beta_P \sec \beta_{tP} \sin(\theta_t + \beta_{tP})\ \partial t/\partial P = 0$$

where

$$\tan \beta_{tE} = (t\partial\theta_t/\partial E)/(\partial t/\partial E) \tag{216}$$

$$\tan \beta_{tP} = (t\partial\theta_t/\partial P)/(\partial t/\partial P) \tag{217}$$

Similarly, the real and imaginary parts of equation (208) can also be written using equations (210) and (212) as follows

$$t \cos \theta_t - q_l \sec \beta_{tT} \cos(\theta_t + \beta_{tT})\ \partial t/\partial T \tag{218}$$

$$+ s_l \sec \beta_{tn} \cos(\theta_t + \beta_{tn})\ \partial t/\partial n = t^a - q^a\ \partial t^a/\partial T + s^a\ \partial t^a/\partial n$$

$$t \sin \theta_t - q_l \sec \beta_{tT} \sin(\theta_t + \beta_{tT})\ \partial t/\partial T \tag{219}$$

$$+ s_l \sec \beta_{tn} \sin(\theta_t + \beta_{tn})\ \partial t/\partial n = 0$$

where

$$\tan \beta_{tT} = (t\partial\theta_t/\partial T)/(\partial t/\partial T) \tag{220}$$

$$\tan \beta_{tn} = (t\partial\theta_t/\partial n)/(\partial t/\partial n) \tag{221}$$

These are the equations that describe partially coherent time.

B. Incoherent Time Equation.

If the processes occur relatively slowly then the change in linear time is much greater than the rotated component $dt >> td\theta_t$ or $L_{inc} >> L_{coh}$ and equation (208) becomes

$$\mathcal{L}\bar{t} \sim (1 + L_{inc})\bar{t} = t^a - \beta_E^a\ \partial t^a/\partial E^a \tag{222}$$

which gives $\theta_t = 0$ and yields the scalar equation

$$t - \beta_E\ \partial t/\partial E + 3\beta_P\ \partial t/\partial P = t^a - \beta_E^a\ \partial t^a/\partial E^a \tag{223}$$

which is the equation for incoherent time presented in the previous sections.

C. Coherent Time Equation.

For an ultrafast process, time changes mainly coherently by rotation in internal space so that the rotational time change is much greater than the linear time change $td\theta_t >> dt$ and $L_{coh} >> L_{inc}$ in equation (208) which becomes

$$\mathcal{L}\bar{t} \sim \bar{t}(1 + jL_{coh}) = t^a - \beta_E^a \, \partial t^a/\partial E^a \tag{224}$$

which can be rewritten as either of the following two equations

$$\bar{t}(1 - j\beta_E \, \partial\theta_t/\partial E + 3j\beta_P \, \partial\theta_t/\partial P) = t^a - \beta_E^a \, \partial t^a/\partial E^a \tag{225}$$

$$\bar{t}(1 - jq_1 \, \partial\theta_t/\partial T + js_1 \, \partial\theta_t/\partial n) = t^a - q^a \, \partial t^a/\partial T + s^a \, \partial t^a/\partial n \tag{226}$$

Combining equation (20) with equations (225) and (226) and taking the real and imaginary parts of these two equations yields the two equations for the coherent time state

$$\tan\theta_t = \beta_E \, \partial\theta_t/\partial E - 3\beta_P \, \partial\theta_t/\partial P \tag{227}$$

$$= q_1 \, \partial\theta_t/\partial T - s_1 \, \partial\theta_t/\partial n$$

$$t = \cos\theta_t \, (t^a - \beta_E^a \, \partial t^a/\partial E^a) \tag{228}$$

$$= \cos\theta_t \, (t^a - q^a \, \partial t^a/\partial T + s^a \, \partial t^a/\partial n)$$

Equation (227) is a nonlinear differential equation that determines the internal phase angle of time θ_t , and equation (228) determines the renormalized incoherent linear time in terms of the unrenormalized time t^a . The measured incoherent linear time is given by

$$t_m = t\cos\theta_t = \cos^2\theta_t \, (t^a - \beta_E^a \, \partial t^a/\partial E^a) \tag{229}$$

Coherent time becomes important when processes occur so fast on a linear time scale that $dt << td\theta_t$ and the rotated time describes the time in a coherent state of matter.

D. High Temperature Superconductivity.

In the accompanying paper on electromagnetism it is shown that the broken symmetry form of Ohm's law is

$$R_{\alpha m} = R_{\alpha c}\{1 - \tan\theta_t \tan[2(\theta_\alpha - \theta_t)]\}\cos^2\theta_t \tag{230}$$

for $\alpha = x, y, z$ and where $R_{\alpha m}$ = measured resistance in the α direction, and

$R_{\alpha c}$ = conventionally calculated resistance in the α direction. For the high temperature superconducting state $R_{\alpha m} = 0$ which gives $\theta_t = \pi/2 - 2(\theta_\alpha - \theta_t)$ which combined with the free s - pair condition $\theta_\alpha = 2\theta_t$ gives $\theta_t = \pi/6$ and $\theta_\alpha = \pi/3$. Therefore, whereas the standard Bardeen-Cooper-Schrieffer (BCS) theory predicts superconductivity to occur when $R_{\alpha c} = 0$, the high-T_c superconducting state is associated with $R_{\alpha m} = 0$ which occurs when a coherent time state exists within the planar copper oxides with $\theta_t = \pi/6$ when $T < T_c$.

For the coherent time state with $\theta_t = \pi/6$ equation (227) gives

$$1/\sqrt{\quad} = q_1^c(\partial\theta_t/\partial T)_{T_c} - s_1^c(\partial\theta_t/\partial n)_{T_c} \tag{231}$$

which determines the critical temperature T_c in terms of the particle number density of matter, $T_c = T_c(n)$, provided the internal phase angle $\theta_t(T,n)$ is known. Equivalently, equation (231) determines the pressure dependence of the critical transition temperature $T_c = T_c(P)$. For a coherent time superconducting state, equation (228) gives the relativistic linear time as

$$t = \sqrt{3}/2(t^a - \beta_E^a\, \partial t^a/\partial E^a) \tag{232}$$

$$= \sqrt{3}/2(t^a - q^a\, \partial t^a/\partial T + s^a\, \partial t^a/\partial n)$$

The measured linear time for the high-T_c superconducting state is given by

$$t_m = 3/4(t^a - \beta_E^a\, \partial t^a/\partial E^a) \tag{233}$$

In a high-T_c superconductor processes run faster than is predicted by conventional calculations.

Two renormalized time scales must be distinquished in matter: a) linear time or incoherent time t , and b) circular or coherent time $t\theta_t$ for a coherent time state. Therefore in a coherent time state physical processes, such as electron-muthon scattering, occur during the time $t\theta_t$, while in an incoherent time state physical processes, such as the electron-phonon scattering of the conventional BCS theory, occur on the linear time scale t . These two time scales are associated with corresponding energy scales. The characteristic relative energy scale for superconductors is the normalized superconductivity energy gap[32-38]

$$\Delta' = 2\Delta/(kT_c) \tag{234}$$

where Δ = superconductivity energy gap and T_c = transition temperature. For a coherent time state high-T_c superconductor the energy gap is designated by Δ_{ct} , while the superconductivity energy gap for an incoherent time (conventional BCS) superconductor is designated by Δ_{it} . Then using the Heisenberg uncertainty principle with the normalized superconductivity energy gaps gives

$$\Delta'_{ct}\, t\theta_t = \tau \qquad \Delta'_{it}\, t = \tau \qquad \tau = \hbar^3/(m_p e^4) \tag{235}$$

so that

$$\Delta'_{ct} = \Delta'_{it}/\theta_t = 6/\pi\ \Delta'_{it} \sim 1.91\Delta'_{it} \tag{236}$$

because $\theta_t = \pi/6$ for the high-T_c superconducting state. The larger normalized superconductivity energy gap predicted for high temperature superconductors by equation (236) has been experimentally observed.[38]

8. CONCLUSION. A renormalization group equation for time is developed which suggests that the reaction rates of processes occurring in an ambient thermodynamic medium depend on the state equation of the medium. Physical processes run more slowly when they occur in a real medium with $PV \neq \alpha U$ than they would if they occurred in an ideal thermodynamic medium or vacuum. These results follow from a gauge theory of time that is based on the Minkowski space-time metric. The dimensions of space and time also depend on the thermodynamic state equation of the ambient matter. Theoretical predictions of reaction rates and the dimensions of space and time can be made for solids, Fermi liquids, and the real classical gases by solving the coupled renormalization group equations for energy, time and the dimensions of space and time. The renormalization group equation for coherent time is developed and applied to the problem of describing the high-T_c superconducting state. It is concluded that high-T_c superconductors represent a coherent time state of matter which can be described by the renormalization group equation for coherent time. This time state occurs in matter with free pairs of electrons when the internal phase angle of time is given by $\theta_t = \pi/6$. This implies that the normalized superconductivity energy gap for high-T_c planar copper oxides should be $6/\pi$ times the magnitude of the normalized energy gap of conventional BCS superconductors.

ACKNOWLEDGEMENT

The author wishes to thank Elizabeth K. Klein for typing this paper.

REFERENCES

1. Prigogine, I., From Being to Becoming, Freeman, New York, 1980.

2. Davies, P., The Physics of Time Asymmetry, University of California Press, Berkeley, 1977.

3. Davies, P., Space and Time in the Modern Universe, Cambridge University Press, Cambridge, 1977.

4. Gold, T., ed., The Nature of Time, Cornell University Press, Ithaca, 1967.

5. Misner, C. W., Thorne, K. S., and Wheeler, J., Gravitation, Freeman, San Francisco, 1973.

6. Reichenbach, Hans, The Philosophy of Space and Time, Dover, New York, 1957.

7. Gal-Or, B., Modern Developments in Thermodynamics, Wiley, New York, 1974.

8. Landsberg, P. T., The Enigma of Time, Adam Hilger, Bristol, 1982.

9. Whitrow, G. J., The Nature of Time, Holt, Rinehart and Winston, New York, 1972.

10. Weyl, H., Philosophy of Mathematics and Natural Science, Princeton University Press, Princeton, 1949.

11. Morris, R., Time's Arrows, Simon & Schuster, New York, 1984.

12. Whittaker, E., From Euclid to Eddington, Dover, New York, 1958.

13. Barker, P. and Shugart, C. G., After Einstein, Memphis State University Press, Memphis, 1981.

14. Weyl, H., Space-Time-Matter, Dover, New York, 1922.

15. Schrödinger, E., Space-Time Structure, Cambridge University Press, Cambridge, 1950.

16. Schrödinger, E., Expanding Universes, Cambridge University Press, Cambridge, 1956.

17. Weinberg, S., Gravitation and Cosmology, Wiley, New York, 1972.

18. Berry, M., Principles of Cosmology and Gravitation, Cambridge University Press, Cambridge, 1976.

19. Blum, H. F., Time's Arrow and Evolution, Princeton University Press, Princeton, 1955.

20. Winfree, A. T., The Geometry of Biological Time, Springer-Verlag, New York, 1980.

21. Winfree, A. T., When Time Breaks Down, Princeton University Press, Princeton, 1987.

22. Margenau, H., The Nature of Physical Reality, McGraw-Hill, New York, 1950.

23. Schilpp, P. A., Albert Einstein: Philosopher-Scientist, Tudor Publishing Co., New York, 1951.

24. Eddington, A. S., The Nature of the Physical World, MacMillan, New York, 1928.

25. Schlegel, R., Superposition and Interaction, University of Chicago Press, Chicago, 1980.

26. Whitrow, G. J., Time in History, Oxford University Press, New York, 1989.

27. Fraser, J. T., Time the Familar Stranger, University of Massachusetts Press, Amherst, 1987.

28. Alvarez, E., "Quantum Gravity: An Introduction to Some Recent Results," Rev. Mod. Phys., 61, 561, July 1989.

29. Weiss, R. A., Relativistic Thermodynamics, Vols. 1 & 2, Exposition Press, New York, 1976.

30. Weiss, R. A., Gauge Theory of Thermodynamics, K&W Publications, Vicksburg, MS, 1989.

31. Murphy, G. M., Ordinary Differential Equations and Their Solutions, Van Nostrand, New York, 1960.

32. Phillips, J. C., Physics of High-T_c Superconductors, Academic, San Diego, 1989.

33. Bednorz, J. G. and Müller, K. A., "Perovskite Oxides - The New Approach to High-T_c Superconductivity," Revs. Mod. Phys., Vol. 60, 585, July 1988.

34. Pickett, W. E., "Electronic Structure of the High-Temperature Oxide Superconductors," Rev. Mod. Phys., Vol. 61, 433, April 1989.

35. Micnas, R., Ranninger, J. and Robaszkiewicz, S., "Superconductivity in Narrow-Band Systems with Local Nonretarded Attractive Interactions," Rev. Mod. Phys., Vol. 62, 113, January 1990.

36. Little, W. A., "Experimental Constraints on Theories of High-Transition Temperature Superconductors," Science, Vol. 242, 1390, 9 December 1988.

37. Geballe, T. H. and Hulm, J. K., "Superconductivity - The State that Came in from the Cold," Science, Vol. 239, 367, 22 January 1988.

38. Margaritondo, G., Huber, D. L. and Olson, C. G., "Photoemission Spectroscopy of the High-Temperature Superconductivity Gap," Science, Vol. 246, 770, 10 November 1989.

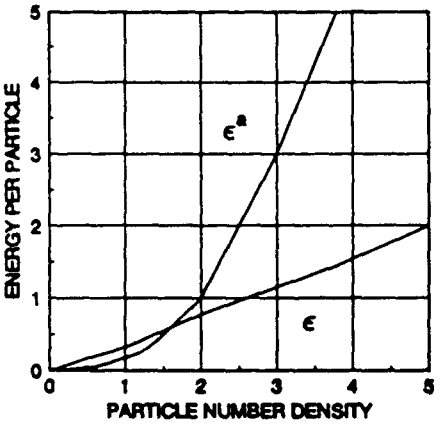


Figure 1

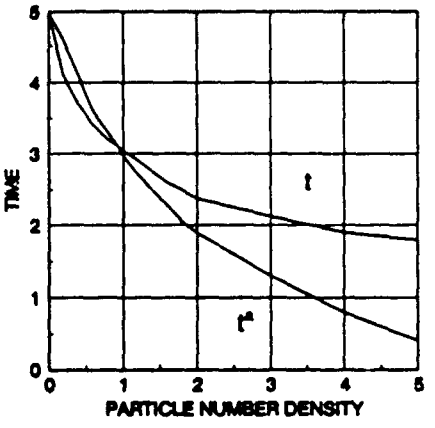


Figure 2

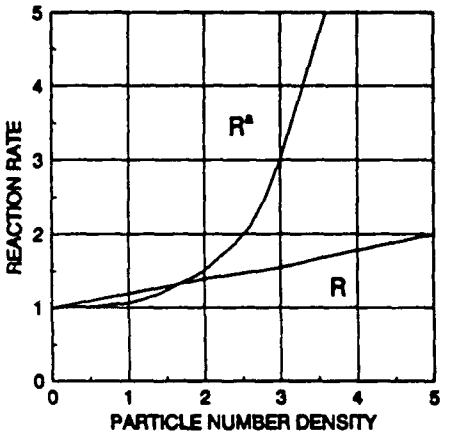


Figure 3

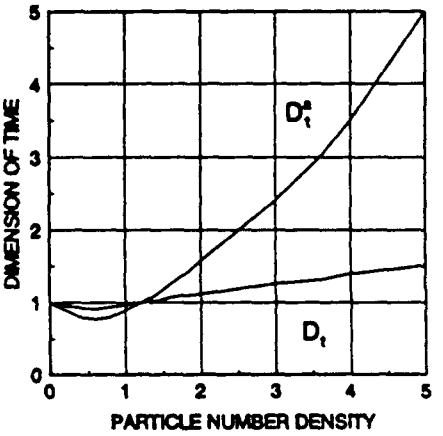


Figure 4

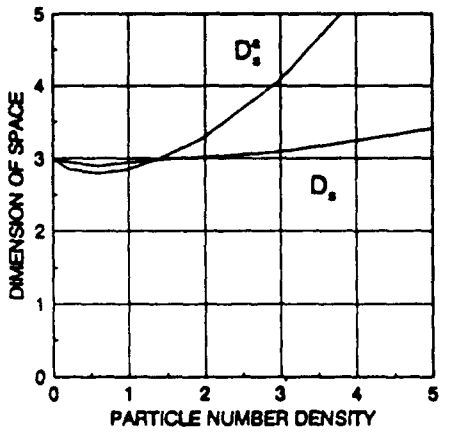


Figure 5

Figure 1. Sketch of renormalized (ϵ) and unrenormalized (ϵ^a) average energy per particle for a neutron gas in terms of particle number density.

Figure 2. Sketch of renormalized (t) and unrenormalized (t^a) time in terms of particle number density.

Figure 3. Sketch of renormalized (R) and unrenormalized (R^a) reaction rate in terms of particle number density.

Figure 4. Sketch of renormalized (D_t) and unrenormalized (D_t^a) dimension of time in terms of particle number density.

Figure 5. Sketch of renormalized (D_s) and unrenormalized (D_s^a) dimension of space in terms of particle number density.

THERMAL RADIATION OF HIGH-T_c SUPERCONDUCTORS

Richard A. Weiss
U. S. Army Engineer Waterways Experiment Station
Vicksburg, Mississippi 39180

ABSTRACT. The theory of a photon gas with broken internal symmetry is developed. The broken symmetry of time induces a broken symmetry in the photon frequency so that Planck's heat radiation law must be written in terms of a complex number spectral energy density. Over long distances the broken symmetry of time can be produced either by gravity or by some special nuclear, atomic or molecular structure that causes a coherent time state to exist in matter. For the case where the gravity of a star or planet produces the broken symmetry of time the internal phase angle of the frequency is essentially constant and the thermal radiation of matter is incoherent and the Stefan-Boltzmann law is valid. But for the case of a high-T_c superconductor with $T < T_c$ the presence of an incoherent time state for the crystal lattice and a coherent time state for the electron (hole) pairs requires the emitted thermal energy to have both an incoherent blackbody radiation component and a coherent non-blackbody component that is calculated by integrating over the internal phase angles of the frequency. This paper calculates the thermal energy spectrum and thermal energy density for high-T_c superconductors and suggests that the measured thermal energy density of high-T_c superconductors may serve as a test for the theory that the planar copper oxides in their superconducting state represent matter in a coherent time state.

1. INTRODUCTION. Broken symmetries play a major role in the understanding of basic phenomena in both particle and bulk matter physics.[1] The broken symmetry vacuum is similar to the ground state of a many-body system.[2] The physical vacuum has non-vanishing fields, and is analogous to the ground state of a ferromagnet. In both cases the Hamiltonians describing these systems are rotationally invariant. This is the case of spontaneously broken symmetry where the ground state does not have the symmetry of the Hamiltonian.[2] The vacuum is thought to have the Higgs field extending throughout all space and having a non-zero value for the ground state. It is similar to the broken symmetry associated with the Ginzburg-Landau order parameter that describes the broken symmetry superconducting state in which the Cooper electron pairs break the symmetry of the ground state and produce a macroscopic coherent state of matter.[2]

The broken symmetry of the vacuum affects the state equations of interacting bulk matter by requiring the state functions such as pressure, internal energy and entropy to be complex numbers having internal phase angles.[3,4] The effects of the broken symmetry vacuum on matter and energy are determined by solving a complex number relativistic trace equation.[4] The broken symmetry of the vacuum state requires that the coordinates of space and time be complex numbers.[4] This requirement affects the calculations of such diverse scientific disciplines as mechanics, electromagnetism and thermodynamics. In par-

ticular, it affects the calculation of the gravitational equilibrium of stars and planets through an intimate connection between the internal phase angle of the pressure and the internal phase angles of the space and time coordinates.[4] The broken symmetry of the time coordinate requires that the frequency of light in matter also has an internal phase angle so that the elementary formulas of atomic physics, such as Planck's radiation law, must have asymmetric forms and exhibit broken internal symmetries.[4] The broken symmetries of the space and time coordinates are induced over macroscopic distances in incoherent matter mainly by gravity, but atomic and molecular structure can also create a coherent state of space and time coordinates of macroscopic dimensions as perhaps in the case of high-T_c superconductors. This paper describes the broken symmetry forms of Planck's law and the Stefan-Boltzmann law for the thermal radiation of matter in a gravitational field and for the thermal radiation from the surface of a high-T_c superconductor.

The effects of the broken symmetry of the Minkowski vacuum on the thermodynamic state functions of matter are calculated from the following complex number relativistic trace equation[4]

$$\bar{U} + T(d\bar{U}/dT)_{\bar{P}V} - 3Vd/dV(\bar{P}V)_{\bar{U}} = U^a + T(dU^a/dT)_{P^aV} \tag{1}$$

where $\bar{U}$ and $\bar{P}$ = complex number renormalized values of the internal energy and pressure respectively, U^a and P^a = unrenormalized values of the internal energy and pressure respectively, T = absolute temperature and V = volume of a fixed number of particles. The renormalized internal energy and pressure are written as

$$\bar{U} = Ue^{j\theta_U} \qquad \bar{P} = Pe^{j\theta_P} \tag{2}$$

A many-body theory is used to determine $U^a(V,T)$ and $P^a(V,T)$ for an interacting system, and then equation (1) is used to determine the renormalized pressure P, θ_P and internal energy U, θ_U. For a noninteracting system $P^aV = c_1U^a$ where c_1 = constant, and it follows from equation (1) that $U = U^a$, $P = P^a$, $\theta_U = 0$ and $\theta_P = 0$. The complex number values of the internal energy and pressure require that the coordinates of space and time are also complex numbers which can be written as[4]

$$\bar{\alpha} = \alpha e^{j\theta_\alpha} \qquad \bar{t} = te^{j\theta_t} \tag{3}$$

for $\alpha = x, y, z$. The internal phase angles of the space and time coordinates θ_α and θ_t are determined from the laws of mechanics such as Euler's equation of fluid motion.[4] From the broken symmetry representation of time in equation (3) it follows that the period and frequency of vibrations can be written as

$$\bar{\tau} = \tau e^{j\theta_\tau} \qquad \bar{\nu} = \nu e^{j\theta_\nu} \tag{4}$$

where because $\bar{\nu} = 1/\bar{\tau}$ it follows that

$$\nu = 1/\tau \qquad \theta_\nu = -\theta_\tau = -\theta_t \tag{5}$$

and therefore the internal phase angle of the frequency of vibration of a system follows directly from the internal phase angle of the time. The measured frequency and period are given by $\nu_m = \nu \cos \theta_\nu$ and $\mathfrak{T}_m = \mathfrak{T} \cos \theta_t$ so that $\nu_m \neq 1/\mathfrak{T}_m$.[4]

Planck's law for the symmetric photon gas gives the thermal spectral energy density as[5-9]

$$E^a_\nu = A/(e^{h\nu/kT} - 1) \qquad A = 8\pi h\nu^3/c^3 \tag{6}$$

where E^a_ν = unrenormalized spectral energy density, h = Planck's constant and k = Boltzmann's constant. From equation (6) it follows that the Stefan-Boltzmann radiation law can be written as[5-9]

$$E^a_r = \int_0^\infty E^a_\nu d\nu = \sigma T^4 \tag{7}$$

where E^a_r = unrenormalized thermal energy density and σ = Stefan-Boltzmann constant given by[5-9]

$$\sigma = (\pi^4/15)(8\pi k^4)/(c^3 h^3) \tag{8}$$

Neglecting the density dependence of the index of refraction gives the radiation pressure as[3-9]

$$P^a_r = 1/3E^a_r = 1/3\sigma T^4 \tag{9}$$

Because equation (9) describes a noninteracting gas of photons, an immediate application of the trace equation (1) gives the renormalized radiation energy density as

$$\bar{E}_r = E^a_r \qquad E_r = E^a_r \qquad \theta_{Er} = 0 \tag{10}$$

as expected for a noninteracting system.[3]

For thermal radiation in matter with broken internal symmetry, due either to gravity or to nuclear, atomic or molecular structure, the procedure for calculating the renormalized radiation density is simple in principle but difficult in practice. The first step is the calculation of the unrenormalized energy of the photon-matter system, in which the radiation internal energy is written as $U^a_r = VE^a_r = VE^a_r(n,T)$ and the matter internal energy as $U^a(n,T)$ where n = particle number density. The energy density of radiation is now a function of n and T . The trace equation (1) is then solved with $U^a(n,T) + U^a_r(n,T)$ used as the source term, with the result that the renormalized internal energies $U(n,T)$, θ_U , $U_r(n,T)$, θ_{Ur} are obtained along with the renormalized values of the Grüneisen parameters $\gamma(n,T)$, θ_γ , γ_r and $\theta_{\gamma r}$ for matter and radiation respectively.[4] The calculation of the unrenormalized radiation internal energy $U^a_r(n,T)$ is very complicated if gravitational effects in matter are included. Even the calculation of the unrenormalized internal energy $U^a_r(T)$ for thermal radiation in empty space with gravitation is very complicated.

An approximate solution to the problem of calculating the energy of thermal radiation in matter with broken internal symmetries has been suggested that utilizes the implication of the trace equation (1) that the renormalized radiation density must be a complex number.[4] The broken symmetry nature of time and frequency as described by equations (3) and (4) respectively suggest the following complex number generalization of Planck's radiation law[4]

$$\bar{E}_\nu = \bar{A}/[e^{h\bar{\nu}/(kT)} - 1] \qquad \bar{A} = 8\pi h\bar{\nu}^3/c^3 \tag{11}$$

where $\bar{E}_\nu$ = complex number value of the renormalized spectral energy density. Placing equation (4) into (11) gives[4]

$$\bar{E}_\nu = E_\nu e^{j\theta E\nu} = A(B + jC)/D \tag{12}$$

$$E_\nu = A/D\ (B^2 + C^2)^{1/2} \tag{13}$$

$$\tan \theta_{E\nu} = C/B \tag{14}$$

where A is given by equation (6) and where[4]

$$D = e^{2x} - 2 \cos y\ e^x + 1 \tag{15}$$

$$B = \cos (3\theta_\nu)\ (\cos y\ e^x - 1) + \sin (3\theta_\nu) \sin y\ e^x \tag{16}$$

$$C = \sin (3\theta_\nu)\ (\cos y\ e^x - 1) - \cos (3\theta_\nu) \sin y\ e^x \tag{17}$$

$$x = h\nu/(kT) \cos \theta_\nu \tag{18}$$

$$y = h\nu/(kT) \sin \theta_\nu \tag{19}$$

When $\theta_\nu = 0$ equation (13) reduces to the standard Planck spectral energy density.

The complex number integrated energy density is given by[4]

$$\bar{E}_r = E_r e^{j\theta Er} = \int \bar{E}_\nu\, d\bar{\nu} = \int_0^\infty E_\nu \sec \beta_{\nu\nu}\ e^{j(\theta E\nu + \theta_\nu + \beta_{\nu\nu})}\, d\nu \tag{20}$$

where

$$\tan \beta_{\nu\nu} = \nu d\theta_\nu/d\nu \tag{21}$$

$$d\bar{\nu} = e^{j\theta}\nu(d\nu + j\nu d\theta_\nu) = \sec \beta_{\nu\nu}\, d\nu\, e^{j(\theta_\nu + \beta_{\nu\nu})} \tag{22}$$

The complex number thermal radiation energy density given by equation (20) has the following real and imaginary parts

$$E_{rR} = E_r \cos \theta_{Er} = \int_o^\infty E_\nu \sec \beta_{\nu\nu} \cos \Phi_\nu \, d\nu \tag{23}$$

$$E_{rI} = E_r \sin \theta_{Er} = \int_o^\infty E_\nu \sec \beta_{\nu\nu} \sin \Phi_\nu \, d\nu \tag{24}$$

where

$$\Phi_\nu = \theta_{E\nu} + \theta_\nu + \beta_{\nu\nu} \tag{25}$$

and where E_ν and $\theta_{E\nu}$ are given by equations (13) and (14) respectively. The magnitude and internal phase angle of the integrated radiation density are given by

$$E_r = (E_{rR}^2 + E_{rI}^2)^{1/2} \tag{26}$$

$$\tan \theta_{Er} = E_{rI}/E_{rR} \tag{27}$$

The measured thermal radiation density is given by the real part of the complex number integrated radiation density so that

$$E_{rm} = E_{rR} \tag{28}$$

and therefore the integral in equation (23) must be evaluated.

Further insight into the meaning of the complex number integral in equation (20) can be obtained by combining equations (20) and (22) to get

$$\bar{E}_r = \bar{E}^r_{inc} + \bar{E}^r_{coh} \tag{29}$$

where

$$\bar{E}^r_{inc} = \int_o^\infty \bar{E}_\nu \, e^{j\theta_\nu} \, d\nu \tag{30}$$

$$\bar{E}^r_{coh} = j\int \bar{E}_\nu \, \bar{\nu} \, d\theta_\nu \tag{31}$$

where $\bar{E}^r_{inc}$ = incoherent radiation energy density and $\bar{E}^r_{coh}$ = coherent radiation energy density. The complex number radiation density given in equation (20) has not been evaluated for arbitrary values of the frequency internal phase angle $\theta_\nu(\nu)$ due to the difficulty of evaluating the integral. The same is true for the measured energy density given in equation (23). Also there is the question of what the function $\theta_\nu(\nu)$ should be for radiation in matter with broken internal symmetries due to gravity or structure. In this paper two special cases of equation (20) are considered. The first case is that of incoherent radiation with $\theta_\nu = \theta_\nu^c$ = constant and the frequency ν being an integration variable, the second case is that of coherent radiation with $\nu = \nu^c$ = constant and the internal phase angle θ_ν is the variable of integration. For

these two cases equation (20) or equivalently equations (30) through (31) can be written as

$$\bar{E}^{r}_{inc} = e^{j\theta^{c}_{\nu}} \int_{o}^{\infty} \bar{E}_{\nu}\, d\nu \qquad \theta_{\nu} = \theta^{c}_{\nu} = \text{const.} \qquad d\bar{\nu} = e^{j\theta^{c}_{\nu}}\, d\nu \tag{32}$$

$$\bar{E}^{r}_{coh} = j\nu_{c} \int \bar{E}_{\nu}\, e^{j\theta_{\nu}}\, d\theta_{\nu} \qquad \nu = \nu_{c} = \text{const.} \qquad d\bar{\nu} = j\bar{\nu} d\theta_{\nu} \tag{33}$$

This paper assumes that the incoherent radiation from ordinary incoherent matter in a gravitational field can be described by equation (32), and that coherent and incoherent time states exist in high-T_c superconductors and the calculation of the radiation energy density for these materials with $T < T_c$ includes contributions from both equations (32) and (33). Equation (32) gives the radiation contribution from the time incoherent lattice of the high-T_c superconductors, and equation (33) gives the radiation contribution from the coherent time states of the electron (hole) pair condensate. Other forms of matter may require $\theta_{\nu}(\nu)$ to have a more general behaviour thus making equations (20) and (23) difficult to evaluate.

The general form of the integral in equation (20) can be rewritten by introducing the following change of variables

$$\bar{\xi} = \xi e^{j\theta_{\xi}} = h\bar{\nu}/(kT) \tag{34}$$

$$\xi = h\nu/(kT) \qquad \theta_{\xi} = \theta_{\nu} = -\theta_{t} \tag{35}$$

The equation (20) can be written as

$$\bar{E}_{r} = (8\pi k^{4}T^{4})/(c^{3}h^{3})\ \bar{I} \tag{36}$$

where

$$\bar{I} = \int \bar{\xi}^{3} (e^{\bar{\xi}} - 1)^{-1}\, d\bar{\xi} \tag{37}$$

The standard technique for evaluating the integral in equation (37) begins by using the following expansion[5-9]

$$(e^{\bar{\xi}} - 1)^{-1} = e^{-\bar{\xi}} + e^{-2\bar{\xi}} + e^{-3\bar{\xi}} + \cdots \tag{38}$$

Then

$$\bar{I} = \sum_{n=1}^{\infty} \bar{I}^{n}$$

where n = integer and

$$\bar{I}^{n} = \int \bar{\xi}^{3}\, e^{-n\bar{\xi}}\, d\bar{\xi} \tag{40}$$

Then the thermal radiation energy density is

$$\bar{\varepsilon}_r = (8\pi k^4 T^4)/(c^3 h^3) \sum_{n=1}^{\infty} \bar{I}^n \tag{41}$$

Thus far no assumption of incoherent or coherent radiation has been made.

The thermal radiation energy density for matter with broken internal symmetries is calculated both for the incoherent and coherent cases. Section 2 determines the incoherent thermal radiation energy density for the case of constant internal phase angle of time as is found for example in a uniform gravitational field, and for the case of coherent thermal radiation which is associated with the coherent time states of the electron (hole) pair fluid in high-T_c superconductors. For coherent time superconductors with $T < T_c$ the thermal radiation density is a weighted average of a blackbody Stefan-Boltzmann component associated with the atomic lattice, and a coherent non-blackbody component which is associated with the electron (hole) Cooper pairs.

2. ASYMMETRIC THERMAL RADIATION. This section calculates the integrated thermal radiation density for: a) the case of incoherent radiation with constant internal phase angle of time which perhaps describes radiation in the presence of a gravity field, and b) the case where electrons and radiation are in a coherent time state in which transitions occur so fast that a change in time occurs as a rotation in time, as perhaps in high-T_c superconductors, rather than as a linear time variation as in ordinary matter and conventional superconductors.

A. Incoherent Thermal Radiation.

The integral in equation (20) can be easily evaluated for the case when $\theta_\nu = -\theta_t$ = constant. This can be done by evaluating the integral in equation (37) for the special case $\theta_\xi = \theta_\xi^c = \theta_\nu^c = -\theta_t^c$ = constant, and $d\bar{\xi} = d\xi\, e^{j\theta_\xi^c}$. In this case the integral in equation (40) can be written as

$$\bar{I}^n = e^{j4\theta_\xi^c} \int_0^\infty \xi^3 e^{-\bar{n}\xi}\, d\xi = e^{j4\theta_\xi^c}\, 6/\bar{n}^4 = 6/n^4 \tag{42}$$

where $\bar{n} = ne^{j\theta_\xi^c}$ and $\bar{I}^n$ is a real number and[5-9]

$$\bar{I}^n = I^n = \int_0^\infty \xi^3 e^{-n\xi}\, d\xi = 6/n^4 \tag{43}$$

Then equation (39) gives the following standard result[5-9]

$$\bar{I} = I = \sum_{n=1}^{\infty} 6/n^4 = \pi^4/15 \tag{44}$$

Therefore for $\theta_\xi = -\theta_t^c$ = constant the integrals in equations (36) and (37) are real numbers and the expression for the incoherent thermal energy density for broken symmetry radiation is

$$\bar{E}_r = E^r_{inc} = \sigma T^4 \qquad \sigma = (8\pi^5 k^4)/(15c^3 h^3) \tag{45}$$

which is the standard Stefan-Boltzmann law. For this case the integral in equation (32) is a real number giving the result in equation (45).

B. Thermal Radiation Spectrum of Coherent Time States.

The accompanying papers on electromagnetism and the gauge theory of time suggests that high-T_c superconductors are coherent time states of matter. In a manner analogous to the Planck concept of radiation being composed of harmonic oscillators, the radiation in a high-T_c superconductor is envisioned as being composed of complex number frequency harmonic oscillators each having internal degrees of freedom. For radiation in high-T_c superconductors, each oscillator of a fixed frequency undergoes internal vibrations. The calculation of the thermal radiation energy density for high-T_c superconductors with $T < T_c$ therefore involves an integration over the internal degrees of freedom (the internal phase angles of the frequency). For these internal vibrations $\nu = \nu_c = \text{constant}$ and from equations (34) and (35) $\bar{\xi} = \xi_c \exp(j\theta_\xi)$ with $\xi_c = \text{constant}$ and

$$\xi_c = h\nu_c/(kT) \qquad d\bar{\xi} = j\bar{\xi}d\theta_\xi \qquad d\theta_\xi = d\theta_\nu = -d\theta_t \tag{46}$$

The integral in equation (40) for these conditions can then be written as

$$\bar{I}^n = j\xi_c^4 \int_o^{-\pi/6} e^{j4\theta_\xi}\, e^{-n\bar{\xi}}\, d\theta_\xi \tag{47}$$

$$= \xi_c^4 \int_o^{-\pi/6} e^{-n\xi_c \cos\theta_\xi}\, (F_1^n + jF_2^n)\, d\theta_\xi$$

$$= \xi_c^4 (I_1^n + jI_2^n)$$

where

$$I_1^n = \int_o^{-\pi/6} F_1^n\, e^{-n\xi_c \cos\theta_\xi}\, d\theta_\xi \tag{48}$$

$$I_2^n = \int_o^{-\pi/6} F_2^n\, e^{-n\xi_c \cos\theta_\xi}\, d\theta_\xi \tag{49}$$

and where

$$F_1^n = -\sin(4\theta_\xi - n\xi_c \sin\theta_\xi) = \sin(4\theta_t - n\xi_c \sin\theta_t) \tag{50}$$

$$= \cos(4\theta_\xi)\sin(n\xi_c \sin\theta_\xi) - \sin(4\theta_\xi)\cos(n\xi_c \sin\theta_\xi)$$

$$F_2^n = \cos(4\theta_\xi - n\xi_c \sin\theta_\xi) = \cos(4\theta_t - n\xi_c \sin\theta_t) \tag{51}$$

$$= \cos(4\theta_\xi)\cos(n\xi_c \sin\theta_\xi) + \sin(4\theta_\xi)\sin(n\xi_c \sin\theta_\xi)$$

The limits of integration in equation (47) will be explained subsequently. The integrals in equations (48) and (49) can be rewritten as

$$I_1^n = G_1^n - G_2^n \tag{52}$$

$$I_2^n = G_3^n + G_4^n \tag{53}$$

where

$$G_1^n(\xi_c) = \int_0^{-\pi/6} e^{-n\xi_c \cos\theta_\xi} \cos(4\theta_\xi)\sin(n\xi_c \sin\theta_\xi)\, d\theta_\xi \tag{54}$$

$$G_2^n(\xi_c) = \int_0^{-\pi/6} e^{-n\xi_c \cos\theta_\xi} \sin(4\theta_\xi)\cos(n\xi_c \sin\theta_\xi)\, d\theta_\xi \tag{55}$$

$$G_3^n(\xi_c) = \int_0^{-\pi/6} e^{-n\xi_c \cos\theta_\xi} \cos(4\theta_\xi)\cos(n\xi_c \sin\theta_\xi)\, d\theta_\xi \tag{56}$$

$$G_4^n(\xi\ ) = \int_0^{-\pi/6} e^{-n\xi_c \cos\theta_\xi} \sin(4\theta_\xi)\sin(n\xi_c \sin\theta_\xi)\, d\theta_\xi \tag{57}$$

Using the phase angle relationship given in equation (35) allows equations (54) through (57) to be written as

$$G_1^n(\xi_c) = \int_0^{\pi/6} e^{-n\xi_c \cos\theta_t} \cos(4\theta_t)\sin(n\xi_c \sin\theta_t)\, d\theta_t \tag{58}$$

$$G_2^n(\xi_c) = \int_0^{\pi/6} e^{-n\xi_c \cos\theta_t} \sin(4\theta_t)\cos(n\xi_c \sin\theta_t)\, d\theta_t \tag{59}$$

$$G_3^n(\xi_c) = -\int_0^{\pi/6} e^{-n\xi_c \cos\theta_t} \cos(4\theta_t)\cos(n\xi_c \sin\theta_t)\, d\theta_t \tag{60}$$

$$G_4^n(\xi_c) = -\int_0^{\pi/6} e^{-n\xi_c \cos\theta_t} \sin(4\theta_t)\sin(n\xi_c \sin\theta_t)\, d\theta_t \tag{61}$$

where ξ_c is given by equation (46).

The limits of integration in equations (54) through (57) need some explanation. In the accompanying paper on electromagnetism and gravity it is shown that the broken symmetry form of Ohm's law gives the measured resistance of a body as

$$R_{\alpha m} = R_{\alpha c}\{1 - \tan\theta_t \tan[2(\theta_\alpha - \theta_t)]\}\cos^2\theta_t \tag{62}$$

where $\alpha = x, y$ and z specifies the orientation of the conductor, and where $R_{\alpha m}$ = measured resistance in the α direction, and $R_{\alpha c} = W_{\alpha m}/I_{\alpha m}$ = conventionally calculated resistance in the α direction, where $W_{\alpha m}$ and $I_{\alpha m}$ = measured voltage and measured current respectively in the α direction. The high-T_c superconducting state is described by $R_{\alpha m} = 0$ due to the broken symmetry portion of equation (62) which occurs when $\theta_t + 2(\theta_\alpha - \theta_t) = \pi/2$. This condition combined with the free s - pair condition $\theta_\alpha = 2\theta_t$ gives the result $\theta_t = \pi/6$ and $\theta_\alpha = \pi/3$ for the coherent time high-T_c superconducting state which exists when $T < T_c$. In this scheme conventional superconductivity occurs when $R_{\alpha c} = 0$. Conventional and high-T_c superconductors are distinct, for instance experiments show that the normalized superconductivity energy gap for high-T_c superconductors is about twice the normalized energy gap for conventional superconductors.[10]

From equations (41), (46) and (47) it follows that the coherent thermal radiation energy density for high-T_c superconductors with $T < T_c$ is given by

$$\bar{E}^r_{coh}(\nu_c, T) = 8\pi h\nu_c^4/c^3\, \bar{B}_{coh}(\xi_c) \tag{63}$$

where

$$\bar{B}_{coh}(\xi_c) = \sum_{n=1}^{\infty}(I_1^n + jI_2^n) \tag{64}$$

$$= \sum_{n=1}^{\infty}[G_1^n - G_2^n + j(G_3^n + G_4^n)]$$

The real and imaginary parts and magnitude of $\bar{B}_{coh}$ are given by

$$B_{cohR} = \sum_{n=1}^{\infty}(G_1^n - G_2^n) \qquad B_{cohI} = \sum_{n=1}^{\infty}(G_3^n + G_4^n) \tag{65}$$

$$B_{coh} = (B_{cohR}^2 + B_{cohI}^2)^{1/2} \tag{66}$$

The choice of the constant frequency ν_c is related to the superconducting transition temperature by

$$h\nu_c = kT_c \tag{67}$$

so that equation (35) and (67) give

$$\xi_c = T_c/T \tag{68}$$

and equation (63) can be rewritten as

$$\bar{E}^r_{coh}(T_c, T) = 15\sigma/\pi^4\, T_c^4 \bar{B}_{coh}(T_c/T) \tag{69}$$

which gives the coherent-time portion of the thermal radiation energy density that is associated with the Cooper pairs of electrons and holes.

The total thermal radiation energy density emitted by a high-T_c superconductor for $T < T_c$ is the weighted average of the incoherent-time radiation energy density from the solid lattice of the copper oxide superconductor and the coherent-time radiation energy density from the condensate of Cooper electron (hole) pairs and therefore

$$\bar{E}_r = \alpha_{inc}\bar{E}^r_{inc} + \alpha_{coh}\bar{E}^r_{coh} \tag{70}$$

$$= \sigma T^4[\alpha_{inc} + \alpha_{coh}15/\pi^4(T_c/T)^4\bar{B}_{coh}(T_c/T)]$$

where α_{inc} = particle number density fraction of lattice atoms (ions) and where α_{coh} = particle number density fraction of electron (hole) pairs, which satisfy the following relations

$$\alpha_{coh}/\alpha_{inc} = n_p = n_e/2 \qquad \alpha_{inc} + \alpha_{coh} = 1 \tag{71}$$

$$\alpha_{inc} = (1 + n_p)^{-1} = 2(2 + n_e)^{-1} \tag{72}$$

$$\alpha_{coh} = n_p(1 + n_p)^{-1} = n_e(2 + n_e)^{-1} \tag{73}$$

where n_p = average number of electron (hole) pairs per lattice site, and n_e = average number of electrons (holes) per lattice site. Therefore the radiant energy density for a high-T_c superconductor with $T < T_c$ is the weighted average of a blackbody term associated with the lattice of atoms and ions, and a non-blackbody term associated with the coherent time state of the electron (hole) pair fluid. From equation (70) it follows that

$$E_{rR} = \alpha_{inc}\sigma T^4 + \alpha_{coh}15\sigma/\pi^4\ T_c^4 B_{cohR}(T_c/T) \tag{74}$$

$$E_{rI} = \alpha_{coh}15\sigma/\pi^4\ T_c^4 B_{cohI}(T_c/T) \tag{75}$$

$$E_r = (E^2_{rR} + E^2_{rI})^{1/2} \tag{76}$$

$$\tan\theta_{Er} = E_{rI}/E_{rR} \tag{77}$$

The measured value is the real part of the thermal radiation density given in equation (74).

Equation (70) is valid for a blackbody high-T_c superconductor. For real copper oxide materials a surface emissivity factor ε needs to be introduced into equations (70) through (77). Thus equation (74) becomes for $T < T_c$

$$E_{rR} = \varepsilon\sigma T^4[\alpha_{inc} + \alpha_{coh}15/\pi^4(T_c/T)^4 B_{cohR}(T_c/T)] \tag{78}$$

$$= \varepsilon_{eff}\sigma T^4$$

where the emissivity ε may be temperature dependent, and where the effective emissivity for high-T_c superconductors with $T < T_c$ is given by

$$\varepsilon_{eff} = \varepsilon[\alpha_{inc} + \alpha_{coh}15/\pi^4(T_c/T)^4 B_{cohR}(T_c/T)] \tag{79}$$

For matter in an incoherent time state the thermal radiation energy density is given by

$$E^r_{inc} = \varepsilon\sigma T^4 \tag{80}$$

which is also the case for high-T_c copper oxides with $T > T_c$. Therefore from equations (78) and (80) it follows that at the transition temperature $T = T_c$ there is a sudden drop in the thermal radiation energy density as the temperature is lowered across the transition temperature given by

$$\Delta E^r = [E_{rR} - E^r_{inc}]_{T_c} = -\varepsilon\sigma T_c^4[1 - \alpha_{inc} - \alpha_{coh}15/\pi^4 B_{cohR}(1)] \tag{81}$$

where from equations (58), (59) and (65) $B_{cohR}(1) < 0$. Equivalently, at the transition temperature there is an abrupt drop in the value of the emissivity given by

$$\Delta\varepsilon = [\varepsilon_{eff} - \varepsilon]_{T_c} = -\varepsilon[1 - \alpha_{inc} - \alpha_{coh}15/\pi^4 B_{cohR}(1)] \tag{82}$$

For $T < T_c$ equations (78) and (79) show that the thermal radiation energy density and emissivity are given approximately by

$$E_{rR} \sim \alpha_{inc}\varepsilon\sigma T^4 \tag{83}$$

$$\varepsilon_{eff} \sim \alpha_{inc}\varepsilon$$

because the exponential functions in $B_{cohR}(\xi_c)$ attenuate the factor $\xi_c^4 B_{cohR}(\xi_c)$ rapidly for $T < T_c$. The value of the emissivity approaches the value $\alpha_{inc}\varepsilon(0)$ for $T \to 0$.

3. CONCLUSION. It has been suggested that a high-T_c superconductor with $T < T_c$ is composed of two states of matter: a) a time-incoherent lattice of atoms and ions, and b) a fluid of Cooper pairs of electrons (or holes) which in this paper is assumed to be in a coherent-time state. The thermal radiation from such a two-state system is itself separable into a universal incoherent blackbody component arising from the atomic and ionic lattice, and a coherent non-blackbody radiation component which arises from the time-coherent electron or hole pairs. The coherent non-blackbody radiation component is not universal because it depends on the particular transition temperatures of the planar

copper oxide compounds. This coherent thermal energy density is determined by integrating the broken symmetry Planck spectrum over the internal phase angles of time. Thus gauge rotated or circular time describes the internal phase motion of oscillators in a coherent time state. A measurement of the non-blackbody thermal radiation component for $T < T_c$ in the planar copper oxides would represent proof that high-T_c superconductivity is described by a coherent time state of matter and is completely different from conventional BCS superconductivity.

ACKNOWLEDGEMENT

The author would like to thank Elizabeth K. Klein for typing this paper.

REFERENCES

1. Collins, P. D. B., Martin, A. D. and Squires, E. J., Particle Physics and Cosmology, John Wiley, New York, 1989.

2. Aitchison, I. J. R. and Hey, A. J. G., Gauge Theories in Particle Physics, Adam Hilger, Bristol, 1982.

3. Weiss, R. A., Relativistic Thermodynamics, Exposition, New York, 1976.

4. Weiss, R. A., Gauge Theory of Thermodynamics, K&W Publications, Vicksburg, 1989.

5. Planck, M., Theory of Heat, MacMillan, New York, 1949.

6. Planck, M., The Theory of Heat Radiation, Dover, New York, 1959.

7. Richtmyer, F. K. and Kennard, E. H., Introduction to Modern Physics, McGraw-Hill, New York, 1947.

8. Joos, G., Theoretical Physics, Hafner, New York, 1950.

9. Page, L., Introduction to Theoretical Physics, Van Nostrand, New York, 1952.

10. Margaritondo, G., Huber, D. L. and Olson, C. G., "Photoemission Spectroscopy of the High-Temperature Superconductivity Gap," Science, Vol. 246, 770, 10 November 1989.

SOME RESULTS ON NUMERICAL SOLUTION OF PARTIAL INTEGRO–DIFFERENTIAL EQUATIONS

Lars B. Wahlbin
Department of Mathematics
Cornell University
Ithaca, NY 14853

ABSTRACT.

The aim of this note is to describe briefly some recent developments in the numerical analysis of the finite element method applied to partial integro–differential equations.

1. INTRODUCTION.

Consider the following linear partial integro–differential problem of "parabolic" type for $u = u(t,x)$,

$$(1.1)\qquad \begin{cases} u_t - \Delta u = \int_0^t B(t,s)u(s)ds + f(t,x), t > 0, x \in \Omega, \\ u(t) = 0 \text{ on } \partial\Omega, \\ u(0) = v \text{ given.} \end{cases}$$

Here Ω is a bounded domain in R^d with sufficiently smooth boundary, Δ is the Laplace operator (for simplicity) and $B(t,s)$ is a second order (at most) partial differential operator in the spatial variables, with smooth coefficients for now. As a general reference to this and similar problems we give RENARDY, HRUSA and NOHEL [8].

Supported by the Army Research Office through the Mathematical Sciences Instiute, Cornell University and the National Science Foundation.

Let S_h , $0 < h < 1$, be a family of finite element spaces on Ω of the "usual" kind in $H_0^1(\Omega)$. Using the weak formulation of (1.1) one seeks the semidiscrete (time–continuous) approximation $u_h(t)$ in S_h by the formula

$$
\begin{cases}
(u_{h,t}, \chi) + D(u_h,\chi) \\
\qquad = \int_0^t B(t,s;\, u_h(s), \chi)ds + (f,\chi), \text{ for } \chi \in S_h, \\
u_h(o) = v_h \in S_h.
\end{cases} \tag{1.2}
$$

Here $D(v,w)$ is the Dirichlet form $(\nabla v, \nabla w)$ with $(v,w) = \int_\Omega vw\, dx$ the L_2–inner product. $B(t,s;v,w)$ denotes the bilinear form on H_0^1 obtained in the natural way from the partial differential operator $B(t,s)$ via integration by parts, if necessary.

Numerical solution of the problem (1.1) via finite differences goes back to DOUGLAS and JONES [3]. In the present finite element context a seminal paper is that of YANIK and FAIRWEATHER [12], cf. GREENWELL [5]. The techniques used there treated the case when the partial differential operator is of at most first order.

THOMÉE and ZHANG [10] adapted the techniques of WHEELER [11] from the parabolic case, $B \equiv O$. With $R_h u$ the (fictitious) Ritz projection into S_h defined by

$$
D(R_h u - u, \chi) = 0, \text{ for } \chi \in S_h, \tag{1.3}
$$

one proceeds to write an equation for $\theta = u_h - R_h u$. (One knows a lot about the error in $R_h u - u$.) The equation for θ turns out to be fairly complicated and far from easy to handle, but Thomée and Zhang managed. To describe a typical result, assume that the finite element spaces S_h have optimal approximation order r , $r \geq 2$ integer, i.e.,

$$
\min_{\chi \in S_h} \|v - \chi\|_{L_2(\Omega)} \leq C h^r \|v\|_{W_2^r(\Omega)} \tag{1.4}
$$

for $v \in W_2^r \cap H_0^1$. Here $r-1$ can be thought of as the basic piecewise polynomial degree of the finite element shape functions and h as the diameter of a typical element. Then,

assuming that the solution u of (1.1) is smooth enough, it is shown in [10] that

$$\|(u_h - u)(t)\|_{L_2(\Omega)} \leq C(T)\, h^r, \quad \text{for } 0 \leq t \leq T. \tag{1.5}$$

I.e., we have an error estimate in L_2 of optimal order.

2. RITZ–VOLTERRA PROJECTIONS.

CANNON and LIN [1] proposed an alternative mode of analysis to that in [10]. An operator $V_h(t)$ into S_h is now introduced via the prescription

$$D(V_h(t)u - u(t), \chi) = \int_0^t B(t,s; V_h(s)u - u(s), \chi)\, ds, \quad \text{for } \chi \in S_h, \tag{2.1}$$

It turns out that the equation for $u_h - V_h u$ is more manageable than that for $u_h - R_h u$. Of course, some more work now has to be done in analyzing the error in $V_h u - u$, and typically this is split as $V_h u - u = (V_h u - R_h u) + (R_h u - u)$ and an equation for the first part on the right is derived and analyzed. It is fair to say that the introduction of V_h splits the analysis into more "balanced" and manageable parts than that of merely considering R_h.

This technique just outlined is applied e.g. to deriving optimal order pointwise error estimates in LIN, THOMÉE and WAHLBIN [7]. (This paper also introduced the name Ritz–Volterra projection for V_h; the original paper [1] called it a "nonclassical H^1" projection.)

There are few changes in applying the technique to a "hyperbolic" integro–differential equation.

$$\begin{cases} u_{tt} - \Delta u = \int_0^t B(t,s)u(s)\, ds + f, \\ u(t) = 0 \text{ on } \partial\Omega, \\ u(0) = v, \; u_t(0) = w. \end{cases} \tag{2.2}$$

Again the introduction of the Ritz–Volterra projection cuts the analysis into balanced parts.

The Ritz–Volterra projection can in some cases in itself be viewed as the finite element solution of the Sobolev (aka pseudoparabolic) equation. For, assume that $B(t,s;\cdot,\cdot)$ in (2.1) actually does not depend on t. Differentiating (2.1) with respect to t,

$$D((V_h u)_t - u_t, \chi) = B(t; V_h u - u, \chi), \text{ for } \chi \epsilon S_h. \tag{2.3}$$

In this way we see the connection with finite element approximation of the typical Sobolev equation

$$\begin{cases} \Delta u_t = B(t)u, \ t>o, \ \chi \epsilon \Omega, \\ u(t) = o \text{ on } \partial\Omega, \\ u(o) = v. \end{cases} \tag{2.4}$$

For an early analysis of finite element methods in such problems, see FORD [6].

In CHEN, THOMÉE and WAHLBIN [2] the Ritz–Volterra technique is applied to a problem with a singular kernel,

$$u_t - \Delta u = \int_o^t K(t-s)\, Bu(s)\, ds + f \tag{2.5}$$

where $|k(t)| \simeq t^{-\alpha}$, $o<\alpha<1$. Results matching provable regularity for the solution u are derived.

3. DISCRETIZATION IN TIME.

So far we have been considering the semidiscrete (time–continuous) approximation given by the system of Volterra ordinary integro–differential equations (1.2). This is

clearly an intermediate step in the analysis since that system needs to be further discretized in time to arrive at a practical method.

To outline some issues, let us assume that we have in mind as a "basic" time–stepping method the backward Euler method (for simplicity in writing). Then write, as a preliminary step, with $U^n \simeq u_h(nk)$, k the (uniform) timestep,

$$(3.1) \qquad \begin{cases} \left(\dfrac{U^n - U^{n-1}}{k}, \chi\right) + D(U^n, \chi) \\ = \int_0^t B(t,s;u_h(s), \chi)\, ds + (f(nk), \chi), \text{ for } \epsilon\chi S_h. \end{cases}$$

The integral in time on the right needs to be further discretized. It seems natural to let that discretization involve only time levels $t_j = jk$. However, note that in general <u>all</u> time levels used in the quadrature of the integral need to be stored. With each time level involving perhaps $10^4 - 10^6$ degrees of freedom (in many space dimenisions) it may be prudent not to involve <u>all</u> previous levels. Rather, use a quadrature rule which is of higher order than the basic backward Euler rule. This being first order, O(k), use, as an example, Simpson's rule for the quadrature of the integral. This in turn being second order, $O(k_1^2)$ in the step k_1 used, it is natural to try to match $k_1 \simeq k^{1/2}$. Of course, if nk is not an integral multiple of k_1, *something special has to be done for a short* ($\leq k^{1/2}$) *last segment of the* integral.

It is easy to see that the storage requirements for such a combination is $O(k^{-1/2})$, as opposed to $O(k^{-1})$ if all previous time–levels were used in the quadrature.

The hard part is now to show that the resulting method is of $O(h^r + k)$ accuracy, of SLOAN and THOMÉE [9].

Various other combinations of basic time–stepping methods and "thinned" quadrature formulae are given in ZHANG [13].

In [2] the quadrature of the singular kernel in (2.5) is accomplished by a product

integration rule.

In the special case of $B(t,s) \equiv \Delta$ in (1.1), differentiation with respect to time leads to the strongly damped wave equation,

$$u_{tt} - \Delta u_t = \Delta u + f. \tag{3.2}$$

A study of finite element approximations for this can be found in LARSSON,THOMÉE and WAHLBIN [6]. An interesting point is that while the corresponding semigroup is analytic in time, it is only mildly smoothing in space (and then only with respect to compatibility conditions, not with respect to regularity). The consequences of this for space and time discretizations are elucidated.

4. CONCLUSION.

The partial integro–differential equation (1.1), although called of "parabolic" type by "analogy", has several differences with the heat equation, $B \equiv O$. A satisfactory mathematical theory for its numerical approximation has to proceed along lines specific to such equations. The introduction of Ritz–Volterra projections as an intermediate step in the analysis appears promising.

In practice, questions of storage limitations will have to be addressed by the theory for numerical solution of partial integro–differential equations.

REFERENCES.

1. Cannon, J.R., and Lin, Y., Nonclassical H^1 projection and Galerkin methods for nonlinear parabolic integro–differential equations, Calcolo 25, 1988, 187–201.

2. Chen, C., Thomée, V., and Wahlbin, L.B., Finite element approximation of a parabolic integro–differential equation with a weakly singular kernel, 1990, to appear.

3. Douglas, J. Jr., and Jones, B.F. Jr., Numerical methods for integro–differential equations of parabolic and hyperbolic types, Numer. Math. 4, 1962, 96–102.

4. Ford, W.H., Galerkin approximations to nonlinear pseudoparabolic partial differential equations, Aequationes Math. 14, 1976, 271–291.

5. Greenwell, C.E., Finite element methods for partial integro–differential equations, Ph.D. Thesis, University of Kentucky, Lexington, 1982.

6. Larsson, S., Thomée, V. and Wahlbin, L.B., Finite element methods for a strongly damped wave equation, IMA J. Numer. Anal., to appear.

7. Lin, Y.–P., Thomée, V. and Wahlbin, L.B., Ritz–Volterra projections to finite element spaces and applications to integro–differential and related equations, SIAM J. Numer. Anal., to appear.

8. Renardy, M., Hrusa, W.J. and Nohel, J.A., Mathematical Problems in Viscoelasticity, Pittman Monographs and Surveys in Pure and Applied Mathematics 35, 1987.

9. Sloan, I. and Thomée, V., Time discretization of an integro–differential equation of parabolic type, SIAM J. Numer. Anal. 23, 1986, 1052–1061.

10. Thomée, V., and Zhang, N.–Y., Error estimates for semidiscrete finite element methods for parabolic integro–differential equations, Math. Comp. 53, 1989, 121–139.

11. Wheeler, M.F., A. priori L_2 error estimates for Galerkin approximations to parabolic partial differential equations, SIAM J. Numer. Anal. 10, 1973, 723–759.

12. Yanik, E.G., and Fairweather, G., Finite element methods for parabolic and hyperbolic partial integro–differential equations, Nonlinear Anal. 12, 1988, 758–809.

13. Zhang, N.–Y., On the discretization in time and space of parabolic integro–differential equations, Ph.D. Thesis, Chalmers University of Technology 1990.

On Solving Cauchy Singular Integral Equations by Using General Quadrature-Collocation Nodes

R. P. Srivastav and Fenggang Zhang

Department of Applied Mathematics and Statistics,
SUNY at Stony Brook. Stony Brook, New York 11794

Abstract. We show that the specific nodes are not necessary for solving Cauchy singular integral equations by using quadrature-collocation methods. The solvability of the discrete system is proved for arbitrary selection of quadrature and collocation nodes. We also propose several special choices of these nodes. Especially, a weighted minimum norm-least square method is discussed.

1. Introduction

The classical theory for solving Cauchy Singular Integral Equations (CSIE) is based on the properties of sectionally holomorphic functions, which enable us to reduce the singular equation to a Fredholm equation of the second kind. (See N. I. Muskhelishvili [1] and F. D. Gakhov [2].) In many physical problems, when numerical solution is necessary, direct methods are often preferable. A method is called direct if the singular integral is replaced by a numerical approximation without resorting to regularization. Such methods initiated by F. Erdogan [3] and F. Erdogan and G. D. Gupta [4] have been developed subsequently by F. Erdogan, G. D. Gupta and T. S. Cook [5], P. S. Theocaris and N. I. Loakimidis [6].

Suppose the CSIE has the form

$$a(x)g(x) + \frac{b(x)}{\pi}\int_{-1}^{1}\frac{g(t)dt}{t-x} + \int_{-1}^{1}k(t,x)g(t)dt = f(x), \qquad -1 < x < 1. \tag{1.1}$$

All these numerical methods use Gauss-type formulae after expressing the unknown singular function as the product of a weight function $(1-x)^{\alpha-1}(1+x)^{\beta-1}$ and a smooth function to be computed. α and β are determined by Noether's index theorems.

Let $Pn^{(\alpha,\beta)}(x)$ be the Jacobi polynomial of degree n orthogonal with respect to the weight function $(1-x)^{\alpha-1}(1+x)^{\beta-1}$. The zeros of $P_n^{(\alpha,\beta)}(x)$ are used as quadrature nodes in the Gauss-Jacobi integral formula and the zeros of another related Jacobi polynomial are used as the corresponding collocation points. For example, when $\alpha = \beta = \frac{1}{2}$, the Jacobi polynomial $P_n^{(\alpha,\beta)}$ is $T_n(x)$, the Chebyshev polynomial of the first kind, and the related polynomial is $U_{n-1}(x)$, the Chebyshev polynomial of the second kind.

There are three factors, which influence the size of the error in the computed solution: (a) accuracy of the quadrature formula, (b) choice of the collocation nodes, (c) "condition" of the linear algebraic system. The rate of convergence depends on the "Lebesgue constants" of the collocation and quadrature nodes, and the smoothness of the functions. Certain sets of quadrature-collocation nodes are inadequate to represent the intrinsic features of the problems, especially of the problems arising from the oscillatory behavior of $f(x)$ or the kernel $k(t,x)$), or the problems due to large derivatives of these functions. For example, for the function

$$f(x) = \frac{e^x}{(x-c)^2 + a^2},$$

Research supported in part by NSF Grant No.DMS-8901893 and US Army Research Office Grant No.DAAL03-90-G-0019.

Typeset by $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX

for very small values of α, a collocation point in the immediate vicinity of c is essential. As discussed by Gerasoulis and Srivastav in [7], the methods based on orthogonal polynomials will require the quadrature rules of an excessively high degree, leading to extremely large systems of linear algebraic equations.

The paper by Tsamysphyros and Theocaris [11] appears to be the first to ask the rhetoric question "Are special collocation nodes necessary for the numerical solution of singular integral equations?". The examples are given showing that it is not so for the Gauss-Chebyshev quadrature and collocation. Our objective in this paper is to analyse the linear algebraic systems for solvability. The approximation characteristics of the computed solution will be discussed in a subsequent paper.

In order to make the paper self-sufficient, some wellknown results are included here. We organize the sections as follows:

Section 2 is used to construct the theory of orthogonal polynomials. The similar results can be found in S. Welstead's Ph.D thesis [12].

In Section 3 we prove the solvability of the system of equations derived from general quadrature-collocation nodes.

Section 4 is to discuss several optional selections of these nodes.

The paper is concluded with a numerical example in section 5.

2. Two Related Sequences of Orthogonal Polynomials

Consider the dominant part of the Cauchy singular integral operator U:

$$Ug(x) := ag(x) + \frac{b}{\pi}\int_{-1}^{1}\frac{g(t)}{t-x}dt, \tag{2.1}$$

where a and b are assumed constants and $|\,a\,| \neq |\,b\,|$. For the general case of variable $a(x)$ and $b(x)$, Welstead has discussed in [12]. Here, we only rehearse some related results.

The operator U is defined on the space of Hölder continuous functions. By using the sectionally holomophic functions, we introduce

$$\phi(z) = \frac{1}{2\pi i}\int_{-1}^{1}\frac{g(t)}{t-z}dt,$$

where $z = x + iy$, x and y are real numbers. Let

$$\phi^+(x) = \lim_{y\to 0^+}\phi(x+iy), \qquad \phi^-(x) = \lim_{y\to 0^-}\phi(x+iy).$$

We have

$$g(x) = \phi^+(x) - \phi^-(x),$$

$$\frac{1}{\pi i}\int_{-1}^{1}\frac{g(t)}{t-x}dt = \phi^+(x) + \phi^-(x)$$

and

$$Ug(x) = (a+ib)\phi^+(x) - (a-ib)\phi^-(x).$$

Denote

$$G = \frac{a-ib}{a+ib} = \frac{a^2-b^2-i2ab}{a^2+b^2} = e^{-i2\theta},$$

we have

$$|G|=1, \qquad \tan\theta=\frac{b}{a}, \qquad 0\leq|\theta|\leq\frac{\pi}{2}, \qquad |\theta|\neq\frac{\pi}{4}.$$

Let

$$\Gamma(x)=-\frac{\theta}{\pi}\log\frac{1-x}{1+x},$$

$$Z(x)=(1-x)^{\lambda_1}(1+x)^{\lambda_{-1}}e^{\Gamma(x)}=(1-x)^{\lambda_1-\frac{\theta}{\pi}}(1+x)^{\lambda_{-1}+\frac{\theta}{\pi}}.$$

If $a\geq 0$ and $b\geq 0$, then $0\leq\frac{\theta}{\pi}\leq\frac{1}{2}$. λ_1 and λ_{-1} can be chosen as following so that both $Z(x)$ and $Z^{-1}(x)$ are integrable on $(-1,1)$.

(1) $\lambda_1=0$ and $\lambda_{-1}=-1$;

(2) $\lambda_1=0$ and $\lambda_{-1}=0$ or $\lambda_1=1$ and $\lambda_{-1}=-1$;

(3) $\lambda_1=1$ and $\lambda_{-1}=0$.

The index of the operator U is defined as $\chi=-(\lambda_1+\lambda_{-1})$.

In the case of 1-index,

$$Z(x)=(1-x)^{\frac{\theta}{\pi}}(1+x)^{-1+\frac{\theta}{\pi}}=\frac{1}{\sqrt{1-x^2}}\left(\frac{1-x}{1+x}\right)^{\frac{1}{2}-\frac{\theta}{\pi}}. \tag{2.2}$$

Let $\{p_n(x)\}_0^\infty$ be a sequence of monic orthogonal polynomials with respect to the weight function $Z(x)$ on the interval $(-1,1)$, i.e.

$$\int_{-1}^{1}Z(x)p_n(x)p_m(x)dx=c_n\delta_{nm}, \tag{2.3}$$

where c_n is a constant, δ_{nm} is the Kronecker notation. According to the general properties of orthogonal polynomials, there is a recursive formula between every three consecutive polynomials:

$$\begin{aligned}
p_0(x)&=1,\\
p_1(x)&=(x-\alpha_0)p_0(x),\\
p_2(x)&=(x-\alpha_1)p_1(x)-\beta_1p_0(x),\\
&\cdots\cdots,\\
p_{n+1}(x)&=(x-\alpha_n)p_n(x)-\beta_np_{n-1}(x),
\end{aligned}$$

where $\{\alpha_k\}_0^n$ and $\{\beta_k\}_1^n$ are two sequences of numbers. Now, we can construct a new sequence of monic polynomials $\{q_n(x)\}_0^\infty$ by using the $\{\alpha_n\}_0^\infty$ and $\{\beta_n\}_0^\infty$:

$$\begin{aligned}
q_0(x)&=1,\\
q_1(x)&=(x-\alpha_1)q_0(x),\\
q_2(x)&=(x-\alpha_2)q_1(x)-\beta_2q_0(x),\\
&\cdots\cdots,\\
q_{n+1}(x)&=(x-\alpha_{n+1})q_n(x)-\beta_{n+1}q_{n-1}(x).
\end{aligned}$$

Our main arguments are

PROPOSITION 2.1.

$$U(Z(x)p_0(x)) = 0,$$
$$U(Z(x)p_k(x)) = \mu q_{k-1}(x), \text{ for } k = 1, 2, \cdots$$

where

$$\mu = \sqrt{a^2 + b^2}. \tag{2.4}$$

PROPOSITION 2.2. $\{q_n(x)\}_0^\infty$ *is a sequence of orthogonal polynomials with respect to the weight function* $Z^{-1}(x)$ *on* $(-1, 1)$.

Consider the adjoint operator V of U:

$$Vg(x) := ag(x) - \frac{b}{\pi}\int_{-1}^{1} \frac{g(t)dt}{t-x}. \tag{2.5}$$

We have a similar result of V as the Proposition 2.1.

PROPOSITION 2.3. *For* $k = 1, 2, \cdots$

$$V(\frac{q_{k-1}(x)}{Z(x)}) = \mu p_k(x).$$

3. THE DISCRETE SCHEME OF GENERAL QUADRATURE NODES AND COLLOCATION POINTS

For the problem with 1-index, we shall find the solution $g(x)$ satisfying

$$\begin{cases} ag(x) + \frac{b}{\pi}\int_{-1}^{1} \frac{g(t)dt}{t-x} = f(x), \\ \frac{1}{\pi}\int_{-1}^{1} g(x)dx = c. \end{cases} \tag{3.1}$$

Let $g(x) = Z(x)\phi(x)$. We choose $x_1 < x_2 < \cdots < x_n$ as quadrature nodes and $y_1, y_2, \cdots, y_{n-1}$ as collocation points. The $\{x_j\}_1^n$ and the $\{y_k\}_1^{n-1}$ are independent of the weight function $Z(x)$.

Denote

$$X(x) = \prod_{j=1}^{n}(x - x_j), \tag{3.2}$$

$$Y(y) = \prod_{k=1}^{n-1}(y - y_k). \tag{3.3}$$

The quadrature coefficients are defined as

$$W_j = \int_{-1}^{1} \frac{Z(x)X(x)dx}{(x - x_j)X'(x_j)}, \tag{3.4}$$

$$V_k = \int_{-1}^{1} \frac{Y(y)dy}{Z(y)(y - y_k)Y'(y_k)}. \tag{3.5}$$

We can construct the system of equations as following:

$$A\tilde{\phi} = \tilde{f}, \tag{3.6}$$

where

$$\tilde{\phi} = (\tilde{\phi}(x_1), \tilde{\phi}(x_2), \cdots, \tilde{\phi}(x_n))^T,$$
$$\tilde{f} = (f(y_1), f(y_2), \cdots, f(y_{n-1}), c)^T$$

and

$$A = (a_{kj})_{n\times n},$$

$$\begin{cases} a_{kj} = \frac{1}{x_j - y_k}\left(\frac{b}{\pi}W_j - \frac{1}{X'(x_j)}U(Z(x)X(x))|_{x=y_k}\right), \text{ for } 1 \le k \le n-1, \qquad 1 \le j \le n, \\ a_{nj} = \frac{1}{\pi}W_j, \text{ for } 1 \le j \le n. \end{cases} \tag{3.7}$$

We can also construct the corresponding matrix B,

$$B = (b_{kj})_{n\times n},$$

$$\begin{cases} b_{kj} = \frac{1}{x_k - y_j}\left(\frac{b}{\pi}V_j + \frac{1}{Y'(y_j)}V(\frac{Y(y)}{Z(y)})|_{y=x_k}\right), \text{ for } 1 \le k \le n, \qquad 1 \le j \le n-1, \\ b_{kn} = \mu b, \text{ for } 1 \le k \le n. \end{cases} \tag{3.8}$$

In order to prove the solvability of (3.6) and to find the inverse of (3.7), we introduce two general lemmas first.

LEMMA 3.1. *If $A = (a_{kj})$ is an $n \times n$ matrix satisfying*

$$\sum_{j=1}^{n} a_{kj}\phi(x_j) = U(Z(x)\phi(x))|_{x=y_k}, \qquad k = 1, 2, \cdots, n-1,$$
$$\sum_{j=1}^{n} a_{nj}\phi(x_j) = \frac{1}{\pi}\int_{-1}^{1} Z(x)\phi(x)dx$$

for any polynomial $\phi(x)$ of degree $\le n-1$, then A is invertible.

LEMMA 3.2. *If $B = (b_{kj})$ is an $n \times n$ matrix satisfying*

$$\sum_{j=1}^{n-1} b_{kj}\phi(y_j) = V(\frac{\phi(y)}{Z(y)})|_{y=x_k}, \qquad k = 1, 2, \cdots, n$$
$$b_{kn} = \mu b,$$

for any polynomial $\phi(y)$ of degree $\le n-2$ and A is an $n \times n$ matrix satisfying the conditions of Lemma 3.1, then

$$AB = \mu^2 I.$$

That is

$$A^{-1} = \frac{1}{\mu^2}B.$$

Next, we need to check that the matrix of (3.7) satisfies the conditions of Lemma 3.1 and the matrix of (3.8) satisfies the conditions of Lemma 3.2.

PROPOSITION 3.3. *The matrix A in (3.7) satisfies the conditions of Lemma 3.1. A is invertible.*

PROPOSITION 3.4. *The matrix B in (3.8) satisfies the conditions of Lemma 3.2. The inverse of the matrix A in (3.7) has a closed form as $\frac{1}{\mu^2}B$.*

4. Several Special Choices of the Quadrature and Collocation Nodes

Sometime we ne i a simpler form of the system of equations, or we require higher accurate quadrature rule and or we want the two groups of nodes to be the same. We can make various options about the quadrature and the collocation nodes. In this sections we provide four different choices of these nodes.

1. Gauss-Jacobi Scheme. Let $\{x_1, x_2, \cdots, x_n\} = \{\xi_1, \xi_2, \cdots, \xi_n\}$ and $\{y_1, y_2, \cdots, y_{n-1}\} = \{\eta_1, \eta_2, \cdots, \eta_{n-1}\}$, where $\{\xi_k\}_1^n$ and $\{\eta_k\}_1^{n-1}$ are the zeros of the $p_n(x)$ and $q_{n-1}(y)$ respectively. We have $X(x) = p_n(x)$ and $Y(y) = q_{n-1}(y)$.

Denote

$$W_j^* = \int_{-1}^{1} \frac{Z(x)p_n(x)dx}{(x-\xi_j)p_n'(\xi_j)}, \tag{4.1}$$

$$V_j^* = \int_{-1}^{1} \frac{q_{n-1}(y)dy}{Z(y)(y-\eta_j)q_{n-1}'(\eta_j)}. \tag{4.2}$$

The matrix A in (3.7) will become the form

$$A = (a_{kj}^*)_{n\times n}, \tag{4.3}$$

where

$$a_{kj}^* = \frac{1}{\xi_j - \eta_k}\left(\frac{b}{\pi}W_j^* - \frac{1}{p_n'(\xi_j)}U(Z(x)p_n(x))|_{x=\eta_k}\right)$$

$$= \frac{1}{\xi_j - \eta_k}\left(\frac{b}{\pi}W_j^* - \frac{1}{p_n'(\xi_j)}q_{n-1}(\eta_k)\right) = \frac{b}{\pi}\frac{W_j^*}{\xi_j - \eta_k}, \text{ for } 1 \le k \le n-1, \qquad 1 \le j \le n.$$

The matrix B in (3.8) has also a simpler form,

$$B = (b_{kj}^*)_{n\times n}, \tag{4.4}$$

where

$$b_{kj}^* = -\frac{1}{\eta_j - \xi_k}\left(\frac{b}{\pi}V_j^* + \frac{1}{q_{n-1}'(\eta_j)}V\left(\frac{q_{n-1}(y)}{Z(y)}\right)|_{y=\xi_k}\right)$$

$$= \frac{1}{\xi_k - \eta_j}\left(\frac{b}{\pi}V_j^* + \frac{1}{q_{n-1}'(\eta_j)}p_n(\xi_k)\right) = \frac{b}{\pi}\frac{V_j^*}{\xi_k - \eta_j}, \text{ for } 1 \le k \le n, \qquad 1 \le j \le n-1.$$

Similar to Proposition 3.3 , we have

Proposition 4.1. *For any polynomial $\phi(t)$ of degree $\le 2n$, the matrix A in (4.3) satisfies*

$$\sum_{j=1}^{n} a_{kj}^*\phi(\xi_j) = U(Z(x)\phi(x))|_{x=\eta_k}, \text{ for } 1 \le k \le n-1,$$

$$\sum_{j=1}^{n} a_{kj}^*\phi(\xi_j) = \frac{1}{\pi}\int_{-1}^{1} Z(x)\phi(x)dx.$$

Similar to Proposition 3.4 we have

PROPOSITION 4.2. *For any polynomial $\phi(t)$ of degree $\leq 2n-2$, the matrix B in (4.4) satisfies*

$$\sum_{j=1}^{n-1} b_{kj}^{*}\phi(\eta_j) = V(\frac{\phi(x)}{Z(x)})|_{x=\xi_k}, \text{ for } k = 1, 2, \cdots, n.$$

Since the Gauss quadrature formulae have higher algebraic accuracy. This scheme may have higher approximation rate of convergence.

2. Lobatto-Jacobi Scheme. Let $\{-1, -\eta_{n-1}, -\eta_{n-2}, \cdots, -\eta_1, 1\}$ be quadrature nodes and let $\{-\xi_n, -\xi_{n-1}, \cdots, -\xi_1\}$ be collocation points. We have

$$X(x) = (x+1)(x-1)\prod_{j=1}^{n}(x+\eta_j) \tag{4.5}$$

$$= (-1)^n(1-x)(1+x)\prod_{j=1}^{n-1}(-x-\eta_j)$$

$$= (-1)^n(1-x)(1+x)q_{n-1}(-x),$$

$$Y(y) = \prod_{j=1}^{n}(y+\xi_j) = (-1)^n p_n(-y). \tag{4.6}$$

Considering our assumption of $a \geq 0$, $b > 0$ and $\theta \geq 0$ in the definition of $Z(x) = (1-x)^{-\frac{\theta}{\pi}}(1+x)^{-1+\frac{\theta}{\pi}}$, we can represent the quadrature coefficients W_j and V_j by using the W_j^* and V_j^*.

$$W_0 := \int_{-1}^{1} \frac{Z(x)X(x)dx}{(x+1)X'(-1)} \tag{4.7}$$

$$= \frac{1}{2q_{n-1}(1)}\int_{-1}^{1} \frac{(1-x)^{-\frac{\theta}{\pi}}(1+x)^{-1+\frac{\theta}{\pi}}(1-x)(1+x)q_{n-1}(-x)dx}{x+1}$$

$$= \frac{1}{2q_{n-1}(1)}\int_{-1}^{1}(1-x)^{-\frac{\theta}{\pi}}(1+x)^{1-\frac{\theta}{\pi}}q_{n-1}(x)dx$$

$$= -\frac{1}{2q_{n-1}(1)}\int_{-1}^{1}\frac{q_{n-1}(x)dx}{(x-1)Z(x)}$$

$$= -\frac{1}{2q_{n-1}(1)}\frac{\pi}{b}(\frac{aq_{n-1}(1)}{Z(1)} - \mu p_n(1))$$

$$= \frac{\pi\mu p_n(1)}{2bq_{n-1}(1)},$$

where we use the fact that $\frac{1}{Z(1)} = 0$ because of $\theta \geq 0$. For $j = 1, 2, \cdots, n-1$, we have

$$W_j := \int_{-1}^{1} \frac{Z(x)X(x)dx}{(x+\eta_{n-j})X'(\eta_{n-j})} \tag{4.8}$$

$$= -\int_{-1}^{1} \frac{(1-x)^{-\frac{\theta}{\pi}}(1+x)^{-1+\frac{\theta}{\pi}}(1-x)(1+x)q_{n-1}(x)dx}{(x+\eta_{n-j})q'_{n-1}(\eta_{n-j})(1-\eta_{n-j}^2)}$$

$$= \frac{1}{1-\eta_{n-j}^2}\int_{-1}^{1} \frac{(1+x)^{1-\frac{\theta}{\pi}}(1-x)^{\frac{\theta}{\pi}}q_{n-1}(x)dx}{(x-\eta_{n-j})q'_{n-1}(\eta_{n-j})}$$

$$= \frac{V^*_{n-j}}{1-\eta_{n-j}^2}.$$

and

(4.9)
$$W_n := \int_{-1}^{1} \frac{Z(x)X(x)dx}{(x-1)X'(1)}$$

$$= -\frac{1}{2q_{n-1}(-1)}\int_{-1}^{1} \frac{(1-x)^{-\frac{\theta}{\pi}}(1+x)^{-1+\frac{\theta}{\pi}}(1-x)(1+x)q_{n-1}(x)dx}{x-1}$$

$$= \frac{1}{2q_{n-1}(-1)}\int_{-1}^{1} \frac{q_{n-1}(x)dx}{Z(x)(x+1)}$$

$$= \frac{1}{2q_{n-1}(-1)}\frac{\pi}{b}\left(\frac{aq_{n-1}(-1)}{Z(-1)} - \mu p_n(-1)\right)$$

$$= -\frac{\pi\mu}{2b}\frac{p_n(-1)}{q_{n-1}(-1)},$$

where we also use $\frac{1}{Z(-1)} = 0$ because of $\theta \geq 0$.

For $j = 1, 2, \cdots, n$, we have

(4.10)
$$V_j := \int_{-1}^{1} \frac{Y(y)dy}{(y+\xi_{n-j+1})Z(y)Y'(-\xi_{n-j+1})}$$

$$= -\int_{-1}^{1} \frac{p_n(-y)dy}{(y+\xi_{n-j+1})Z(y)p'_n(\xi_{n-j+1})}$$

$$= \int_{-1}^{1} \frac{(1+y)^{\frac{\theta}{\pi}}(1-y)^{1-\frac{\theta}{\pi}}p_n(y)dy}{(y-\xi_{n-j+1})p'_n(\xi_{n-j+1})}$$

$$= \int_{-1}^{1} \frac{Z(y)(1-y^2)p_n(y)dy}{(y-\xi_{n-j+1})p'_n(\xi_{n-j+1})}$$

$$= (1-\xi_{n-j+1}^2)W^*_{n-j+1} - \int_{-1}^{1} \frac{Z(y)p_n(y)(y+\xi_{n-j+1})dy}{p'_n(\xi_{n-j+1})}$$

$$= (1-\xi_{n-j+1}^2)W^*_{n-j+1}$$

We can construct the system of equations as following:

Let

$$\tilde{\phi} = (\tilde{\phi}(-1), \tilde{\phi}(-\eta_{n-1}), \cdots, \tilde{\phi}(-\eta_1), \tilde{\phi}(1))^T,$$

$$\tilde{f} = (c, f(\xi_n), f(-\xi_{n-1}), \cdots, f(-\xi_1))^T.$$

We have

$$A\tilde{\phi} = \tilde{f}, \tag{4.11}$$

where

$$A = (a_{kj})_{(n+1)\times(n+1)}, \qquad 0 \le k \le n, \qquad 0 \le j \le n. \tag{4.11}$$

$$\begin{aligned}
a_{kj} &= \frac{b}{\pi}\frac{V^*_{n-j}}{1-\eta^2_{n-j}}\frac{1}{\xi_{n-k+1}-\eta_{1,1-j}}, \text{ for } 1 \le k \le n, \qquad 1 \le j \le n-1,\\
a_{k0} &= -\mu\frac{p_n(1)}{q_{n-1}(1)}\frac{1}{1-\xi_{n-k+1}}, \text{ for } 1 \le k \le n,\\
a_{kn} &= -\mu\frac{p_n(-1)}{q_{n-1}(-1)}\frac{1}{1+\xi_{n-k+1}}, \text{ for } 1 \le k \le n,\\
a_{0j} &= \frac{1}{\pi}\frac{V^*_{n-j}}{1-\eta^2_{n-j}}, \text{ for } 1 \le j \le n-1,\\
a_{00} &= \frac{\mu}{2b}\frac{p_n(1)}{q_{n-1}(1)},\\
a_{0n} &= -\frac{\mu}{2b}\frac{p_n(-1)}{q_{n-1}(-1)}.
\end{aligned} \tag{4.12}$$

Similar to Proposition 4.1, we have

PROPOSITION 4.3. *For any polynomial $\phi(t)$ of degree $\le 2n-3$, the matrix A in (4.12) satisfies*

$$\begin{aligned}
a_{k0}\phi(-1) + \sum_{j=1}^{n-1} a_{kj}\phi(-\eta_{n-j}) + a_{kn}\phi(1) &= U(Z(x)\phi(x))|_{x=-\xi_{n-k+1}}, \text{ for } 1 \le k \le n,\\
a_{00}\phi(-1) + \sum_{j=1}^{n-1} a_{0j}\phi(-\eta_{n-j}) + a_{0n}\phi(1) &= \frac{1}{\pi}\int_{-1}^{1} Z(x)\phi(x)dx.
\end{aligned}$$

Similar to Proposition 4.2, we introduce a matrix

$$B = (b_{kj})_{(n+1)\times(n+1)}, \qquad 0 \le k \le n, \qquad 0 \le j \le n,$$

where

$$\begin{aligned}
b_{kj} &= \frac{b}{\pi}\frac{(1-\xi^2_{n-j+1})W^*_{n-j+1}}{\xi_{n-i+1}-\eta_{n-k}}, \text{ for } 1 \le k \le n, \qquad 1 \le j \le n,\\
b_{0j} &= \frac{b}{\pi}\frac{(1-\xi^2_{n-j+1})W^*_{n-j+1}}{\xi_{n-j+1}+1}, \text{ for } 1 \le j \le n,\\
b_{nj} &= \frac{b}{\pi}\frac{(1-\xi^2_{n-j+1})W^*_{n-j+1}}{\xi_{n-j+1}-1}, \text{ for } 1 \le j \le n,\\
b_{k0} &= b\mu, \text{ for } 0 \le k \le n.
\end{aligned} \tag{4.13}$$

PROPOSITION 4.4. *For any polynomial $\phi(t)$ of degree $\leq 2n-1$, the matrix B in (4.13) satisfying*

$$\sum_{j=1}^{n} b_{kj}\phi(-\xi_{n-j+1}) = \begin{cases} V(\frac{\phi(y)}{Z(y)})|_{y=-1}, & \text{for } k=0, \\ V(\frac{\phi(y)}{Z(y)})|_{y=-\eta_{n-j}}, & \text{for } 1 \leq k \leq n-1, \\ V(\frac{\phi(y)}{Z(y)})|_{y=1}, & \text{for } k=n. \end{cases}$$

By using the same way as that in proof of Lemma 3.1 and Lemma 3.2, we can show that the coefficient matrix A of Gauss-Jacobi scheme in (4.3) and the coefficient matrix A of Lobatto-Jacobi scheme in (4.12) are invertible and that the their inverses are of the closed form $\frac{1}{\mu^2}B$, B is given in (4.4) and (4.13) respectively.

These two schemes can approximate the singular integral operator in higher algebraic accuracy. Are there other groups of quadrature nodes or collocation nodes which have similar properties? For example, can we construct Radau-Jacobi scheme? In the case of $a=0$ and $b=1$, we found four different choices of the nodes (see [9]).

3.Coincidence of Quadrature Nodes and Collocation Points.

If we choose the $n-1$ collocation points the same as the first $n-1$ quadrature nodes, we also can construct an invertible linear system of equations.

Let $y_j = x_j$, for $1 \leq j \leq n-1$, we have

$$X(x) = \prod_{j=1}^{n}(x-x_j),$$

$$Y(y) = \prod_{j=1}^{n-1}(y-x_j).$$

The discrete system of equations has the form

$$A\tilde{\phi} = \tilde{f}.$$

where

$$\tilde{\phi} = (\tilde{\phi}(x_1), \tilde{\phi}(x_2), \cdots, \tilde{\phi}(x_n))^T,$$

$$\tilde{f} = (f(x_1), f(x_2), \cdots, f(x_{n-1}), c)^T,$$

$$A = (a_{kj})_{n\times n}, \qquad 1 \leq k \leq n, \qquad 1 \leq j \leq n,$$

$$a_{kj} = \frac{b}{\pi X'(x_j)} \frac{W_j X'(x_j) - W_k X'(x_k)}{x_j - x_k}, \text{ for } 1 \leq k \leq n-1, \qquad 1 \leq j \leq n, \qquad , k \neq j,$$

$$a_{kk} = aZ(x_k) + \frac{b}{\pi X'(x_k)} \int_{-1}^{1} \frac{Z(t)X(t)dt}{(t-x_k)^2}, \text{ for } 1 \leq k \leq n-1, \tag{4.14}$$

$$a_{nj} = \frac{1}{\pi} W_j, \text{ for } 1 \leq j \leq n$$

PROPOSITION 4.5. *For any polynomial $\phi(t)$ of degree $\leq n-1$, the matrix A in (4.14) satisfies*

$$\sum_{j=1}^{n} a_{kj}\phi(x_j) = U(Z(x)\phi(x))|_{x=x_k}, \text{ for } 1 \leq k \leq n-1,$$

$$\sum_{j=1}^{n} a_{nj}\phi(x_j) = \frac{1}{\pi}\int_{-1}^{1} Z(x)\phi(x)dx.$$

By Lemma 3.1, the matrix A in (4.14) is nonsingular. Next, we introduce a matrix B,

$$B = (b_{kj})_{n\times n} \qquad 1 \le k \le n, \qquad 1 \le j \le n,$$

$$\begin{aligned} b_{kj} &= -\frac{b}{\pi Y'(x_j)} \frac{V_j Y'(x_j) - V_k Y'(x_k)}{x_j - x_k}, \text{ for } 1 \le k \le n, \qquad 1 \le j \le n-1, \qquad k \ne j, \\ b_{kk} &= \frac{a}{Z(x_k)} - \frac{b}{\pi Y'(x_k)} \int_{-1}^{1} \frac{Y(t)dt}{Z(t)(t-x_k)^2}, \text{ for } 1 \le k \le n-1, \\ b_{kn} &= b\mu, \text{ for } 1 \le k \le n. \end{aligned} \tag{4.15}$$

PROPOSITION 4.6. *For any polynomial $\phi(t)$ of degree $\le n-2$, the matrix B in (4.15) satisfies*

$$\sum_{j=1}^{n-1} b_{kj}\phi(x_j) = V(\frac{\phi(x)}{Z(x)})|_{x=x_k}, \text{ for } 1 \le k \le n.$$

By using Lemma 3.2 , we know that the inverse of A in (4.14) is $\frac{1}{\mu^2}B$.

Note: The collocation points are not necessary the FIRST $n-1$ quadrature nodes. In fact, they can be arbitrary $n-1$ nodes. If we take n collocation points, we will obtain a system of $n+1$ equations of n unknowns. The rank of the coefficient matrix is n. We can use the regularization method to find it's solution.

4.Minimum Norm-Least Square Scheme.

Let $\{x_1, x_2, \cdots, x_n\} = \{y_1, y_2, \cdots, y_n\} = \{\xi_1, \xi_2, \cdots, \xi_n\}$, then $X(x) = Y(x) = p_n(x)$. We can construct a system of n equations of n unknowns only from the integral equation $U(Z(x)\phi(x)) = f(x)$ without considering the additional condition $\frac{1}{\pi}\int_{-1}^{1} Z(x)\phi(x) = c$. We are going to show that the rank of the coefficient matrix is $n-1$ and to find a closed form of it's generalized inverse.

Denote the discrete system as

$$A\tilde{\phi} = \tilde{f},$$

where

$$\begin{aligned} \tilde{\phi} &= (\tilde{\phi}(\xi_1), \tilde{\phi}(\xi_2), \cdots, \tilde{\phi}(\xi_n))^T, \\ \tilde{f} &= (f(\xi_1), f(\xi_2), \cdots, f(\xi_n))^T, \\ A &= (a_{kj})_{n\times n}, \end{aligned}$$

$$\begin{aligned} a_{kj} &= \frac{b}{\pi p_n'(\xi_j)} \int_{-1}^{1} \frac{Z(t)p_n(t)dt}{(t-\xi_j)(t-\xi_k)}, \text{ for } 1 \le k, j \le n, \qquad k \ne j, \\ a_{kk} &= aZ(\xi_k) + \frac{b}{\pi p_n'(\xi_k)} \int_{-1}^{1} \frac{Z(t)p_n(t)dt}{(t-\xi)^2}, \text{ for } 1 \le k \le n. \end{aligned} \tag{4.16}$$

Similarly, we introduce a matrix B,

$$B = (b_{kj})_{n\times n},$$

$$\begin{aligned} b_{kj} &= -\frac{b}{\pi p_n'(\xi_j)} \int_{-1}^{1} \frac{p_n(t)dt}{Z(t)(t-\xi_j)(t-\xi_k)} \text{ for } 1 \le k, j \le n, \qquad k \ne j, \\ b_{kk} &= \frac{a}{Z(\xi_k)} - \frac{b}{\pi p_n'(\xi_k)} \int_{-1}^{1} \frac{p_n(t)dt}{Z(t)(t-\xi_k)^2}, \text{ for } 1 \le k \le n. \end{aligned} \tag{4.17}$$

PROPOSITION 4.7. *For any polynomial $\phi(t)$ of degree $\leq 2n$, the matrix A in (4.16) satisfies*

$$\sum_{j=1}^{n} a_{kj}\phi(\xi_j) = U(Z(x)\phi(x))|_{x=\xi_k}, \text{ for } 1 \leq k \leq n.$$

PROPOSITION 4.8. *For any polynomial $\phi(t)$ of degree $\leq 2n$, the matrix B in (4.17) satisfies*

$$\sum_{j=1}^{n} b_{kj}\phi(\xi_j) = V(\frac{\phi(x)}{Z(x)})|_{x=\xi_k}, \text{ for } 1 \leq k \leq n.$$

The proofs are the same as that of Proposition 4.5, however, we can not use the general Lemma 3.1 and Lemma 3.2 anymore.

Denote

$$P = \begin{pmatrix} p_0(\xi_1) & p_1(\xi_1) & \cdots & p_{n-1}(\xi_1) \\ p_0(\xi_2) & p_1(\xi_2) & \cdots & p_{n-1}(\xi_2) \\ \cdots & \cdots & \cdots & \cdots \\ p_0(\xi_n) & p_1(\xi_n) & \cdots & p_{n-1}(\xi_n) \end{pmatrix}$$

and

$$Q = \begin{pmatrix} 0 & q_0(\xi_1) & \cdots & q_{n-2}(\xi_1) \\ 0 & q_0(\xi_2) & \cdots & q_{n-2}(\xi_2) \\ \cdots & \cdots & \cdots & \cdots \\ 0 & q_0(\xi_n) & \cdots & q_{n-2}(\xi_n) \end{pmatrix}.$$

We have

$$AP = \mu Q.$$

Since P is nonsingular, $rank(A) = rank(Q) = n - 1$.

Denote

$$\hat{P} = \begin{pmatrix} p_1(\xi_1) & p_2(\xi_1) & \cdots & p_{n-1}(\xi_1) & 0 \\ p_1(\xi_2) & p_2(\xi_2) & \cdots & p_{n-1}(\xi_2) & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ p_1(\xi_n) & p_2(\xi_n) & \cdots & p_{n-1}(\xi_n) & 0 \end{pmatrix}$$

and

$$\hat{Q} = \begin{pmatrix} q_0(\xi_1) & q_1(\xi_1) & \cdots & q_{n-1}(\xi_1) \\ q_0(\xi_2) & q_1(\xi_2) & \cdots & q_{n-1}(\xi_2) \\ \cdots & \cdots & \cdots & \cdots \\ q_0(\xi_n) & q_1(]xi_n) & \cdots & q_{n-1}(\xi_n) \end{pmatrix}.$$

We have

$$B\hat{Q} = \mu\hat{P}.$$

Since Q is nonsingular, $rank(A) = rank(\hat{p}) = n - 1$.

Introduce

$$\hat{B} = \frac{1}{\mu^2}B,$$

we have

PROPOSITION 4.9. *$\hat{B}$ is the $M-N$ generalized inverse of A in (4.16), that is $\hat{B}$ and A satisfy*

$$A\hat{B}A = A, \tag{4.18.1}$$

$$\hat{B}A\hat{B} = \hat{B}, \tag{4.18.2}$$

$$(A\hat{B})^T M = M(A\hat{B}), \tag{4.18.3}$$

$$(\hat{B}A)^T N = N(\hat{B}A), \tag{4.18.4}$$

where $M = (\hat{Q}\hat{Q}^T)^{-1}$ and $N = (PP^T)^{-1}$.

Proof. Let

$$J = \begin{pmatrix} 0 & I_{n-1} \\ 0 & 0 \end{pmatrix},$$

where I_{n-1} is an identity matrix of order $n-1$. It is obvious that J^T and J satisfy

$$JJ^TJ = J, \tag{4.19.1}$$

$$J^TJJ^T = J^T, \tag{4.19.2}$$

$$(JJ^T)^T = JJ^T, \tag{4.19.3}$$

$$(J^TJ)^T = J^TJ. \tag{4.19.4}$$

Since $\hat{P} = pJ^T$ and $Q = \hat{Q}J$, we have

$$A = \mu\hat{Q}JP^{-1},$$

$$\hat{B} = \frac{1}{\mu}PJ^T\hat{Q}^{-1}.$$

Therefore

$$A\hat{B}A = \mu(\hat{Q}JP^{-1})(PJ^T\hat{Q}^{-1})(\hat{Q}JP^{-1}) = \mu(\hat{Q}JJ^TJP^{-1} = A,$$

$$\hat{B}A\hat{B} = \frac{1}{\mu}(PJ^T\hat{Q}^{-1})(\hat{Q}JP^{-1})(PJ^T\hat{Q}^{-1}) = \frac{1}{\mu}PJ^TJJ^T\hat{Q}^{-1} = \hat{B},$$

$$(A\hat{B})^TM = (\hat{Q}JP^{-1}PJ^T\hat{Q}^{-1})^{-1}(\hat{Q}\hat{Q}^T)^{-1} = \hat{Q}^{-T}JJ^T\hat{Q}^T(\hat{Q}^{-T}\hat{Q}^{-1})$$

$$= \hat{Q}^{-T}JJ^T\hat{Q}^{-1} = (\hat{Q}^{-T}\hat{Q}^{-1})(\hat{Q}JJ^T\hat{Q}^{-1}) = M(A\hat{B}),$$

$$(\hat{B}A)^TN = (PJ^T\hat{Q}^{-1}\hat{Q}JP^{-1})^T(PP^T)^{-1} = (P^{-T}J^TJP^T)(P^{-T}P^{-1})$$

$$= P^{-T}J^TJP^{-1} = (PP^T)^{-1}(PJ^TJP^{-1}) = N(\hat{B}A)^T.$$

According to C. R. Rao and S. K. Mitra [13] (pp. 52), $\hat{B}$ is the $M-N$ generalized inverse of A. Q.E.D.

PROPOSITION 4.10. *$\tilde{x} = \hat{B}\tilde{f}$ is the minimum $N-$norm $M-$ least square solution of the $n \times n$ system of equations $A\tilde{\phi} = \tilde{f}$. That is, the $\tilde{x}$ satisfies*

$$\| \tilde{x} \|_N = \min \| \tilde{y} \|_N,$$

where $\tilde{y}$ makes

$$\| A\tilde{y} - \tilde{f} \|_M = \min_{\tilde{z} \in R^n} \| A\tilde{z} - \tilde{f} \|_M.$$

The $\| \tilde{x} \|_M$ and $\| \tilde{x} \|_n$ are defined as $\sqrt{\tilde{x}^T M \tilde{x}}$ and $\sqrt{\tilde{x}^T N \tilde{x}}$ respectively.

5. Numerical example

We have considered a numerical example for the equation (1.1) with $a = 0$, $b = 1$, $k = 0$ and $f(x) = \frac{1}{(x-c)^2+c^2}$. As c is quite small (e.g. $c = 0.01$), $f(x)$ has a peak value 10^4 at $x = c$. The exact solution

$$\phi(x) = \frac{(1+4c^4)^{\frac{1}{4}}}{c} \frac{(x-c)\cos\frac{\theta}{2} + c\sin\frac{\theta}{2}}{(x-c)^2+c^2},$$

where

$$\tan\theta = 2c^2.$$

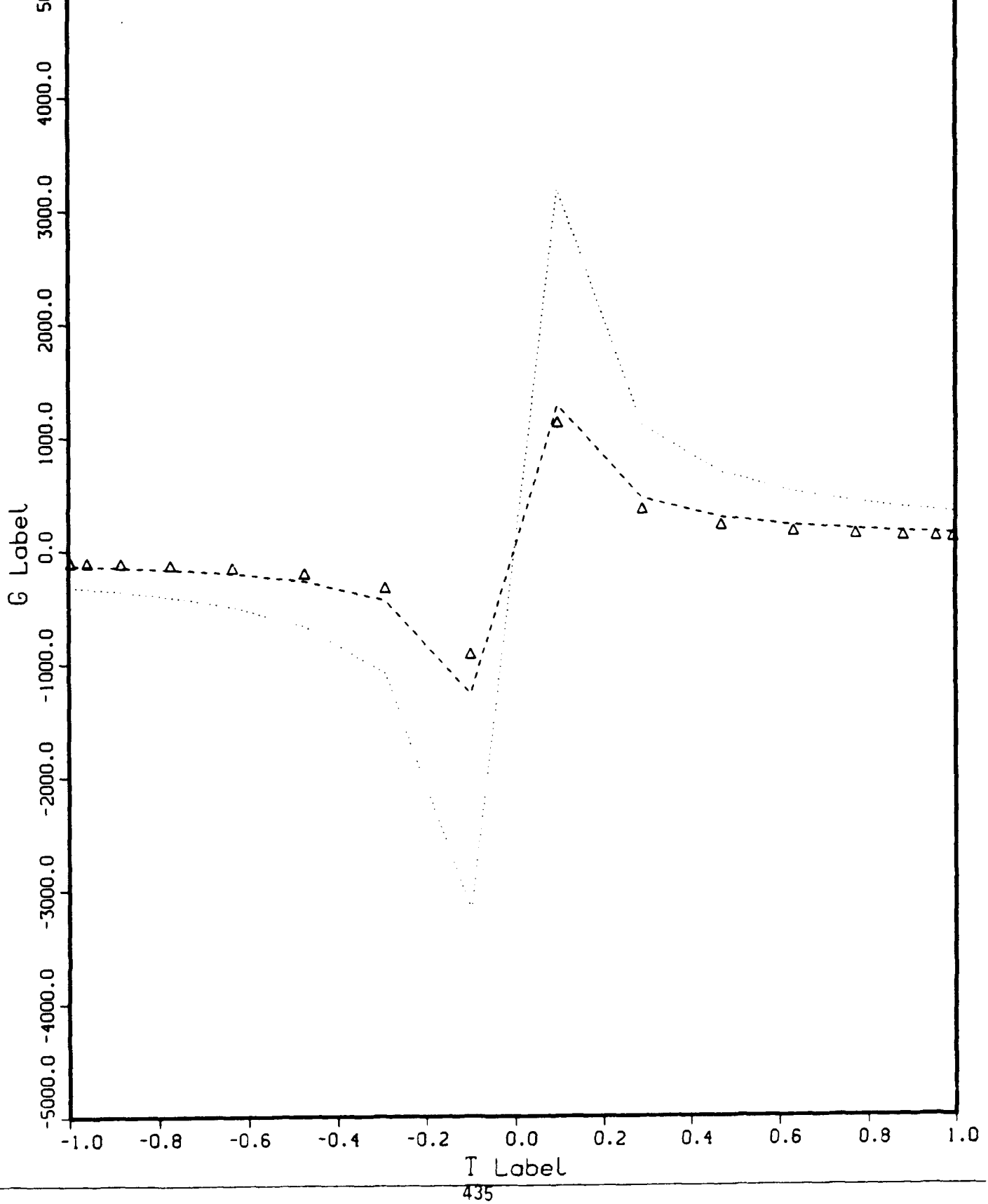
It has also a pair of maximum and minimum at the vicinity of 0. Since the unusual behavior, the values of $f(x)$ at Chebyshev collocation points $\{s_j\}$ can not represent the feature of $f(x)$ and the classical Gauss-Chebyshev scheme failes to provide a satisfactory approximate solution. The curve (dot line) in the following figure shows unacceptable errors.

Another way is to choose collocation points flexibly. For example, if we take $x = -0.01$ as collocation point instead of $x = s_{fracn2} = 0$, we obtain a general quadrature colloaction scheme. The curve (dash line) in the figure shows that it gives a much better approximation to the exact solution (points labeled by small triangles in the figure) than the Gauss-Chebyshev scheme does.

As n increases, Chebyshev nodes become dense. Since the Lebesgue constant of the interpolation at Chebyshev nodes are bounded, a more satisfying approximation may be obtained. However, under the restriction of the number n, the flexible choice of collocation nodes can lead to a better result at a lower computational cost.

References

1. N. I. Muskhelishivili, "Singular Integral Equations," Noordhoff, Groninggen, 1953.
2. F. D. Gakhov, "Boundary Value Problems," Pergammon, London, 1966.
3. F. Erdogan, SIAM J. Appl. Math. **17**, 1041-1052.
4. F. Erdogan and G. D. Gupta, Quart. Appl. Math. **30** (1972), 525-534.
5. F. Erdogan, G. D. Gupta and T. S. Cook, Mechanics of Fracture **1** (1973), 368-425.
6. P. S. Theocaris and N. I. Ioakimidis, Quart. Appl. Math. **35** (1977), 173-183.
7. A. Gerasoulis and R. P. Srivastav, Int. J. Computer Mathematics **12** (12), 59-75.
8. R. P. Srivastav and Erica Jen, Applicable Analysis **14** (1983), 275-285.
9. R. P. Srivastav, IMA. J. Numerical Analysis **3** (1983), 305-318.
10. R. P. Srivastav and Fenggang Zhang, *On numerical Methods for Solving Cauchy Singular Equations*, Appl. Math. Lett. **2**, 267-231.
11. G. Tsamysphyros and P. S. Theocaris, Fracture Mechanics (1981),.
12. S. Welstead, "Ph.D. Thesis" (1982), Purdue University.
13. C. R. Rao and S. K. Mitra, "Generalized Inverse of Matrices and it's Applications.," John Willey and Sons Inc., New York, London, Sydney, Toronto, 1971.


The Curve of g(t) with General nodes ym=-0.01
5000.0
4000.0
3000.0
2000.0
1000.0
0.0
-1000.0
-2000.0
-3000.0
-4000.0
-5000.0
G Label
-1.0
-0.8
-0.6
-0.4
-0.2
0.0
0.2
0.4
0.6
0.8
1.0
T Label

On a Hyperbolic Tangent Quadrature Rule for Solving Singular Integral Equations with Hadamard Finite Part Integrals

FENGGANG ZHANG

Department of Applied Mathematics and Statistics
SUNY at Stony Brook.

Abstract. Stenger's formula is adapted for singular integrals defined as Hadamard finite part. Convergence of the revised quadrature rule is studied. A scheme, for solving singular integral equations with Hadamard finite part integrals, is proposed based on the revised hyperbolic tangent quadrature rule. The integral equation is reduced to a system of linear equations, by taking the same points as quadrature nodes and collocation points. The coefficient matrix of the system is shown to be nonsingular.

1. INTRODUCTION

Many physical problems from diverse areas such as aerodynamics, hydrodynamics and elasticity may be reduced to integral equations of the form

$$\int_{-1}^{1} \frac{f(t)}{(t-x)^2}dt + \int_{-1}^{1} k(x,t)f(t)dt = F(x), -1 < x < 1, \tag{1.1}$$

The first term denotes the Hadamard finite part integral (J. Hadamard [7]).

$$\int_{-1}^{1} \frac{f(t)}{(t-x)^2}dt = F.P.\lim_{\epsilon\to 0}(\int_{x+\epsilon}^{1} + \int_{-1}^{x-\epsilon})\frac{f(t)dt}{(t-x)^2}, \tag{1.2}$$

where the initial "F.P." means extracting only the terms which remain finite as ϵ goes to zero. The second term on the left side of (1.1) is the Fredholm integral.

The Hadamard finite part integral also can be regarded as a distribution defined on $C_0^\infty(-\infty,\infty)$ (A. Zemanian [8]), where $C_0^\infty(-\infty,\infty)$ is a set of infinitely differentiable functions with compact support. Therefore, we have

$$\int_{-1}^{1} \frac{f(t)}{(t-x)^2}dt = \frac{d}{dx}\int_{-1}^{1} \frac{f(t)}{t-x}dt. \tag{1.3}$$

The right side of (1.3) is the distributional derivative of the Cauchy principal value integral, which is often regular.

The numerical methods for solving (1.1) have been discussed by Kutt [1], Ioakimidis [2], Kaya [3] and Golberg [4]. In [5] Venturino has proposed a technique for solving Cauchy singular integral equations. Venturino's scheme utilized the quadrature formulas which were proposed by F. Stenger[6] and were called the "hyperbolic tangent quadrature rule". Although there is a close relation between the two types of singular equations, the supersingularity of the Hadamard finite part integral prevents direct application of the original hyperbolic tangent quadrature rule. Stenger's formula in favorable

This research was completed under the guidance of Professor R. Srivastav. The author thanks him for his significant advice and very helpful comments. Research supported in part by NSF Grant No.DMS-8901800 and US Army Research Office Grant No.DAAL03-90-G-0019

Typeset by $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX

cases has exponential convergence. It seems to be of interest to modify it so that it can be used for supersingular integrals.

In this paper, we first generalize the trapezoid quadrature rule to the Hadamard finite part integral on $(-\infty,\infty)$. Then, we propose the revised scheme of the hyperbolic tangent quadrature rule and estimate the error bound of this scheme. In section 4, we introduce the concept of the generalized Cauchy matrix and prove the coefficient matrix of the discrete system is invertible. Finally, we give two numerical examples to check how well the revised scheme works

For the sake of brevity, we only discuss the canonical equation of the form

$$\int_{-1}^{1} \frac{f(t)}{(t-x)^2}dt = F(x), -1 < x < 1. \tag{1.4}$$

2. Hadamard Finite Part Integral on $(-\infty,\infty)$ and the Revised Trapezoid Quadrature Rule

According to [6], the trapezoid quadrature rule can be introduced by using the Whittaker Cardinal function.

Let Ω_d be a strip in the complex plane:

$$\Omega_d := \{x+iy, \mid y \mid < d\}, \tag{2.1}$$

where $d > 0$.

Let B_d be a family of functions $f(z)$, that are analytic in Ω_d, satisfying

$$\int_{-d}^{d} \mid f(x+iy) \mid dy \to 0, \text{ as } x \to \pm\infty \tag{2.2}$$

and

$$N(f,\Omega_d) := \lim_{y\to d^-} \int_{-\infty}^{\infty} (\mid f(x+iy) \mid + \mid f(x-iy) \mid)dx < \infty. \tag{2.3}$$

Denote

$$\eta(f) := \int_{-\infty}^{\infty} f(t)dt - h\sum_{k=-\infty}^{\infty} f(kh), \tag{2.4}$$

$$\eta_M(f) := \int_{-\infty}^{\infty} f(t)dt - h\sum_{k=-M}^{M} f(kh). \tag{2.5}$$

In [6], it is proved that if $f \in B_d$, then

$$\| \eta(f) \|_\infty \leq \frac{e^{-\frac{\pi d}{h}}}{2\sinh\frac{\pi d}{h}} N(f,\Omega_d). \tag{2.6}$$

If also

$$\mid f(x) \mid \leq c_1 e^{-\alpha|x|}, \tag{2.7}$$

then

$$\mid \eta_M(f) \mid \leq c_2 e^{-\sqrt{2\pi d\alpha M}} \tag{2.8}$$

by taking $h = \sqrt{\frac{2\pi d}{\alpha M}}$, where $c_1, c_2, \cdots$, are constants independent of M.

Now, for the Hadamard finite part integral on $(-\infty,\infty)$,

$$I(f) := \int_{-\infty}^{\infty} \frac{f(t)}{(t-s)^2}dt, \tag{2.9}$$

there is an associated infinite series, whose partial sums may be used for numerical integration. The error of this approximation is easy to estimate. The main result in this section is

THEOREM 2.1. *If* $f(x) \in B_d$, $| f(x) | \leq ce^{-\alpha|x|}$, $\| f'' \|_\infty \leq c_2$ *and*

$$\sigma(f) := h \sum_{k=-\infty}^{\infty} \frac{f(x+kh)}{(k-0.5)(k+0.5)h^2}, \tag{2.10}$$

$$\epsilon(f) := I(f) - \sigma(f). \tag{2.11}$$

$$\epsilon_M(f) := I(f) - h \sum_{k=-M}^{M} \frac{f(x+kh)}{(k-0.5)(k+0.5)h^2}, \tag{2.12}$$

then

$$| \epsilon(f) | \leq \frac{c_1 e^{-\frac{\pi d}{h}}}{\alpha \sinh \frac{\pi d}{h}} + c_2 h,$$

$$| \epsilon_M(f) | \leq c_3 (e^{-\alpha M h} + h)$$

by taking $M = \frac{2\pi d}{\alpha h^2}$, *where* c, c_1, c_2 *and* c_3 *are constants independent of* M *and* h,

$$c_1 = \frac{4(1+\alpha+\alpha d^2)}{d^2}, \tag{2.13}$$

$$c_3 = \max\left(\frac{4c_1}{\alpha}, c_2 + \frac{\alpha c}{4\pi d}\right). \tag{2.14}$$

We prove the theorem through several lemmas.

LEMMA 2.2. *If* $f(t) \in B_d$ *and* $\| f \|_\infty \leq c$, *then the Hadamard finite part integral* $I(f)$ *can be represented as*

$$I_1 = \int_{-\infty}^{\infty} g(t)dt$$

where

$$g(t) = \begin{cases} \dfrac{f(x+t)+f(x-t)-2f(x)}{2t^2}, & \text{as } t \neq 0 \\ \dfrac{f''(x)}{2}, & \text{as } t = 0. \end{cases}$$

Here $g(t) \in B_d$ *and*

$$N(g, \Omega_d) \leq \frac{2}{d^2} N(f, \Omega_d) + 8\left(\frac{1}{d^2} + 1\right) \| f \|_\infty .$$

Proof.

$$I(f) = \int_x^\infty \frac{f(t)}{(t-x)^2} dt + \int_{-\infty}^x \frac{f(t)}{(t-x)^2} dt$$

$$= \int_0^\infty \frac{f(x+t)+f(x-t)}{t^2} dt$$

$$= F.P. \lim_{\epsilon \to 0} \left(\int_\epsilon^\infty \frac{f(x+t)+f(x-t)-2f(x)}{t^2} dt + \int_\epsilon^\infty \frac{2f(x)}{t^2} dt \right)$$

$$= \int_0^\infty 2g(t)dt + f(x)F.P.\lim_{\epsilon\to 0}\int_\epsilon^\infty \frac{1}{t^2}dt$$

$$= I_1,$$

because $f(t)$ is analytic, 0 is no longer the singular point of $g(x)$, $g(t)$ is even on $(-\infty,\infty)$ and the second term is equal to zero. Furthermore,

$$N(g,\Omega_d) = \lim_{y\to d^-}\int_{-\infty}^{\infty}(\mid g(t+iy)\mid + \mid g(t-iy)\mid)dt$$

$$= \lim_{y\to d^-}\int_{-\infty}^{\infty}(\frac{\mid f(x+t+iy)+f(x-t-iy)-2f(x)\mid}{\mid t+iy\mid^2}$$

$$+\frac{\mid f(x+t-iy)+f(x-t+iy)-2f(x)\mid}{\mid t-iy\mid^2})dt$$

$$\le \lim_{y\to d^-}(2\int_{-\infty}^{\infty}\frac{\mid f(x+t+iy)\mid + \mid f(x-t-iy)\mid}{\mid y\mid^2}dt$$

$$+4f(x)(\int_{|t|\le 1}\frac{1}{\mid y\mid^2}dt + \int_{|t|>1}\frac{1}{t^2}dt)$$

$$\le \frac{2}{d^2}N(f,\Omega_d) + 4\parallel f\parallel_\infty(\frac{2}{d^2}+2). \qquad Q.E.D.$$

LEMMA 2.3. *If $f\in B_d$ and $\mid f(x)\mid \le ce^{-\alpha|x|}$,*

$$\epsilon(g) := I_1 - \frac{h}{2}(\sum_{k=-\infty,k\neq 0}^{\infty}\frac{f(x+kh)+f(x-kh)-2f(x)}{k^2h^2} + f''(x)),$$

then

$$\mid \epsilon(g)\mid \le \frac{c_1 e^{-\frac{\pi d}{h}}}{\alpha\sinh\frac{\pi d}{h}},$$

where c_1 is defined in (2.13).

Proof. Since

$$N(f,\Omega_d) \le 2c\int_{-\infty}^{\infty}e^{-\alpha|t|}dt = \frac{4c}{\alpha},$$

by using (2.6) and the Lemma 2.2, we obtain

$$\mid \epsilon(g)\mid \le \frac{e^{-\frac{\pi d}{h}}}{2\sinh\frac{\pi d}{h}}N(g,\Omega_d)$$

$$\le \frac{e^{-\frac{\pi d}{h}}}{2\sinh\frac{\pi d}{h}}(\frac{2}{d^2}N(f,\Omega_d) + 8c(\frac{1}{d^2}+1))$$

$$\le \frac{4ce^{-\frac{\pi d}{h}}}{\alpha d^2\sinh\frac{\pi d}{h}}(1+\alpha+\alpha d^2). \qquad Q.E.D.$$

LAMMA 2.4. *If* $f \in B_d$, $\quad | f''(x) | \leq c_2$ *and*

$$\epsilon^*(g) := I_1 - \frac{h}{2} \sum_{k=-\infty}^{\infty} \frac{f(x+kh) + f(x-kh) - 2f(x)}{(k-0.5)h(k+0.5)h},$$

then

$$| \epsilon^*(g) | \leq | \epsilon(g) | + c_2 h.$$

Proof. Using the mean value theorem and the identity

$$\sum_{k=1}^{\infty} \frac{1}{k^2 - 0.25} = 2,$$

we find that

$$| \epsilon^*(g) - \epsilon(g) | = | h \sum_{k=1}^{\infty} \frac{f(x+kh) + f(x-kh) - 2f(x)}{h^2} (\frac{1}{(k-0.5)(k+0.5)} - \frac{1}{k^2}) - \frac{h}{2} f''(x) |$$

$$\leq c_2 h \sum_{k=1}^{\infty} k^2 (\frac{1}{k^2 - 0.25} - \frac{1}{k^2}) + \frac{h}{2} c_2$$

$$= c_2 h (\frac{1}{4} \sum_{k=1}^{\infty} \frac{1}{k^2 - 0.25} + 0.5) = c_2 h. \qquad Q.E.D.$$

The proof of the Theorem 2.1:
By the Lemma 2.2, the Lemma 2.3 ,the Lemma 2.4 and the identity

$$\sum_{k=-\infty}^{\infty} \frac{1}{(k-0.5)(k+0.5)} = 0,$$

we find that

$$\epsilon(f) = I(f) - \sigma(f) = I_1 - \sigma(f)$$

$$= \epsilon^*(g) + \frac{h}{2} \sum_{k=-\infty}^{\infty} \frac{f(x+kh) + f(x-kh) - 2f(x)}{(x-0.5h)(x+0.5h)} - \sigma(f)$$

$$= \epsilon^*(g) - \frac{1}{h} \sum_{k=-\infty}^{\infty} \frac{f(x)}{(k-0.5h)(k+0.5h)} = \epsilon^*(g)$$

and

$$| \epsilon_M(f) | \leq | \epsilon(f) | + \frac{h}{2} \sum_{|k|>M} \frac{f(x+kh)}{(k-0.5)(k+0.5)h^2}$$

$$\leq | \epsilon(f) | + \frac{c}{4(M+0.5)h}$$

$$\leq \frac{c_1 e^{-\frac{\pi d}{h}}}{\alpha \sinh \frac{\pi d}{h}} + c_2 h + \frac{c}{(2M+1)h}.$$

Since $M = \frac{2\pi d}{\alpha h^2}$, for M large

$$\frac{1}{\sinh \frac{\pi d}{h}} \leq 4 e^{-\frac{\pi d}{h}},$$

we have

$$| \epsilon_M(f) | \leq \frac{4c_1}{\alpha} e^{-\alpha M h} + (c_2 + \frac{c\alpha}{4\pi d}) h$$

$$\leq c_3 (e^{-\alpha M h} + h). \qquad Q.E.D.$$

3. Hadamard Finite Part Integral on the Interval $(-1,1)$ and the Revised Hyperbolic Tangent Quadrature Rule

In [6] the hyperbolic tangent quadrature rule is introduced to approximate the regular and the Cauchy principal value integrals on $(-1,1)$. Let us recall the main results of [6] before we discuss its modification.

Let Ω be the open unit disk in the complex plane:

$$\Omega := \{z \mid \mid z \mid < 1\}$$

and B_d be a family of functions $f(z)$, which are analytic in Ω and satisfy

$$N(f,\Omega) := \lim_{r\to 1^-} \int_0^{2\pi} \mid f(re^{i\theta}) \mid d\theta < \infty. \tag{3.1}$$

If $f(x) \in B_d$ and

$$x_k = \tanh\frac{kh}{2} = \frac{e^{kh}-1}{e^{kh}+1}, \tag{3.2}$$

then

$$\mid \eta(f) \mid := \mid \int_{-1}^{1} f(x)dx - \frac{h}{2}\sum_{k=-\infty}^{\infty}(1-x_k^2)f(x_k) \mid \leq \frac{e^{-\frac{\pi^2}{2h}}}{2\sinh\frac{\pi^2}{2h}} N(f,\Omega); \tag{3.3}$$

also, if $\mid f(x) \mid \leq c(1-x^2)^{\alpha-1}$ on $(-1,1)$ and $h = \frac{\pi}{\sqrt{\alpha M}}$, then

$$\mid \eta_M(f) \mid := \mid \int_{-1}^{1} f(x)dx - \frac{h}{2}\sum_{k=-M}^{M}(1-x_k^2)f(x_k) \mid \leq \frac{c}{\alpha}e^{-\sqrt{\alpha M}}. \tag{3.4}$$

Let

$$I(f) := \int_{-1}^{1} \frac{f(t)}{(t-x)^2}dt, \qquad -1 < x < 1, \tag{3.5}$$

and

$$\sigma(f) := \frac{h}{2(1-x^2)}\sum_{k=-\infty}^{\infty} \frac{f(\frac{x_k+x}{1+x_kx})}{s_{k-0.5}s_{k+0.5}}, \tag{3.6}$$

where

$$s_k = \sinh\frac{kh}{2}. \tag{3.7}$$

Similar to the section 2, let

$$\sigma_M(f) := \frac{h}{2(1-x^2)}\sum_{l=-M}^{M} \frac{f(\frac{x_k+x}{1+x_kx})}{s_{k-0.5}s_{k+0.5}}, \tag{3.8}$$

$$\epsilon(f) := I(f) - \sigma(f),$$

$$\epsilon_M(f) := I(f) - \sigma_M(f).$$

A fractional linear transform of the complex variable defined by

$$t = \frac{s+x}{1+sx} \qquad \text{or } s = \frac{t-x}{1-tx}. \tag{3.9}$$

is introduced. It is a one-to-one analytic mapping from the open unit disk onto itself and leaves $-1,1$ unchanged. We obtain

$$I(f) = \int_{-1}^{1} \frac{f(\frac{x+s}{1+sx})}{s^2(1-x^2)} ds. \tag{3.10}$$

Let

$$g(s,x) = \begin{cases} \dfrac{f(\frac{s+x}{1+sx}) + f(\frac{-s+x}{1-sx}) - 2f(x)}{2s^2} & \text{if } s \neq 0 \\ \dfrac{(1-x^2)f''(x)}{2} & \text{if } s = 0. \end{cases} \tag{3.11}$$

Our main result in this section is

THEOREM 3.1. *If $f(x) \in B_d$, $\| f \|_\infty \leq c_0$ and*

$$| g(s,x)(1-s^2) | \leq c_2(x), \text{ for } -1 < x < 1, \quad -1 < s < 1, \tag{3.12}$$

then

$$| \epsilon(f) | \leq c_1(x)(e^{-\frac{\pi^2}{h}} + h^2)$$

and by taking $M = \frac{\pi^2}{h^2}$, we have

$$| \epsilon_M(f) | \leq c_3(x)(2e^{-Mh} + h^2).$$

where $c_1(x)$, $c_2(x)$ and $c_3(x)$ are functions of x, but independent on s, h and M,

$$c_1(x) = \max\left(2\pi c_0\left(\frac{1}{d^2(x)} + \frac{1}{1-x^2}\right), \left(\frac{1}{8}c_2(x) + \frac{1}{12}c_0\right)\frac{1}{1-x^2}\right), \tag{3.13}$$

$$c_3(x) = \max\left(c_1(x), \frac{2c_0}{1-x^2}\right) \tag{3.14}$$

and

$$d(x) = \min(1-x, 1+x).$$

We prove this theorem through several lemmas.

LEMMA3.2. *If $f(t) \in B_d$, then the Hadamard finite part integral $I(f)$ can be represented as the sum of I_1 and I_2,*

$$I_1 = \frac{1}{1-x^2}\int_{-1}^{1} g(s,x)ds, \tag{3.15}$$

$$I_2 = \frac{-2f(x)}{1-x^2}. \tag{3.16}$$

Here I_1 is a regular integral, $g(s,x) \in B_d$ and

$$N(g(s,x),\Omega_d) \le \frac{2(1-x^2)}{d^2(x)}N(f,\Omega_d) + 4\pi \mid f(x) \mid .$$

Proof. From the [8] (page 62), the Hadamard finite part integral can be regarded as a pseudofunction or a functional defined on the test space $C_0^\infty(-\infty,\infty)$, that is, for any function $\phi(s) \in C_0^\infty(-\infty,\infty)$,

$$\int_0^1 \frac{\phi(s)}{s^2}ds := \int_0^1 \frac{\int_0^s \phi''(\tau)(s-\tau)d\tau}{s^2}ds - \phi(0)$$

$$:= \int_0^1 \frac{\phi(s)-\phi(0)-\phi'(0)s}{s^2}ds - \phi(0)$$

and

$$\int_{-1}^1 \frac{\phi(s)}{s^2}ds = \int_0^1 \frac{\phi(s)}{s^2}ds + \int_{-1}^0 \frac{\phi(s)}{s^2}ds$$

$$:= \int_0^1 \frac{\phi(s)-\phi(0)-s\phi'(0)}{s^2}ds - \phi(0) + \int_0^1 \frac{\phi(-s)-\phi(0)+s\phi'(0)}{s^2}ds - \phi(0)$$

$$= \int_{-1}^1 \frac{\phi(s)+\phi(-s)-2\phi(0)}{2s^2}ds - 2\phi(0).$$

This functional can be generalized to any continuous function with finite support in $(-\infty,\infty)$. Particularly,

$$\int_{-1}^1 \frac{1}{s^2} = -2.$$

In fact, let

$$f(s) = \begin{cases} 1, \text{ if } \mid s \mid < 1+\alpha, \\ 0, \text{ if } \mid s \mid \ge 1+2\alpha, \\ \frac{s+1+2\alpha}{\alpha}, \text{ if } -1-2\alpha < s \le -1-\alpha, \\ -\frac{s-1-2\alpha}{\alpha}, \text{ if } 1+\alpha \le s < 1+2\alpha. \end{cases}$$

We can construct a sequence $\{\phi_n(s)\}$, $\phi_n(s) \in C_0^\infty(-\infty,\infty)$ and

$$\phi_n(s) \to 1, \qquad \phi_n'(s) \to 0, \qquad \phi_n''(s) \to 0 \text{ uniformly on } [-1,1] \text{ as } n \to \infty,$$

where

$$\phi_n(s) = \int_{-\infty}^{\infty} f(\tau)\gamma_n(s-\tau)d\tau,$$

$$\gamma_n(t) = \frac{\zeta(nt)}{\int_{-\infty}^{\infty} \zeta(nt)dt},$$

and

$$\zeta(t) = \begin{cases} 0, \text{ if } \mid t \mid \geq 1, \\ e^{\frac{1}{t^2-1}}, \text{ if } \mid t \mid < 1. \end{cases}$$

Therefore,

$$\int_{-\infty}^{\infty} \frac{1}{s^2} ds = \lim_{n\to\infty} \int_{-1}^{1} \frac{\phi_n(s)}{s^2} ds$$

$$= \lim_{n\to\infty} \left(\int_{-1}^{1} \frac{\int_0^s \phi_n''(\tau)(s-\tau)d\tau}{s^2} ds - 2\phi_n(0) \right) = -2.$$

Let $\phi(s) = f(\frac{x+s}{1+sx})$, we have

$$I(f) = \frac{1}{1-x^2} \int_{-1}^{1} \frac{f(\frac{s+x}{1+sx})}{s^2} ds \tag{3.17}$$

$$= \frac{1}{1-x^2} \int_{-1}^{1} \frac{f(\frac{s+x}{1+sx}) + f(\frac{-s+x}{1-sx})ds - 2f(x)}{2s^2} ds - 2\frac{f(x)}{1-x^2}$$

$$= I_1 + I_2.$$

Since (3.4) is a fractional linear transform from the open unit disk onto itself, $f(\frac{s+x}{1+sx})$ is analytic with respect to s in the $\mid s \mid < 1$ and the $g(s,x)$ is too. Furthermore,

$$N(g(s,x), \Omega_d) = \lim_{r\to 1^-} \int_0^{2\pi} \mid g(re^{i\theta}, x) \mid d\theta \tag{3.18}$$

$$\leq \lim_{r\to 1^-} \left(\int_0^{2\pi} \mid f(\frac{re^{i\theta}+x}{1+xr^{i\theta}}) \mid d\theta + \int_0^{2\pi} \mid f(\frac{-re^{i\theta}+x}{1-xr^{i\theta}}) \mid d\theta \right) + 4\pi \mid f(x) \mid,$$

where $s = re^{i\theta}$. Let $t = \rho e^{i\beta}$, we have

$$ds = \frac{1-x^2}{(1-tx)^2} dt,$$

$$d\theta = \frac{t(1-x^2)}{(t-x)(1-tx)} d\beta$$

and

$$\lim_{r\to 1^-} \int_0^{2\pi} \mid f(\frac{re^{i\theta}+x}{1+xre^{i\theta}}) \mid d\theta = \lim_{r\to 1^-} \int_0^{2\pi} \mid f(\rho e^{i\beta}) \mid \mid \frac{(1-x^2)\rho e^{i\beta}}{(\rho e^{i\beta}-x)(1-x\rho e^{i\beta})} d\beta \mid$$

$$\leq \frac{1-x^2}{d^2(x)} N(f, \Omega_d).$$

We can obtain the same estimation for $\lim_{r\to 1^-} \int_0^{2\pi} \mid f(\frac{-re^{i\theta}+x}{1-xre^{i\theta}}) \mid d\theta$. Therefore,

$$N(g(s,x), \Omega_d) \leq \frac{2(1-x^2)}{d^2(x)} N(f, \Omega_d) + 4\pi \mid f(x) \mid . \qquad Q.E.D.$$

LEMMA 3.3. *If* $| g(s,x)(1-s^2) | \leq c_2(x)$ *for* $-1 < x < 1$ *and*

$$\epsilon(g) = \int_{-1}^{1} g(s,x)ds - \frac{h}{2}\sum_{k=-\infty}^{\infty} \frac{g(x_k,x)}{s_{k-0.5}s_{k+0.5}},$$

then

$$| \epsilon(g) | \leq \frac{e^{-\frac{\pi^2}{2h}}}{2\sinh\frac{\pi^2}{2h}} N(g(s,x),\Omega_d) + \frac{1}{8}c_2(x)h^2.$$

Proof. By (3.3), we have

$$| \int_{-1}^{1} g(s,x)ds - \frac{h}{2}\sum_{k=-\infty}^{\infty} g(x_k,x)(1-x_k^2) | \leq \frac{e^{-\frac{\pi^2}{2h}}}{2\sinh\frac{\pi^2}{2h}} N(g(s,x),\Omega_d).$$

By using the identity

$$\sum_{k=-\infty}^{\infty} \frac{1}{s_{k-0.5}s_{k+0.5}} = -\frac{2}{\sinh\frac{h}{2}},$$

we obtain

$$\sum_{k=-\infty}^{\infty} (1 - \frac{1}{(1-x_k^2)s_{k-0.5}s_{k+0.5}}) = \frac{1-\cosh\frac{h}{2}}{2}\sum_{k=-\infty}^{\infty} \frac{1}{s_{k-0.5}s_{k+0.5}}$$

$$= (-\sinh^2\frac{h}{4})(\frac{-2}{\sinh\frac{h}{2}}) = \tanh\frac{h}{4} \leq \frac{h}{4}.$$

Therefore,

$$| \frac{h}{2}\sum_{k=-\infty}^{\infty} g(x_k,x)(1-x_k^2) - \frac{h}{2}\sum_{k=-\infty}^{\infty} \frac{g(x_k,x)}{s_{k-0.5}s_{k+0.5}} | \leq \frac{1}{8}c_2(x)h^2.$$

By using the triangle inequality, we can obtain the estimation of $\epsilon(g)$. Q.E.D.

Proof of the Theorem 3.1

Since $| f(x) | \leq c_0$, by using (3.1) and the Lemma 3.2, we have

$$N(f,\Omega_d) \leq 2\pi c_0$$

and

$$N(g,\Omega_d) \leq (\frac{1-x^2}{d^2(x)} + 1)4\pi c_0.$$

Now, writing the $\sigma(f)$ in the sum of σ_1 and σ_2,

$$\sigma_1 = \frac{h}{2(1-x^2)}\sum_{k=-\infty}^{\infty} \frac{f(\frac{x_k+x}{1+x_kx}) + f(\frac{-x_k+x}{1-x_kx}) - 2f(x)}{2s_{k-0.5}s_{k+0.5}}$$

$$= \frac{h}{2(1-x^2)}\sum_{k=-\infty}^{\infty} \frac{g(x_k,x)}{s_{k-0.5}s_{k+0.5}},$$

$$\sigma_2 = \frac{h}{2(1-x^2)}f(x)\sum_{k=-\infty}^{\infty} \frac{1}{s_{k-0.5}s_{k+0.5}}$$

using the Lemma 3.2,

$$\epsilon(f) = \frac{1}{1-x^2}\epsilon(g) - \frac{f(x)}{1-x^2}(2 + \frac{h}{2}\sum_{k=-\infty}^{\infty}\frac{1}{s_{k-0.5}s_{k+0.5}}),$$

and using the estimation

$$\mid 2 + \frac{h}{2}\sum_{k=-\infty}^{\infty}\frac{1}{s_{k-0.5}s_{k+0.5}} \mid = \mid 2 - \frac{h}{\sinh\frac{h}{2}} \mid \leq \frac{h^2}{12},$$

we have

$$\mid \epsilon(f) \mid \leq \frac{1}{1-x^2} \mid \epsilon(g) \mid + \frac{c_0}{1-x^2}\frac{h^2}{12}$$

$$\leq \frac{1}{1-x^2}((\frac{1}{1-x^2}+1)4\pi c_0\frac{e^{-\frac{\pi^2}{2h}}}{2\sinh\frac{\pi^2}{2h}} + \frac{1}{8}c_2(x)h^2) + \frac{c_0}{1-x^2}\frac{1}{12}h^2$$

$$\leq c_1(x)(e^{-\frac{\pi^2}{h}} + h^2),$$

where $c_1(x)$ is defined in (3.13). Furthermore

$$\mid \epsilon_M(f) \mid \leq \mid \epsilon(f) \mid + \frac{h}{2(1-x^2)}\sum_{|k|>M}\frac{f(\frac{x_k+x}{1+x_kx})}{s_{k-0.5}s_{k+0.5}}$$

$$\leq \mid \epsilon(f) \mid + \frac{c_0h}{2(1-x^2)}\frac{2}{\sinh\frac{h}{2}}(\frac{1}{\tanh(M+0.5)\frac{h}{2}} - 1)$$

$$\leq c_3(x)(e^{-\frac{\pi^2}{h}} + h^2 + e^{-Mh}),$$

where $c_3(x)$ is defined in (3.14). By taking $M = \frac{\pi^2}{h^2}$,

$$\mid \epsilon_M(f) \mid \leq c_3(x)(2e^{-Mh} + h^2). \qquad Q.E.D.$$

Note, the upper bound $c_2(x)$ in the (3.12) is dependent on the behavior of the function $f(t)$. $c_2(x)$ may become unbounded when x goes to ± 1.

4. The Solvability of the Discrete System of Equations

For the finite part integral equation (1.1), if $k(x,t) = 0$, the discrete functional equation given by the revised hyperbolic tangent quadrature rule will be

$$\frac{h}{2(1-x^2)}\sum_{k=-M}^{M}\frac{f(\frac{x_k+x}{1+x_kx})}{s_{k-0.5}s_{k+0.5}} = F(x), \tag{4.1}$$

where $x_k = \tanh\frac{kh}{2}$.

We can choose $x = x_j$ as the collocation points. Since

$$\frac{x_k + x_j}{1 + x_kx_j} = x_{k+j}, \tag{4.2}$$

the collocation points are the same as the quadrature nodes. Let $l = k + j$, we have

$$\frac{h}{2}\sum_{l=-M+j}^{M+j}\frac{f(x_l)}{s_{l-j-0.5}s_{l-j+0.5}} = (1-x_j^2)F(x_j),\ \text{for } j=-M,-M+1,\cdots,-1,0,1,\cdots,M. \tag{4.3}$$

There are $2M+1$ equations. In order to keep the number of the unknowns $f(x_l)$ the same as the number of equations, we prefer to ignore those quadrature nodes which are too close to the ends -1 or 1. It is reasonable to take the range of x_l as from x_{-M} to x_M and we have

$$\frac{h}{2}\sum_{l=-M}^{M}\frac{f(x_l)}{s_{l-j-0.5}s_{l-j+0.5}} = (1-x_j^2)F(x_j),\ \text{for } j=-M,\cdots,-1,0,1,\cdots,M. \tag{4.4}$$

Let

$$\tilde{\phi} = \frac{h}{2}(f(x_{-M}), f(x_{-M+1}),\cdots,f(x_M))^T$$

and

$$\tilde{F} = ((1-x_{-M}^2)F(x_{-M}),(1-x_{-M+1}^2)F(x_{-M+1}),\cdots,(1-x_M^2)F(x_M))^T.$$

The discrete linear system of equations is

$$A\tilde{\phi} = \tilde{F}, \tag{4.5}$$

where

$$A = (a_{jl})_{(2M+1)\times(2M+1)},$$

$$j=-M,-M+1,\cdots,-1,0,1,\cdots,M, \qquad l=-M,-M+1,\cdots,-1,0,1,\cdots,M,$$

$$a_{jl} = \frac{1}{s_{l-j-0.5}s_{l-j+0.5}}. \tag{4.6}$$

The following steps are concentrated on proving that A is invertible.

Since

$$a_{jl} = \frac{4}{(e^{(l-j-0.5)h/2}-e^{-(l-j-0.5)h/2})(e^{(l-j+0.5)h/2}-e^{-(l-j+0.5)h/2})}$$

$$= \frac{4\rho^{2l}}{(\rho^{l-0.5}-\rho^j)(\rho^{l+0.5}-\rho^j)},$$

where

$$\rho = e^h > 0.$$

The matrix A can represented as the product of B and D,

$$D = 4diag(\rho^{-2M},\rho^{2(-M+1)},\cdots,\rho^{2M}),$$

$$B = (b_{jl})_{(2M+1)\times(2M+1)},$$

$$b_{jl} = \frac{1}{(\rho^{l-0.5}-\rho^j)(\rho^{l+0.5}-\rho^j)}. \tag{4.7}$$

We only need to prove that B is nonsingular.

If B is an $n \times n$ matrix, $b_{ij} = \frac{1}{x_i - y_j}$, where all $\{x_i\}_1^n$ and $\{y_j\}_1^n$ are distinct, then

$$det(B) \neq 0.$$

B is called the Cauchy matrix [9]. In fact,

$$det(B) = (-1)^{\frac{n(n-1)}{2}} \frac{\prod_{1 \leq i < j \leq n}(x_i - x_j)(y_i - y_j)}{\prod_{1 \leq i,j \leq n}(x_i - y_j)}. \tag{4.8}$$

The set of functions $\{\frac{1}{x-y_j}\}_1^n$ is linearly independent, that is, if there are constants $c_1, c_2, \cdots, c_n$, such that

$$\sum_{j=1}^{n} \frac{c_j}{x - y_j} = 0, \tag{4.9}$$

then $c_1 = c_2 = \cdots = c_n = 0$.

By [10], we can say that the set of functions $\{\frac{1}{x-y_j}\}_1^n$ satisfies the Harr condition. Now we generalize these concepts.

DEFINITION 4.1. *If $\{x_i\}_0^n$ and $\{y_j\}_1^n$ are two groups of numbers, we call the matrix*

$$B = (b_{ij})_{n \times n} = (\frac{1}{(x_{i-1} - y_j)(x_i - y_j)})_{n \times n} \tag{4.10}$$

the generalized Cauchy matrix.

LEMMA 4.2. *If all $\{x_i\}_0^n$ and $\{y_j\}_1^n$ are distinct, then the generalized Cauchy matrix is nonsingular.*

Proof. If the B in (4.10) is singular, then there are $c_1, c_2, \cdots, c_n$ not all zeros such that

$$\sum_{i=1}^{n} \frac{c_i}{(x_{i-1} - y_j)(x_i - y_j)} = 0, \text{ for } j = 1, 2, \cdots, n.$$

Let

$$p(t) = \prod_{i=0}^{n}(x_i - t).$$

$p(t)$ is a polynomial of degree $n + 1$. Consider

$$\phi(t) = \sum_{i=1}^{n} \frac{c_i p(t)}{(x_{i-1} - t)(x_i - t)},$$

where $\phi(t)$ is a polynomial of the degree $n - 1$. Since $\phi(y_j) = 0$ for $j = 1, 2, \cdots, n$, we have

$$\phi(t) \equiv 0.$$

That is

$$\sum_{i=1}^{n} \frac{c_i}{(x_{i-1} - t)(x_i - t)} \equiv 0,$$

or

$$\sum_{i=1}^{n} \frac{c_i}{x_i - x_{i-1}} \left(\frac{1}{x_{i-1} - t} - \frac{1}{x_i - t} \right) \equiv 0.$$

By rearrangement of the summation, we obtain

$$\frac{c_0}{x_1 - x_0} \frac{1}{x_0 - t} + \sum_{i=1}^{n-1} \left(\frac{c_{i+1}}{x_{i+1} - x_i} - \frac{c_i}{x_i - x_{i-1}} \right) \frac{1}{x_i - t} - \frac{c_n}{x_n - x_{n-1}} \frac{1}{x_n - t} \equiv 0. \tag{4.11}$$

Since the set of functions $\{\frac{1}{x_i - t}\}_0^n$ is linearly independent, (it satisfies the Harr condition,) all

$$\frac{c_1}{x_1 - x_0}, \left\{ \frac{c_{i+1}}{x_{i+1} - x_i} - \frac{c_i}{x_i - x_{i-1}} \right\}_{i=1}^{i=n-1}, -\frac{c_n}{x_n - x_{n-1}}$$

must be zeros, that is

$$c_1 = c_2 = \cdots = c_n = 0. \qquad Q.E.D.$$

Since the matrix B in (4.7) is a generalized Cauchy matrix, the $\{\rho^{l-0.5}\}$ and the $\{\rho^j\}$ are two groups of distinct points. By using this lemma, B must be nonsingular. Hence the following theorem is obvious.

THEOREM 4.3. *The matrix A in (4.6) is invertible.*

5. NUMERICAL RESULTS

We give here two applications of the method presented in the preceding sections. Although the matrix A is a symmetric matrix, we use the general Gaussian elimination algorithm with single precision.

Example 1. Consider the following *Hadamard finite part integral equation*,

$$\frac{1}{\pi} \int_{-1}^{1} \frac{f(t)dt}{(t-x)^2} = F(x),$$

where $F(x) = x^2 + 4x - 0.25$. The exact analytic solution is $f(t) = -\frac{\sqrt{1-t^2}}{3}(t^2 + 6t - 0.25)$. We select $M = 16$ and $h = 0.8$. The results are shown in Table 1.

Example 2. Consider the same equation with the right hand side $F(x) = \frac{2}{\pi}(x - 1 - x \ln \frac{1-x}{1+x})$. The exact analytic solution is $f(t) = 1 - t^2$. We choose $M = 10$, $h = 1.0$ and choose $M = 16$, $h = 0.8$. The results are shown in Table 2 and Table 3. Since the solution is an even function, we only show the half data.

Table 1.

Nodes	Exact solution	Numerical solution	Absolute error	Relative error
−0.999995	0.005336	0.004071	−0.001265	23.71%
−0.999988	0.008523	0.007710	−0.000814	9.55%
−0.999973	0.012860	0.012462	−0.000397	3.09%
−0.999940	0.019163	0.019203	0.000040	0.21%
−0.999865	0.028797	0.029051	0.000253	0.88%
−0.999699	0.042951	0.043598	0.000647	1.51%
−0.999330	0.064019	0.065190	0.001171	1.83%
−0.998509	0.095427	0.097277	0.001849	1.94%
−0.996683	0.142066	0.144885	0.002819	1.98%
−0.992632	0.210851	0.215146	0.004295	2.04%
−0.983675	0.310984	0.317533	0.006549	2.11%
−0.964028	0.452289	0.462418	0.010129	2.24%
−0.921669	0.637646	0.653708	0.016062	2.52%
−0.833655	0.838912	0.864813	0.025901	3.09%
−0.664037	0.945410	0.984484	0.039074	4.13%
−0.379949	0.735484	0.777139	0.041656	5.66%
−0.000000	0.083333	0.091056	0.007723	9.27%
0.379949	−0.670339	−0.708051	−0.037712	5.63%
0.664037	−1.040588	−1.087179	−0.046591	4.48%
0.833655	−1.002749	−1.036427	−0.033677	3.36%
0.921668	−0.792702	−0.813875	−0.021173	2.67%
0.964028	−0.572672	−0.585929	−0.013257	2.31%
0.983675	−0.397081	−0.405581	−0.008500	2.14%
0.992631	−0.270259	−0.275803	−0.005544	2.05%
0.996682	−0.182408	−0.186038	−0.003630	1.99%
0.998508	−0.122649	−0.125002	−0.002353	1.92%
0.999329	−0.082330	−0.083799	−0.001469	1.78%
0.999699	−0.055227	−0.056053	−0.000826	1.50%
0.999865	−0.037031	−0.037354	−0.000322	0.87%
0.999939	−0.024833	−0.024696	0.000137	0.55%
0.999973	−0.016643	−0.016028	0.000615	3.69%
0.999988	−0.011177	−0.009917	0.001260	11.27%
0.999994	−0.007492	−0.005239	0.002253	30.07%

Table 2.

Nodes	Exact solution	Numerical Solution	Absolute error	Relative error
−0.99991	0.000180	0.000158	−0.000022	12.36%
−0.99975	0.000493	0.000482	−0.000011	2.17%
−0.99933	0.001340	0.001336	−0.000004	0.28%
−0.99818	0.003641	0.003638	−0.000002	0.06%
−0.99505	0.009866	0.009870	0.000004	0.04%
−0.98661	0.026592	0.026631	0.000039	0.14%
−0.96403	0.070650	0.070934	0.000284	0.40%
−0.90515	0.180706	0.182594	0.001888	1.04%
−0.76159	0.419974	0.430501	0.010527	2.50%
−0.46212	0.786448	0.825407	0.038960	4.95%
0.00000	1.000000	1.065347	0.065347	6.53%
0.46212	0.786448	0.825408	0.038960	4.95%
0.76159	0.419974	0.430501	0.010527	2.50%
0.90515	0.180707	0.182594	0.001888	1.04%
0.96403	0.070651	0.070934	0.000283	0.40%
0.98661	0.026592	0.026630	0.000038	0.14%
0.99505	0.009866	0.009868	0.000002	0.02%
0.99818	0.003641	0.003636	−0.000005	0.14%
0.99933	0.001341	0.001332	−0.000009	0.69%
0.99975	0.000494	0.000478	−0.000015	3.11%
0.99991	0.000182	0.000154	−0.000027	15.06%

Table 3.

Nodes	Exact solution	Numerical solution	Absolute error
0.0000000	1.0000000	1.0411291	0.0411291
0.3799490	0.8556388	0.8853009	0.0296621
0.6640366	0.5590555	0.5713608	0.0123053
0.8336545	0.3050202	0.3086018	0.0035816
0.9216684	0.1505274	0.1513890	0.0008616
0.9640275	0.0706509	0.0708401	0.0001892
0.9836748	0.0323839	0.0324238	0.0000400
0.9926315	0.0146828	0.0146910	0.0000082
0.9966823	0.0066243	0.0066258	0.0000015
0.9985079	0.0029820	0.0029820	0.0000000
0.9993293	0.0013410	0.0013406	−0.0000005
0.9996985	0.0006029	0.0006021	−0.0000008
0.9998645	0.0002710	0.0002701	−0.0000009
0.9999391	0.0001218	0.0001207	−0.0000012
0.9999726	0.0000547	0.0000535	−0.0000012
0.9999877	0.0000247	0.0000228	−0.0000018
0.9999945	0.0000111	0.0000086	−0.0000025

We can give some final comments on the numerical results. First, since both the collocation points and the quadrature nodes cluster around the endpoints of the interval, the absolute error in the middle points seems a little bigger. By increasing the number of nodes, the accuracy can be improved. Second, although the nodes are very dense around the end points, the relative error at the nodes of the most extreme positions is large. The reason is that the outside quadrature nodes have been

neglected when we construct the system of equations. However, the points on the second and the third nodal positions are so close to the points of the extreme positions that it will not affect the total figure to discard the evaluated values at the most extreme positions.

REFERENCES

1. H. R. Kutt, Numer. Math. **24** (1975), 205-210.
2. N. K. Ioakimidis, Acta Mechanica **45** (1982), 31-47.
3. A. C. Kaya and F. Erdogan, Q. Appl. Math **45** (1987), 105-199.
4. M. A. Golberg, J. Integral Equation **5** (1983), 329-340.
5. E. Venturino, Math. Comp. **47** (1986), 159-167.
6. F. Stenger, J. of Approximation Theory **17** (1976), 222-240.
7. J. Hadamard, "Lectures on Cauchy Problems in Linear Partial Differential Equations," Yale University Press, 1923.
8. A. H. Zemanian, "Distribution Theory and Transform Analysis," McGraw-Hill Book Company, 1965.
9. D. E. Knuth, "The Art of Computer Programming," Addison-Wesley Publishing Company, 1973.
10. E. W. Cheney, "Introduction to Approximation Theory," McGraw-Hill Book Company, 1966.

DOMAIN DECOMPOSITION METHODS FOR NONSELFADJOINT OPERATORS *

ZBIGNIEW LEYK†

Abstract. We consider the problem of solving the system of equations, arising from the discretization of nonselfadjoint elliptic boundary value problems via the finite element method. It can be solved with the aid of exact or approximate solvers for the same equations restricted to subregions. The interactions between the subregions, to enforce appropriate continuity requirements, are handled by an iterative method. If we want to use the conjugate gradient method, we must transform the primary system of equations as to get a symmetric system of equations. We show how this can be done so as to get a small condition number of the transformed system of equations. We also discuss computational aspects of the problem.

Key Words. Domain decomposition, additive Schwarz method, nonsymmetric linear equations, finite elements, elliptic equations

AMS(MOS) subject classifications: 65N30, 65F10.

1. Introduction. Domain decomposition methods have recently become one of the most powerful methods for solving partial differential equations. Most of them are different variants of the additive Schwarz method, see [8], which in turn is very simple and easy to implement on computers (including parallel computers). The additive Schwarz method was introduced in [6] and [9] and subsequently studied in [5, 7, 8]. In those papers only symmetric equations were considered, whereas in [3] nonsymmetric equations have been analyzed.

The main property of the additive Schwarz method (*ASM*) is to divide the given problem into a number of smaller equivalent problems. The interaction between these smaller problems is handled by an iterative method. We choose a division of the primary problem such that the condition number of the tranformed (preconditioned) problem is independent of the number of the smaller problems and of some other parameters like H (size of a coarse mesh) and h (size of a fine mesh). If the primary problem is symmetric, then the conjugate gradients method is used. But if it is nonsymmetric, it is rather difficult to decide on the best iterative method. In [3], where the nonsymmetric positive definite differential problems have been considered, the GMRES method is used. Moreover, for symmetric problems of the form $Aw = f$, the inner product $(w, v)_A = (Aw, v)$ has been used ($(\cdot, \cdot)$ is the usual L^2 inner product). This inner product cannot be applied when A is nonsymmetric and we must use another one. In [3] the $(w, v)_{A_S} =$

* This work was supported by the U.S. Army Research Office through the Mathematical Sciences Institute of Cornell University

† MSI, 409 College Avenue, Cornell University, Ithaca, N.Y. 14853; on leave from University of Warsaw, Institute of Computer Science, 00-901 Warsaw, PKiN, p. 850, Poland

$(A_S w, v)$ inner product has been used, where A_S is the symmetric part of A. We show in this paper that for *ASM* there is another convenient inner product of the form $(w,v)_{R^{-1}} = (R^{-1}w, v)$, where R is an operator related to the decomposition of the primary problem.

The outline of this paper is as follows. We present in Sect. 3 an iterative method to solve nonsymmetric preconditioned problems. It is the CG method with a special inner product applied to a symmetrized preconditioned problem. The way of transforming nonsymmetric equations into symmetric ones (symmetrization) is described in [2]. We prove that the iterative method is convergent and the rate of convergence is given. In Sect. 4 we show that for the system of linear equations resulting from the finite element method applied for Dirichlet boundary value problem (a primary problem) the rate of convergence of this method is independent of the number of smaller problems (or subregions) and of H and h. Note that we do not impose the condition that the coarse mesh should be fine enough to get the convergence of the iterative methods as in [3] (we have no upper bound on H). In Sect. 5 we consider some numerical examples.

2. Preliminaries. Consider a linear equation of the form

$$Aw = f, \tag{2.1}$$

where $A : W \to W$ is a linear operator, $w, f \in W$ and W is a finite-dimensional Hilbert space with an inner product $(\cdot,\cdot)$ and a norm $\|\cdot\|$. Finite element or finite differences discretization of an elliptic boundary value problem lead to problems of this type.

Let $D : W \to W$ be a linear, symmetric and positive definite operator. Define the inner product $(w,v)_D = (Dw, v)$ and the norm $\|v\|_D = \sqrt{(Dv,v)}$.

We assume that the operator A satisfies the following conditions:

$$m\|w\|_D^2 \le (Aw, w) \le M\|w\|_D^2, \quad \forall w \in W, \tag{2.2}$$

$$(D^{-1}Aw, Aw) \le K\,(Aw, w), \quad \forall w \in W, \tag{2.3}$$

where $0 < m \le M$, $K > 0$. Note that A need not be symmetric. We introduce the norm $\|v\|_{D^{-1}}$ defined as

$$\|v\|_{D^{-1}} = \sup_{w \in W} \frac{|(v,w)|}{\|w\|_D} \tag{2.4}$$

and hence for any $v, w \in W$

$$|(v,w)| \le \|v\|_{D^{-1}} \|w\|_D. \tag{2.5}$$

We have the equality $\|v\|_{D^{-1}} = \sqrt{(D^{-1}v, v)}$. Note that (2.3) can be rewritten as $\|Aw\|_{D^{-1}}^2 \le K\,(Aw, w)$.

Remark 2.1. Condition (2.3) is equivalent to

$$(D^{-1}w, w) \leq K\,(A^{-1}w, w), \quad \forall w \in W.$$

If A is symmetric then $K = M$ (it follows from (2.2)). □

Remark 2.2. Let A^T be the transpose of A with respect to $(\cdot, \cdot)$, i.e., $(A^T u, v) = (u, Av)$ for any $u, v \in W$. It is evident that $A = A_S + A_N$, where $A_S = \frac{1}{2}(A + A^T)$ (the symmetric part of A) and $A_N = \frac{1}{2}(A - A^T)$ (the skew-symmetric part of A). Assume that

$$(D^{-1}A_N w, A_N w) \leq K_1(Aw, w), \quad \forall w \in W, \tag{2.6}$$

where $K_1 \geq 0$. We show that K depends on K_1 and M.

First we notice that $(A_S w, w) = (Aw, w)$ for any $w \in W$. We have

$$(D^{-1}Aw, Aw) = (D^{-1}A_S w, A_S w) + 2(D^{-1}A_S w, A_N w) + (D^{-1}A_N w, A_N w). \tag{2.7}$$

It is easy to show that $(A_S v, v) \leq M(Dv, v)$ is equivalent to $(D^{-1}v, v) \leq M(A_S^{-1}v, v)$ for any $v \in W$. Hence

$$(D^{-1}A_S w, A_S w) \leq M(A_S^{-1}A_S w, A_S w) = M(Aw, w). \tag{2.8}$$

Further

$$(D^{-1}A_S w, A_N w) \leq (D^{-1}A_S w, A_S w)^{1/2}(D^{-1}A_N w, A_N w)^{1/2}. \tag{2.9}$$

Combining (2.7) with (2.6), (2.8) and (2.9) we get

$$(D^{-1}Aw, Aw) \leq M(Aw, w) + 2\sqrt{MK_1}(Aw, w) + K_1(Aw, w) = (\sqrt{M} + \sqrt{K_1})^2(Aw, w).$$

From the above it follows that we can set $K = (\sqrt{M} + \sqrt{K_1})^2$. □

Remark 2.3. If we have the following estimates

$$m_1\|w\|_D^2 \leq (Aw, w), \quad \forall w \in W \tag{2.10}$$

and

$$|(Aw, v)| \leq M_1\|w\|_D\|v\|_D, \quad \forall w, v \in W, \tag{2.11}$$

then we can take $m = m_1$, $M = M_1$ and $K = m_1^{-1}M_1^2$ in (2.2) and (2.3). We show that the inequalities (2.10) and (2.11) imply (2.3). Taking $v = D^{-1}Aw$ in (2.11) we get

$$|(Aw, D^{-1}Aw)| \leq M_1\|w\|_D\|D^{-1}Aw\|_D.$$

But $\|D^{-1}Aw\|_D = \|Aw\|_{D^{-1}}$ and hence

$$|(Aw, D^{-1}Aw)| \leq M_1^2\|w\|_D^2.$$

Since $\|w\|_D^2 \le m_1^{-1}(Aw, w)$, then

$$|(Aw, D^{-1}Aw)| \le m_1^{-1} M_1^2 (Aw, w).$$

Usually the constant K is smaller then $m_1^{-1}M_1^2$.

It is also possible to show that (2.2) and (2.3) implies (2.11). We have

$$(Au, v) \le \|Au\|_{D^{-1}} \|v\|_D \le \sqrt{KM}\|u\|_D \|v\|_D,$$

since $(D^{-1}Au, Au) \le K(Au, u) \le KM\|u\|_D^2$. We can take $M_1 = \sqrt{KM}$ and $m_1 = m$. From the above considerations it follows that the conditions (2.2) and (2.3) are equivalent to (2.10) and (2.11). The condition (2.3) is important only for nonsymmetric A. If A is symmetric it follows from (2.2). □

In order to decompose the space W we choose linear subspaces $W_i \subset W$, $i = 0, \ldots, k$, such that

$$W = W_0 + \ldots + W_k, \tag{2.12}$$

i.e., any $w \in W$ can be represented (usually non-uniquely) as $w = w_0 + \ldots + w_k$, where $w_i \in W_i$. We assume that there exists a constant $\delta > 0$ such that for any $v \in W$ there exists such a decomposition $v = \sum_{i=0}^k v_i$, $v_i \in W_i$, $i = 0, \ldots, k$, that

$$\sum_{i=0}^{k} \|v_i\|_D^2 \le \delta \|v\|_D^2. \tag{2.13}$$

Note that δ may depend on k and $\dim W_i$, $i = 0, \ldots, k$.

Let S_i, $i = 0, \ldots, k$, be the orthogonal projectors onto W_i with respect to $(\cdot, \cdot)_D$, that is, for any $w \in W$

$$(S_i w, v_i)_D = (w, v_i)_D, \quad \forall v_i \in W_i. \tag{2.14}$$

We assume that there exists a constant γ such that for any $v \in W$

$$\sum_{i=0}^{k} \|S_i v\|_D^2 \le \gamma \|v\|_D^2. \tag{2.15}$$

We consider that γ is the smallest constant satisfying (2.15) for any $v \in W$.

Remark 2.4. Since $\|S_i v\|_D \le \|v\|_D$, then $\sum_{i=0}^k \|S_i v\|_D^2 \le (k+1)\|v\|_D^2$ and we can always take $\gamma = k + 1$. But for some decompositions of W we find that γ does not depend on k. □

We now introduce *the local decomposition operators* $R_i : W \to W_i$, $i = 0, \ldots, k$, defined for any $w \in W$ as follows

$$(R_i w, v_i)_D = (w, v_i), \quad \forall v_i \in W_i. \tag{2.16}$$

The global decomposition operator is a linear combination of R_i, $i = 0, \ldots, k$, that is

$$R = \sum_{i=0}^{k} R_i.$$

Note that $R : W \to W$, since W is a linear space. Denote by $Q_i : W \to W_i$, $i = 0, \ldots, k$, the orthogonal projectors onto W_i, i.e., for any $w \in W$

$$(Q_i w, v_i) = (w, v_i), \quad \forall v_i \in W.$$

From (2.16) we get that for any $w, v \in W$

$$(DR_i w, Q_i v) = (w, Q_i v)$$

and, since Q_i is symmetric and $R_i w = Q_i R_i w$, we find that for any $v \in W$

$$(Q_i D Q_i R_i w, v) = (Q_i w, v).$$

This implies that

$$D_i R_i w = Q_i w,$$

where $D_i = Q_i D Q_i$. Hence $R_i w = D_i^{-1} Q_i w$ and

$$R = \sum_{i=0}^{k} D_i^{-1} Q_i \tag{2.17}$$

Note that R_i, $i = 0, \ldots, k$, is symmetric with respect to $(\cdot, \cdot)$, but it is not a projector ($R_i^2 \neq R_i$).

We now show that R is equivalent to D^{-1}, i.e.,

$$\delta^{-1}(D^{-1}w, w) \leq (Rw, w) \leq \gamma(D^{-1}w, w),$$

where δ and γ are defined in (2.13) and (2.15) respectively. From the above it follows that $1 \leq \delta\gamma$.

LEMMA 2.1. *If (2.13) is satisfied, then for any* $w \in W$

$$\delta\,(Rw, w) \geq \|w\|^2_{D^{-1}}, \tag{2.18}$$

Proof. For any $v \in W$ we have a decomposition $v = \sum_{i=0}^{k} v_i$, $v_i \in W_i$, satisfying (2.13). Note that $(w, v_i) = (R_i w, v_i)_D$, see (2.16). Hence

$$|\,(w, v)\,| = \left|\sum_{i=0}^{k} (w, v_i)\right| \leq \sum_{i=0}^{k} |\,(R_i w, v_i)_D\,|\,. \tag{2.19}$$

Further,

$$\sum_{i=0}^{k} |(R_i w, v_i)_D| \leq \sum_{i=0}^{k} \|R_i w\|_D \|v_i\|_D \tag{2.20}$$
$$\leq \left(\sum_{i=0}^{k} \|R_i w\|_D^2\right)^{1/2} \left(\sum_{i=0}^{k} \|v_i\|_D^2\right)^{1/2}$$

Since $(R_i w, R_i w)_D = (w, R_i w)$, see (2.16), then from (2.19), (2.20) and (2.13) we obtain

$$|(w, v)| \leq (w, Rw)^{1/2} \delta^{1/2} \|v\|_D.$$

We divide the above by $\|v\|_D$ and take the supremum over $v \in W$. Using (2.4) we obtain

$$\|w\|_{D^{-1}} \leq \delta^{1/2} (Rw, w)^{1/2}, \tag{2.21}$$

which completes the proof. □

Since D^{-1} is positive definite Lemma 2.1 implies

COROLLARY 2.1. *R is invertible.*

We now obtain the estimate of (Rw, w) from above.

LEMMA 2.2. *Let (2.15) be satisfied. Then for any* $w \in W$

$$(Rw, w) \leq \gamma \|w\|_{D^{-1}}^2.$$

Proof. From (2.16) and (2.14) we find that for any $v_i \in W_i$, $i = 0, \ldots, k$,

$$(R_i D w, v_i)_D = (Dw, v_i) = (S_i w, v_i)_D.$$

Hence for any $w \in W$

$$R_i D w = S_i w. \tag{2.22}$$

Since $S_i = S_i^2$

$$(RDv, Dv) = \sum_{i=0}^{k} (R_i D v, v)_D = \sum_{i=0}^{k} (S_i v, v)_D = \sum_{i=0}^{k} \|S_i v\|_D^2 \leq \gamma \|v\|_D^2.$$

Taking $Dv = w$ we obtain $(Rw, w) \leq \gamma \|w\|_{D^{-1}}^2$. □

3. Symmetrization. It is evident that a nonsymmetric equation can be transformed to an equivalent equation with a symmetric operator. See [2], where two kinds of symmetrization of nonsymmetric equations are described. We shall use one of them.

Consider the following equation

$$Tw = z, \tag{3.1}$$

where $T = RA^TRA$, $z = RA^TRf$ and A^T is transpose of A with respect to $(\cdot, \cdot)$, i.e., $(A^Tu, v) = (u, Av)$ for any $u, v \in W$. Notice that T is symmetric with respect to $(\cdot, \cdot)_{R^{-1}}$, since $(Tw, v)_{R^{-1}} = (RAw, Av)$ and R is symmetric. We can use the conjugate gradient method (CG) to solve (3.1), see [2]. It is rather easy to find Rw for any $w \in W$, but it is hard to compute $R^{-1}w$. Thus we introduce some changes in CG algorithm. We use the version of CG studied in [10], see also [1].

We apply the following algorithm to solve (3.1). We assume that we are given an initial approximation w_0 to the solution w of (3.1).

Algorithm S

$\tilde{r}_0 := A^TR(f - Aw_0)$; $p_0 := R\tilde{r}_0$; $i := 0$;
while *error* $>$ *tolerance* **do**
begin

$$
\begin{aligned}
\alpha_i &:= \frac{(R\tilde{r}_i, \tilde{r}_i)}{(p_i, A^TRAp_i)}; \\
w_{i+1} &:= w_i + \alpha_i p_i; \\
\tilde{r}_{i+1} &:= \tilde{r}_i - \alpha_i A^TRAp_i; \\
\beta_i &:= \frac{(R\tilde{r}_{i+1}, \tilde{r}_{i+1})}{(R\tilde{r}_i, \tilde{r}_i)}; \\
p_{i+1} &:= R\tilde{r}_{i+1} + \beta_i p_i; \\
i &:= i + 1
\end{aligned}
\tag{3.2}
$$

end.

Here we have introduced $R\tilde{r}_i$ instead of r_i used in [10] and we have used the $(\cdot, \cdot)_{R^{-1}}$ inner product instead of the standard $(\cdot, \cdot)$, since T is symmetric with respect to the $(\cdot, \cdot)_{R^{-1}}$ inner product and our estimates are given in $\|\cdot\|_{R^{-1}}$ norm. Note that R can be considered as a preconditioner for the equation $A^TRAw = A^TRf$. We get the same algorithm and the same rate of convergence. On each iterative step we use the operator R twice: to compute A^TRAp_i and $R\tilde{r}_{i+1}$.

We now study the rate of convergence of CG applied to (3.1). We have

THEOREM 3.1.

$$
\|w - w_n\|_{R^{-1}} \leq 2\left(1 - \frac{2m}{\delta\gamma\sqrt{KM} + m}\right)^n \|w - w_0\|_{R^{-1}},
$$

where w_n is an approximation of w produced by the CG method after n iterative steps.

Proof. We show that

$$
\delta^{-2}m^2\|w\|^2_{R^{-1}} \leq (Tw, w)_{R^{-1}} \leq \gamma^2 KM\|w\|^2_{R^{-1}} \quad \text{for all } w \in W. \tag{3.3}
$$

We have

$$
(Tw, w)_{R^{-1}} = (RAw, Aw).
$$

By Lemma 2.2

$$\begin{aligned} \|w\|_D^2 &= (RDw, w)_{R^{-1}} \leq (RDw, Dw)^{1/2} \|w\|_{R^{-1}} \\ &\leq \gamma^{1/2} \|Dw\|_{D^{-1}} \|w\|_{R^{-1}} = \gamma^{1/2} \|w\|_D \|w\|_{R^{-1}}. \end{aligned}$$

Hence

$$\|w\|_D^2 \leq \gamma \|w\|_{R^{-1}}^2. \tag{3.4}$$

Using Lemma 2.2, (2.3), (2.2) and (3.4) we get

$$(RAw, Aw) \leq \gamma \|Aw\|_{D^{-1}}^2 \leq \gamma K(Aw, w) \leq \gamma KM \|w\|_D^2 \leq \gamma^2 KM \|w\|_{R^{-1}}^2.$$

Thus we have proved the second inequality in (3.3). We now prove the first inequality. By Lemma 2.1

$$(RAw, Aw) \geq \delta^{-1} \|Aw\|_{D^{-1}}^2. \tag{3.5}$$

We now show that $\|v\|_D^2 \geq \delta^{-1} \|v\|_{R^{-1}}^2$. We have

$$(R^{-1}v, v) = (R^{-1}v, Dv)_{D^{-1}} \leq (R^{-1}v, R^{-1}v)_{D^{-1}}^{1/2} (Dv, v)^{1/2}.$$

Using Lemma 2.1 ($w = R^{-1}v$) we find that

$$(R^{-1}v, R^{-1}v)_{D^{-1}}^{1/2} \leq \delta^{1/2} (R^{-1}v, v)^{1/2}.$$

Hence

$$(R^{-1}v, v) \leq \delta (Dv, v). \tag{3.6}$$

From (2.4) and (2.2) we get

$$\|Aw\|_{D^{-1}} = \sup_{v \in W} \frac{|(Aw, v)|}{\|v\|_D} \geq \frac{|(Aw, w)|}{\|w\|_D} \geq m \|w\|_D. \tag{3.7}$$

Using (3.7) and (3.6) we obtain

$$\|Aw\|_{D^{-1}} \geq m \delta^{-1/2} \|w\|_{R^{-1}}.$$

Combining the above and (3.5) we finally have

$$(RAw, Aw) \geq \delta^{-2} m^2 \|w\|_{R^{-1}}^2.$$

We conclude that the condition number κ of T is bounded by $\delta^2 \gamma^2 KM/m^2$. It is well known that for the CG method, see [2, 11],

$$\|w - w_n\|_{R^{-1}} \leq 2 \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^n \|w - w_0\|_{R^{-1}}.$$

Since $\kappa \leq \delta^2 \gamma^2 KM/m^2$, then

$$\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} = 1 - \frac{2}{\sqrt{\kappa} + 1} \leq 1 - \frac{2m}{\delta \gamma \sqrt{KM} + m}.$$

□

4. Application. In this section we apply the theory given in the previous sections to the Dirichlet boundary value problem.

4.1. Differential problem. Let Ω denote an open bounded polygon in R^2 with a boundary $\partial\Omega$. Consider the following Dirichlet boundary value problem

$$\left\{ \begin{array}{rcll} Lu & = & \tilde{f} & \text{in } \Omega, \\ u & = & 0 & \text{on } \partial\Omega, \end{array} \right. \tag{4.1}$$

where L is the elliptic operator of the form

$$Lu(x) = -\sum_{i,j=1}^{2} \frac{\partial}{\partial x_i}\left(a_{ij}(x)\frac{\partial u(x)}{\partial x_j}\right) + \sum_{i=1}^{2} b_i(x)\frac{\partial u(x)}{\partial x_i} + c(x)u(x),$$

and $a_{ij}(x) = a_{ji}(x)$ for $i,j = 1,2$ and any $x \in \Omega$. We suppose that L is strongly elliptic, that is, there exists a constant $c_0 > 0$ such that

$$\sum_{i,j=1}^{2} a_{ij}(x)\xi_i\xi_j \geq c_0 \sum_{i=1}^{2} \xi_i^2, \quad \forall \xi_1, \xi_2 \in R, \quad \forall x \in \Omega.$$

We will consider the equation (4.1) in the weak form: Find $u \in H_0^1(\Omega)$ such that

$$(\tilde{A}u, v) = (\tilde{f}, v), \qquad \forall v \in H_0^1(\Omega), \tag{4.2}$$

where

$$(\tilde{A}u, v) = \sum_{i,j=1}^{2} \int_\Omega a_{ij}\frac{\partial u}{\partial x_i}\frac{\partial v}{\partial x_j}\,dx + \sum_{i=1}^{2} \int_\Omega b_i \frac{\partial u}{\partial x_i} v\,dx + \int_\Omega cuv\,dx$$

and

$$(\tilde{f}, v) = \int_\Omega \tilde{f} v\,dx.$$

Here $(\cdot, \cdot)$ means the $L^2(\Omega)$ inner product.

We assume that $\tilde{f} \in L^2(\Omega)$ and the coefficients a_{ij}, b_i and c are such that the operator $\tilde{A}$ is well defined. Moreover, we suppose that the equation (4.2) has the unique solution $u \in H_0^1(\Omega)$ for any fixed $\tilde{f}$.

4.2. The finite element method. We approximate the solution of the equation (4.2) by means of the finite element method with piecewise linear triangular elements. We first introduce a two level triangulation of $\Omega \in R^2$ and the finite element spaces.

For a given polygonal region Ω, we define $\{\Omega_i\}_{i=1}^k$ to be a regular finite element triangulation of Ω, that is, $\{\Omega_i\}$ is a set of non-overlapping triangles such that

$$\Omega = \bigcup_{i=1}^{k} \overline{\Omega_i}.$$

and moreover, no vertex of a triangle lies on the edge of another triangle.

Let H_i be the diameter of Ω_i and $\hat{H}_i$ be the diameter of the largest inscribed ball in Ω_i. We assume that the ratio $H_i/\hat{H}_i$ is uniformly bounded from above, that is, we have a regular triangulation. Let $H = \max\{H_1, \ldots, H_k\}$. We call Ω_i the coarse mesh of Ω. Note that usually k depends on H ($k = cH^{-2}$), but this fact will not be used further.

Next we divide each Ω_i into smaller triangles, denoted by τ_j^i, $j = 1, \ldots, n_i$. We assume that $\Omega_i = \bigcup_{j=1}^{n_i} \tau_j^i$ and the triangulation $\{\tau_j^i\}$ is regular. We also assume that no vertex of a smaller triangle lies on the edge of any other smaller triangler. Let h_j^i be the diameter of τ_j^i and

$$h = \max_{i,j} h_j^i.$$

We call $\{\tau_j^i\}$ the fine mesh of Ω. We now introduce the piecewise linear finite element spaces. Let

$$V^H = \{v^H | \text{ continuous on } \Omega, \text{ and } v^H{}_{|\Omega_i} \text{ is linear on } \Omega_i,\ v^H = 0 \text{ on } \partial\Omega\ \}$$

and

$$V^h = \{v^h | \text{ continuous on } \Omega, \text{ and } v^h{}_{|\tau_j^i} \text{ is linear on } \tau_j^i,\ v^h = 0 \text{ on } \partial\Omega\ \}.$$

Note that $V^H \subset V^h$.

The finite element method is defined as follows: Find $u_h \in V^h$ such that

$$(\tilde{A}u_h, v_h) = (\tilde{f}, v_h), \quad \forall v^h \in V^h \tag{4.3}$$

We assume that the coefficients a_{ij}, b_i and c are such, that the conditions (2.2) and (2.3) are satisfied with constants $0 < m \leq M$, $K > 0$. From the Lax-Milgram Theorem, see e.g. [4], it follows that the equation (4.3) has a unique solution $u_h \in V_h$ for any fixed $\tilde{f}$.

4.3. The additive Schwarz method. Let U_h be the orthogonal projector from $L^2(\Omega)$ onto $V^h \in H_0^1(\Omega)$ with respect to $(\cdot, \cdot)$. We define $A = U_h \tilde{A} U_h$ and $f = U_h \tilde{f}$. Usually we choose $D = A_S$, where A_S is the symmetric part of A, i.e., $A_S = \frac{1}{2}(A + A^T)$. Then we have $m = M = 1$. Note that the operator $A : V^h \to V^h$ is the finite dimensional operator and satisfies the conditions (2.2) and (2.3). We have the equation (2.1). We now define the space W and its decomposition.

We extend each Ω_i to a larger region $\tilde{\Omega}_i$ such that $\Omega_i \subset \tilde{\Omega}_i$. We assume that there exists a constant $\alpha > 0$ such that

$$\text{dist}\,(\partial\tilde{\Omega}_i, \partial\Omega_i) \geq \alpha H_i, \quad \forall i = 1, \ldots, k.$$

Moreover, we suppose that the boundary $\partial\tilde{\Omega}_i$ does not cut through any h-level elements. We cut off the parts of $\tilde{\Omega}_i$ that are outside of Ω. We define for $i = 1, \ldots, k$

$$W_i = V^h \cap H_0^1(\tilde{\Omega}_i),$$

where we extend each function of W_i by zero to the complement of $\tilde{\Omega}_i$. It is known that this extension is continuous. Thus W_i, $i = 1, \ldots, k$, are the subspaces of W. We also define

$$W_0 = V^H.$$

Finally the space W is defined as follows:

$$W = W_0 + W_1 + \cdots + W_k.$$

It is clear that $W = V^h$.

We check now that the conditions (2.13) and (2.15) are satisfied. It can be shown that γ is independent of k. Also the constant δ is independent of k, h and H. The proof of this is given in [5]. As a conclusion we get

THEOREM 4.1. *The rate of convergence of the method (3.2) is independent of k, h and H.*

5. Numerical results. Consider the following problem defined on $\Omega = [0,1] \times [0,1] \subset R^2$

$$Lu = -\frac{\partial}{\partial x}\left(a_1 \frac{\partial u}{\partial x}\right) - \frac{\partial}{\partial y}\left(a_2 \frac{\partial u}{\partial y}\right) + b_1 \frac{\partial u}{\partial x} + b_2 \frac{\partial u}{\partial y} + cu = f, \tag{5.1}$$

with $u = 0$ on $\partial\Omega$. The function f is defined so that the solution of (5.1) in $H_0^1(\Omega)$ is $u = xe^{xy}\sin(\pi x)\sin(\pi y)$. The coefficients are as follows:

$$a_1 = 1 + x^2 + y^2,\ a_2 = e^{xy},\ b_1 = x + y,\ b_2 = 1/(1 + x + y) \text{ and } c = 0.5 \tag{5.2}$$

or

$$a_1 = \sigma,\ a_2 = \sigma,\ b_1 = 1,\ b_2 = 1 \text{ and } c = 1, \tag{5.3}$$

σ will be specified later. This test problem is similar to that in [3]. The stopping criterion for iterative methods is $\|r_i\|_{R^{-1}} \leq 10^{-3}\|r_0\|_{R^{-1}}$. The operator $D = A_S$, where A_S is the symmetric part of A. The program run in double precision on a Sun SPARC station machine.

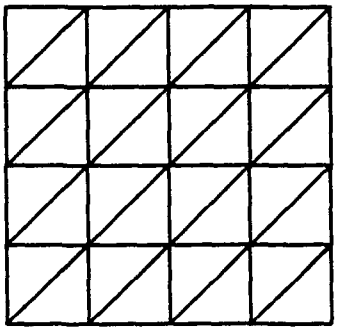


FIG. 5.1 *Coarse mesh of* Ω

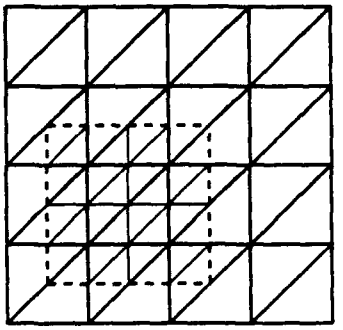


FIG. 5.2 *Subregion* $\tilde{\Omega}_i$ *with fine mesh inside*

Let triangles $\{\Omega_i\}_{i=1}^{2k}$ be the division of Ω (coarse mesh), see Fig. 5.1. We extend each $\Omega_i \cup \Omega_{i+1}$, $i = 1, 3, \ldots, 2k-1$ to a larger rectangle $\tilde{\Omega}_i$, $i = 1, \ldots, k$, such that each side of $\tilde{\Omega}_i$ is parallel to its corresponding axis, see Fig. 5.2. We divide each Ω_i into smaller triangles (fine mesh) such that $\partial\tilde{\Omega}_i$ does not cut through any h–level triangles. The width of the strip $\tilde{\Omega}_i \setminus (\Omega_i \cup \Omega_{i+1})$ in h-units will be called the size of overlap. We do not extend $\tilde{\Omega}_i$ outside Ω.

Test 1.
We examine the test problem (5.1). For different h and H we will study the effectiveness of the algorithm and test whether it depends on the mesh sizes h and H. The overlapping factor α is fixed. We will try to keep $\alpha \approx 1/3$.

Let **ite** denote the number of iterative steps required by the method (3.1), **max err** and L^2 denote respectively the maximum error and error in L^2-norm between the numerical solution given by (3.1) and the exact solution. The mark $*$ in the tables means that the overlapping factor α is less then $1/3$.

h	H	ite	H	ite	max err	L^2
1/15	1/3	15*	1/5	18	3.8×10^{-3}	1.4×10^{-3}
1/30	1/3	16*	1/5	18	9.2×10^{-4}	3.8×10^{-4}
1/45	1/3	16	1/5	18	5.0×10^{-4}	1.8×10^{-4}
1/60	1/3	16	1/5	18	4.0×10^{-4}	1.3×10^{-4}

TABLE. 5.1 *Problem (5.1), (5.2)*

h	H	ite	H	ite	max err	L^2
1/15	1/3	15*	1/5	15	9.0×10^{-3}	4.2×10^{-3}
1/30	1/3	16*	1/5	16	2.3×10^{-3}	1.1×10^{-3}
1/45	1/3	17	1/5	16	1.1×10^{-3}	4.9×10^{-4}
1/60	1/3	17	1/5	16	7.5×10^{-4}	3.0×10^{-4}

TABLE. 5.2 *Problem (5.1), (5.3) with* $\sigma = \sqrt{2}/30$

In Tab. 5.1 and Tab. 5.2, respectively, the problem (5.1) with coefficients (5.2) and the problem (5.1), (5.3) with $\sigma = \sqrt{2}/30$ are tested. From these tables it is seen that the method does not degenerate with h when H is fixed.

Test 2.

We test how the method (3.1) depends on H. We keep $h = 1/30$ and vary the parameter H. In Tab. 5.3 results are given for the problem (5.1) with coefficients (5.2). In Tab. 5.4 and Tab. 5.5 we give results for the problem with coefficients (5.3) and with $\sigma = \sqrt{2}/30$ and $\sigma = \sqrt{2}/100$ respectively. It can be seen that **ite** is in agreement with the theory where an upper bound on the number of iterative steps is independent of h and H. A particular number of iterative steps may vary in some range.

H	1/3	1/5	1/6	1/10
ite	16*	18	18*	17

TABLE. 5.3 *Problem (5.1), (5.2)*

H	1/3	1/5	1/6	1/10
ite	16*	16	15*	14

TABLE. 5.4 *Problem (5.1), (5.3) with* $\sigma = \sqrt{2}/30$

H	1/3	1/5	1/6	1/10
ite	22*	22	17*	19

TABLE. 5.5 *Problem (5.1), (5.3) with* $\sigma = \sqrt{2}/100$

Acknowledgments. The author thanks Professor James H. Bramble for his stimulating discussion and valuable remarks.

REFERENCES

[1] O. Axelsson and V. A. Barker. *Finite element solution of boundary value problems.* Academic Press, 1984.

[2] J. H. Bramble and J. E. Pasciak. Preconditioned iterative methods for nonselfadjoint or indefinite elliptic boundary value problems. In H. Kardestuncer, editor, *Unification of Finite Element Methods*. Elsevier Science Publishers B. V., 1984.

[3] X.-C. Cai. An additive Schwarz algorithm for nonselfadjoint elliptic equations. In T. F. Chan et al., editors, *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 232–244. SIAM, Philadelphia, 1990.

[4] P. G. Ciarlet. *The finite element method for elliptic problems*. North–Holland, Amsterdam, 1978.

[5] M. Dryja. An additive Schwarz algorithm for two- and three-dimensional finite element elliptic problems. In T. F. Chan et al., editors, *Domain decomposition methods*, pages 168–172. SIAM, Philadelphia, 1989.

[6] M. Dryja and O. B. Widlund. An additive variant of the Schwarz alternating method for the case of many subregions. Technical Report 339, Dept. of Comp. Sci., Courant Institute, 1987.

[7] M. Dryja and O. B. Widlund. Some domain decomposition algorithms for elliptic problems. In L. Hayes and D. Kincaid, editors, *Iterative Methods for Large Linear Systems*. Academic Press, 1990.

[8] M. Dryja and O. B. Widlund. Towards a unified theory of domain decomposition algorithms for elliptic problems. In T. F. Chan et al., editors, *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 3–21. SIAM, Philadelphia, 1990.

[9] A. M. Matsokin and S. V. Nepomnyaschikh. A Schwarz alternating method in a subspace. *Soviet Mathematics*, 29:78–84, 1985.

[10] J. K. Reid. On the method of conjugate gradients for the solution of large sparse systems of equations. In J. K. Reid, editor, *Large Sparse Sets of Linear Equations*, pages 231–254. Academic Press, New York, 1971.

[11] A. A. Samarskii and E. S. Nikolaev. *Numerical Methods for Grid Equations*. Birkhauser Verlag, Basel, 1989.

INELASTIC MICROSTRUCTURE IN RAPID GRANULAR FLOWS OF SMOOTH DISKS

by

Mark A. Hopkins
Thayer School of Engineering
Dartmouth College
Hanover, New Hampshire 03755

and

Michel Y. Louge†
Department of Mechanical Engineering
Cornell University
Ithaca, New York 14853

ABSTRACT

Computer simulations of two-dimensional rapid granular flows of uniform smooth inelastic disks under simple shear reveal a dynamic microstructure characterized by the local, spatially anisotropic agglomeration of disks. A spectral analysis of the concentration field suggests that the formation of this inelastic microstructure is correlated with the magnitude of the total stresses in the flow. The simulations confirm the theoretical results of Jenkins and Richman (*J. of Fluid Mech.* **192**, pp. 313-328, 1988) for the kinetic stresses in the dilute limit and for the collisional stresses in the dense limit, when the size of the periodic domain used in the simulations is a small multiple of the disk diameter. However the kinetic and, to a lesser extent, collisional stresses both increase significantly with the size of the periodic domain, thus departing from the predictions of the theory that assumes spatial homogeneity and isotropy. The corresponding paper discussing the simulation technique and the formation of inelastic microstructure is presently under review by *Phys. Fluids A* (1990).

† Author to whom correspondence should be addressed.

CURVE DESIGN AND ANALYSIS USING SPLINES AND WAVELETS *

Charles K. Chui*
Department of Mathematics
Texas A&M University
College Station, TX 77843

ABSTRACT. *While it is well known that spline functions* provide a very useful tool for designing and representing curves, this paper initiates an application of wavelets for analyzing their shapes. An iterative design and analysis procedure is recommended for yielding satisfactory results. The local optimal-order interpolatory scheme introduced in our earlier work provides a moving average algorithm for the design purpose, while the decomposition formula in wavelets yields a decomposition algorithm. After necessary modifications are made, a reconstruction algorithm is used to restore a more desirable curve. This algorithm is a consequence of the two-scale relation of B-splines and that of the compactly supported B-wavelets we recently constructed.

1. INTRODUCTION. Many curve design schemes are available in the literature. In particular, when spline functions are used, these schemes are especially efficient and the resulting spline representations of curves are usually very satisfactory. However, much less has been studied on the analysis of the "shapes" of these spline curves. This paper initiates an application of spline wavelets for such an analysis. The decomposition sequences of splines into splines with fewer knots and the orthogonal *spline wavelet complements* are used to yield a wavelet decomposition algorithm, while the pair of two-scale relations give rise to a reconstruction algorithm.

Let us first recall that for each positive integer m, the m^{th} order B-spline N_m, defined recursively by

$$N_m(x) = (N_{m-1} * N_1)(x) = \int_0^1 N_{m-1}(x-t)dt, \tag{1}$$

with $N_1 = \chi_{[0,1)}$, generates a multiresolution analysis of $L^2 = L^2(-\infty, \infty)$ (cf. M [10,11]). In particular, if V_k is the L^2-closure of the linear span of

$$N_{m;k,j}(x) := N_m(2^k x - j), \quad j \in \mathbf{Z}, \tag{2}$$

then $\{V_k\}$ is a nested sequence of closed subspaces of L^2, whose closure is all of L^2, and whose intersection is the zero function. For each $k \in \mathbf{Z}$, let W_k denote the orthogonal complementary subspace of V_{k+1} relative to V_k; that is,

* Supported by SDIO/IST managed by the U.S. Army Research Office under Contract Numbers DAAL 03-87-K-0025 and DAAL 03-90-G-0091

$$W_k \perp V_k \quad \text{and} \quad V_{k+1} = V_k + W_k,$$

and we use the standard notation

(3) $$V_{k+1} = V_k \oplus W_k.$$

It is also well known (cf. M [10,11]) that there exists some function ψ_m in W_0 that generates all the spaces W_k; that is, by setting

(4) $$\psi_{m;k,j}(x) = \psi_m(2^k x - j),$$

then each W_k is the closure of the linear span of $\psi_{m;k,j}$, $j \in \mathbf{Z}$. Here, ψ_m is called a *wavelet* corresponding to the B-spline N_m. In M [10,11], however, since an orthonormalization procedure is used to give ψ_m, this wavelet has to have infinite support. By introducing a new approach to the construction of wavelets in CW [6,7], we have now a compactly supported spline wavelet

(5) $$\psi_m(x) = \frac{1}{2^{m-1}} \sum_{j=0}^{2m-2} (-1)^j N_{2m}(j+1) N_{2m}^{(m)}(2x - j).$$

Since this wavelet has minimum support (cf. CW [8]), we may call it a B-wavelet of order m associated with the B-spline N_m (cf. Fig. 1 and Fig. 2). The m^{th} order spline/wavelet pair

(6) $$(N_m, \psi_m),$$

given in (1) and (5), will be used for the design/analysis of curves.

2. DESIGN, DECOMPOSITION, AND RECONSTRUCTION ALGORITHMS. The design/analysis scheme is an iterative procedure that requires three "moving-average" algorithms. The first one is an interpolation/approximation moving average algorithm derived from the optimal order real-time interpolatoin scheme discussed in CD [5] (see also C [2] and CD [4]). The second algorithm is a so-called decomposition algorithm. It is derived from the decomposition formula

(7) $$N_m(2x - \ell) = \sum_n a_{\ell-2n} N_m(x - n) + \sum_n b_{\ell-2n} \psi_m(x - n).$$

The validity of (7) follows from the decomposition of V_1 as an orthogonal sum of V_0 and W_0 as given by (3). The third algorithm comes from the pair of two-scale relations:

(8) $$N_m(x) = \sum_n p_n N_m(2x - n)$$

and

(9) $$\psi_m(x) = \sum_n q_n N_m(2x - n).$$

Here, (8) and (9) are consequences of $V_0 \subset V_1$ and $W_0 \subset V_1$, respectively. The four sequences $\{a_n\}, \{b_n\}, \{p_n\}$, and $\{q_n\}$ are easily computed by using the results in CW [7]. Note that since both N_m and ψ_m have compact supports, $\{p_n\}$ and $\{q_n\}$ are finite sequences. The sequences $\{a_n\}$ and $\{b_n\}$ have exponential decay, and by a duality principle established in CW [7], the two pairs $(\{a_n\}, \{b_n\})$ and $(\{p_n\}, \{q_n\})$ are interchangeable. Details and generalities can be found in CW [6,7,8].

2.1. DESIGN ALGORITHM. We employ the real-time interpolatory scheme discussed in CD [5] (see also C [2], and CD [4]). However, since the B-splines must be used for decomposition purposes, we must change the real-time table-look-up scheme in CD [5] to a moving average scheme. We emphasize that this interpolation scheme has the optimal order of approximation. The construction introduced in CD [4] can be summarized as follows. Quasi-interpolation formulae are constructed using finite blocks of data information. Next, a completely local interpolation scheme is used, and finally a blending procedure is employed to yield the desired interpolatory scheme that utilizes only finite blocks of data information but gives the optimal order of approximation. That is, we have a finite sequence $\{w_{-k_m}^{(m)}, \ldots, w_{k_m}^{(m)}\}$ such that

(10) $$(S_M f)(x) = \sum_j \left\{ \sum_i w_{j+\frac{m}{2}-i\ell_m}^{(m)} f\Big(\frac{i}{2^M}\ell_m\Big) \right\} N_m(2^M x - j)$$

satisfies:

(11) $$(S_M f - f)\Big(\frac{j}{2^M}\ell_m\Big) = 0, \quad j \in \mathbf{Z},$$

and

(12) $$\|S_M f - f\|_\infty = O\Big(\frac{1}{2^{mM}}\Big), \quad M \to \infty,$$

for all $f \in C^m$. Here, $w_p^{(m)} := 0$ if $|p| > k_m$; ℓ_m is a positive integer depending on the order m of the splines (e.g. $\ell_2 = 1$ and $\ell_4 = 2$, cf. C [2,3]); and $M \in \mathbf{Z}$ with $\ell_m/2^M$ denoting

the sampling rate. Note that $S_M f$ is an m^{th} order spline with knots at $2^{-M}\mathbf{Z}$. Also, the length of the "weight" sequence $\{w_i^{(m)}\}_{i=-k_m}^{k_m}$ also depends on m. It is trivial that $k_2 = 0$ and $w_0^{(2)} = 1$ for linear splines. For cubic splines, we have $k_4 = 4$ (i.e. $w_p^{(4)} = 0$ for $|p| > 4$) and

$$\begin{cases} w_{-4}^{(4)} = w_4^{(4)} = \dfrac{1}{48} \\ w_{-3}^{(4)} = w_3^{(4)} = -\dfrac{1}{12} \\ w_{-2}^{(4)} = w_2^{(4)} = -\dfrac{1}{8} \\ w_{-1}^{(4)} = w_1^{(4)} = \dfrac{7}{12} \\ w_0^{(4)} = \dfrac{29}{24} \end{cases}$$

(cf. CD [4]).

2.2. DECOMPOSITION ALGORITHM. From the decomposition formula (7) and the scale-change notation in (2) and (4), we have

$$N_{m;k+1,j} = \sum_n a_{j-2n} N_{m;k,n} + \sum_n b_{j-2n} \psi_{m;k,n}. \tag{13}$$

Suppose that the samples

$$\left\{ f\left(\frac{j\ell_m}{2^M}\right) \right\}, \quad j \in \mathbf{Z} \tag{14}$$

are used to design the interpolatory m^{th} order spline curve with knots at $2^{-M}\mathbf{Z}$. Here, M is a sufficiently large positive integer. Set

$$\begin{cases} f_M(x) = (S_M f)(x) = \sum_j c_j^M N_{m;M,j}(x) \\ \mathbf{c}^M = \{c_j^M\}, \ j \in \mathbf{Z}. \end{cases}$$

Recall from (11) and (12) that f_M interpolates f at $\{j\ell_m/2^M\}$, $j \in \mathbf{Z}$, and yields the m^{th} order of approximation to f. Hence, we have

$$c_j^M = \sum_i w_{j+\frac{m}{2}-i\ell_m}^{(m)} f\left(\frac{i\ell_m}{2^M}\right) \tag{15}$$

where we recall that $w_p^{(n)} = 0$ for $|p| > k_m$, and m is even since we only use even order splines for interpolation. For $k < M$, we also set

$$\text{(16)}\qquad \begin{cases} f_k(x) = \sum_j c_j^k N_{m;k,j}(x) \\ \mathbf{c}^k = \{c_j^k\},\ j \in \mathbf{Z}. \end{cases}$$

where f_k is the projection of f_{k+1} from V_{k+1} to V_k as indicated by (3). Let $g_k \in W_k$ be its orthogonal complement; that is, $g_k \perp f_k$ and $f_{k+1} = f_k + g_k$. Write

$$\text{(17)}\qquad \begin{cases} g_k(x) = \sum_j d_j^k \psi_{m;k,j}(x) \\ \mathbf{d}^k = \{d_j^k\},\ j \in \mathbf{Z}. \end{cases}$$

Then $\{\mathbf{c}^k\}$ and $\{\mathbf{d}^k\}$, $k = M-1, \ldots, M-L$ (where L is arbitrary) can be computed by applying the following recursive formula:

$$\text{(18)}\qquad \begin{cases} c_j^k = \sum_n a_{n-2j} c_n^{k+1} \\ d_j^k = \sum_n b_{n-2j} c_n^{k+1}. \end{cases}$$

Hence, starting from $\mathbf{c}^M$ in (15), we can easily compute $\mathbf{c}^{M-1}, \mathbf{d}^{M-1}; \mathbf{c}^{M-2}, \mathbf{d}^{M-2}; \ldots; \mathbf{c}^{M-L}, \mathbf{d}^{M-2}$ by applying (18). That is, by using (16) and (17), we have a (mutually) orthogonal decomposition:

$$\text{(19)}\qquad f_M(x) = g_{M-1}(x) + \cdots + g_{M-L}(x) + f_{M-L}(x).$$

where $f_M(x)$ interpolates $f(x)$ at $\{j\ell_m/2^M\}$, $j \in \mathbf{Z}$, and approximates $f(x)$ with the m^{th} order of approximation, and where

$$\text{(20)}\qquad \|f_{M-L}\|_\infty \to 0, \quad L \to \infty.$$

It will be clear that the orthogonal decomposition (19) has significant information on the curve $y = f_M(x)$. To verify (18), we need the following result.

LEMMA 1. *Let $\xi \in L^1 \cap L^2$ on $(-\infty, \infty)$ be non-trivial. Then $\{\xi(\cdot - j)\colon j \in \mathbf{Z}\}$ is ℓ^2-linearly independent, in the sense that*

$$\text{(21)}\qquad \sum_{j \in \mathbf{Z}} a_j \xi(\cdot - j) = 0 \text{ a.e.},$$

where $\{a_j\} \in \ell^2$, *implies that* $a_j = 0$ *for all* j.

The proof of this lemma easily follows by using Fourier transform argument. Indeed, the Fourier transform of the left-hand side of (21) is the product of the symbol (or Fourier series) of $\{a_j\}$ and the Fourier transform $\hat{\xi}$ of ξ. Since $\hat{\xi}$ is nontrivial, then (21) holds if and only if the symbol of $\{a_j\}$ identically vanishes, or $a_j = 0$ for all j.

Now, we are ready to verify (18). From (13) and the definitions in (16) and (17), we observe that $f_{k+1}(x) = f_k(x) + g_k(x)$ if and only if

$$\sum_n \sum_j c_n^{k+1} a_{n-2j} N_{m;k,j}(x) + \sum_n \sum_j c_n^{k+1} b_{n-2j} \psi_{m;k,j}(x)$$
$$= \sum_j c_j^k N_{m;k,j}(x) + \sum_j d_j^k \psi_{m;k,j}(x).$$

Now, by Lemma 1, both sets $\{N_{m;k,j}\colon\ j \in \mathbf{Z}\}$ and $\{\psi_{m;k,j}\colon\ j \in \mathbf{Z}\}$ are ℓ^2-linearly independent. Hence, since these two sets are orthogonal to each other, we have verified (18).

2.3. RECONSTRUCTION ALGORITHM. Once we have obtained the orthogonal decompositoin (19), we can judge which portion of the curve $y = f_M(x)$ needs adjustment. Note that $g_k(x)$ contains the information on the portion of $f_M(x)$ with slope proportional to 2^k, since

$$\frac{d}{dx}(\psi_{m;k,j}(x)) = 2^k \psi'_m(2^k x - j).$$

Furthermore, this information is localized, in view of

$$\operatorname{supp}(\psi'_{m;k,j}) = \operatorname{supp}(\psi_{m;k,j}) = \left[\frac{j}{2^k}, \frac{j(2m-1)}{2^k}\right].$$

Of course, the magnitude of this slope is further multiplied by the coefficients d_j^k in addition to the 2^k factor. The orthogonality of this decomposition implies that the localized information is meaningful.

After certain portions of the curve $y = f_M(x)$ have been adjusted, say by replacing $\mathbf{d}^k$ with $\tilde{\mathbf{d}}^k$ and $\mathbf{c}^{M-L}$ with $\tilde{\mathbf{c}}^{M-L}$, we may reconstruct $\tilde{\mathbf{c}}^M = \{\tilde{c}_j^M\}$, which gives the modified spline curve

$$\tilde{f}_M(x) = \sum_j \tilde{c}_j^M N_{m;M,j}(x),$$

by applying the recursive formula:

$$\tilde{c}_j^{k+1} = \sum_n (p_{j-2n} \tilde{c}_n^k + q_{j-2n} \tilde{d}_n^k) \tag{22}$$

$n = M - L, \ldots, M - 1$.

To prove (22), we note from the pair of two-scale relations (8) and (9) that

$$(23) \qquad \begin{cases} N_{m;k,j}(x) = \sum_n p_n N_{m;k+1,2j+n}(x) = \sum_n p_{n-2j} N_{m;k+1,n}(x) \\ \psi_{m;k,j}(x) = \sum_n q_n N_{m;k+1,2j+n}(x) = \sum_n q_{n-2j} N_{m;k+1,n}(x). \end{cases}$$

Hence, $\tilde{f}_{k+1}(x) = \tilde{f}_k(x) + \tilde{g}_k(x)$ if and only if

$$\begin{aligned} &\sum_j \tilde{c}_j^{k+1} N_{j;k+1,j}(x) = \sum_j (\tilde{c}_j^k N_{m;k,j}(x) + \tilde{d}_j^k \psi_{m;k,j}(x)) \\ &= \sum_j \sum_n (\tilde{c}_j^k p_{n-2j} + \tilde{d}_j^k q_{n-2j}) N_{m;k+1,n}(x) \\ &= \sum_j \left\{ \sum_n (\tilde{c}_n^k p_{j-2n} + \tilde{d}_n^k q_{j-2n}) \right\} N_{m;k+1,j}(x). \end{aligned}$$

So, (22) follows by an application of Lemma 1.

3. THE RECONSTRUCTION AND DECOMPOSITION SEQUENCES. From (8), (9), and the definition of the spline wavelet $\psi_m(x)$ in (5), it is easy to derive the reconstrution sequences $\{p_n\}$ and $\{q_n\}$ for each fixed positive integer m. In fact, we have:

$$(24) \qquad p_n = \begin{cases} 2^{-m+1}\binom{m}{n} & \text{for } 0 \le n \le m \\ 0 & \text{otherwise} \end{cases}$$

and

$$(25) \qquad q_n = \begin{cases} \frac{(-1)^n}{2^{m-1}} \sum_{j=0}^{m} \binom{m}{j} N_{2m}(n - j + 1) & \text{for } 0 \le n \le 3m - 2 \\ 0 & \text{otherwise.} \end{cases}$$

To verify (25), one simply applies the B-spline identity

$$N_{2m}^{(m)}(x) = \sum_{j=0}^{m} (-1)^j \binom{m}{j} N_m(x - j)$$

(cf. [1, chap. 1]). To describe the decomposition sequences $\{a_n\}$ and $\{b_n\}$, we need the Euler-Frobenius polynomials

$$\Pi_{2m-1}(z) = (2m-1)! \sum_{n=0}^{2m-2} N_{2m}(n+1) z^n$$

(which are polynomials of degree $2m-2$). It is well known (cf. S [12]) that $\Pi_{2m-1}(z)$ has $2m-2$ distinct (real) negative roots which are in reciprocal pairs; that is, arranging $\{r_j\}$ in decreasing order, we have

$$r_{2m-2} < r_{2m-1} < \cdots < r_m < -1 < r_{m-1} < \cdots < r_1 < 0$$

with $r_1 r_{2m-2} = \cdots = r_{m-1} r_m = 1$. In particular, since $\Pi_{2m-1}(z) \neq 0$ for $|z| = 1$, its reciprocal has an absolutely convergent Fourier series expansion:

$$\frac{1}{\Pi_{2m-1}(z)} = \frac{1}{(2m-1)!} \sum_{n \in \mathbf{Z}} \alpha_n^{(m)} z^{n-m+1}, \qquad |z| = 1, \tag{26}$$

where

$$|\alpha_n^{(m)}| = 0(|r_m|^{-|n|}), \qquad |n| \to \infty. \tag{27}$$

The *decomposition sequences* $\{a_n\}$ and $\{b_n\}$ are given by

$$a_n = \frac{1}{2\pi i} \int_{|z|=1} \frac{(1+z)^m}{2^m} \frac{\Pi_{2m-1}(z)}{\Pi_{2m-1}(z^2)} z^{n-2} dz \tag{28}$$

and

$$b_n = \frac{1}{2\pi i} \int_{|z|=1} \frac{-(2m-1)!}{2^m} \frac{(1-z)^m}{\Pi_{2m-1}(z^2)} z^{n-2} dz. \tag{29}$$

Hence, in view of (27), we have

$$|a_n|, |b_n| = 0(|r_m|^{-\frac{|n|}{2}}), \qquad |n| \to \infty. \tag{30}$$

Here, the larger the order m of splines we use, the smaller is the exponent $|r_m|$ (cf. S [11]).

It is possible to interchange the pair $(\{a_n\}, \{b_n\})$ of decomposition sequences and the pair $(\{p_n\}, \{q_n\})$ of reconstruction sequences. This is the so-called duality principle introduced in CW [7]. To do so, we simply replace the B-splines $N_m(x-j)$ and wavelets $\psi_m(x-j)$ by (an appropriate shift of) their dual bases. Details are given in CW [7].

4. FINAL REMARKS. Although we have obtained the *same* wavelet decomposition (19) as M [10,11], we have uesd different bases $\{N_{m;k,j}\}$ and $\{\psi_{m;k,j}\}$. In M [10.11], in order to use Fourier methods, the B-splines $\{N_{m;k,j}\}$ are orthonormalized, and thus orthonormal wavelets are also obtained. For curve design, however, it is much more efficient to use the B-spline series in order to keep the local structure. As a bonus, we have two finite sequences $\{p_n\}$ and $\{q_n\}$ for reconstruction. If the decomposition sequences $\{a_n\}$ and $\{b_n\}$ are also required to be finite, then it is proved in CW [8] that we no longer have spline curves. In fact, it is proved in CW [8] that this requirement is achieved only by the compactly supported orthonormal wavelets of Daubechies (cf. D [9]) which have no explicit formulations and are not as smooth as the spline wavelets $\psi_m(x)$ in this paper.

REFERENCES

1. Chui, C.K., *Multivariate Splines*, CBMS-NSF Series in Applied Math. #54, SIAM Publications, Philadelphia, 1988.
2. Chui, C.K., Vertex splines and their applications to interpolation of discrete data, in *Computation of Curves and Surfaces*, Ed. by W. Dahmen, M. Gasca, and C.A. Micchelli, Kluwer Academic Publishers, 1990, 137-181.
3. Chui, C.K., Construction and applications of interpolation formulas, in *Multivariate Approximation and Interpolation*, Ed. by W. Haussman and K. Jetter, ISNM Series in Math., Birkhäuser Verlag, Basel. To appear.
4. Chui, C.K. and Diamond, H., A general framework for local interpolation, CAT Report #190, Texas A&M University, College Station, 1989.
5. Chui, C.K. and Diamond, H., Approximation and interpolation formulas for real-time applications, Proc. Seventh Army Conf. on Appl. Math. and Comp. (1990), 765-772.
6. Chui, C.K. and Wang, J.Z., A cardinal spline approach to wavelets, CAT Report #211, Texas A&M University, College Station, 1990.
7. Chui, C.K. and Wang, J.Z., On compactly supported spline wavelets and a duality principle, CAT Report #213, Texas A&M University, College Station, 1990.
8. Chui, C.K. and Wang, J.Z., A general framework of compactly supported splines and wavelets, CAT Report #219, Texas A&M Uniersity, College Station, 1990.
9. Daubechies, I., Orthonormal bases of compactly supported wavelets, Comm. Pure and Appl. Math. **41** (1988), 909-996.
10. Mallat, S.G., Multiresolution approximation and wavelet orthonormal bases of $L^2(\mathbf{R})$, Trans. Amer. Math. Soc. **315** (1989), 69-87.
11. Mallat, S.G., Multifrequency channel decompositions of images and wavelet models, IEEE Trans. ASSP **37** (1989), 2091-2110.
12. Schoenberg, I.J., *Cardinal Spline Interpolation*, CBMS-NSF Series in Applied Math. #12, SIAM Publications, Philadelphia, 1973.

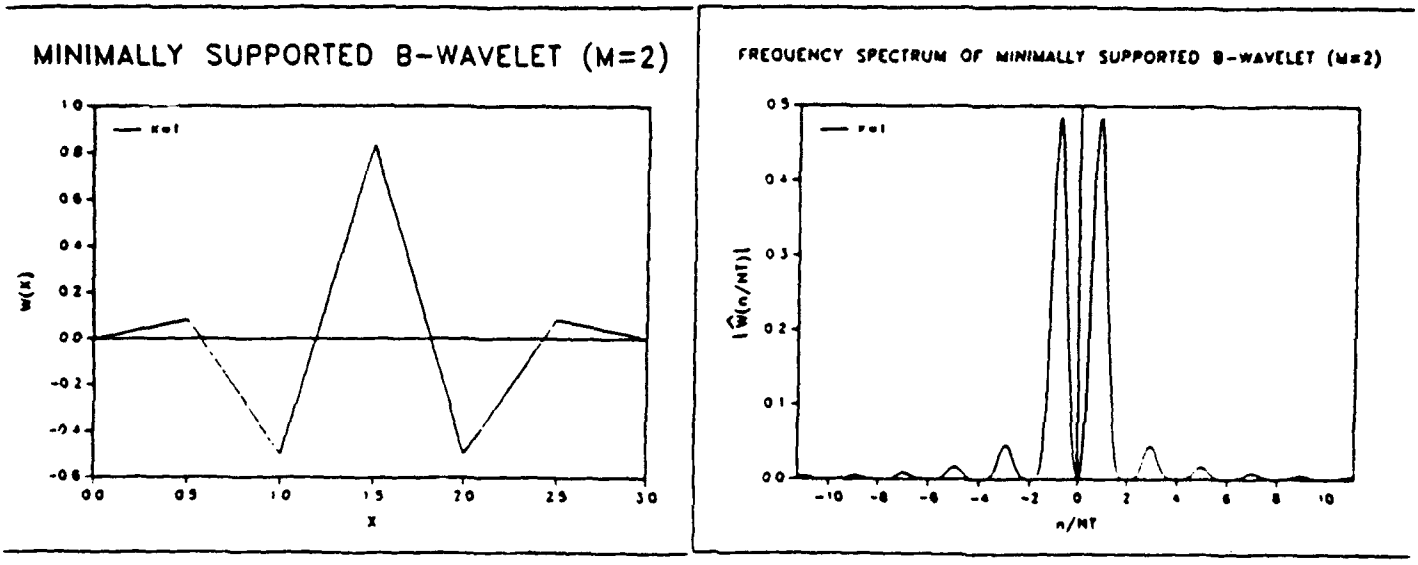


Fig. 1 (Linear B-wavelet and its magnitude spectrum)

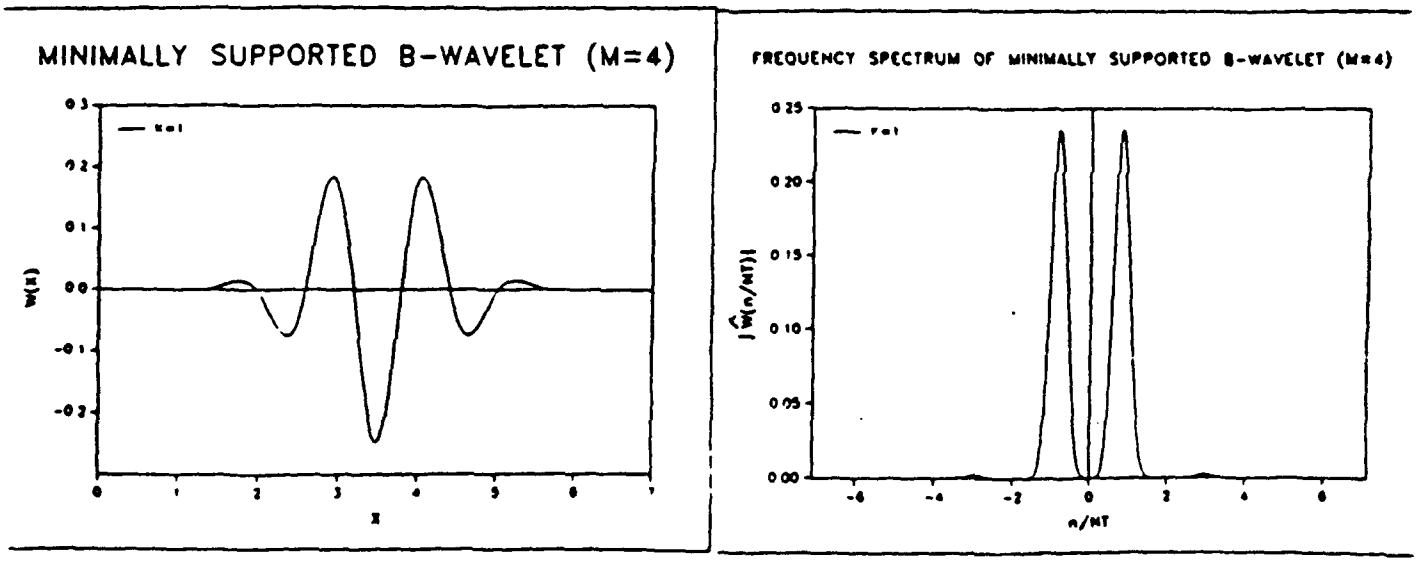


Fig. 2 (Cubic B-wavelet and its magnitude spectrum)

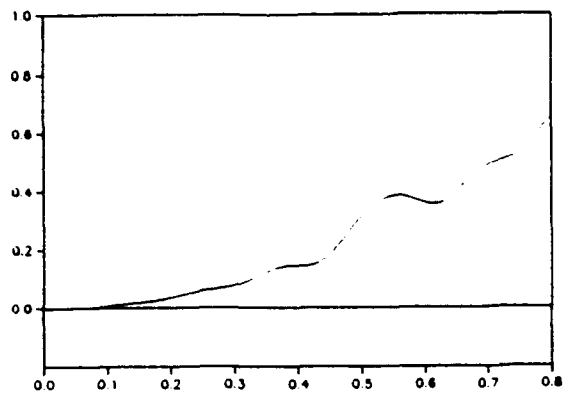
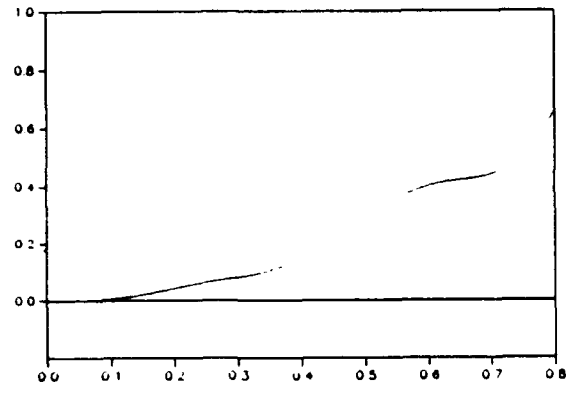
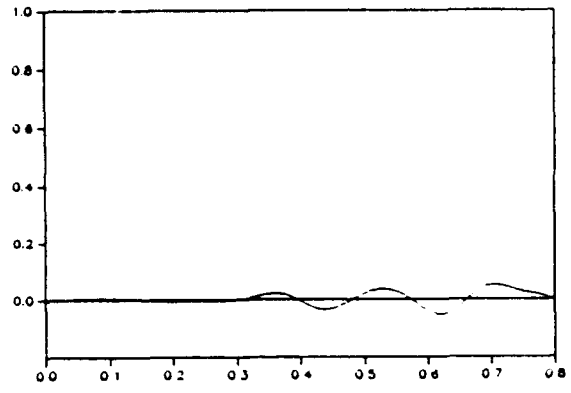
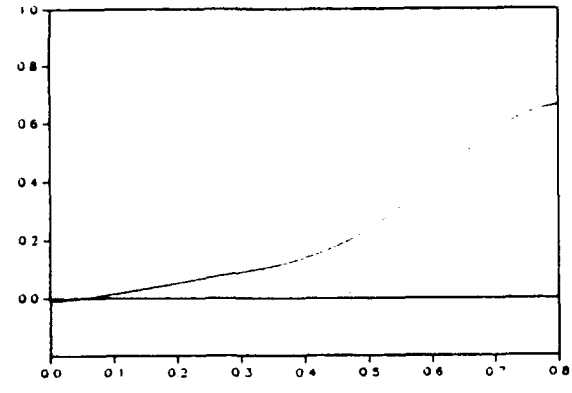
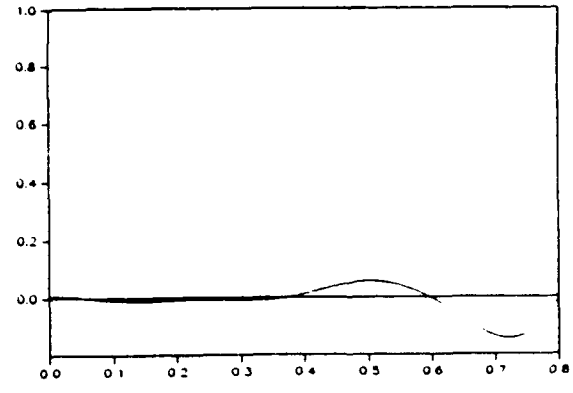


Fig. 3 (Cubic Spline/Wavelet decomposition of curves)









Multiple Families of Engineering Analyses Interrogating a Single Geometric Model

Michael John Muuss

U. S. Army Ballistic Research Laboratory
Aberdeen Proving Ground
Maryland 21005-5066 USA

ABSTRACT

As a new product is being designed, there are generally two or more different kinds of engineering analyses that need to be performed. It is still often the case that in the course of developing a new product design, several geometric models may have to be constructed, because different engineering analysis software packages generally require different forms of input. Irrespective of the specific representation used for a three-dimensional geometry/material database, rarely does an application code read that database directly. Rather, a specific interrogation method is used to pass particular geometric/material attributes to the application code.

In this paper, the strategy used in the BRL-CAD Package is presented. A variety of procedural interfaces have been provided so that diverse analysis codes can be driven from a single, central geometric model. These interfaces include the ability to produce a wireframe representation of the model, intersect rays with the model, tessellate the model into a 3-D surface mesh, and perform topological feature extraction. In addition, extensions to the software will be discussed, including the ability to approximate the model as a 3-D finite-element volume mesh, and converting the model to a homogeneous trimmed B-Spline representation.

June 19, 1990

Multiple Families of Engineering Analyses Interrogating a Single Geometric Model

Michael John Muuss

U. S. Army Ballistic Research Laboratory
Aberdeen Proving Ground
Maryland 21005-5066 USA

1. Introduction

As a new product is being designed, there are generally two or more different kinds of engineering analyses that need to be performed. For example, vulnerability analysis, structural analysis, thermal analysis, and computational fluid dynamics (CFD), as well as predictive radar, infra-red (IR), and X-ray signatures. At any stage of the design, almost any kind of CAD system can be used to make engineering drawings. However, drawings are suitable only for interpretation by human beings, not for automatic computerized analysis. Therefore, computerized drawings can not be used to provide the geometric input required for these analysis codes.

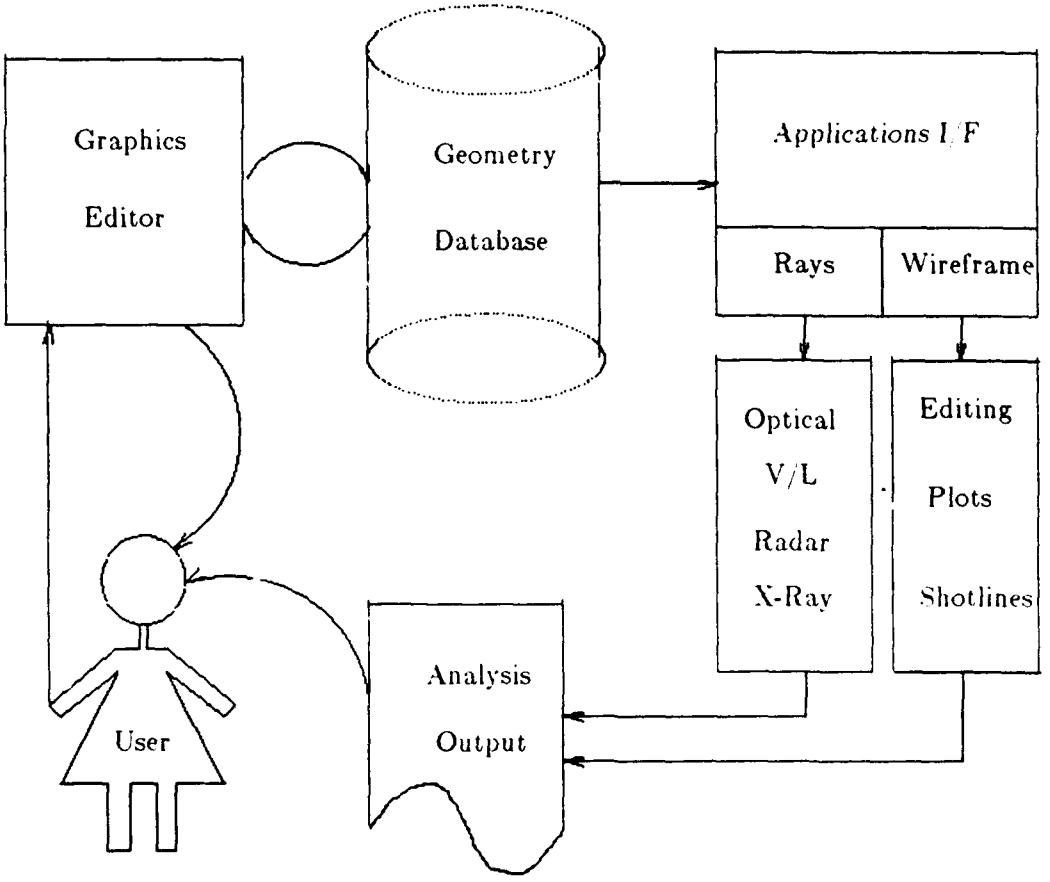


Figure 1 — The Design Loop

If the product is designed, not on a computer drafting system, but instead on a full 3-D solid

modeling system, then in addition to producing drawings, the model can be subjected to numerous engineering analyses, allowing the effects of varying many parameters to be studied in a controlled and automatic way. Thus, the real payoff from building a solid geometric model comes when it is time to analyze it. This capability is so powerful that it ordinarily justifies any extra time or equipment investments needed to support the construction of the 3-D solid model.

2. Solid Modeling and the Design Process

For more than twenty years, the Ballistic Research Laboratory (**BRL**) has been utilizing solid geometric modeling methods to support vulnerability/lethality and neutron transport studies of military targets. [1,2] In such item-level studies target geometry and material information are passed to various application codes to derive certain measures-of-performance. Building on the general paradigm, the BRL and many others have extended the general techniques to support many predictive signature models[3] including optical, millimeter wave (MMW), infra-red (IR), magnetic and X-ray.

It is important to note that this type of analysis must generally be supported by a solid geometric model. A solid model[4] is a computer description of closed, solid, three-dimensional shapes represented by an analytical framework within which the three-dimensional material can be completely and unambiguously defined. Properly utilized, the solid model becomes the central element in the iterative process of taking a design from idea to prototype design to working design to optimized design. This iterative process is termed the "design loop". The early form of the design loop is illustrated in Figure 1.

In a full scale solid modeling system, there is no need for initial drawings; the designer expresses the initial structures directly into the modeling system's editor, just as a modern author creates his "rough draft" directly into a word processor. At the completion of each version of the design, the model is subjected to a battery of analyses appropriate to the function of the object being designed. Strength, volume, weight, level of protection, and other similar evaluations can be reported, along with the production of a variety of images and/or drawings. These automated analyses help identify weaknesses or deficiencies in a design *early in the design process*. By detecting flaws early, the designer has the opportunity to correct his plans before having invested too much time in a bad design, or the designer can switch to an entirely different approach which may seem more promising than the original one. In this way, the solid modeling system allows the designer to concentrate on the important, creative aspects of the design process. Freeing the designer of routine analysis permits designs to be finished in less time than previously required, or allows much more rigorously optimized designs to be delivered in comparable timeframes and at the same cost as unoptimized designs created using older techniques.[5] Furthermore, the modeling system allows sweeping design changes to be made quickly and cheaply, allowing great flexibility in the face of ever changing requirements and markets. The time needed to create a new product can be further decreased by re-utilizing elements of earlier models and then modifying them as appropriate. If an existing component already in inventory is entirely suitable for use in a new design, significant manufacturing and inventory savings will be realized.

Two major families of solid model representations exist, each with several unique advantages. The first is the Combinatorial Solid Geometry **Rep**resentation (**CSG-Rep**).[6] Solid models of this type are expressed as boolean combinations of primitive solids which are geometric entities described by some set of parameters and occupying a fixed volume in space. The second is the **B**oundary **Rep**resentation (**B-Rep**), of which there are two sub-types, *explicit*, where each solid is described by an explicit enumeration of the extent of the surface of the solid, and *implicit*, where the surface of the solid is described by an analytic function such as a Coons patch, Bezier patch, B-spline, etc. Hybrid systems such as the BRL-CAD Package[7] also exist.

The objective of a given application will to a large degree determine the most "natural" form in which the model might be presented. For example, extracting just the *edges* of the objects in a model would be suitable for a program attempting to construct a wire-frame display of the model. Another family of applications exists which needs to be able to find the intersection of small object paths (eg. photons) with the model. Generally, these alternatives are motivated by

the representation of a physical process being simulated, and each alternative is useful for a whole family of applications.

Unfortunately, it is not often the case that building only *one* 3-D model of the product is enough. Each of the different engineering analysis software packages needed to perform the analyses usually requires a different form of input. As a result, more than one kind of geometric model may have to be constructed. However, rarely do these application codes read the three-dimensional geometry/material data base directly. Rather, each application has a specific interrogation method that is invoked to obtain geometric and material attributes from a source or reference file. The physical simulation techniques used in the application software are therefore constrained by the available techniques for extracting geometric information from the model. As a result, each analysis package often requires a unique form of input. Without a central geometry database that can drive all the analysis packages, the designer can be forced into having to create many different representations of each design, one for each distinctly different type of analysis code. This can be very costly and time consuming; the time needed to create a single model ranges between 1 week and several man-years, depending on the complexity of the design. Having to spend the effort to manually create the same design in different formats to drive several analysis codes is an unfortunate and expensive necessity.

The philosophy adopted at BRL has been to develop a broad set of analyses which are supported from the ne geometry database.[3] These analyses cover the spectrum from engineering decision aids, to design validators, to signature prediction codes, to the generation of wireframe drawings, to high-resolution image generation for management comprehension and sales advantage. Key analysis capabilities have been developed to assess the strength, weight, protection, and performance levels offered by the structures represented by a solid model. Using this analysis information and additional domain-specific applications tools makes it possible to produce highly detailed designs constructed with a philosophy of *system optimization* right from the start.[8] This facilitates the rapid development of products with the desired levels of performance at the best attainable price.

To accomplish all these goals, the BRL-CAD Package[7] provides a variety of procedural interfaces so that the diverse collection of analysis codes can be driven from a single, central geometric model. These procedural interfaces follow the natural object-oriented programming interface. An application program retrieves one or more objects from the model database, and then requests those objects to either interrogate themselves in the desired way, or convert themselves into the desired representation. This applications interface is depicted in Figure 2.

3. Wireframe Representation

The interactive model editor **mged** program primarily employs 3-D wireframe outlines of the various solid objects, in order to maintain the highest possible speed of user interaction. The conversion of database objects into wireframe drawings is the simplest of the application interfaces provided by **librt**.

After the user specifies which objects from the model database should be displayed, **mged** retrieves the necessary database records and invokes the **ft_plot()** interface. **ft_plot()** passes the database object to the appropriate object-specific wireframe converter, which generates a wireframe outline of that object.

The wireframe is comprised of a collection of 3-D virtual pen-plotter *move* and *draw* operations, returned to the application as a linked list of **vlist** structures attached to the application provided **vlhead** structure. Each **vlist** structure has three elements, vl_pnt, the XYZ coordinates of a point in space, vl_draw, a flag which indicates whether the virtual pen should be moved invisibly from the current position to vl_pnt (vl_draw=VL_CMD_LINE_MOVE) or moved visibly, drawing a line from the current position to vl_pnt (vl_draw=VL_CMD_LINE_DRAW). There is also a vl_forw pointer to provide the linkage to the next **vlist** structure in the list.

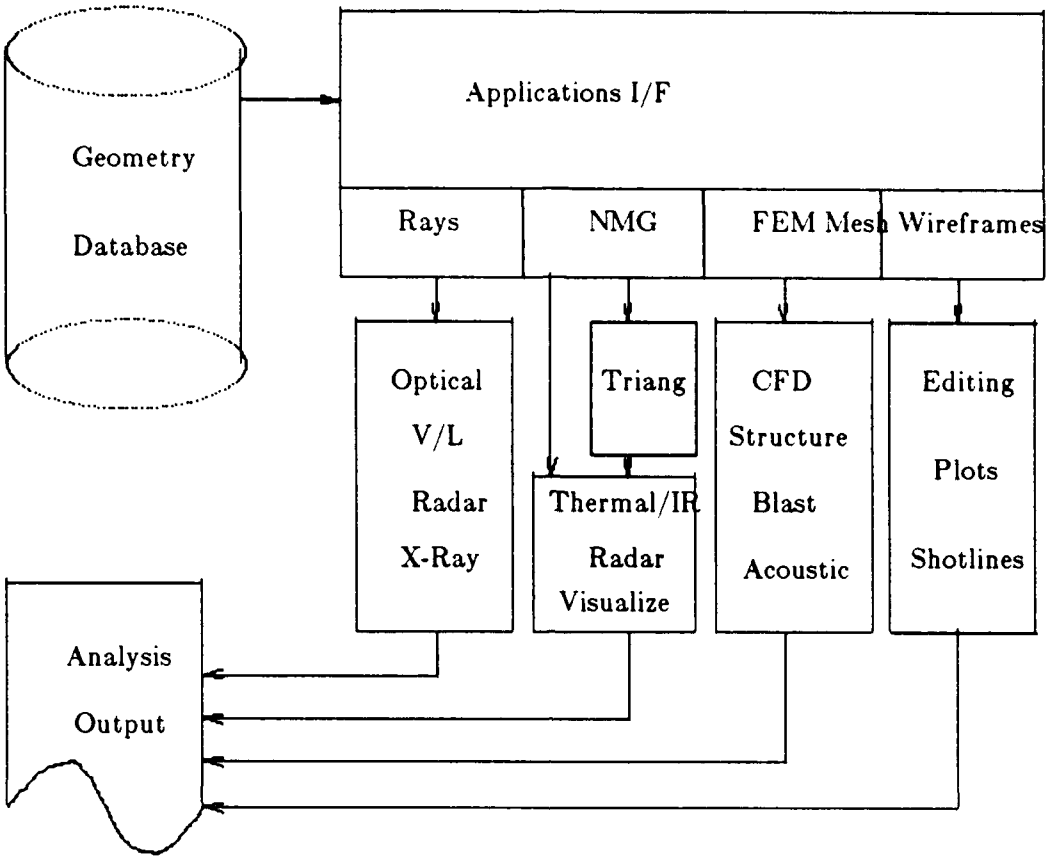


Figure 2 — The Applications Interface

4. Ray Tracing

Many phenomena that are ordinarily difficult to model can be handled simply and elegantly with ray-tracing. For example, an illumination model based on ray-tracing merely needs to fire a ray at each light source to determine the total light energy at each point. Ray-tracing also makes it easy to deal with objects that are partly or entirely reflective, and with transparent objects that have varying refractive indices. Furthermore, by applying the proper sorts of dither,[9] motion-blur, shadow penumbra, depth-of-field, translucency, and other effects are easily achieved.

The power of the lighting model code can further extended by making a provision to record the paths of all the rays followed when computing the light intensity for each pixel in an auxiliary file. This capability allows one to follow the path of the light rays passing through lenses reflecting from mirrors while performing image rendering, with no additional computation. Studying the paths of light rays as they are repeatedly bent by passing from air to glass and back again has traditionally been a painstaking manual procedure for lens designers. By modeling, it becomes possible to predict lens behavior, including making a determination of the exact focal length, finding the precise influence of spherical distortions and edge effects, determining the amount of image distortion due to internal reflection and scattering, and finding the level of reflections from the lens mounting hardware. Furthermore, experiments can be conducted to determine the effects of adding or removing baffles, irises, special lens coatings, etc.

Rays begin at a point $\vec{P}$, and proceed infinitely in a given direction $\vec{D}$. Any point $\vec{A}$ on a ray may be expressed as a linear combination of $\vec{P}$ and $\vec{D}$:

$$\vec{A} = \vec{P} + t * \vec{D}$$

where valid solutions for t are in the range $[0, \infty)$. (D_x, D_y, D_z) are the *direction cosines* for the

ray, being the cosine of the angle between the ray and the appropriate axis. While not necessary, in this analysis it is assumed that $\vec{D}$ is of unit length, i.e. $|\vec{D}|=1$.

The most traditional approach to ray-tracing has been batch-oriented, with the user defining a set of "viewing angles", turning loose a big batch job to compute all the ray intersections, and then post-processing all the ray data into some meaningful form. However, the major drawback of this approach is that the application has no immediate control over ray paths, making another batch run necessary for each level of reflection, etc.

In order to be successful, applications need: (1) interactive control of ray paths, to naturally implement reflection, refraction, and fragmenting into multiple subsidiary rays, and (2) the ability to fire rays in arbitrary directions from arbitrary points. Nearly all non-batch implementations have closely coupled a specific application (typically a model of illumination) with the ray-tracing code, allowing efficient and effective control of the ray paths. The most flexible approach of all is to provide the ray-tracing capability through a general-purpose library, and make the functionality available as needed to any application.

4.1. RT Library Interface

In order to give all applications interactive control over the ray paths, and to allow the rays to be fired in arbitrary directions from arbitrary points, BRL has implemented its third generation ray-tracing capability as a set of library routines. The RT library exists to allow application programs to intersect rays with model geometry. There are two parts to the interface: preparation routines and the actual ray-tracing routine. Three "preparation" routines exist; the first routine which must be called is **rt_dirbuild()**, which opens the database file, and builds the in-core database table of contents. The second routine to be called is **rt_gettree()**, which adds a database sub-tree to the active model space. **rt_gettree()** can be called multiple times to load different parts of the database into the active model space. The third routine is **rt_prep()**, which computes the space partitioning data structures, and does other initialization chores, prior to actual ray-tracing. Calling this routine is optional, as it will be called by **rt_shootray()** if needed. **rt_prep()** is provided as a separate routine to facilitate more accurate timing of the preparation and ray-tracing phases of applications.

To compute the intersection of a ray with the geometry in the active model space, the application must call **rt_shootray()** once for each ray. Ray behaviors such as perspective, reflection, refraction, etc, are entirely determined by the applications program logic, and not by the ray-tracing library. The ray-path specification determined by the applications program is passed as a parameter to **rt_shootray()** in the **application** structure, which contains five major elements: the vector a_ray.r_pt ($\vec{P}$) which is the starting point of the ray to be fired, the vector a_ray.r_dir ($\vec{D}$) which is the unit-length direction vector of the ray, the pointer *a_hit() which is the address of an application-provided routine to call on those rays where some geometry is hit by the ray, the pointer *a_miss() which is the address of an application-provided routine to call on those rays where the ray does not hit any geometry, the flag a_onehit, plus various locations for applications to store state (recursion level, colors, etc).

When the a_onehit flag is set to zero, the ray exhibits classical behavior, and is traced through the entire model. Applications such as lighting models may often only be interested in the first object hit: in this case, a_onehit may be set to the value one to stop ray-tracing as soon as the ray has intersected at least one piece of geometry. Similarly, if only the first three hits are required (such as in the routine that refracts light through glass), then a_onehit may be given the value of three. Then, at most three hit points will be returned, an in-hit, an out-hit, and a subsequent in-hit. When only a limited number of intersections are required, the use of this flag can provide a significant savings in run-time.

The return from the application provided a_hit()/a_miss() routine is the formal return of the function **rt_shootray()**. The **rt_shootray()** function is prepared for full recursion so that the application provided a_hit()/a_miss() routines can themselves fire additional rays by calling **rt_shootray()** recursively before deciding their own return value. In addition, the function **rt_shootray()** is fully prepared to be operating in parallel with other instances of itself in the

same address space, allowing the application to take advantage of parallel hardware capabilities where such exist.

4.2. Sample RT Application

A simple application program that fires one ray at a model and prints the result is included below, to demonstrate the simplicity of the interface to **librt**.

```
struct application ap;
struct rt_i *rtip;
main() {
    rtip = rt_dirbuild("model.g");
    rt_gettree(rtip, "car");
    rt_prep(rtip);
    VSET( ap.a_point, 100, 0, 0 );
    VSET( ap.a_dir, -1, 0, 0 );
    ap.a_hit = &hit_geom;
    ap.a_miss = &miss_geom;
    ap.a_rt_i = rtip;
    rt_shootray( &ap );
}
hit_geom(app, part)
struct application *app;
struct partition *part;
{
    printf("Hit %s", part->pt_forw->pt_regionp->reg_name);
}
miss_geom(){
    printf("Missed");
}
```

4.3. Ray Intersection Data

If a given ray hits some model geometry, the application provided routine indicated in the a_hit() pointer is called. A pointer to the head of a doubly-linked list of **partition** structures is provided. Each **partition** structure contains information about a line segment along the ray; the partition has both an "in" and an "out" hit point. Each hit point is characterized by the hit distance **hit_dist**, which is the distance t from the starting point **r_pt** along the ray to the hit point. The linked list of **partition** structures is sorted by ascending values of **hit_dist**.

As a result of this definition, the "line-of-sight" distance between any two hit points can be determined simply by subtracting the two **hit_dist** values. This will give the distance between the hit points, in millimeters.

If the flag a_onehit was set non-zero, then only the first a_onehit hit points along the partition list are guaranteed to be correct; any additional hit points provided should be ignored. This is usually important only when a_onehit was set to an odd number; the value of pt_outhit in the last **partition** structure may not be accurate, and should be ignored.

If the actual 3-space coordinates of the hit point are required, they can be computed into the **hit_point** element with the macro:

```
struct xray     *rayp;
struct hit      *hitp;
VJOIN1( hitp->hit_point, rayp->r_pt, hitp->hit_dist, rayp->r_dir );
```

which is the C-language version of

$$\vec{A} = \vec{P} + t * \vec{D}$$

4.4. Surface Normals

As an efficiency measure, only the hit distances are computed when a ray is intersected with the model geometry. For any hit point, the surface normal at that point can be easily acquired by executing the C macro:

```
struct hit      *hitp;
struct soltab   *stp;
struct xray     *rayp;
RT_HIT_NORM( hitp, stp, rayp );
```

In addition to providing the unit-length outward-pointing surface normal in hitp->hit_normal, this macro also automaticly computes the 3-space coordinates of the hit point in hitp->hit_point.

4.5. Gaussian Curvature

For any hit point, after the surface normal has been computed, the Gaussian surface curvature at that hit point can be acquired by executing the C macro:

```
struct curvature *curvp;
struct hit      *hitp;
struct soltab   *stp;
RT_CURVE( curvp, hitp, stp );
```

A **curvature** structure has three elements, **crv_pdir** the principle direction, **crv_c1** the curvature in the principle direction, and **crv_c2** the curvature in the other direction. The principal direction vector **crv_pdir** has unit length. |c1| <= |c2|, i.e. c1 is the most nearly flat principle curvature. A positive curvature indicates that the surface bends toward the (outward pointing) normal vector at that point. c1 and c2 are the inverse radii of curvature. The other principle direction is implied, and can be found by taking the cross product of the normal with **crv_pdir**, ie, pdir2 = normal x pdir1.

4.6. U-V Mapping

Both the U and V coordinates range from 0.0 to 1.0 inclusive. A given (U,V) coordinate may appear at more than one place on the surface of the object. The (U,V) coordinate of the hit point is returned in **uvcoord** structure elements uv_u and uv_v.

In addition, the approximate "beam coverage" of the ray, in U-V space, is returned in the structure elements uv_du and uv_dv. These approximate values are based upon the ray's initial beam radius (a_rbeam) and beam divergence per millimeter (a_diverge) as specified in the application structure. These delta-U and delta-V values can be helpful for anti-aliasing or filtering areas of the original texture map to produce an "area sample" value for the hit point.

For any hit point, after the value of **hit_point** has been computed, the U-V coordinates of that point can be acquired by executing the C macro:

```
struct application       *ap;
struct hit      *hitp;
struct soltab   *stp;
struct uvcoord  *uvp;
RT_HIT_UVCOORD( ap, stp, hitp, uvp );
```

For some simple lighting-model applications, it is sometimes desirable to create a mapping between the coordinate system on the surface of an object to the coordinate system of a square, the so-called "U-V" coordinates. This is generally used to drive simple 2-D *texture mapping* algorithms. The most common application is to extract a "paint" color from a rectangular RGB image file at coordinates (U,V), and apply this color to the surface of an object. These parameters can also be used to simulate the effect of minor surface roughness using the *bump mapping* technique. Here, the U and V coordinates index into a rectangular file of perturbation angles; the surface normal returned by RT_HIT_NORM() is then modified by up to plus or minus 90 degrees

each in both the U and V directions, according to the stored perturbation.

5. 3-D Surface Mesh

Combinatorial Solid Geometric (CSG) models are formed by the boolean combination of "primitive" solids.[4] For example, a plate with a hole is most easily modeled as a plate primitive minus a cylinder primitive. It is important to note that in CSG models, there is no explicit representation of the surfaces of the solids stored; indeed, for complex boolean combinations of complex primitives, some of the resultant shapes may have very convoluted topology and surfaces that may be at best high degree polynomials.

There are many applications that would benefit from being able to express an *approximation* of these complex shapes created using CSG modeling as a collection of planar N-gons which together enclose roughly the same volume of space as the original CSG solid. The most obvious such application is to drive polygon-based rendering routines (lighting modules) for predictive optical signatures. On many modern workstations there is direct hardware/firmware support for high-speed rendering of polygons. In addition, there are whole collections of 2.5-D infra-red predictive signature programs and 3-D polygon radar codes. The very best predictive radar signatures can be calculated using the Method of Moments, which requires having a 3-D surface tessellation to sub-wavelength resolution of the entire model.

A sensible strategy for converting a CSG model to the equivalent approximate 3-D surface mesh is to tackle the problem in two parts. First, a routine has to be written to convert each of the primitive solids into tessellated form. Second, a routine has to be written to take two tessellated objects and combine them according to a boolean operation (union, intersection, or subtraction) back into a consistent set of solid tessellated objects. Until very recently, it has been this second step that has proven extremely difficult.[10] The topology of solid tessellated objects has traditionally been represented using the "winged-edge" data structure. The major breakthrough is due to Kevin Weiler,[11,12] who noted that the "winged-edge" data structure was unable to represent non-3-Manifold conditions that often occur when performing boolean operations. Weiler proposed changing from the "winged-edge" data structure to the "radial-edge" data structure, suitable for representing all the Non-Manifold Geometric (**NMG**) and topological configurations that boolean operations might produce. Thus, NMG objects are closed under boolean operations.

Employing the NMG representation for faceted solid objects gives rise to the rich set of possibilities diagramed in Figure 3. From this diagram it should be clear that the final evaluated NMG solid object can be employed in a variety of ways. The primary use will be for input to analysis codes that need an approximate 3-D surface mesh of the solid model. In this case, the NMG objects are sent across the interface, either directly into an application, or via a triangulator that turns the planar N-gon faces of the NMG objects into simple triangle lists, and thence to the application. However, a very powerful second use will be to create new faceted shapes which are then stored back in the database as new geometric objects, suitable for future editing or analysis.

While a detailed description of the NMG data structures is beyond the scope of this paper, there are several advantageous properties of the NMGs that are worth mentioning. The NMG representation maintains full topology information, so that the relationships between vertices, edges, loops, faces, and shells are continuously available. The geometry information associated with a planar face is the plane equation (which includes the outward-pointing surface normal); the plane equation does not have to be re-derived from the vertices. For applications that would prefer visual realism rather than geometric fidelity, there is room in the vertex geometry structure to carry around a "phony" normal for each vertex, suitable for use in Gouraud shading[13] algorithms.

One of the most exciting current research projects at BRL is the extension of the NMG framework to permit faces either to be planar N-gons, or trimmed non-uniform rational B-splines ("trimmed NURBS"). This will permit many of the tessellation operations to be implemented exactly, rather than as approximations. This will also permit solids to enjoy the economy of having most faces be represented as planar N-gons, which are very compact and efficient to process, while those few faces that require sculptured surface shape control can be represented as trimmed

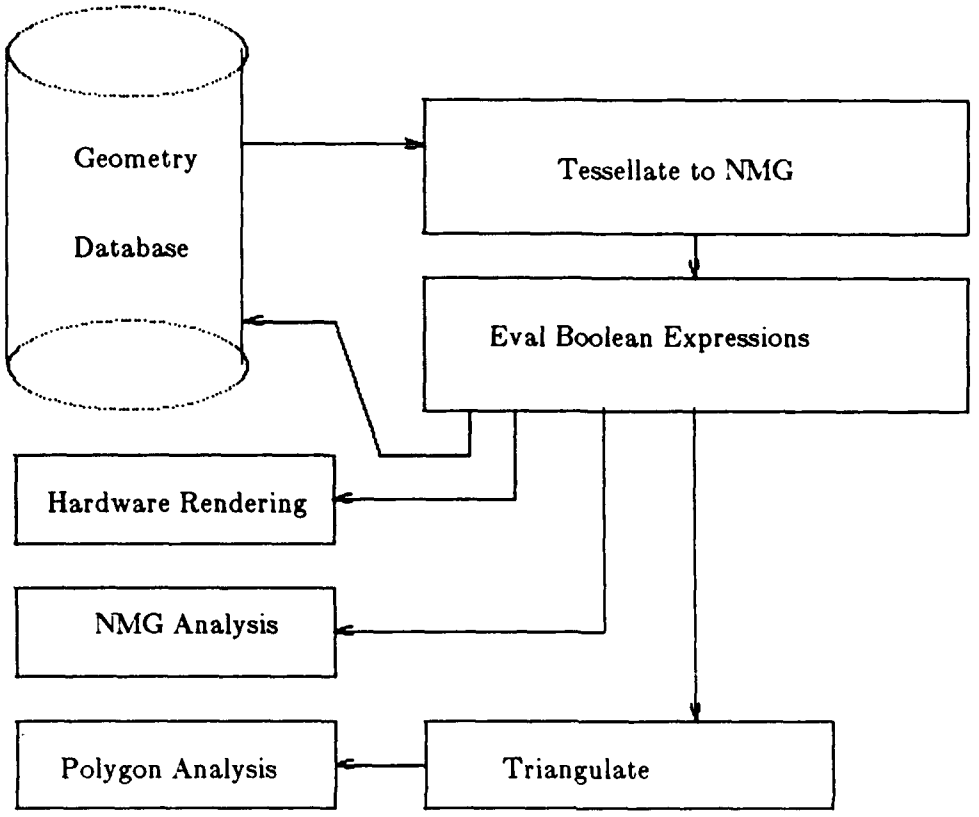


Figure 3 — NMG Wiring Diagram

NURBS. This combination provides both efficiency and full shape control in the rich non-Manifold topological framework; a combination that does not exist in any current commercial CAD system.

6. Topological Representation

Some predictive radar signature codes, such as the TRACK code of GTRI,[14] do not operate directly on a geometric representation of an object. Instead, they rely on the fact that large radar returns occur primarily due to the existence of dihedral and trihedral structures in the object. The original model is analyzed to locate all instances of the topological features of interest, for example, planar face elements, edges where two locally planar elements join to make a dihedral, edges where three locally planar elements join to make a trihedral, etc. Then this list of topological features is used as input to the feature-based analysis code.

Due to the rather broad set of possible interpretations of the term "feature", each kind of topological feature extraction is itself considered an application program, rather than being implemented as a standard part of the interrogation library. The process of topological feature extraction is currently programmed using the interrogation features described above.

7. Extensions

To date, most BRL-CAD applications programs have been implemented using the ray-tracing paradigm, because of ray-tracing having a 20-year head start. By choosing the ray sampling density within the Nyquist limit for a given spatial resolution, many applications based on ray-tracing are well satisfied by extracting ray/geometry intersection information. However, a mathematical ray has as its cross section a point, while physical objects have significant cross-sectional area. This lack of cross-sectional area will always lead to some sampling inaccuracies.

Applications which simulate particles or small rocks approaching the model might benefit from having a direct cylinder/geometry intersection capability, and applications which shine beams of light on the model such as spotlights or even highly collimated light such as laser light might benefit from cone/geometry intersection capabilities.[15,16] Applications which are attempting to simulate wave effects might be well expressed in terms of plane/geometry intersection curves, and structural analysis routines would probably prefer to obtain the geometry as a collection of connected hyperpatches.

While very recent research has begun to explore techniques for intersecting cylinders, cones, and planes with geometry,[17] ray-tracing and polygon-based techniques are by far the most well developed approaches. However, there are several additional type of interface to the model geometry that are likely to be of general applicability. These interfaces are candidates for implementation in a future release of the BRL-CAD Package.

7.1. 3-D Finite-Element Volume Mesh

Many forms of energy flow analysis, such as heat flow, vibrational analysis (acoustic energy flow), and stress analysis require the use of 3-D Finite-Element Mesh (FEM) techniques. While there has been some work on using the ray-tracing paradigm to construct finite element and finite difference meshes[18] it has been difficult to deal with high spatial frequency (fine detail) portions of the model. In particular, meshing small diameter pipes is problematic; undersampling can cause the pipe to incorrectly be separated into multiple pieces.

In order to improve on the current state of affairs, it seems necessary to provide support for the generation of volume meshes directly as part of the application interface. This would provide the meshing algorithm to have unrestricted access to the underlying geometry, the space partitioning tree, and other internal data in order to perform a better job.

Even more promising still would be a strategy that takes advantage of the NMG support. A first pass might tessellate the model and evaluate the booleans to produce a surface mesh. The second pass would then take the surface mesh and fill the interior (or exterior) volumes with appropriately chosen volume elements. A very good fit could probably be achieved using only parallelepiped ("brick") elements and 20-node "superelements". The brick elements would be used to fill interior volume that does not border on a face, and the superelements would be used for volume that contacts a face. Recourse could be made back to the underlying geometry (perhaps via firing a few well chosen rays) to get the curvature of the superelement faces to match the curvature of the underlying primitive, rather than having to relay strictly on the NMG planar-face approximation.

7.2. 3-D Volume Elements (Voxels)

A representation which is similar to the finite-element mesh is based on **Volume Elements** (**VOXELS**). There are two distinct kinds of voxels. The first kind of voxels can be considered a special case of volume meshing discussed previously, in which the model is "diced" into a large collection of homogeneous parallelepiped ("brick") elements. As one example, ERIM has a utility program which uses ray-tracing to covert BRL CSG-format geometry to this kind of voxel representation to feed a first-principles IR model.[19]

A distinctly different form of voxel representation is based upon the use of 8-way binary space subdivision stored using an "oct-tree" data structure. In this technique, the model is enclosed in a bounding box. The bounding box is evenly split along the X, Y, and Z axes to form eight smaller boxes. This algorithm is applied recursively so that all boxes which are neither entirely full nor entirely empty are repeatedly split, until the size of the voxels satisfies some termination condition. In this way, small voxels that lie along the surface of objects can fit arbitrarily tightly to the surface, while the interior of an isomorphic region will be contained primarily in a single large voxel. The oct-tree representation provides the application program with a homogeneous geometric representation based entirely on cubes of varying size. Having such a homogeneous representation can often greatly ease the task of algorithm development. On the other hand, achieving a good approximation of curved objects using cubes requires a huge number of

voxels to be used, resulting in very large voxel datasets, and an exponential increase (order N**6) in the number of element-to-element equations to be solved. The oct-tree approach to IR signature generation is employed in the BRL-CAD program, **lgt**.[20]

7.3. Homogeneous Trimmed B-Splines

When support for trimmed NURB faces has been added to the NMG capability, it will be possible to represent all existing primitives either with exact rational B-spline versions, or with very good rational B-spline approximations. This could be done even for faces that were completely planar.

This offers the hope that it might be possible (albeit memory intensive) to convert an entire CSG solid model into a homogeneous collection of non-uniform rational B-spline faces organized in a non-manifold topological data structure. In addition to the conceptual simplicity afforded by having a uniform representation for shape, this affords the opportunity to create new analysis codes that can process curved surfaces, yet at least initially only have to deal with one kind of shape. This would also provide a very direct and natural interface to spline-based[21] and Bezier-patch[22] based modeling systems.

7.4. Analytic Analysis

Given a homogeneous geometric representation such as the Trimmed B-Splines just discussed which also has an analytic representation, a further processing capability arises. Rather than interrogating the data base by means sampling or subdivision techniques, the direct mathematical manipulation of the source geometry through its parametric representation becomes possible. Calculations of physical properties requiring integration over a surfaces can often be evaluated with greater accuracy using an explicit analytic calculation than would be provided by numerical evaluation. While this may be difficult in general due to the complexity of the parametric expression, some classes of surface representations good candidates. Splines, for example, are piecewise-polynomial functions which have relatively simple Fourier transform representations. Since 2-D spatial Fourier transforms arise frequently in far-field electromagnetic scattering calculations, exploitation of the parametric spline representation is of interest in predictive scattering calculations. Direct use of spline parameters in a Physical Optics scattering model is part of the methodology used at the Aircraft Division, Northrop Corporation.

With the rapidly developing potential of symbolic calculation, treatment of seemingly impossible formulas resulting from the geometry/physics interaction may become tenable. This could help to reduce the trend towards employing numerical methods at the onset of a problem and avoid some of the accompanying instabilities and errors.

8. Summary

Much of the power and flexibility of the BRL-CAD Package comes from the diversity of shapes that can be represented, and the variety of analysis interfaces that are available. Having a diversity of interface possibilities has permitted a wide variety of analysis codes to be driven from a single, central geometric model.

9. Acknowledgments

This work has been supported in part by the Joint Technical Coordinating Group-Munitions Effectiveness (JTCG-ME), Smart Munitions Working Group, under grants monitored by Julian A. Chernick, US Army Materiel Systems Analysis Activity, Aberdeen Proving Ground, MD, 21005-5066.

References

1. P. H. Deitz, *Solid Geometric Modeling - The Key to Improved Materiel Acquisition from Concept to Deployment*, Defense Computer Graphics 83, Washington DC (10-14 October 1983).

2. P. H. Deitz, *Modern Computer-Aided Tools for High-Resolution Weapons System Engineering*, DoD Manufacturing Technology Advisory Group MTAG-84 Conference, Seattle WA (25-29 November 1984).

3. P. H. Deitz, *Predictive Signature Modeling via Solid Geometry at the BRL*, Sixth KRC Symposium on Ground Vehicle Signatures, Houghton MI (21-22 August 1984).

4. M. J. Muuss, "Understanding the Preparation and Analysis of Solid Models," in *Techniques for Computer Graphics*, ed. D. F. Rogers, R. A. Earnshaw, Springer-Verlag (1987).

5. P. H. Deitz, *The Future of Army Item-Level Modeling*, Army Operations Research Symposium XXIV, Ft. Lee VA (8-10 October 1985).

6. MAGI, *A Geometric Description Technique Suitable for Computer Analysis of Both Nuclear and Conventional Vulnerability of Armored Military Vehicles*, MAGI Report 6701, AD847576 (August 1967).

7. M. J. Muuss and P. Dykstra, K. Applin, G. Moss, P. Stay, C. Kennedy, *Ballistic Research Laboratory CAD Package, Release 3.0, A Solid Modeling System and Ray-Tracing Benchmark*, BRL Internal Publication (October 1988).

8. P. Deitz, W. Mermagen Jr, and P. Stay, "An Integrated Environment for Army, Navy, and Air Force Target Description Support," *Proceedings of the Tenth Annual Symposium on Survivability/Vulnerability*, (April 1988).

9. R. Cook and Porter, L. Carpenter, "Distributed Ray Tracing," *Computer Graphics (Proceedings of Siggraph '84)* **18**(3) pp. 137-145 (July 1984).

10. David H. Laidlaw, W. Benjamin Trumbore, and John F. Hughes, "Constructive Solid Geometry for Polyhedral Objects," *Computer Graphics* **120**(4)Proceedings of SIGGRAPH 86, (August 1986).

11. Kevin J. Weiler, "Edge-based Data Structures for Solid Modeling in Curved-Surface Environments," *IEEE Computer Graphics and Applications* **5**(1) pp. 21-40 (January 1985).

12. Kevin J. Weiler, "The Radial Edge Structure: a Topological Representation for Non-Manifold Geometric Modeling," in *Geometric Modeling for CAD Applications*, ed. M. Wozny, H. McLaughlin, and J. Encarnacao, Springer Verlag (December 1987).

13. H. Gouraud, "Continuous Shading of Curved Surfaces," *IEEE Transactions on Computers* **C-20**(6) pp. 623-628 (June 1971).

14. John Pelfer, *Georgia Tech Research Institute Radar Cross Section Modeling Software*, Modeling and Analysis Division, Georgia Tech Research Institute (October 1986).

15. J. Amanatides, "Ray Tracing with Cones," *Computer Graphics (Proceedings of Siggraph '84)* **18**(3)(July 1984).

16. D. B. Kirk, "The Simulation of Natural Features Using Cone Tracing," pp. 129-144 in *Advanced Computer Graphics*, ed. T. L. Kunii, Springer-Verlag (1986).

17. J. T. Kajiya, "New Techniques for Ray Tracing Procedurally Defined Objects," *Transactions of Graphics* **2**(3) pp. 161-181 (July 1983).

18. G. Laguna, *Recent Advances in 3D Finite Difference Mesh Generation Using the BRL-CAD Package*, BRL-CAD Symposium '89, Aberdeen Proving Ground, MD (24-25 October, 1989).

19. B. E. Morey, S. R. Stewart, and K. E. Gunderson, "Infrared Image Simulation of Threat Vehicles," *The Proceedings of the Test Technology Symposium*, (18-20 April 1989).

20. Gary S. Moss, "The "lgt" Lighting Model," *BRL-CAD Symposium '88*, p. 11 (28-29 June 1988).

21. D. F. Rogers and J. A. Adams, *Mathematical Elements for Computer Graphics, 2nd ed.*, McGraw-Hill, New York (1990).

22. P. E. Bezier, *Mathematical and Practical Possibilities of UNISURF*, Academic Press, New York (1974).

PARALLEL ALGORITHMS FOR FLUID INTERFACE PROBLEMS *

Yuefan Deng, James Glimm,
Yi Wang and Qiang Zhang

Department of Applied Mathematics and Statistics
The University at Stony Brook
Stony Brook, NY 11794-3600

Abstract

In this report, we present and analyze the results of applying parallel algorithms to a two dimensional gas dynamics code using front tracking. We also discuss the ideas to generalize the algorithm to three dimensions. The main purpose of this paper is to demonstrate that parallel computations can be applied to complex algorithms. We take front tracking as an example.

A number of computations for two dimensional fluid flow have been successful to simulate chaotic mixing, shock interactions and oil reservoir simulation. In most of these cases, further progress depends upon three dimensional computations. Performing systematic studies of chaotic mixing in three dimensions will require the extensive computer resources which parallel computations can offer.

In order to develop a reliable parallel programming paradigm for three dimensional studies, we first parallelize the relatively simpler serial two dimensional gas dynamics code on two representative distributed-memory MIMD parallel supercomputers, the iPSC/860 and the NCUBE/2.

The main algorithmic issues for front tracking in three dimensions are: (1) the construction of surface grids, (2) the construction of volume grids which are adapted to (i.e. which respect, or do not overlap with) specified surfaces, (3) the efficient computation of interface topology, (4) the resolution of self intersections in a tangled interface, and (5) parallel computation. Most of these issues are important for numerous methods of computation. We present methods for addressing these issues which are appropriate to the front tracking context.

*Supported by the Applied Mathematics Subprogram of the U.S. Department of Energy DE-FG02-90ER25084, the National Science Foundation, grant DMS-89018844, and the AFSOR, grant 90-0075.

1 Introduction

Significant effort has been devoted to developing parallel algorithms for scientific problems with natural load distribution. We concentrate on exploring parallelization for problems, where such uniform load distributions do not exist, such as the front tracking algorithms for fluid interfaces.

Front tracking [3, 7, 9, 10] is an algorithm which preserves, and explicitly recognizes, discontinuity surfaces as computational degrees of freedom. Marker points are introduced to define the location of an interface between computational domains.

This method has given very high quality solutions; the price to pay is probably the complexity of its implementation. Indeed, the value of this method has been demonstrated by a series of computations for two dimensional fluid flow. However, is it possible to generalize such a method to three dimensions? Is it feasible to introduce parallel computations to this method?

In order to answer these questions, we have performed a series of timing studies on three typical different environments for three dimensional front tracking algorithms [4]. The algorithms studied were interface topology algorithms which arise as a subproblem of gas dynamics simulations. To address the issue of feasibility of developing parallel computations is the main point of this paper in which we report initial results for the decomposition of a two dimensional fluid interface into sub-interfaces located on computational subdomains.

Techniques of generating a surface grid and an unstructured interior volume grid for three dimensional computational domains are also discussed. Several on-going aspects of this project will also be mentioned briefly.

It is typical that physical systems involve a number of materials that are separated by interfaces. Decomposing such a system into disjoint components separated by lower-dimensional physical objects (curves in two dimensions and surfaces in three dimensions), specifying, modifying, and moving dynamically such interfaces is the basis of front tracking or interface methods. The application of these algorithms to two dimensional hyperbolic systems was described by Glimm and McBryan [11] in 1985. The code has since then been used in a number of physical problems [2, 5, 6, 8].

2 The Parallelization Method: Domain Decomposition

Parallelizing this code is of immediate interest to prove the applicability of parallel computations in front tracking. Decomposing the self-contained interface data structures into sub-domains and then mapping these sub-domains onto the participating processors is our first step towards parallelization. Several requirements are imposed on the decomposition: At initialization, individual processors are activated to generate correctly positioned sub-interfaces while at re-start from an existing full interface, individual processors are to pick up the right portion of the interface. The distribu-

tion of computational load is rarely uniform; a heavier load appears mostly where the interface occurs. Moreover, a uniform distribution of the interface is not common. Therefore conventional uniform domain decomposition is not adequate for a good speedup. To decrease load im-balance, we consider non-uniform subdomains. For a number of problems in the scientific applications, we are considering interfaces that frequently cluster in a horizontal (vertical) direction, which leads us to decompose the interfaces into vertical (horizontal) strip-wise sub-interfaces.

Clipping a sub-interface from a full interface is conceptually straightforward. First, we determine the border curves that bound the sub-interface and the find all curves that intersect these border curves. Then, we split these intersected curves into two pieces, one of which lies within the border, and is inserted into the sub-interface. Finally, we find the intersection points and insert them as boundary nodes into the sub-interface. The locally "interior" elements (nodes and curves) are inserted unchanged.

The following four figures are included to show two typical decompositions we performed on an interface obtained from a study of bubble growth for Rayleigh-Taylor unstable interfaces This interface (Figure 1) is generated in a domain with $0 \leq x \leq 10$ and $0 \leq y \leq 20$ on a 120×240 mesh at the 750th time step. Starting from this interface, we first use 16 processors to decompose it into 15 sub-interfaces (one processor idle) with each being on a 40×48 mesh. In Figure 2, we draw the sub-interface clipped by processor 10. Of course, this represents a random choice. In Figure 3, we "glue" all 15 sub-interfaces obtained by the 15 processors in one frame with size comparable to the original interface for comparison. In Figure 4, we show a vertically strip-wise decomposition, with each sub-interface lying in a 20×240 mesh. As we can see from Figures 3 and 4, gathering the scattered sub-interfaces produces a perfectly identical interface to the orginal one (Figure 1). A quantitative comparison, though fairly tedious, also proves exact consistency.

We must realize that an unconventional domain decomposition, which may help reduce the load imbalance overhead, can cause significant communication penalty. We are currently studying the possibility for a optimal case. The answer, being machine-dependent, is not meaningful in general. We need to tune for possible optimizations for a particular machine.

In a message-passing computing environment, communication *per se* is important and interesting. We will report results on communication in a separate paper shortly. Here, we just outline the main ideas for reducing communication costs: (1) exchange minimum boundary information, which is possible due to the locality of the interface algorithm, transfer data extended two mesh blocks beyond a subdomain which is typically about 40 mesh blocks wide in each dimension (as in our example case above). (2) map, in the hypercube communication logic, the nearest-neighbor subdomains onto the possible nearest neighbor processors to reduce unnecessary costs due to crossing processors.

Another issue was software support for portability and ease of programming. We have tried to build the code on two representative distributed-memory MIMD parallel supercomputers: iPSC/860 and NCUBE/2 as well as a cluster of workstations, with

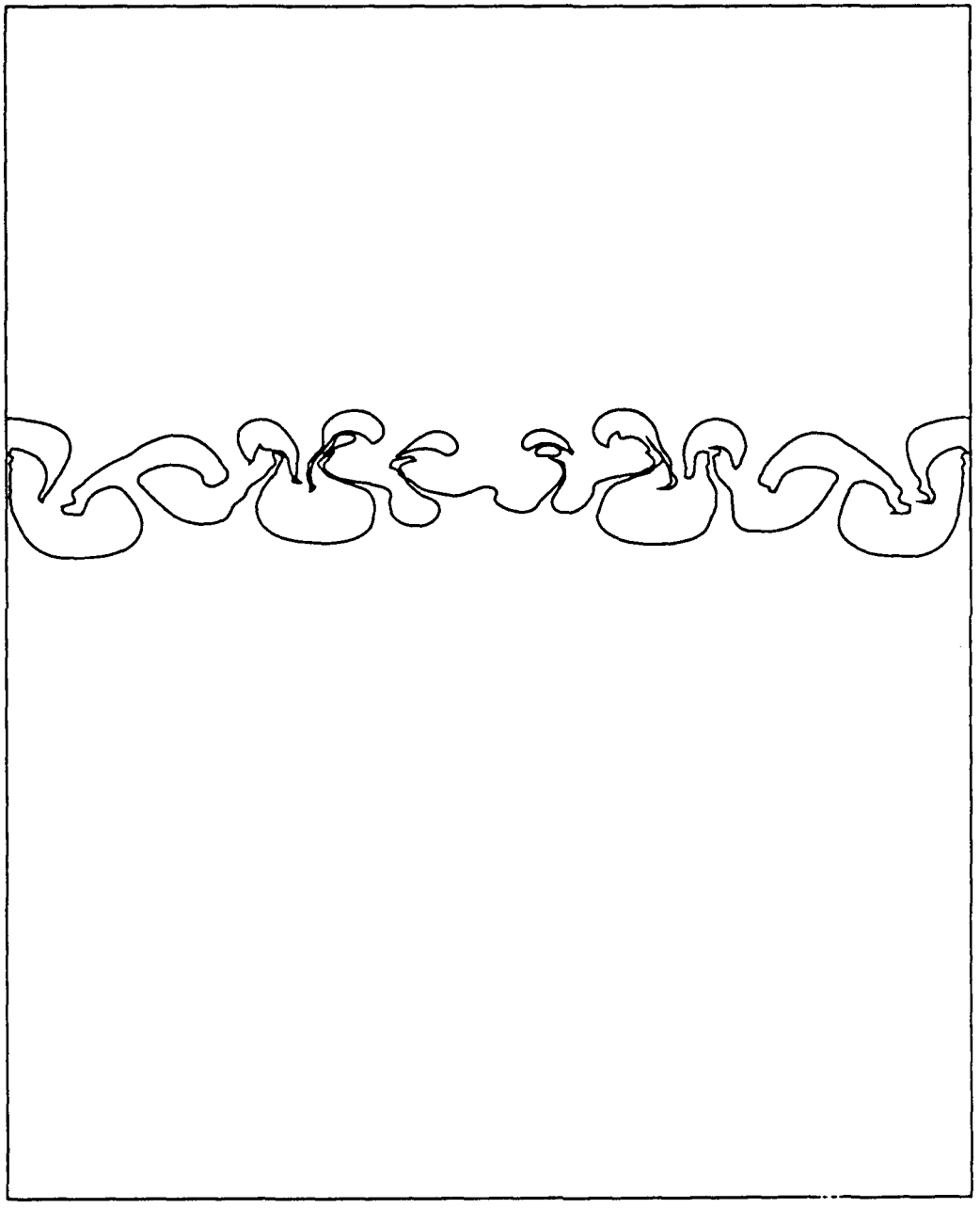


Figure 1: A full material interface after 750 time steps.

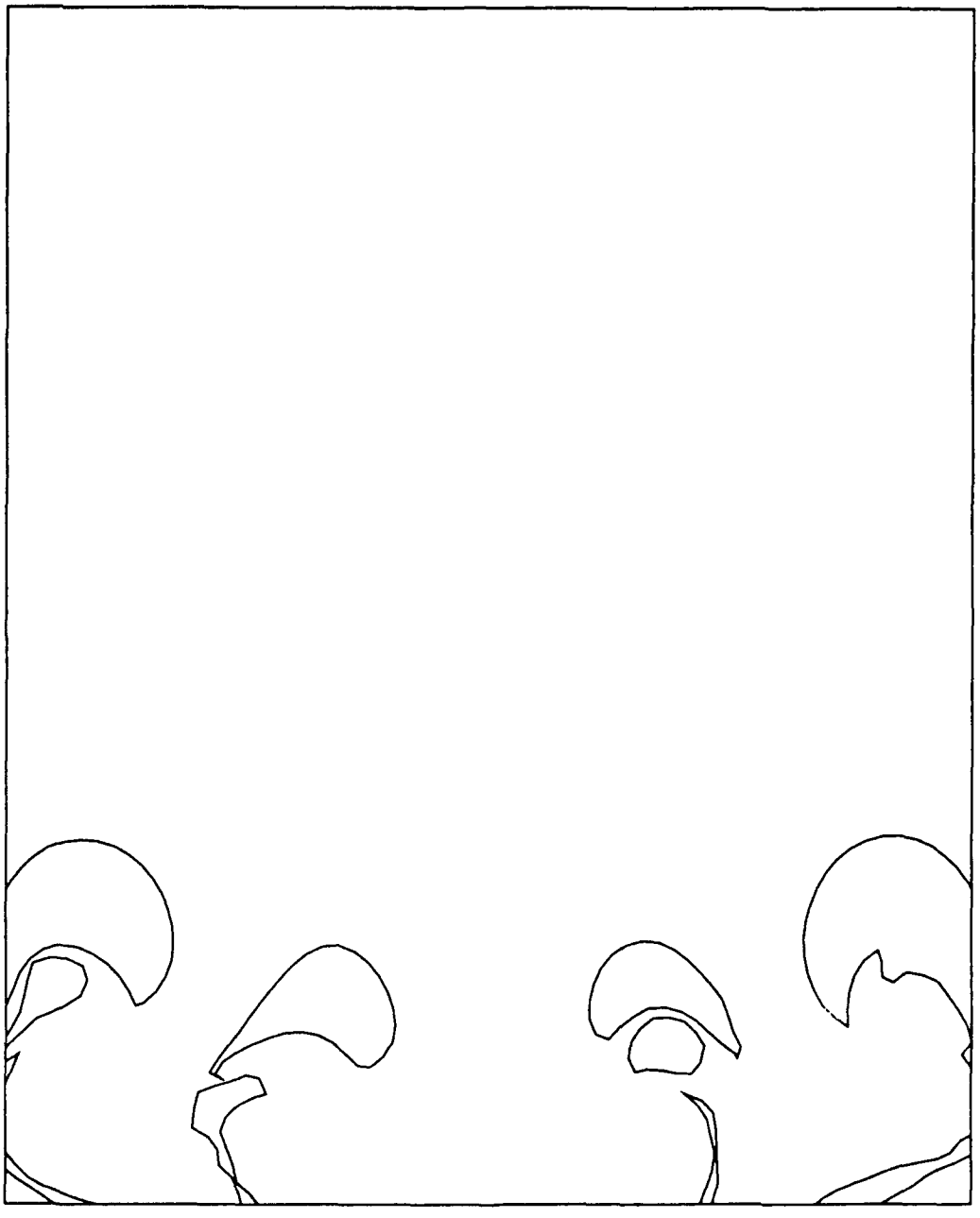


Figure 2: A representative material interface decomposed from processor 10.

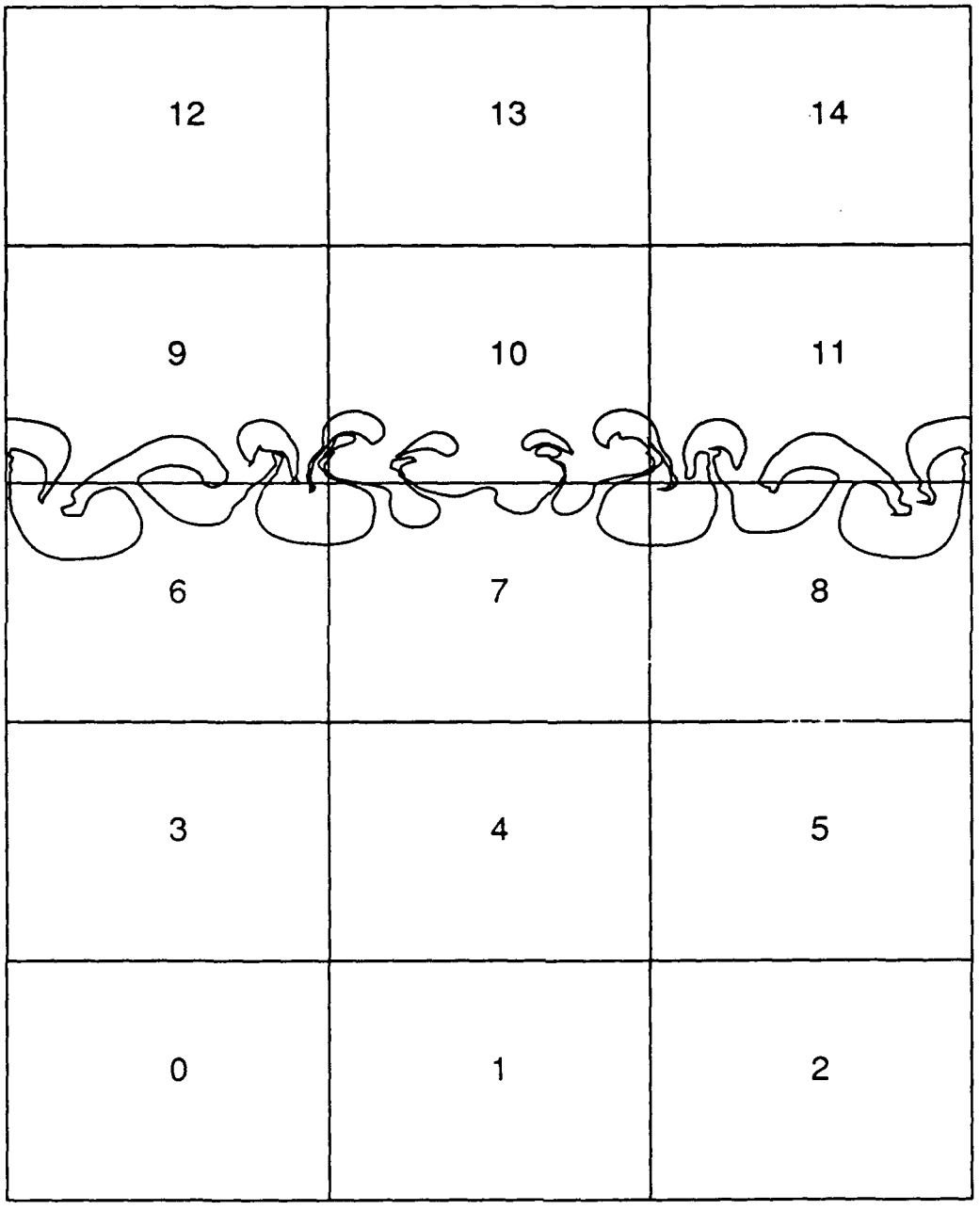


Figure 3: A material interface is decomposed into 15 subdomains (box-wise). The numbers in the sub-interfaces denote the processor ID's.

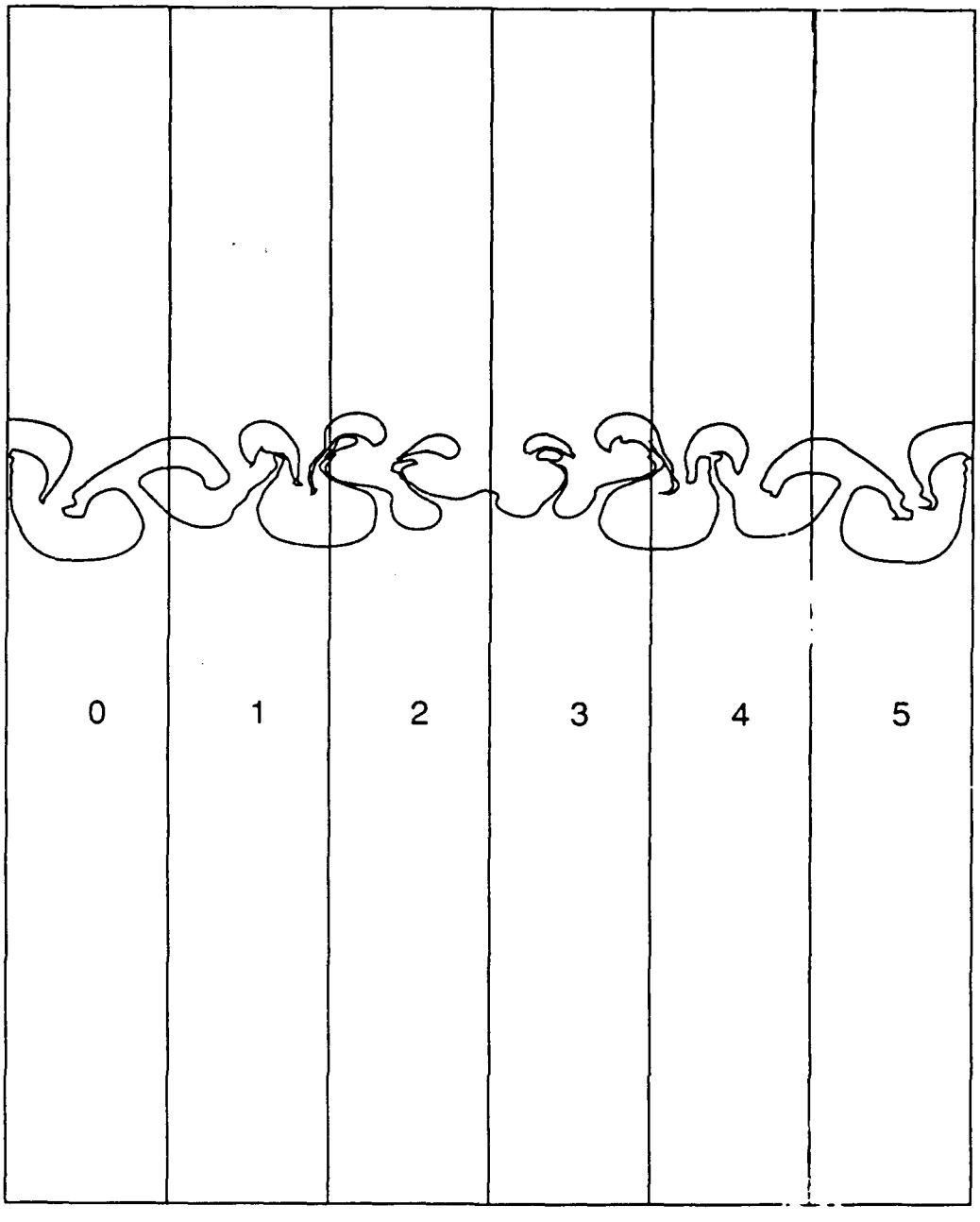


Figure 4: A material interface is decomposed into 6 subdomains (strip-wise). The numbers in the sub-interfaces denote the processor ID's.

a home-made, self-contained "interface" library. It appears to be sufficient.

Several deeper computer science issues have also been addressed. For example, a fairly new idea of dynamically managing the local physical memories with chunks of objects is implemented in this code to broadcast interfaces among processors. It would have been a simple matter if this were done on conventional computers or even shared-memory machines with "flat" memory space. The major operation involved in communicating objects among distributed memory processors is the packing, in the source memory, and the unpacking, at the destination processor, of the objects. On the destination processors, we restore the copied objects by computing their new pointers.

3 Surface and Volume Grid Generation in 3D

The interface data structures for three dimensions are an extension [12] of those of two dimensions. It is conceptually easy to understand. The basic data structures in three dimensions include COMPONENT, POINT, NODE, TRI(ANGLE), BOND, CURVE, SURFACE, and INTERFACE.

Decomposing an interface system into disjoint components separated by lower-dimensional physical objects (curves in two dimensions and surfaces three dimensions) and specifying, modifying and moving dynamically such interfaces is the basis of interface methods. A package of routines that provide facilities for defining, building, and modifying such decompositions and for efficiently solving various physical problems comprise the main algorithmic issues of the interface methods.

The surfaces are defined as a collection of contiguous triangles. After propagation, regridding may be needed to achieve triangles of approximately uniform size and good aspect ratios. Local interface operations of merger or refinement of triangles can accomplish this. For volume grids, a frontal construction is more suitable than a Delaunay triangulation, as the latter does not respect given surfaces unless additional points are inserted in the surface. Interface topology is computed on the basis of a precomputation of hashed lists. Self intersections are detected with the help of these hashed lists and are resolved by coboundary component information. Interface topology was addressed in; we discuss here only the ideas of generating a surface grid and an unstructured interior volume grid.

3.1 Surface Grid Generation

First, let us consider the surface grid generation. In three dimensions, an interface consists of a set of surfaces. Each surface contains a set of triangles and is bounded by curves. A curve contains a set of doubly linked line segments called bonds. For a valid interface, there are no intersections between surfaces and no self-intersections on each surface. Surfaces may connect along boundary curves only. Each triangle has three pointers. They point to three neighboring triangles connected at its edges. If any edge of a triangle is on the boundary of the surface, the corresponding pointer

points to the bond of a boundary curve which coincides with that boundary edge.

We consider surfaces presented parametrically by a mapping onto a planar domain. We generate the initial surfaces in three steps: (1) approximate each surface by its parametric plan surface and generate coarse triangles for the plane surface; (2) repeatedly divide each large triangle into two small ones of same size until all triangles have size less than the desired value; (3) map the coordinates of the vertices of the triangles on the planar surface to obtain the desired non-planar surface shape. The algorithm works for multi-connected surface as well. Due to the coarseness of the triangles, the time spent in step (1) is less than that in steps (2) or (3). Both steps (2) and (3) are O(N) operations. Here N is the total number of triangles on the interface at the end of the surface grid generation. Therefore the overall algorithm is O(N). For the reason of numerical stability, one prefers to generate triangles with a good aspect ratio. We achieved this goal by generating the coarse triangles with a good aspect ratio in step (1) and dividing each triangle at the middle of its longest edge in step (2). When a divided edge is a boundary edge, we also divided all triangles (on other surfaces) which are connected to the bond which coincides with the boundary edge. If the parametric map defining the surface has significant influence on the aspect ratio or the size of the triangles, one should process step (2) again after the mapping. The overall operation is still O(N). The algorithm described above applies to certain classes of interfaces for which a parametric mapping exists. For surfaces of revolution, we apply a different algorithm. A surface of revolution is defined by a curve in three dimensions, an axis of rotation and the angle of revolution. The desired triangle size determines the bond length on the curve. Then bonds are generated for the curve. The trace of a bond on the surface is a curved strip. The curved strip will be covered by triangles and the vertices of the triangles lie on the edges of the strip. We calculate the arc length of the curved edges of the strip, *i.e.*, the arc length of traces of the end points of the bonds on the surface of revolution. From the arc length, we determine the positions of the vertices of the triangles. The triangulation of the strip starts from the bond. The bond will be a (boundary) edge of the first triangle on the strip. There are two points adjacent to the bond, one from each of the edge curves of the strip. By choosing an adjacent point, we form one of two possible triangles on the strip. We select the adjacent point which provides the better aspect ratio. In the selected triangle, there is only one edge which cuts through the strip. That edge will be one of the edges in the next triangle to be generated. Then in the process of generating the second triangle, that edge plays the role of the bond in the first triangle. We apply same selection criterion recursively to obtain next triangle. This process is repeated until we reach the end of one the curved edges of the strip. We connect all points left on the other curved edge of the strip to the end point of the finished edge of the strip to form the rest of the triangles. Then the strip is completely covered by the triangles. This algorithm is O(N) also. The combination of these two algorithms is sufficient to generate initial interfaces for many problems we will consider. The physical quantities are initialized by the analytic solution of the linearized governing equations in the case of fluid instability problems.

3.2 Volume Grid Generation

Now let us discuss the unstructured interface fiting volume grid generation. This is a method to generate an unstructured mesh in three dimensions (*i.e.*, a set of tetrahedra) in the mesh blocks cut by the interface. The interface is a collection of outer and inner boundaries. In three dimensions, these boundaries are surfaces of arbitrary shape, approximated by a set of triangles, which lie inside the regular Cartesian grid and may divide the computational domain into any number of regions. Grid generation is intended to fill with tetrahedra those rectangular blocks which are cut by the interface. Its purpose is to allow interpolation of state values defined on the interface and those defined on regular Cartesian nodes. In many computational fluid problems, the state values on these two kinds of nodes are computed separately. The essential feature of the interpolation algorithm is that interpolation averages only combine state from a single component, or a single side of the interface.

The whole computational region is divided by grid lines, parallel to the coordinate axis. One can view the region as being composed of mesh blocks (in two dimensions, mesh rectangles). Unstructured grid generation takes place in a collection of mesh blocks through which the interface cuts. Other blocks in the computational domain are left intact, or more precisely, the grid in this latter case is nothing but the mesh block itself. The name "hybrid grid" can be used: the unstructured grid (tetrahedra) in the set of interface-influenced mesh blocks, and the structured grid (cube) in the complimentary set of blocks.

The major work concerned the unstructured grids. The main difficulty comes from the fact that the interface can be of an arbitrary shape and that the algorithm must be designed in such a way that it is independent of the interface. If Delaunay triangulation [1] is used, more points than those already existing on the interface must be added to preserve the surface geometrical shape. To achieve this goal, the point distribution should be dense enough to specify the surface shape. This is theoretically possible, but technically unattractive and impractical because of the arbitrarity of the interface. Based on this reason, we chose not to use the Delaunay triangulation method. New points are allowed in principle on the interface itself, but not in the interior, or volume region. This restriction is based on the final usage of the grid. As mentioned before, the grid is used for interpolation of state values between two sets of states: those defined at points on the interface and those on the regular Cartesian nodal points. Since the function values are definited only on the above points, interpolation is needed to determine the value whenever a new point is added. This is possible within any triangle of the surface. However, additional points would tend to generate more tetrahedra and complicate the computation thereafter, and we decide to make full use of the given points and disallow any new point being added.

The general approach, rather than a method, stems from the advancing front method [13, 14]. As in the advancing front method, we share the same feature of "advancing" the inner or outer boundaries forward by adding tetrahedra until the region is filled. A base triangle is chosen from the front, then a vertex is determined by some criterion to form a tetrahedra. What is different lies in the vertex determination

procedure. As said earlier, a vertex can only be one of the existing points; furthermore, the tetrahedra thus generated should be valid, i.e., should not cut through other existing tetrahedra or triangles.

The algorithm flow pattern makes the sense clear. The interface is composed of surfaces and the surface is composed of triangles. First, two kinds of intersection points — a surface triangle edge with a mesh block face and a mesh block edge with a triangle interior — are determined. These are zero dimensional objects. Second, from them the one dimensional line segments on mesh blocks face and interior are computed. Then, out come the two dimensional triangles are generated from the segments and finally, the 3-dimensional tetrahedra from two dimensional triangles. Here are the main steps of the method:

Step 1. Zero dimensional points are intersections of interfaces and blocks. Since interfaces are composed of triangles, the points are either the intersections of a triangle with mesh block edge or or of triangle side with a mesh block face.

Step 2. One dimensional segments are located in two places: on block faces and in the interior of mesh blocks. On faces are the intersections of a triangle and a rectangular face. One triangle will produce either one segment on a face, or not at all. In the interior of a mesh block, if there are any segments from a triangle, those segments make a convex polygon, lying on the same plane with the triangle.

Step 3. Triangles are generated from the segments. At the start of this step, the front consists of the sequence of straight line segments which form the convex polygon or the connection of intersection points in Step 1 on the mesh block face. During the generation process, whose detailed description is given later, any straight line segment which is available to form an element side is termed active and kept in a list, whereas any segment which is no longer active is removed from the list. This process ceases when the list containing active segments become empty.

Step 4. The three dimensional tetrahedra generation strategy is a direct extension of that presented above for triangles. Instead of segments, now the front is formed by triangles. The front is updated as each new tetrahedron is generated and the process terminates when the front is empty.

The grid generation process is sketched below:

For the sake of simplicity, only the generation process in Step 3 is given. This is a process in two dimensions. The process in three dimensions of Step 4 is essentially the same, except for two substitutions: segment by triangle and triangle by tetrahedra.

G.0 Initially, all the segments are put into a list.

G.1 Select an active side from the list in the front as a base. In order to prevent large elements from overshadowing the region of small elements, the side with the smallest length in the active side list is chosen.

G.2 Take an existing point as the vertex. If the triangle made by this vertex and a base is valid, i.e., does not intersect other triangles, this point is denoted as a "candidate point" for this particular base. Of all such candidate points, choose the one which: 1. will produce smallest number of new sides; 2. under condition 1, have the shortest distance to the mid-point of base.

G.3 Add new sides (if any) to the list. Delete the old side which has two triangles

attach to it.

G.4 If there are any sides left in the list, go to G.1.

The algorithm was written for general interface problems. It has been tested on several simple interfaces. Before it can be integrated in the overall computational code, further tests are needed as more complicated interfaces become available from the surface generation procedure and time-dependent interface dynamics.

The volume grid construction is performed every time step. It must satisfy two somewhat incompatible criteria: efficiency and robustness. Experience with two dimensional front tracking indicates that these criteria can be satisfied by use of two (or more) algorithm. One algorithm must be fast, but is allowed to fail occasionally. The other may be slow, but is only used when the first algorithm fails. The volume grid construction proposed here appears to be robust but (relative to speed requirement of the application) may be slow. Other volume grid constructions, not reported here, are under investigation which are intended to be fast but not necessarily robust. Practical experience with complex dynamically generated fronts will be needed to assess the success of these algorithms.

4 Conclusions

A previous study justifies the feasibility of performing parallel computations in three dimensions from the point of view of available hardware and timing studies of representative algorithms. This paper re-affirms this possibility from the perspective of programmability by showing results.

References

1. BAKER, T. J. *Generation of Tetrahedral Meshes Around Complete Aircraft.* In *Numerical Grid Generation in Computational Fluid Mechanics* , S. S. et al. Ed. Pineridge Press, 1988, 675–686.

2. BERGER, M. AND JAMESON, T. *Automatic Adaptive Grid Refinement for the Euler Equations. AIAA 23* (1985), 561–568.

3. CHERN, I-L., GLIMM, J., MCBRYAN, O., PLOHR, B. AND YANIV, S. *Front Tracking for Gas Dynamics. J. Comput. Phys. 62* (1985), 83–110.

4. DENG, Y. AND GLIMM, J. *Parallel Computations Using Interface Methods for Fluid Dynamics in Three Dimensions.* In *Parallel CFD: Implementations and Results Using Parallel Computers*, H. D. Simon, Ed. The MIT Press , 1990, to appear.

5. GARDNER, C. L., GLIMM, J., GROVE, J., MCBRYAN, O., MENIKOFF, R., SHARP, D. H. AND ZHANG, Q. *A Study of Chaos and Mixing in Rayleigh-Taylor and Richtmyer-Meshkov Unstable Interfaces.* In *Proceedings of the International Conference on 'The Physics of Chaos and Systems Far from Equilibrium' (CHAOS' 87), Monterey, CA, USA, Jan. 11–14, 1987*, M. Duong-van and B. Nichols, Eds. Special Issue of Nuclear Physics B (proceedings supplements section).

6. GARDNER, C. L., GLIMM, J., MCBRYAN, O., MENIKOFF, R., SHARP, D. AND ZHANG, Q. *The Dynamics of Bubble Growth for Rayleigh-Taylor Unstable Interfaces. Phys. of Fluids 31* (1988), 447–465.

7. GLIMM, J. *Tracking of Interfaces in Fluid Flow: Accurate methods for Piecewise Smooth Problems, Transonic Shock and Multidimensional Flows.* In *Advances in Scientific computing*, R. E. Meyer, Ed. Academic Press, NY, 1982.

8. GLIMM, J., GROVE, J., LINDQUIST, B., MCBRYAN, O. AND TRYGGVASON, G. *The Bifurcation of Tracked Scalar Waves. SIAM J. Sci. Stat. Comput. 9* (1988).

9. GLIMM, J., ISAACSON, E., MARCHESIN, D. AND MCBRYAN, O. *Front Tracking for Hyperbolic Systems. Adv. Appl. Math. 2* (1981), 91–119.

10. GLIMM, J. AND MCBRYAN, O. *Front Tracking for Hyperbolic Conservation Laws.* In *J Proc. of the 1981 Army Numerical Analysis and Computers Conference*. 1981.

11. ——— *A Computational Model for Interfaces. Adv. Appl. Math. 6* (1985), 422-435.

12. GLIMM, J. AND SHARP, D. *An S-matrix Theory for Classical Nonlinear Physics. Found. Phys. 16* (1986), 125–141.

13. LÖHNER, R. *Adaptive Remeshing for Transient Problems with Moving Bodies. AIAA 88* (1988), 3737.

14. PERAIRE, J., VAHDATI, M., MORGAN, K. AND ZIENKIEWICZ, O. C. *Adaptive Remeshing for Compressible Flow Computations. J. Appl. Phys. 72* (1987), 449-466.

Interior pressure discontinuities in compressible viscous steady state flow*

Senhuei E. Chen
Department of Mathematics
Howard University
Washington D.C. 20059
R. B. Kellogg
Institute for Physical Science and Technology
University of Maryland
College Park, Md 20742

Abstract Two dimensional, steady state, viscous, compressible flows with a streamline along which there is a jump in the pressure are considered. The evidence for the existence of such flows is reviewed. In the case of the linearized flow equations, linearized around a smooth ambient flow, detailed information concerning the form of the discontinuity and its behavior near the left and right hand boundary is presented.

1. Introduction The system of equations governing steady state, compressible, viscous flow is not elliptic, because the continuity equation is a hyperbolic equation in the density. Hence there arises the possibility that there are flows with interior discontinuities. The purpose of this note is to review our ongoing work on the existence and properties of these putative flows.

Although existence of such flows has yet to be rigorously established, a picture of such flows is beginning to emerge. The picture includes the features that the flows do not appear "spontaneously" in the interior of the flow field, as do shock waves. Rather, the discontinuities are created by a discontinuity in the pressure on an inflow portion of the boundary. Also, the curve of discontinuity is the streamline of the flow that emanates from the boundary point where the boundary pressure is discontinuous, and (at least in the linearized case), the magnitude of the pressure jump decays as one moves along the streamline into the flow region.

We shall be considering the system

$$L_1(U,V,P) \equiv -(2\mu+\lambda)U_{xx} - \mu U_{yy} - (\mu+\lambda)V_{xy} + (\rho U^2)_x + (\rho UV)_y + P_x = 0, \tag{1.1a}$$

$$L_2(U,V,P) \equiv -(\mu+\lambda)U_{xy} - \mu V_{xx} - (2\mu+\lambda)V_{yy} + (\rho UV)_x + (\rho V^2)_y + P_y = 0, \tag{1.1b}$$

$$L_3(U,V,P) \equiv (\rho U)_x + (\rho V)_y = 0, \tag{1.1c}$$

in a region Ω of the xy plane. The quantities U, V are the components of the velocity field, P represents the pressure, and ρ represents the density. The latter two quantities are linked by the thermodynamic relations

$$\rho = \rho(P), \quad P = P(\rho). \tag{1.1d}$$

Thus, we are neglecting the effect of variations in internal energy on the flow. (We believe that including internal energy as a variable in the problem would not affect the conclusions of this study.) The system (1.1) must be supplemented by boundary conditions to determine the solution. Let Γ denote the boundary of Ω, and let $\mathbf{n}$ denote the outward pointing unit normal to Γ. Our boundary conditions are then

$$U, V \text{ given on } \Gamma, \tag{1.2a}$$

$$P \text{ given on } \Gamma_{\text{in}} := \{(x,y) \in \Gamma : [U,V]^T \cdot \mathbf{n} < 0\}. \tag{1.2b}$$

We note that the boundary region Γ_{in} depends on the specified boundary values of U and V. If, for example, $U = V = 0$ on Γ, then Γ_{in} is empty, and there is no specification of pressure. In such a case, according to our understanding there will be no discontinuous solutions of (1.1), (1.2).

We shall consider, in addition to the system (1.1), the linearization of this system about a given smooth ambient flow. Let U, V, P, and ρ be the ambient flow; that is, suppose these functions satisfy (1.1a-d). Let

* Supported in part by the U. S. Army Research Office.

u, v and p be the linearized dependent variables. We shall regard ρ as a function of P according to (1.1d), and we set $\rho' = d\rho/dP$. The linearized equations are

$$(1.3a) \qquad L_1(u,v,p) \equiv -(2\mu+\lambda)u_{xx} - \mu u_{yy} - (\mu+\lambda)v_{xy} + \rho U u_x + \rho V u_y + p_x + A = 0,$$

$$(1.3b) \qquad L_2(u,v,p) \equiv -(\mu+\lambda)u_{xy} - \mu v_{xx} - (2\mu+\lambda)v_{yy} + \rho U v_x + \rho V v_y + p_y + B = 0,$$

$$(1.3c) \qquad L_3(u,v,p) \equiv \rho' U p_x + \rho' V p_y + \rho(u_x + v_y) + C = 0.$$

In these formulas, $\rho(P)$ and $\rho'(P)$ are evaluated at the ambient pressure, and A, B, and C contain undifferentiated terms involving u, v, and p. For example, $C = \rho_x u + \rho_y v + (U_x + V_y)\rho' p$. Each term of A, B, or C contains derivatives of the ambient variables. Thus, in the case of *uniform* ambient flow, these lowest order terms all vanish.

The appropriate boundary conditions for (1.3) are the specification of u, v, and p on Γ, and the specification of p on Γ_{in}. Note that Γ_{in} depends on the ambient flow, the flow about which the linearization is taken. If the ambient flow vanishes on Γ, $\Gamma_{in} = \emptyset$ and there is no specification of pressure. In such a case, our theory will not give discontinuous solutions; the discontinuities are produced by a discontinuity in the pressure that is specified on Γ_{in}.

Some results on the existence of solutions to (1.1) are contained in Valli [5], and Veiga [2]. They show that if the boundary data is small and smooth, there is a smooth solution to the system. The boundary condition (1.2) is implicitly contained in the treatment Geymonat [9] of the linearized transient problem. D. Hoff [3,4] has considered the time dependent, viscous, compressible flow equations in one space dimension. He shows that there are indeed solutions for which the pressure has a jump discontinuity; however he also shows that this discontinuity decays as time goes to infinity.

2. Jump conditions Suppose the system (1.1) has a solution that takes a jump across a curve $\mathcal{C}$. Are there jump conditions, analogous to the Rankine Hugoniot conditions, that must hold on the curve? To answer this question we must first define a weak solution to the system (1.1) in a domain Ω. Let the smooth curve $\mathcal{C}$ divide Ω into two subdomains, Ω_1 and Ω_2. Suppose $[U, V, P]$ are smooth functions in each of the subdomains Ω_j. We say that $[U, V, P]$ is a weak solution to (1.1) provided that for each $\phi \in C_0^\infty(\Omega)$ one has the integral identities

$$\int\int \{[-(2\mu+\lambda)U\phi_{xx} - \mu\phi_{yy}] - (\mu+\lambda)V\phi_{xy} - \rho U^2\phi_x - \rho UV\phi_y - P\phi_x\}dxdy = 0,$$

$$\int\int \{-(\mu+\lambda)U\phi_{xy} + V[-\mu\phi_{xx} - (2\mu+\lambda)\phi_{yy}] - \rho UV\phi_x - \rho V^2\phi_y - P\phi_y\}dxdy = 0,$$

$$\int\int \{\rho U\phi_x + \rho V\phi_y\}dxdy = 0.$$

Let $x(s), y(s)$ be a parametrization of a curve $\mathcal{C}$ of discontinuity, let $\dot{} = d/ds$, and let δ denote the jump in a quantity across $\mathcal{C}$. As stated in [8], the jump conditions are that $\mathcal{C}$ is a streamline of the flow and that

$$(J1) \qquad U\dot{y} = V\dot{x},$$

$$(J2) \qquad \delta U = \delta V = \delta\dot{U} = \delta\dot{V} = 0,$$

$$(J3) \qquad (2\mu+\lambda)\dot{y}\delta U_x - \mu\dot{x}\delta U_y + (\mu+\lambda)\dot{y}\delta V_y = \dot{y}\delta P,$$

$$(J4) \qquad \mu\dot{y}\delta V_x - (2\mu+\lambda)\dot{x}\delta V_y + (\mu+\lambda)\dot{y}\delta U_y = -\dot{x}\delta P.$$

The condition (J1) means that $\mathcal{C}$ is a streamline of the flow. The conditions (J3) and (J4) say that the net normal fluid stress acting on an infinitesmal element of fluid e that straddles the curve $\mathcal{C}$ is balanced by the net hydrostatic pressure acting on e. Although the width of e is infinitesmal, the jump in pressure creates a

non-zero net hydrostatic pressure, which in turn requires a non-zero net normal stress, and hence a jump in the derivatives of U and V.

The conditions (J1) and (J2) imply that $\dot{x}\delta U_x + \dot{y}\delta U_y = 0$, $\dot{x}\delta V_x + \dot{y}\delta V_y = 0$. These two equations, together with (J2) and (J3), give 4 equations for the jumps in the 4 first derivatives of U and V. Solving these equations and setting $\dot{x} = U, \dot{y} = V$, we obtain

$$\delta U_x = \frac{V^2}{(2\mu+\lambda)(U^2+V^2)}\delta P,$$

$$\delta U_y = \delta V_x = \frac{-UV}{(2\mu+\lambda)(U^2+V^2)}\delta P,$$

$$\delta V_y = \frac{U^2}{(2\mu+\lambda)(U^2+V^2)}\delta P.$$

The jump conditions (J1) - (J4) are derived in [8] for the system (1.1). Similar jump conditions are derived in [7] for the linearized system (1.3).

3. Nonlinear analysis In [1] we have constructed a simplification of the system (1.1) that still retains the "elliptic-hyperbolic" character of the original flow equations, and we have constructed an existence theorem for a discontinuous solution of the simplified system. Here, we briefly describe the results of this analysis.

To obtain the simplified system we set $\mu = -\lambda = 1$ and we drop the convective terms in (1.1a,b). More crucially, we replace the continuity equation by a modified "continuity" equation, $U\rho_x + V\rho_y = 0$. Thus, we have dropped the term $\rho U_x + \rho V_y$ from the continuity equation. This modified "continuity" equation has no physical meaning. Our only justification in considering it is that the elliptic-hyperbolic character of the system is unchanged, and that we are able to demonstrate the existence of a solution of the modified system with an interior discontinuity. With the assumption that the system is barotropic, the density $\rho = \rho(P)$ is a function of pressure only and the system can be simplified further. Writing $\rho_x = (d\rho/dP)P_x$, $\rho_y = (d\rho/dP)P_y$, we obtain from (1.1c) the equation $UP_x + VP_y = 0$. We make a further notational change. The solution that we obtain will be such that the flow is close to uniform flow in the x direction. We therefore replace the unknown U by $1 + U$. This leaves the modified momentum equations unchanged. The modified compressible flow equations studied in this paper are therefore

$$-U_{xx} - U_{yy} + P_x = 0, \tag{3.1a}$$

$$-V_{xx} - V_{yy} + P_y = 0, \tag{3.1b}$$

$$(1+U)P_x + VP_y = 0. \tag{3.1c}$$

The system (3.1) is considered in the strip $D = (0,a) \times (-\infty,\infty)$. The width a is chosen chosen small enough to satisfy certain inequalities. We impose zero boundary conditions for U and V on the sides of the strip D, and we impose the boundary condition for the pressure on the side of the strip D through which the flow enters. Thus, we are led to the boundary conditions:

$$U(0,y) = U(a,y) = 0, \tag{3.2a}$$

$$V(0,y) = V(a,y) = 0, \tag{3.2b}$$

$$P(0,y) = P_0(y). \tag{3.2c}$$

The function $P_0(y)$ is chosen to have a simple jump discontinuity at $y = 0$, to vanish outside $[0,a]$, and to be smooth for $y \geq 0$. We let $\delta P_0 = P_0(+0)$ denote the jump in $P_0(y)$ at $y = 0$. We show in [1] that the

system (3.1), (3.2) has a solution with the property that $P(x,y)$ is discontinuous across a curve $\mathcal{C}$. The curve $\mathcal{C}$ is the characteristic of (3.1c) emanating from the point (0,0), which is the point of discontinuity of P_0. The first derivatives of U and V undergo jump discontinuities across $\mathcal{C}$, and satisfy certain jump conditions analogous to (J1) - (J4).

The existence proof reformulates the problem as a fixed point mapping T in a certain Banach space $\mathcal{A}$ and uses the Schauder fixed point theorem. To describe $\mathcal{A}$, let $C^0(\bar{D})$ be the space of continuous functions in $\bar{D}$. Let

$$\mathcal{A} = \{(U,V) : U \in C^0(\bar{D}), V \in C^0(\bar{D}), ||U|| + ||V|| < \infty\},$$

where

$$||U|| = \sup_{x,y \in D} |U(x,y)| + \sup_{x,y,\bar{y} \in D, y \neq \bar{y}} \frac{|U(x,y) - U(x,\bar{y})|}{|y-\bar{y}|(|\ln|y-\bar{y}|| + 1)},$$

and similarly for $||V||$. It is easy to see that $\mathcal{A}$ is a Banach space with the norm $||(U,V)|| = ||U|| + ||V||$. The nonlinear map T is defined as the composition of two maps, $T = T_2 \circ T_1$, where T_1 and T_2 are defined as follows. Let $T_1 : (U,V) \to P$, where P is obtained by solving

$$(1+U)P_x + VP_y = 0, \quad P(0,y) = P_0(y), \tag{3.3a}$$

and let $T_2 : P \to (\bar{U}, \bar{V})$, where $\bar{U}$, $\bar{V}$ satisfy

$$\begin{aligned} -\bar{U}_{xx} - \bar{U}_{yy} + P_x = 0, \quad \bar{U}(0,y) = \bar{U}(a,y) = 0, \\ -\bar{V}_{xx} - \bar{V}_{yy} + P_y = 0, \quad \bar{V}(0,y) = \bar{V}(a,y) = 0. \end{aligned} \tag{3.3b}$$

The particular form of the norm $||U||$ reflects the elliptic-hyperbolic character of the system. The modulus of continuity of U in the y variable is $r[|\ln|r|| + 1]$. This modulus of continuity is just enough to guarantee the solvability of the hyperbolic equation (3.3a) and to provide a priori estimates that arise from this solution. (The finiteness of $||u||$ and $||v||$ imply that the characteristic equation of (3.3a) satisfies the Osgood criterion [6, Chapter III, Corollary 6.2], and hence is uniquely solvable.) On the other hand, this modulus of continuity is provided by estimates for the weakly singular integrals that occurs in the solution of the elliptic equations (3.1b), and it seems that no better modulus of continuity arises from these weakly singular integrals.

4. Linear analysis - derivation of the transformed equations In [7] we have exhibited a solution to the linearized system (1.3) for which there is an internal pressure discontinuity. We are at present carrying this analysis further to give a detailed description of the discontinuity of the linear problem. We shall now describe this work.

We consider the linearization around a constant ambient flow, $U = \text{const} > 0$, $V = \text{const}$, $P = \text{const}$. In this case the quantities $A = B = C = 0$. In addition we drop the convective terms in (1.3a,b), and we set $\rho(P) = \rho'(P) = 1$. As a result we obtain the system

$$\begin{aligned} -\Delta u + p_x = 0, \\ -\Delta v + p_y = 0, \\ Up_x + Vp_y + u_x + v_y = 0. \end{aligned} \tag{4.1}$$

We consider the system (4.1) in the strip $0 < x < a, -\infty < y < \infty$, with the boundary conditions

$$\begin{aligned} u(0,y) = u_0(y), \quad u(a,y) = u_a(u), \\ v(0,y) = v_0(y), \quad v(a,y) = v_a(y), \\ p(0,y) = p_0(y). \end{aligned} \tag{4.2}$$

We suppose that u_0, v_0, and p_0 are smooth functions except possibly at $y = 0$. Then the line $y = VU^{-1}x$ is a possible discontinuity of the solution, and the solution satisfies the following jump conditions across this

line:

$$
\begin{aligned}
&\delta u = 0, \quad \delta v = 0 \\
&\delta u_x = \frac{V^2}{U^2+V^2}\delta p, \quad \delta v_x = \frac{-UV}{U^2+V^2}\delta p, \\
&\delta u_y = \frac{-UV}{U^2+V^2}\delta p, \quad \delta v_y = \frac{U^2}{U^2+V^2}\delta p.
\end{aligned} \tag{4.3}
$$

To solve the problem (4.1), (4.2) we take the Fourier transform with respect to y. For this, we must recognize that the first derivatives of p and the second derivatives of u and v are defined almost everywhere, but are not the corresponding derivatives of u, v, and p respectively in the distributional sense. To calculate the Fourier transforms of these derivatives, we must take into account the jumps across the line $y = VU^{-1}x$. We use the following fact, established through an integration by parts: if $q(y)$ is a smooth function for $y > y_0$ and for $y < y_0$, and with one sided limits at $y = y_0$, and, letting $q'(y)$ denote the function, defined for all values of y except y_0 by the derivative of q, if q and q' decay suitably at $y = \pm\infty$, then

$$
(\mathcal{F}q')(t) = it\hat{q}(t) - \frac{1}{\sqrt{2\pi}}e^{-ity_0}[q(y_0+0) - q(y_0-0)] = it\hat{q}(t) - \frac{1}{\sqrt{2\pi}}e^{-ity_0}\delta q(y_0). \tag{4.4}
$$

Suppose $w(x,y)$ is smooth everywhere in S except on the line $y = VU^{-1}x$. Then from (4) we obtain

$$
(\mathcal{F}w_y)(x,t) = it\hat{w}(x,t) - \beta\delta w(x), \tag{4.5}
$$

where we have set

$$
(\delta w)(x) = w(x, VU^{-1}x + 0) - w(x, VU^{-1}x - 0), \quad \beta = \frac{1}{\sqrt{2\pi}}e^{-itVU^{-1}x}.
$$

A similar formula may be obtained for $\mathcal{F}w_x$ in the following way. We write

$$
\hat{w}(x,t) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{VU^{-1}x} w(x,y)e^{-ity}dy + \frac{1}{\sqrt{2\pi}}\int_{VU^{-1}x}^{\infty} w(x,y)e^{-ity}dy.
$$

Differentiating with respect to x we obtain

$$
(\mathcal{F}w_x)(x,t) = \hat{w}_x(x,t) + VU^{-1}\beta\delta w(x). \tag{4.6}
$$

Using (4.5) and (4.6) we obtain the transformed equations

$$
\begin{aligned}
-\hat{u}_{xx} + t^2\hat{u} + \hat{p}_x &= \beta\{VU^{-1}\delta u_x - \delta v_y - VU^{-1}\delta p\}, \\
-\hat{v}_{xx} + t^2\hat{v} + it\hat{p} &= \beta\{VU^{-1}\delta v_x - \delta u_y + \delta p\}, \\
U\hat{p}_x + itV\hat{p} + \hat{u}_x + it\hat{v} &= 0.
\end{aligned}
$$

Inserting the jump conditions (4.3) we obtain the same transformed equations as in the case of no jump:

$$
\begin{aligned}
-\hat{u}_{xx} + t^2\hat{u} + \hat{p}_x &= 0, \\
-\hat{v}_{xx} + t^2\hat{v} + it\hat{p} &= 0, \\
U\hat{p}_x + itV\hat{p} + \hat{u}_x + it\hat{v} &= 0,
\end{aligned} \tag{4.7}
$$

with the boundary conditions

$$
\begin{aligned}
&\hat{u}(0,t) = \hat{u}_0(t), \quad \hat{u}(a,t) = \hat{u}_a(t), \\
&\hat{v}(0,t) = \hat{v}_0(t), \quad \hat{v}(a,t) = \hat{v}_a(t), \\
&\hat{p}(0,t) = \hat{p}_0(t).
\end{aligned} \tag{4.8}
$$

5. Linear analysis - solution of the transformed equations We seek a solution of the problem (4.7) with an x dependence of the form $e^{r(t)x}$. This leads to the quintic equation for $r(t)$,

$$(r^2 - t^2)^2(rU + itV + 1) = 0,$$

with roots

$$r = \pm t, \pm t, -\frac{1}{U} - \frac{tV}{U}i.$$

This also leads to the general solution of (4.7) in the form

$$\begin{bmatrix} \hat{u} \\ \hat{v} \\ \hat{p} \end{bmatrix} = \sum_1^5 c_j \hat{z}_j, \tag{5.1}$$

where the $\hat{z}_j$ are given by

$$\hat{z}_1 = \begin{bmatrix} 1 \\ i \\ 0 \end{bmatrix} e^{tx}, \hat{z}_2 = \begin{bmatrix} -1 \\ i \\ 0 \end{bmatrix} e^{-tx}, \hat{z}_3 = \begin{bmatrix} \alpha \\ \beta \\ 1 \end{bmatrix} e^{-(1+itV)U^{-1}x},$$

$$\hat{z}_4 = \begin{bmatrix} -1 - 2t(U + iV) + tx \\ itx \\ 2t \end{bmatrix} e^{tx}, \hat{z}_5 = \begin{bmatrix} 1 - 2t(U - iV) + tx \\ -itx \\ 2t \end{bmatrix} e^{-tx},$$

with

$$\alpha = \frac{-U(1 + itV)}{1 + 2itV - t^2(U^2 + V^2)},$$

$$\beta = \frac{itU^2}{1 + 2itV - t^2(U^2 + V^2)}.$$

The coefficients are chosen to satisfy the boundary conditions (4.8). Imposing these boundary conditions, we are led to the linear system

$$B \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{bmatrix} = \begin{bmatrix} \hat{u}_0 \\ \hat{v}_0 \\ \hat{p}_0 \\ \hat{u}_a \\ \hat{v}_a \end{bmatrix}$$

where

$$B = \begin{bmatrix} 1 & -1 & iUVDt^{-1} & -2(U + iV)t & -2(U - iV)t \\ i & i & -iU^2Dt^{-1} & 0 & 0 \\ 0 & 0 & 1 & 2t & 2t \\ p & -p^{-1} & iUVDt^{-1}q & [-2(U + iV)t + at]p & [-2(U - iV)t + at]p^{-1} \\ ip & ip^{-1} & -iU^2Dt^{-1}q & itap & -itap^{-1} \end{bmatrix}$$

and where

$$p = e^{at}, \quad q = e^{-\gamma a}, \quad \gamma = (1 + itV)U^{-1}, \quad D = (U^2 + V^2)^{-1}.$$

The matrix B is inverted using *Mathematica*. Let $A = B^{-1} = [a_{jk}]$. Because of the fundamental importance of this matrix for the linearized compressible flow equations, we record the following asymptotic forms for the matrix entries, valid for large values of $|t|$. For $t >> 0$, we get

$$a_{11} = \tfrac{1}{2}UD^2qt^{-1}p^{-1}(2U - a)(U + iV)$$

$$a_{12} = -\tfrac{1}{2}UD^2qit^{-1}p^{-1}(2U - a)(U + iV)$$

$$a_{13} = \tfrac{1}{2}UDqt^{-1}p^{-1}(2U - a)$$

$$a_{14} = \tfrac{1}{2}aDp^{-1}(U - iV)$$

$$
\begin{aligned}
a_{15} &= -\tfrac{1}{2}iDp^{-1}[2(U^2+V^2)-a(U-iV)]\\
a_{21} &= U^2D^2t^{-1}(U+iV)\\
a_{22} &= -i\\
a_{23} &= U^2Dt^{-1}\\
a_{24} &= -\tfrac{1}{2}aD(U-iV)p^{-1}\\
a_{25} &= \tfrac{1}{2}iD[2(U^2+V^2)-a(U-iV)]\\
a_{31} &= D(U+iV)\\
a_{32} &= -iD(U+iV)\\
a_{33} &= 1\\
a_{34} &= -Dp^{-1}(a+2iV)\\
a_{35} &= iDp^{-1}(2U-a)\\
a_{41} &= \tfrac{1}{2}UD^2qp^{-1}t^{-2}(U+iV)\\
a_{42} &= -\tfrac{1}{2}iUD^2qp^{-1}t^{-2}(U+iV)\\
a_{43} &= \tfrac{1}{2}UDqp^{-1}t^{-2}\\
a_{44} &= -\tfrac{1}{2}Dt^{-1}p^{-1}(U-iV)\\
a_{45} &= -\tfrac{1}{2}iDt^{-1}p^{-1}(U-iV)\\
a_{51} &= -\tfrac{1}{2}Dt^{-1}(U+iV)\\
a_{52} &= \tfrac{1}{2}iDt^{-1}(U+iV)\\
a_{53} &= -\tfrac{1}{2}UDt^{-2}\\
a_{54} &= \tfrac{1}{2}Dt^{-1}p^{-1}(U+iV+a)\\
a_{55} &= -\tfrac{1}{2}iDt^{-1}p^{-1}(U+iV-a).
\end{aligned}
$$

Similarly, for $t << 0$, we get

$$
\begin{aligned}
a_{11} &= U^2D^2(U-iV)t^{-1}\\
a_{12} &= -i\\
a_{13} &= U^2Dt^{-1}\\
a_{14} &= \tfrac{1}{2}aD(U+iV)p\\
a_{15} &= \tfrac{1}{2}iD[2(U^2+V^2)-a(U+iV)]p\\
a_{21} &= \tfrac{1}{2}UD^2(2U-a)(U-iV)qt^{-1}p\\
a_{22} &= \tfrac{1}{2}iUD^2(2U-a)(U-iV)qt^{-1}p\\
a_{23} &= \tfrac{1}{2}UD(2U-a)qt^{-1}p\\
a_{24} &= -\tfrac{1}{2}aD(U+iV)p\\
a_{25} &= -\tfrac{1}{2}iD[2(U^2+V^2)-a(U+iV)]p\\
a_{31} &= D(U-iV)\\
a_{32} &= iD(U-iV)\\
a_{33} &= 1\\
a_{34} &= D(2iV-a)p\\
a_{35} &= -iD(2U-a)p\\
a_{41} &= -\tfrac{1}{2}D(U-iV)t^{-1}\\
a_{42} &= -\tfrac{1}{2}iD(U-iV)t^{-1}\\
a_{43} &= \tfrac{1}{2}UDt^{-2}\\
a_{44} &= \tfrac{1}{2}D(U-iV+a)t^{-1}p
\end{aligned}
$$

$$
\begin{aligned}
a_{45} &= \tfrac{1}{2} iD(U - iV - a)t^{-1}p \\
a_{51} &= -\tfrac{1}{2} UD^2(U - iV)qt^{-2}p \\
a_{52} &= -\tfrac{1}{2} iUD^2(U - iV)qt^{-2}p \\
a_{53} &= -\tfrac{1}{2} UDqt^{-2}p \\
a_{54} &= -\tfrac{1}{2} D(U + iV)t^{-1}p \\
a_{55} &= \tfrac{1}{2} iD(U + iV)t^{-1}p.
\end{aligned}
$$

We illustrate the use of these formulas by considering three special cases.

Case 1: $p_0 \neq 0, u_0 = v_0 = u_a = v_a = 0.$

In this case, the solution to (4.7), (4.8) is given by

$$
\begin{aligned}
\hat{u} =& \{a_{13}e^{tx} - a_{23}e^{-tx} + \alpha a_{33}e^{-(1+itV)U^{-1}x} \\
& - [-1 - 2t(U + iV) + tx]a_{43}e^{tx} + [1 - 2t(U - iV) + tx]a_{53}e^{-tx}\}\hat{p}_0, \\
\hat{v} =& \{ia_{13}e^{tx} + ia_{23}e^{-tx} + \beta a_{33}e^{-(1+itV)U^{-1}x} + itxa_{43}e^{tx} - itxa_{53}e^{-tx}\}\hat{p}_0, \\
\hat{p} =& \{a_{33}e^{-(1+itV)U^{-1}x} + 2ta_{43}e^{tx} + 2ta_{53}e^{-tx}\}\hat{p}_0.
\end{aligned}
$$

Using the above asymptotic formulas, we obtain asymptotic expressions for $\hat{u}$, $\hat{v}$, and $\hat{p}$. These expressions are arranged into three groups, which we denote the discontinuous part, the left hand part, and the right hand part. In this way we arrive at the formulas

$$
\begin{aligned}
u &= u_D + u_L + u_R + u_{\text{rem}}, \\
v &= v_D + v_L + v_R + v_{\text{rem}}, \\
p &= p_D + p_L + p_R + p_{\text{rem}},
\end{aligned}
$$

where

$$
\begin{aligned}
\hat{u}_D(x,t) &:= UVD\frac{it}{t^2+1}e^{-(1+iVt)U^{-1}x}\hat{p}_0, \\
\hat{u}_L(x,t) &:= -\tfrac{1}{2}UD\{x\frac{1}{\sqrt{t^2+1}} + 2V\frac{it}{t^2+1}\}e^{-x\sqrt{t^2+1}}\hat{p}_0, \\
\hat{u}_R(x,t) &:= -\tfrac{1}{2}UDE\{(a-x)\frac{1}{\sqrt{t^2+1}} + 2V\frac{it}{t^2+1}\}e^{-iaVtU^{-1}}e^{-(a-x)\sqrt{t^2+1}}\hat{p}_0, \\
\hat{v}_D(x,t) &:= -U^2D\frac{it}{t^2+1}e^{-(1+iVt)U^{-1}x}\hat{p}_0, \\
\hat{v}_L(x,t) &:= \tfrac{1}{2}UD(2U+x)\frac{it}{t^2+1}e^{-x\sqrt{t^2+1}}\hat{p}_0, \\
\hat{v}_R(x,t) &:= \tfrac{1}{2}UDE(2U-a+x)\frac{it}{t^2+1}e^{-iaVtU^{-1}}e^{-(a-x)\sqrt{t^2+1}}\hat{p}_0, \\
\hat{p}_D(x,t) &:= e^{-(1+itV)U^{-1}x}\hat{p}_0, \\
p_L(x,t) &:= -UD\frac{1}{\sqrt{t^2+1}}e^{-x\sqrt{t^2+1}}\hat{p}_0, \\
\hat{p}_R(x,t) &:= UDE\frac{1}{\sqrt{t^2+1}}e^{-iaVtU^{-1}}e^{-(a-x)\sqrt{t^2+1}}\hat{p}_0
\end{aligned}
$$

Upon taking an inverse Fourier transform, one obtains for u_D, v_D, and p_D the formulas

$$
\begin{aligned}
u_D(x,y) =& -\pi UVDe^{-U^{-1}x}e^{-(y-VU^{-1}x)}\int_{-\infty}^{y-VU^{-1}x} p_0(t)e^t dt \\
& + \pi UVDe^{-U^{-1}x}e^{-(y-VU^{-1}x)}\int_{y-VU^{-1}x}^{\infty} p_0(t)e^{-t}dt,
\end{aligned}
\tag{5.1a}
$$

$$
\begin{aligned}
v_D(x,y) =& \pi UVDe^{-U^{-1}x}e^{-(y-VU^{-1}x)}\int_{-\infty}^{y-VU^{-1}x} p_0(t)e^t dt \\
&- \pi UVDe^{-U^{-1}x}e^{-(y-VU^{-1}x)}\int_{y-VU^{-1}x}^{\infty} p_0(t)e^{-t}dt,
\end{aligned} \tag{5.1b}
$$

$$
p_D(x,y) = 2\pi e^{-U^{-1}x}p_0(y-VU^{-1}x). \tag{5.1c}
$$

If $p_0(y)$ has a jump discontinuity at $y=0$ and is smooth in $(-\infty,0)$ and $(0,\infty)$, then the formulas (5.1) yield a discontinuity of p_D and the first derivatives of u_D and v_D on the line $y = VU^{-1}x$. It may be verified that the triple $[u_D, v_D, p_D]$ satisfies the jump conditions (4.3). One can also obtain expressions for the left and right functions, involving the convolution of p_0 with certain kernels. We shall not write down these kernels. Instead, we note that because of the presence of the factor $\exp\{-x\sqrt{t^2+1}\}$ in the Fourier transform of the left hand functions, these functions are smooth for $x > 0$, and similarly, because of the presence of the factor $\exp\{-(a-x)\sqrt{t^2+1}\}$ in the right hand functions, these functions are smooth for $x < a$. These functions have the "purpose" of reconciling the the values of the unknowns at the boundary where the jump discontinuity occurs. An analysis of the remainder terms shows that p_{rem} is continuous and $u_{\text{rem}}, v_{\text{rem}}$ are continuously differentiable in the strip. We conclude that (5.1) displays the discontinuous behavior of the solution of (4.1), (4.2) caused by a discontinuity in the inflow pressure.

Case 2: $u_0 \neq 0, v_0 = p_0 = u_a = v_a = 0$.

Again, the asymptotic formulas for the coefficients yield a discontinuous part, a left hand part, a right hand part, and a smoother remainder. The formulas for the discontinuous part are

$$
\hat{u}_D = -UVD^2\{V\frac{1}{\sqrt{t^2+1}} - U\frac{it}{t^2+1}\}e^{-(1+itV)U^{-1}x}\hat{u}_0, \tag{5.2a}
$$

$$
\hat{v}_D = U^2D^2\{V\frac{1}{\sqrt{t^2+1}} - U\frac{it}{t^2+1}\}e^{-(1+itV)U^{-1}x}\hat{u}_0, \tag{5.2b}
$$

$$
\hat{p}_D = D\{U + V\frac{it}{\sqrt{t^2+1}}\}e^{-(1+itV)U^{-1}x}\hat{u}_0. \tag{5.2c}
$$

The formulas (5.2a) and (5.2b) may be inverted to give for u_D and v_D the formulas

$$
\begin{aligned}
u_D(x,y) =& \frac{-1}{\pi}UV^2D^2e^{-U^{-1}x}\int_{-\infty}^{\infty} K_0(|y-s-VU^{-1}x|)u_0(s)ds \\
&- \frac{1}{2}UV^2D^2e^{-U^{-1}x}\int_{-\infty}^{\infty} \operatorname{sgn}(y-s-VU^{-1}x)e^{-|y-s-VU^{-1}x|}u_0(s)ds,
\end{aligned} \tag{5.3a}
$$

$$
\begin{aligned}
v_D(x,y) =& \frac{1}{\pi}U^2VD^2e^{-U^{-1}x}\int_{-\infty}^{\infty} K_0(|y-s-VU^{-1}x|)u_0(s)ds \\
&+ \frac{1}{2}U^2VD^2e^{-U^{-1}x}\int_{-\infty}^{\infty} \operatorname{sgn}(y-s-VU^{-1}x)e^{-|y-s-VU^{-1}x|}u_0(s)ds.
\end{aligned} \tag{5.3b}
$$

The inversion of (5.3) involves a δ function, and so does not have meaning if u_0 has a discontinuity. If u_0 is continuous, but u_0' has a discontinuity at $y=0$, we obtain for p_0 the formula

$$
p_D(x,y) = DUe^{-U^{-1}x}u_0(y-VU^{-1}x) - \frac{2}{\sqrt{2\pi}}DVe^{-U^{-1}x}\int_{-\infty}^{\infty} K_0(|y-s-VU^{-1}x|)u_0'(s)ds. \tag{5.3c}
$$

Suppose u_0 has a jump discontinuity at $y=0$ and is smooth elsewhere. Then the functions u_D and v_D have continuous first derivatives, but their second derivatives become infinite on the line $y = VU^{-1}x$, and the function p_D is continuous, but the first derivatives of p_D become infinite on the line $y = VU^{-1}x$. We

conclude that a jump in u_0' does not produce a pressure discontinuity, but does produce a higher order singularity in the solution on the line $y = VU^{-1}x$.

Case 3: $u_a \neq 0, u_0 = v_0 = p_0 = v_a = 0$.

In this case, the asymptotic formulas yield only a right hand part. There is no discontinuity in the interior resulting from a discontinuity in u_a.

References

1. Senhuei E. Chen and R. Bruce Kellogg, An interior discontinuity of a nonlinear elliptic-hyperbolic system, to appear in SIAM J. Math. Anal.

2. H. Beirão da Veiga, Stationary motions and the incompressible limit for compressible viscous fluids, Houston J. of Math., 13 (1987), 527-544.

3. D. Hoff, Construction of solutions for compressible, isentropic Navier-Stokes equations in one space dimension with nonsmooth initial data, Pro. Royal Soc. Edinburgh Vol. 103A, 1986. pp. 301-315.

4. D. Hoff, Global existence for 1d, compressible, isentropic Navier-Stokes equations with large initial data, Trans. AMS vol. 303, No. 1, 1987, pp. 169-181.

5. A. Valli, On the existence of stationary solutions to compressible Navier-Stokes equations, Analyse Nonlinéaire Vol. 4, No. 1, 1987, pp. 89-113.

6. P. Hartman, Ordinary Differential Equations, 1964, John Wiley and Son.

7. R. B. Kellogg, Discontinuous solutions of the linearized steady state, compressible, viscous Navier-Stokes equations, SIAM J. Math Anal. Vol. 19. No. 3, 1988, pp. 567-579.

8. Senhuei E. Chen, A discontinuous solution to a nonlinear elliptic-hyperbolic system, Ph.D thesis, University of Maryland, College Park, Md., 1989.

9. G. Geymonat and P. Leyland, Transport and propagation of a perturbation of a flow of a compressible fluid in a bounded region, Arch. Rat. Mech. Anal. vol 100, 1987, 53-81.

AN EXTENSION OF MESH EQUIDISTRIBUTION TO TIME-DEPENDENT PARTIAL DIFFERENTIAL EQUATIONS

J.M. Coyle
U.S. Army Armament Research, Development, and Engineering Center
Close Combat Armaments Center
Benet Laboratories
Watervliet, NY 12189-4050

ABSTRACT. Mesh equidistribution for static approximation problems is extended to a domain that admits time dependency. Instead of some quantity being equidistributed over a static mesh, the change in this quantity is equidistributed over a dynamic, moving mesh. Applications are for utilization in numerical methods to solve time-dependent partial differential equations. This mesh moving scheme is incorporated into a finite element code which already refines adaptively. Comparisons are made for stationary and moving meshes.

INTRODUCTION. An equidistributed mesh in one space dimension is a partition of a given domain into subintervals such that some given quantity is uniform over each subinterval. More specifically, given an interval (a,b) and a positive weight function w(x) defined on (a,b), then an equidistributed mesh is a partition

$$\{a = x_0 < x_1 < x_2 < \ldots < x_{M-1} < x_M = b\}$$

such that

$$\int_{x_{j-1}}^{x_j} w(x)dx = \text{constant} = \frac{1}{M}\int_a^b w(x)dx \quad , \quad j = 1,2,\ldots,M \tag{1}$$

The usual application of such a mesh is for approximating functional relationships to a certain accuracy with a minimum number of mesh points by choosing w(x) appropriately [1]. Equidistribution strategies have also been used in numerical methods for solving two-point boundary value problems [2,3]. This is because it has been shown [4,5] that the task of selecting a mesh to minimize the discretization error is asymptotically equivalent to equidistributing the local discretization error.

The successes in the above fields of functional approximation and numerical ordinary differential equations have led some investigators to consider the use of equidistribution strategies for generating moving meshes in the field of numerical partial differential equations (PDEs) [6-8]. The general framework is to simply reconsider Eq. (1) with a time dependency. That is to say, the problem is now to determine a dynamic mesh

$$\{a = x_0 < x_1(t) < x_2(t) < \ldots < x_{M-1}(t) < x_M = b\}$$

at time t so that

$$\int_{x_{j-1}(t)}^{x_j(t)} w(x,t)dx = c(t) = \frac{1}{M}\int_a^b w(x,t)dx \quad , \quad j = 1,2,\ldots,M \qquad (2)$$

where the positive weight function w(x,t) is usually chosen to be a function of the solution of the underlying PDE. For example, w has been chosen to be proportional to the solution's gradient, curvature, and local discretization error.

When applying Eq. (2) in some numerical scheme, most investigators move a fixed number of points so as to follow and resolve local nonuniformities in the solution. In order to guarantee a certain accuracy, they must be sure that this fixed number is large enough to approximate the solution throughout the entire spatial domain for the entire temporal "life" of the solution. Some see this as a limitation since the correct number of fixed points necessary is not generally known a priori.

Also, this moving mesh is not operating in a vacuum. It is being used in conjunction with some numerical solution procedure. Since the accuracy of most such procedures can depend on the shape of the space-time grid, sometimes the equidistribution law can be too dynamic and deform the grid enough so as to introduce a new, even larger source of error. This can happen even if the equidistribution law doesn't demand much moving of its own accord. If a non-equidistributed mesh is used as the initial mesh, then the grid can deform drastically as the moving mesh tries to relocate to the proper equidistributed positions. In order to avoid these difficulties, some investigators have abandoned moving altogether and developed local refinement methods [3].

A local refinement method is a procedure where uniform fine grids are added to coarse grids in regions where the solution is not adequately resolved. Although they can guarantee a solution to a prescribed accuracy, they can be costly, as they involve recomputing the solution, and they are not as good as moving mesh methods at reducing dispersive errors in the vicinity of wavefronts.

The choice here has been to combine local refinement with mesh moving based on equidistribution. The purpose is twofold. First, the refinement procedure is incorporated to avoid any drastic deformation of the grid by the moving mesh as well as guarantee a prescribed accuracy. Second, the mesh moving is applied in order to obtain as accurate a solution as possible for any given discretization so as to put off the need for refinement for as long as possible and thus to reduce the costs involved.

Equation (2) as it stands, however, is not easily partnered with a refinement scheme. It is too dependent on mesh position and the number of extant mesh points. Hence, a refinement step can disrupt the nature of the equidistribution and cause a drastic change in the mesh dynamics similar to that caused by a "bad" initial mesh.

The attempt to overcome the difficulty reported here was to try to extend Eq. (2) in such a way that it worked with the refinement procedure rather than against it. It seemed that the dynamics of Eq. (2) were based on the static spatial nature of Eq. (1) and an extension was needed that incorporated more of the time dependency of the domain and solution process.

In the next section, this extension of Eq. (2) is presented as well as the algorithm for coupling the refinement and moving procedures. Then, in the following section, results on a series of test cases are presented for comparison. In the Discussion section, the characteristics of the extended equidistribution law are discussed in light of the results of the previous section. Finally, in the last section, some conclusions are presented.

PROCEDURES. The basic principle behind this extension of Eq. (2) is to try to equidistribute temporal properties as well as spatial. To this end, consider a typical time interval of interest (0,T) and a discretization

$$\{0 = t_0 < t_0 < t_1 < t_2 < \ldots < t_{N-1} < t_N = T\}$$

of that interval. Then, given any time level t_{n-1} and any mesh

$$\{a = x_0^{n-1} < x_1^{n-1} < x_2^{n-1} < \ldots < x_M^{n-1} = b\}$$

at that time level, require the new mesh

$$\{a = x_0^n < x_1^n < x_2^n < \ldots < x_{M-1}^n < x_M^n = b\}$$

at the next time level t_n to satisfy

$$\int_{x_{j-1}^n}^{x_j^n} w(x,t_n)dx = \int_{x_{j-1}^{n-1}}^{x_j^{n-1}} w(x,t_{n-1})dx + \frac{1}{M}\left\{\int_a^b w(x,t_n)dx - \int_a^b w(x,t_{n-1})dx\right\} ,$$

$$j = 1,2,\ldots,M \qquad (3)$$

Note that Eq. (3) is not an equidistribution law in the sense of Eqs. (1) and (2). No quantity is being held constant over any subinterval by the enforcement of Eq. (3). Rather, it is the change in the quantity w over each new subinterval

$$(x_{j-1}^n, x_j^n)$$

that is allowed to vary by a constant amount (which is proportional to the total change in w from t_{n-1} to t_n) when compared to its values over the old subinterval

$$(x_{j-1}^{n-1}, x_j^{n-1})$$

In a sense then, it is the time change of this quantity that is being equidistributed.

Note also that the relationship between old and new meshes is not as dramatic as in Eq. (2). If

$$\{x_j^{n-1}\}_{j=0}^M$$

is a "bad" mesh, e.g., suppose it was readjusted by a refinement procedure, then Eq. (3) simply requires that

$$\{x_j^n\}_{j=0}^M$$

differs from

$$\{x_j^{n-1}\}_{j=0}^M$$

by a constant amount over each subinterval and does not require any drastic readjustment to a new equidistributed position.

In order to test the performance of Eq. (3) and its postulated properties, it was incorporated into a numerical PDE solver that already implemented an automatic refinement strategy. The overall solution algorithm is as follows:

1. Move mesh to next time level according to Eq. (3).
2. Solve PDE using finite elements in space and finite differences in time.
3. Estimate error that occurred in the solution process.
4. If the error is less than or equal to a prescribed tolerance and the time level is less than T, then go to step 1.
5. If the error is greater than the tolerance, refine in either space or time or both, then go to step 2.
6. If the time level is greater than or equal to T, then stop.

RESULTS. The following PDE was solved numerically for all test cases:

$$u_t - u_x(1 + \frac{1}{10} u) = \frac{1}{200} u_{xx} \quad , \quad 0 < x < 1 \quad , \quad t > 0$$

$$u(x,0) = \tanh 10(x-1) \quad , \quad 0 \leq x \leq 1$$

$$u(0,t) = \tanh 10(-1+t) \quad , \quad t \geq 0$$

$$u(1,t) = \tanh 10t \quad , \quad t \geq 0$$

The exact solution, $u(x,t) = \tanh 10(x-1+t)$, is simply a wavefront that moves through the spatial domain from right to left as time progresses. Optimally, the mesh should try to follow the front as it moves across the interval (0,1).

For all cases, the L_2 norm of the error was prescribed to be less than a tolerance of 0.01, an initial uniform mesh with 11 points (M = 11) and an initial time step of 0.05 ($\Delta t = 0.05$) was input, and the solution process was allowed to proceed for 75 time steps.

Case 1. For this case, no movement was allowed--only refinement. Mesh trajectories are shown in Figure 1. At the end of 75 time steps, N = 37 and Δt = 0.00218.

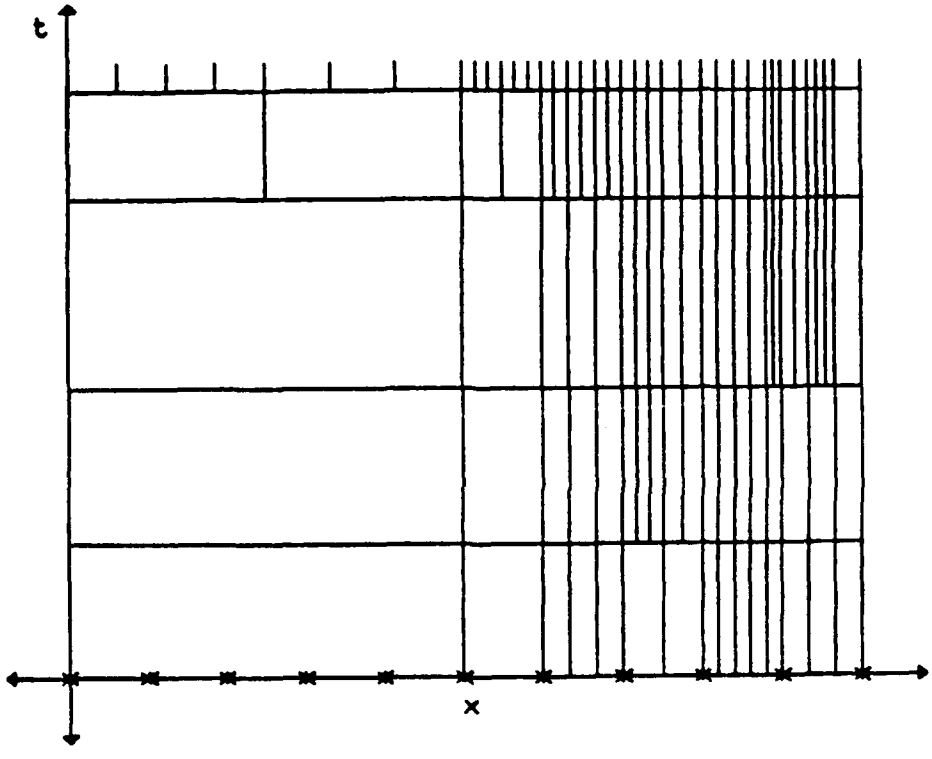


Figure 1

Mesh Trajectories for Case 1. Stars on x-axis indicate the mesh input before any moving or refining. Horizontal lines indicate a temporal refinement has occurred (actual values of Δt are not shown).

Case 2. For this case, movement was based on the first time derivative of the solution ($w(x,t) = |u_t(x,t)|$). Mesh trajectories are shown in Figure 2. At the end of 75 time steps, N = 40 and $\Delta t = 0.00579$.

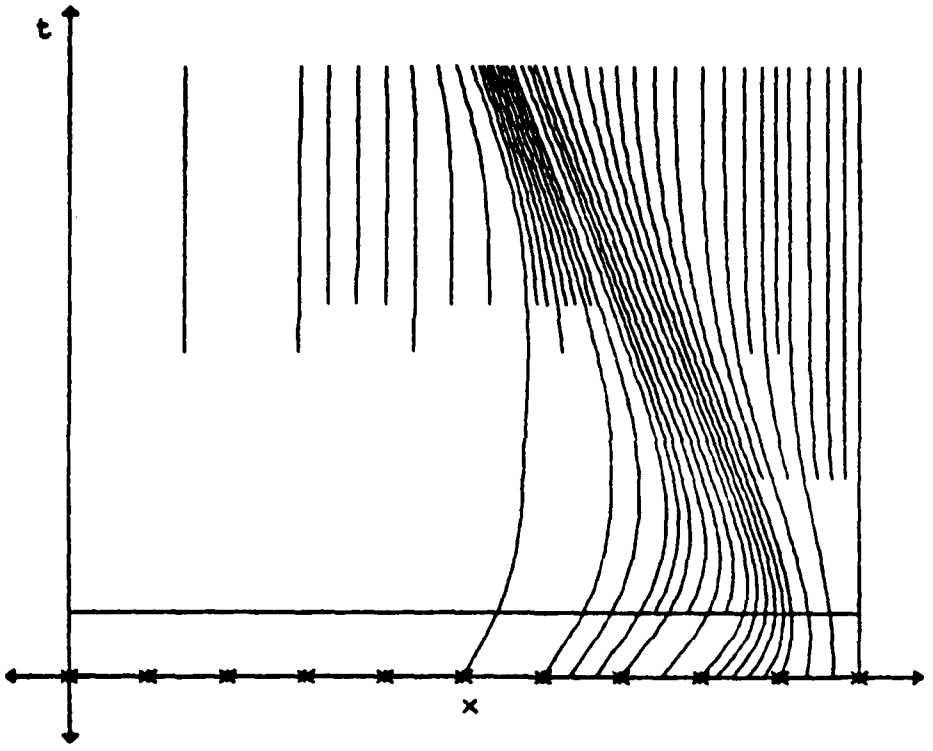


Figure 2

Mesh Trajectories for Case 2. Stars on x-axis indicate the mesh input before any moving or refining. Horizontal lines indicate a temporal refinement has occurred (actual values of Δt are not shown).

Case 3. For this case, movement was based on the second spatial derivative of the solution ($w(x,t) = |u_{xx}(x,t)|$). Mesh trajectories are shown in Figure 3. At the end of 75 time steps, N = 23 and $\Delta t = 0.00201$.

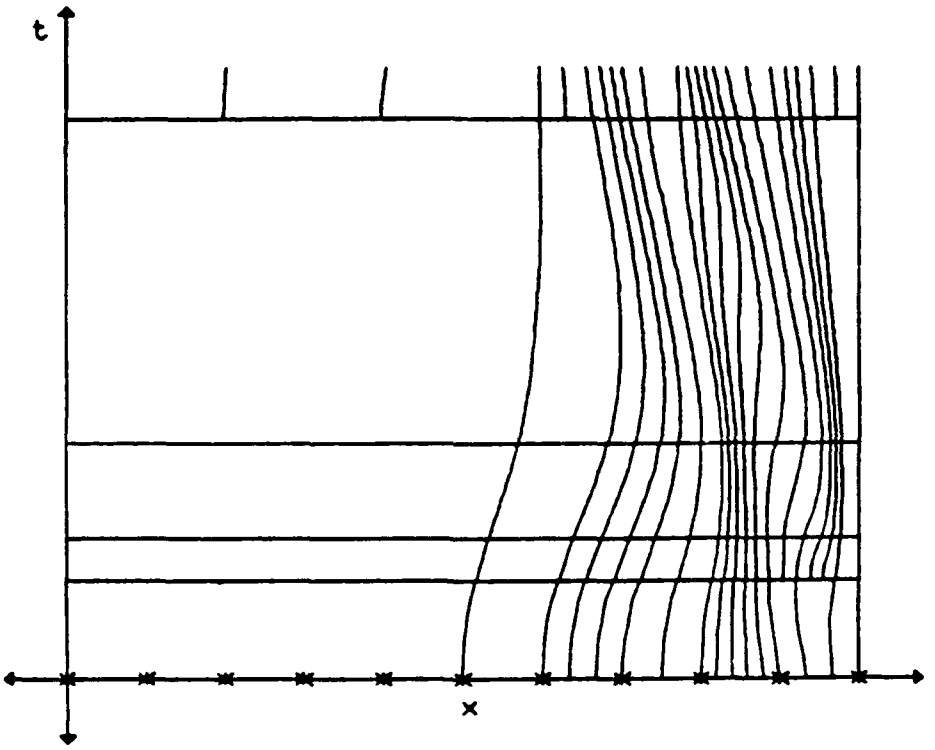


Figure 3

Mesh Trajectories for Case 3. Stars on x-axis indicate the mesh input before any moving or refining. Horizontal lines indicate a temporal refinement has occurred (actual values of Δt are not shown).

DISCUSSION. Overall, the results are very encouraging. The mesh trajectories flow smoothly with whatever solution characteristic the equidistribution law is based. This is true even when the initial mesh is unrelated to the equidistribution rule and when the refinement procedure alters the mesh (see Figures 2 and 3). This is exactly as desired and postulated.

Furthermore, it seems that mesh moving can decrease the amount of refinement necessary for a given problem as hoped. This is evident when comparing case 1 with cases 2 and 3.

In case 2, the level of temporal refinement is less than in case 1 for the same number of time steps and the same tolerance level. This is as expected since the temporal component of the error is proportional to a time derivative of the solution. Hence, movement based on equidistributing this error should reduce the temporal refinement necessary.

Similarly, in case 3, the level of spatial refinement is less than in case 1. Once again, this is as expected since the movement here is based on a quantity proportional to the spatial component of the error.

CONCLUSIONS. Whether or not this new moving scheme will develop into a robust numerical procedure is still uncertain. There are still stability questions to be answered as well as some implementation difficulties not addressed here.

However, the results presented here give credence to the notion that mesh moving and refinement schemes can be successfully combined. Refinement procedures do not have to interfere with mesh movement and mesh movement can be performed to reduce the levels of refinement necessary to solve a problem to a given tolerance.

REFERENCES

1. C. deBoor, A Practical Guide to Splines, Springer-Verlag, New York, 1978.

2. U. Ascher, J. Christiansen, and R.D. Russell, "Collocation Software for Boundary-Value Codes," ACM Trans. Math. Software, Vol. 7, No. 2, June 1981, pp. 209-222.

3. M. Lentini and V. Pereyra, "An Adaptive Finite Difference Solver for Nonlinear Two-Point Boundary Problems With Mild Boundary Layers," SIAM J. Numer. Anal., Vol. 14, No. 1, March 1977, pp. 91-111.

4. M.J. Berger and J. Oliger, "Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations," J. Comput. Phys., Vol. 53, 1984, pp. 484-512.

5. V. Pereyra and E.G. Sewell, "Mesh Selection for Discrete Solution of Boundary Problems in Ordinary Differential Equations," Numer. Math., Vol. 23, No. 3, 1975, pp. 261-268.

6. J.B. Bell and G.R. Shubin, "An Adaptive Grid Finite Difference Method for Conservation Laws," J. Comput. Phys., Vol. 52, 1983, pp. 569-591.

7. S.F. Davis and J.E. Flaherty, "An Adaptive Finite Element Method for Initial-Boundary Value Problems for Partial Differential Equations," SIAM J. Sci. Statist. Comput., Vol. 3, No. 1, 1982, pp. 6-27.

8. H.A. Dwyer, "A Discussion of Some Criteria For the Use of Adaptive Gridding," Adaptive Computational Methods for Partial Differential Equations, (I. Babuska, J. Chandra, and J.E. Flaherty, eds.), SIAM, Philadelphia, PA, 1983, pp. 111-122.

ADAPTIVE METHODS AND PARALLEL COMPUTATION FOR PARTIAL DIFFERENTIAL EQUATIONS*

Rupak Biswas, Messaoud Benantar

Department of Computer Science
Rensselaer Polytechnic Institute
Troy, NY 12180

and

Joseph E. Flaherty

Department of Computer Science
Rensselaer Polytechnic Institute
Troy, NY 12180
and
U.S. Army Armament, Munitions, and Chemical Command
Armament Research, Development, and Engineering Center
Close Combat Armaments Center
Benét Laboratories
Watervliet, NY 12189

ABSTRACT. Consider the adaptive solution of two-dimensional vector systems of hyperbolic and elliptic partial differential equations on shared-memory parallel computers. Hyperbolic systems are approximated by an explicit finite volume technique and solved by a recursive local mesh refinement procedure on a tree-structured grid. Local refinement of the time steps and spatial cells of a coarse base mesh is performed in regions where a refinement indicator exceeds a prescribed tolerance. Computational procedures that sequentially traverse the tree while processing solutions on each grid in parallel, that process solutions at the same tree level in parallel, and that dynamically assign processors to nodes of the tree have been developed and applied to an example. Computational results comparing a variety of heuristic processor load balancing techniques and refinement strategies are presented.

For elliptic problems, the spatial domain is discretized using a finite quadtree mesh generation procedure and the differential system is discretized by a finite element-Galerkin technique with a hierarchical piecewise polynomial basis. Adaptive mesh refinement and order enrichment strategies are based on control of estimates of local and global discretization errors. Resulting linear algebraic systems are solved by a conjugate gradient technique with a symmetric successive over-relaxation

* This research was partially supported by the U. S. Air Force Office of Scientific Research, Air Force Systems Command, USAF, under Grant Number AFOSR-90-0194; by the SDIO/IST under management of the U. S. Army Research Office under Contract Number DAAL03-90-C-0096; and by the National Science Foundation under Grant Numbers CDA-8805910 and CCR-8920694.

preconditioner. Stiffness matrix assembly and linear system solution are processed in parallel with computations scheduled on noncontiguous quadrants of the tree in order to minimize process synchronization. Determining noncontiguous regions by coloring the regular finite quadtree structure is far simpler than coloring elements of the unstructured mesh that the finite quadtree procedure generates. We describe a linear-time complexity coloring procedure that uses a maximum of six colors.

1. INTRODUCTION. Partial differential equations that arise in scientific and engineering applications typically feature solutions that develop, evolve, and decay on diverse temporal and spatial scales. The Fokker-Plank equation of mathematical physics may be used to illustrate this phenomena. Perspective renditions of its solution u as a function of two spatial arguments x and y are shown at four times t in Figure 1 [20]. As time progresses, a single "spike" in the probability density arising from an initial Maxwell-Boltzmann distribution evolves into the two spikes shown. Conventional fixed-step and fixed-order finite difference and finite element techniques for solving such problems would either require excessive computing resources or fail to adequately resolve nonuniform behavior. As a result, they are gradually being replaced by adaptive methods that offer greater efficiency, reliability, and robustness by automatically refining, coarsening, or relocating meshes or by varying the order of numerical accuracy.

Adaptive software for ordinary differential equations has existed for some time and procedures that vary both mesh spacing and order of accuracy are in common use for both initial [17] and boundary [7] value problems. The situation is far more difficult for partial differential equations due to the greater diversity of phenomena that can occur; however, some production-ready adaptive software has appeared for elliptic problems [12]. The state of the art for transient problems lags that for elliptic systems but some projects are underway [16]. Adaptive strategies will either have to be retrofitted into an existing software system for solving partial differential equations or have to be coupled with pre- and post-processing software tools before widespread use occurs.

With an adaptive procedure, an initial crude approximate solution generated on a coarse mesh with a low-order numerical method is enriched until a prescribed accuracy level is attained. Adaptive strategies in current practice are classified as h-, p-, or r-refinement when, respectively, computational meshes are refined or coarsened in regions of the problem domain that require more or less resolution [6, 12], the order of accuracy is varied in different regions [10], or a fixed-topology mesh is redistributed [5]. These basic enrichment methods may be used alone or in combination. The particular combination of h- and p-refinement, for example, has been shown to yield exponential convergence rates in certain situations [9].

Enrichment indicators, which are frequently estimates of the local discretization error of the numerical scheme, are used to control the adaptive process. Resources (finer meshes, higher-order methods, etc.) are introduced in regions having large enrichment indicators and deleted from regions where indicators are low. Using estimates of the discretization error as enrichment indicators also enables the calculation

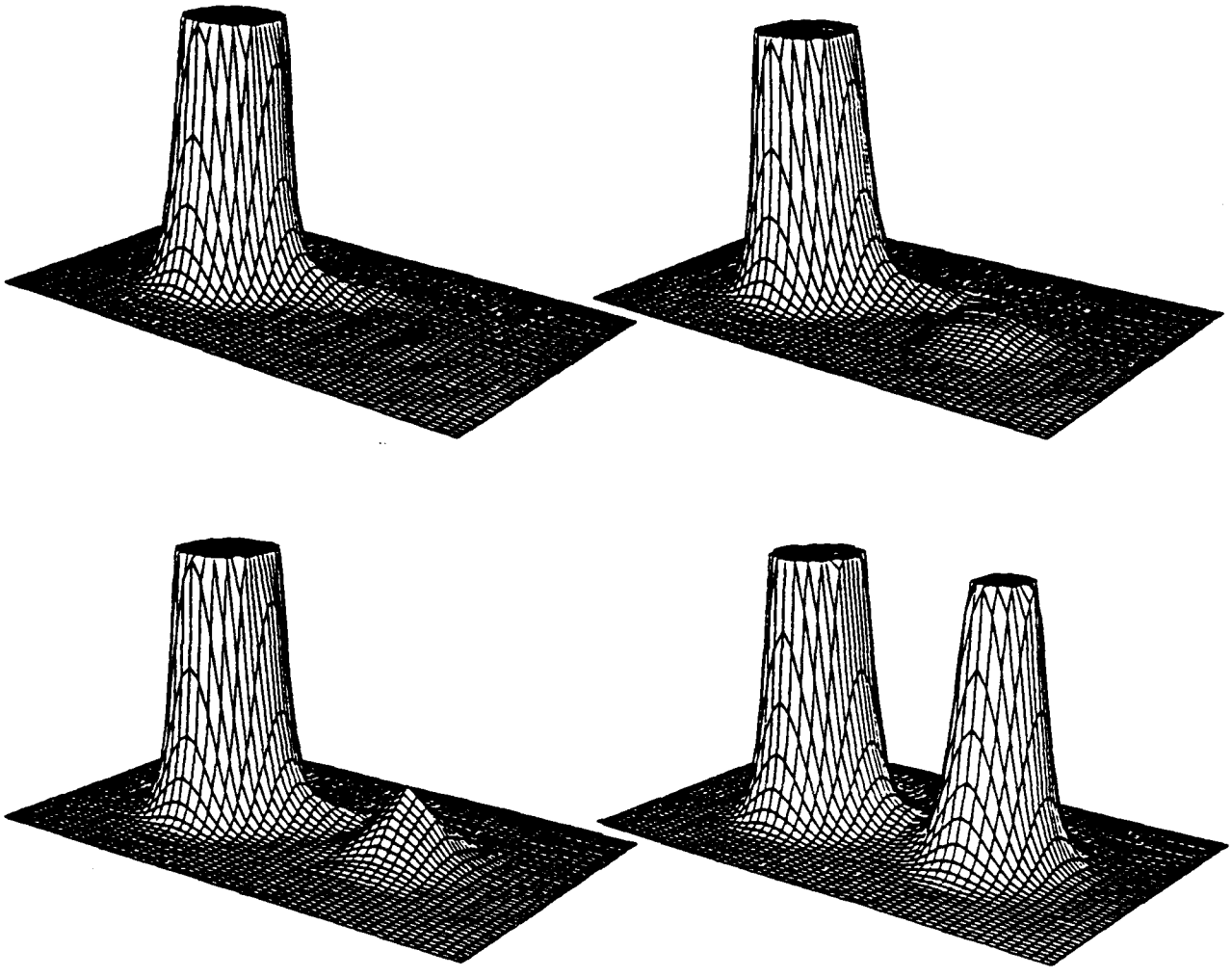


Figure 1. Solution $u(x,y,t)$ of the Fokker-Plank equation at times $t = 4$ (upper left), 10 (upper right), 20 (lower left), and 100 (lower right) obtained by Moore and Flaherty [20]. Solutions having magnitudes greater than 0.1 have been omitted in order to emphasize fine-scale structure.

of local and global accuracy measures which should become a standard part of every scientific computation. Estimates of the local discretization error are typically obtained by using either h- or p-refinement. Thus, one uses the difference between solutions computed either on two meshes or with two distinct orders of accuracy to furnish an error estimate. Special "superconvergence" points where solutions converge faster than they do globally can be used to significantly reduce computational cost [2].

Parallel procedures are becoming increasingly important both as hardware systems become available and as problem complexity increases. Furthermore, the efficiency afforded by adaptive strategies, cannot be ignored in a parallel computational

environment since the demand to model nature more accurately is always beyond hardware capabilities. Models of parallel computation are based on distributed-memory and shared-memory architectures. Distributed-memory systems tend to have large numbers of relatively simple processing elements connected in a network. Available memory on these fine-grained systems is distributed with the processing elements at the nodes of the network, so data access is by message passing. Balancing communication and synchronizing processing is extremely important because processing elements are typically operating in lock-step fashion in order to improve throughput and processor utilization. Shared-memory systems involve a more coarse-grained level of parallelism with relatively few processors operating asynchronously and communicating with a global memory, although variations are common. For example, processing elements may have a local cache memory in order to reduce bus contention and may have vector capabilities; thus, providing a hierarchy of coarse- and fine-grained parallelism.

Our goal is to develop parallel adaptive methods for partial differential equations. Fortunately, our adaptive software utilizes hierarchical (tree) data structures that have many embedded parallel constructs. Transient hyperbolic problems may generally be solved using explicit numerical techniques which greatly simplify processor communication. Experiments, reported in Section 2, with a variety of tree traversal strategies on an adaptive mesh refinement finite difference scheme [6] indicate that the dynamic load balancing scheme of assigning grid-vertex computations at a given tree level to processors as they become available provided the best parallel performance. Static load balancing strategies, that either traverse the tree of grids serially while processing solutions on each grid in parallel or traverse the tree in parallel while processing solutions on grids at the same tree level are also discussed. These alternatives to dynamic processor assignment may provide better performance on hierarchical memory computers.

For elliptic problems, system assembly and solution are processed in parallel with computations scheduled on noncontiguous tree quadrants in order to minimize process synchronization. "Coloring" the elements of a mesh so as to avoid memory contention on a shared-memory computer is far simpler when an underlying tree structure is present than for more general unstructured grids that the finite quadtree structure generates. The six-color procedure, described in Section 3, for the finite element solution scheme on quadtree-structured grids displays a high degree of parallelism when piecewise linear approximations are used. Unfortunately, the same procedure does not do as well with higher-order piecewise polynomial approximations; however, an element edge coloring procedure may improve performance.

2. HYPERBOLIC SYSTEMS. Consider a system of two-dimensional conservation laws in m variables on a rectangular domain Ω having the form

$$\mathbf{u}_t + \mathbf{f}_x(x,y,t,\mathbf{u}) + \mathbf{g}_y(x,y,t,\mathbf{u}) = 0, \quad (x,y) \in \Omega, \quad t > 0, \tag{1a}$$

subject to the initial conditions

$$\mathbf{u}(x,y,0) = \mathbf{u}^0(x,y), \quad (x,y) \in \Omega \cup \partial\Omega, \tag{1b}$$

and appropriate well-posed boundary conditions. The functions $\mathbf{u}$, $\mathbf{f}$, $\mathbf{g}$, and $\mathbf{u}^0$ are m-vectors, x and y are spatial coordinates, t denotes time, and $\partial\Omega$ is the boundary of Ω.

Our research is based on a serial adaptive hr-refinement algorithm of Arney and Flaherty [6]. We forego mesh motion at present and briefly describe an h-refinement procedure that utilizes their strategy. The problem (1) is solved on a coarse rectangular "base" mesh for a sequence of base-mesh time slices of duration Δt_n, $n = 0, 1, \cdots$, by an explicit finite difference, finite volume, or finite element scheme. For a base-mesh time step, say from t_n to $t_{n+1} = t_n + \Delta t_n$, a discrete solution is generated on the base mesh along with a set of local enrichment indicators which, in this case, are refinement indicators. Cells of the mesh where refinement indicators at t_{n+1} fail to satisfy a prescribed tolerance are identified and grouped into rectangular clusters. After ensuring that clusters have an adequate percentage of high-refinement-indicator cells and subsequently enlarging the clusters by a one-element buffer to provide a transition between regions of high and low refinement indicators, cells of the base mesh are bisected in space and time to create finer meshes that are associated with each rectangular cluster. Local problems are solved on the finer meshes and the refinement procedure is repeated until refinement indicators satisfy the prescribed unit-step criteria. After finding an acceptable solution on the base mesh, the integration continues with, possibly, a new base-mesh time step Δt_{n+1}. Data management involves the use of a tree structure with nodes of the tree corresponding to meshes at each refinement level for the current base-mesh time step. The base mesh is the root of the tree and finer grids are regarded as offspring of coarser ones.

With an aim of maintaining generality at the possible expense of accuracy and performance, we discretize (1) using the Richtmyer two-step version of the Lax-Wendroff method [23], which we describe for a one-dimensional problem as follows. Introduce a mesh on Ω having spacing $\Delta x_j = x_{j+1} - x_j$ and let the discrete approximation of $\mathbf{u}(x_j, t_n)$ be denoted as $\mathbf{U}_j^n$. Predicted solutions at cell centers are generated by the Lax-Friedrichs scheme, i.e.,

$$\mathbf{U}_{j+\frac{1}{2}}^{n+\frac{1}{2}} = \tfrac{1}{2}(\mathbf{U}_{j+1}^n + \mathbf{U}_j^n) - \frac{\Delta t_n}{2\Delta x_j}(\mathbf{f}_{j+1}^n - \mathbf{f}_j^n). \tag{2a}$$

This provisional solution is then corrected by the leapfrog scheme

$$\mathbf{U}_j^{n+1} = \mathbf{U}_j^n - \frac{2\Delta t_n}{\Delta x_j + \Delta x_{j-1}}(\mathbf{f}_{j+\frac{1}{2}}^{n+\frac{1}{2}} - \mathbf{f}_{j-\frac{1}{2}}^{n+\frac{1}{2}}). \tag{2b}$$

Following Arney et al. [4, 6], refinement indicators are selected as estimates of the local discretization error obtained by Richardson's extrapolation (h-refinement) on a mesh having half the spatial and temporal spacing of the mesh used to generate the solution. Fine-mesh solutions generated as part of this error estimation process may subsequently be used on finer meshes when refinement is necessary. Initial and boundary data for refined meshes is determined by piecewise bilinear polynomial interpolation from acceptable solutions on the finest available meshes.

Parallel procedures are developed for the adaptive h-refinement solution scheme described above using a P-processor concurrent read exclusive write (CREW) shared-memory MIMD computer. We consider both static and dynamic strategies for balancing processor loading. As the names imply, with static load balancing, processors are assigned tasks a priori with the goal of having them all terminate at approximately the same time, whereas, with dynamic load balancing, available processors are assigned tasks from a task queue. Two possible static load balancing techniques come to mind: (i) serial depth-first traversal of the tree of grids with solutions on each grid being generated in parallel and (ii) parallel generation of solutions on all grids that are at the same tree level. With the depth-first traversal procedure, each grid is statically divided into P subregions and a processor is assigned to each subregion. With the parallel tree traversal procedure, the P processors are distributed among all grids at a particular tree level so as to balance loading. Thus, parallelism occurs both within a grid and across the breadth of the tree with this strategy. In both cases, the parallel solution process proceeds from one base-mesh time step to the next.

Serial depth-first traversal of the tree leads to a highly structured algorithm that has a straight-forward design because the same procedure is used on all grids. Balancing processor loading on rectangular grids is nearly perfect with an explicit finite difference scheme like (2) and similar behavior could be expected for geometrically complex regions. Load imbalance occurs due to differences in the time required to compute initial data. Other than at $t = 0$, initial data is determined by interpolating solutions from the finest grid at the end of the previous base-mesh time step to the present grid. Tree traversal, required to determine the correct solution vertices for the interpolation, would generally take different times in different regions due to variations in tree depth. This defect might be remedied by using either a more sophisticated domain decomposition technique or a more complex data structure to store the tree of grids.

The serial depth-first traversal procedure becomes inefficient when P is of the order of the number of elements in a grid. This situation can be avoided by refining grids by more than a binary factor; thus, maintaining a shallow tree depth. Lowering the efficiency of clusters by including a greater percentage of low-refinement-indicator cells will also increase grid size but diminish optimal grid usage. The inefficiency cited here should not be a factor on data-parallel computers and the serial tree traversal procedure might also be viable there.

The parallel tree-traversal procedure requires complex dynamic scheduling to assign processors to grids. One possibility is to estimate the work remaining to reduce error estimates to prescribed tolerances and to assign processors to subgrids so as to balance this load. Were such a heuristic technique successful, the parallel tree traversal procedure would not degrade in efficiency when the number of elements on a grid is $O(P)$.

Consider a situation where Q processors are used to obtain a solution on a grid at tree level $l - 1$ and suppose that refinement indicators dictate the creation of L grids $G_{l,i}$, $i = 1, 2, \cdots, L$, at level l. Further assume that

i. the prescribed local refinement tolerance at level $l-1$ is τ_{l-1};

ii. the areas of $G_{l,i}$ are $M_{l,i}$, $i = 1, 2, \cdots, L$;

iii. estimates $E_{l,i}$ of the discretization error are available for $G_{l,i}$, $i = 1, 2, \cdots, L$; and

iv. the convergence rate of the numerical scheme is known as a function of the local mesh spacing.

The Richtmyer two-step scheme (2) has a quadratic convergence rate which we use to illustrate the load balancing technique; however, the approach easily extends to other convergence rates.

In order to satisfy the prescribed accuracy criterion, $G_{l,i}$ should be refined by a factor of $(E_{l,i}/\tau_{l-1})^2$. The time step on $G_{l,i}$ must be reduced by a factor of $E_{l,i}/\tau_{l-1}$ in order to satisfy the Courant condition. Hence, the expected work $W_{l,i}$ to find an acceptable solution on $G_{l,i}$ is

$$W_{l,i} = M_{l,i}\left[\frac{E_{l,i}}{\tau_{l-1}}\right]^3. \tag{3}$$

The Q available processors should be allocated so as to balance the time required to complete the expected work on each of the L grids at level l. Thus, assign Q_i processors to grid $G_{l,i}$, $i = 1, 2, \cdots, L$, so that

$$\frac{W_{l,1}}{Q_1} = \frac{W_{l,2}}{Q_2} = \cdots = \frac{W_{l,L}}{Q_L}, \quad \sum_{i=1}^{L} Q_i = Q. \tag{4a,b}$$

The quality of load balancing using this approach will depend on the accuracy of the discretization error estimate. Previous investigations [4, 6] revealed that error estimates were generally better than 80 percent of the actual error for a wide range of mesh spacings and problems. Equation (3) can be used to select refinement factors other than binary and, indeed, to select different refinement levels for different meshes at a given tree level. This consideration combined with over-refinement to a tolerance somewhat less than the prescribed tolerance should maintain a shallow tree depth and enhance parallelism at the expense of grid optimality.

Simple dynamic load balancing can take full advantage of the CREW shared-memory MIMD environment. One just maintains a queue of mesh points at a given tree level and compute solutions at these points as processors become available. Load balancing is perfect except for any inherent system hardware anomalies. Balancing processor loads on geometrically complex regions is as simple as on rectangular regions because mesh points are processed on a first-come-first-serve basis independently of the grid to which they belong. Nonuniformities in initial data also introduce no problems and neither does the relationship of P to the number of cells in a grid. Finally, complex processor scheduling based on accurate error estimates is avoided. This strategy, however, might not be appropriate for hierarchical or distributed-memory computers.

Binary refinement of space-time grids may be optimal in using the fewest mesh points; however, tree depth tends to be large and this introduces serial overhead into a parallel procedure. As previously suggested, serial overhead can be reduced by keeping tree depth shallow and to do this we perform M-ary instead of binary refinement. The value of M is chosen adaptively for different clusters so that the prescribed tolerance is satisfied after a single refinement step. Thus, if τ_0 is the prescribed local discretization error tolerance, then choose M for grid $G_{1,i}$ as the first even integer greater than $E_{1,i}/\tau_0$. Having a good a priori knowledge of the work required on each cluster, processors can be distributed among the grids according to (4) to effectively balance loading. Of course, the refinement tolerance may not be satisfied after performing one level of M-ary refinement. Should this occur, we perform additional levels of 2-ary refinement until accuracy requirements are satisfied. The terms "binary" and "2-ary" refinement have been used to distinguish differences in our methods of checking the refinement condition. With binary refinement, the refinement condition is checked after each of the two finer time steps but with 2-ary refinement, the condition is only checked after the second time step. As a result, the fine grids remain unchanged for both of the two finer time steps with 2-ary refinement.

The efficiency of this mesh refinement strategy and of the serial depth-first traversal and dynamic balancing techniques are appraised in an example. Performance of the parallel traversal procedure was not as good as either of these schemes and results are not presented for it. Computer codes based on these algorithms have been implemented on a 16-processor Sequent Balance 21000 shared-memory parallel computer. Parallelism on this system is supported through the use of a parallel programming library that permits the creation of parallel processes and enforces synchronization and communication using barriers and hardware locks. CPU time and parallel speed up are used as performance measures.

Example 1. Consider the linear scalar differential equation

$$u_t + 2u_x + 2u_y = 0, \quad 0 < x, y < 1, \quad t > 0, \tag{5a}$$

with initial and Dirichlet boundary data specified so that the exact solution is

$$u(x,y,t) = \tfrac{1}{2}[1 - \tanh(100x - 10y - 180t + 10)]. \tag{5b}$$

The solution (5b) is a relatively steep but smooth wave that moves at an angle of 45 degrees across the square domain as time progresses.

Adaptive refinement is controlled by using an approximation of the local discretization error in the L^1 norm as a refinement indicator. Exact errors for this scalar problem are also measured in L^1 as

$$\|e(\cdot,\cdot,t)\|_1 = \iint_{\Omega} |\mathbf{P}u(x,y,t) - U(x,y,t)|\, dxdy, \tag{6}$$

where $U(x,y,t)$ is a piecewise constant representation of the discrete solution and $\mathbf{P}u(x,y,t)$ is a projection onto the space of piecewise constant functions obtained by using values at cell centers.

Our first experiment involves the solution of (5) for $0 < t \leq 0.35$ on 10×10, 25×25, and 45×45 uniform grids having initial time steps of 0.017, 0.007, and 0.004, respectively. No spatial refinement was performed and the static and dynamic load balancing strategies were used. CPU times and parallel speed ups for each base mesh for the two load balancing technique are shown in Figure 2. Speed up with 15 processors and the static load balancing technique (shown in the upper portion of Figure 2) are in excess of 51, 75, and 87 percent of ideal with the 10×10, 25×25, and 45×45 base meshes, respectively. Speed up increases dramatically as the mesh is made finer due to smaller data granularity. Similar speed up data for the three base meshes with the dynamic load balancing technique (shown in the lower portion of Figure 2) are 53, 77, and 90 percent of ideal. The static load balancing strategy takes slightly more time than the dynamic technique, except in the uniprocessor case where they are identical, because of load imbalances on the P subdomains due to differences in the times required to generate initial and boundary data.

Our second experiment involves solving (5) for $0 < t \leq 0.35$ on a 10×10 base mesh having an initial time step of 0.017 using dynamic load balancing and adaptive h-refinement with either binary refinement or M-ary followed by 2-ary refinement. Refinement tolerances of 0.012, 0.006, and 0.003 were selected. The resulting CPU times and parallel speed ups for each adaptive strategy are presented in Figure 3. Maximum speed ups shown in the upper portion of Figure 3 for the binary refinement strategy are in excess of 82, 86, and 72 percent of ideal for tolerances of 0.012, 0.006, and 0.003, respectively. Initially, parallel performance improves as the tolerance is decreased due to the finer data granularity; however, the performance ultimately degrades due to the serial overhead incurred when managing a more complex data structure. Maximum speed ups for the more sophisticated M-ary followed by 2-ary refinement strategy shown in the lower portion of Figure 3 are in excess of 88, 82, and 73 percent of ideal for the three decreasing tolerances. Speed ups for this refinement strategy are only marginally better than those for the binary refinement technique, but the CPU times for the M-ary strategy are much less than those for the binary refinement strategy. For example, CPU times with 15 processors and a tolerance of 0.003 were 226.11 and 182.73 for the binary and M-ary strategies, respectively. Maintaining a shallow tree has clearly increased performance by reducing the serial overhead associated with its management.

Speed up is not an appropriate measure of the complexity required to solve a problem to a prescribed level of accuracy. Tradeoffs occur between the higher degree of parallelism possible with a uniform mesh solution and the greater sequential efficiency of an adaptive procedure. In order to gage the differential, we generated uniform mesh and adaptive mesh solutions of (5) on various processor configurations and to varying levels of accuracy for both static serial tree traversal and dynamic load balancing strategies. Computations on uniform grids ranged from a 5×5 mesh to a 45×45 mesh. All adaptive computations used a 10×10 base mesh, M-ary followed by 2-ary refinement, and tolerances ranging from 0.012 to 0.003.

Results for the global L^1 error as a function of CPU time are presented in Figure 4 for computations performed on 1, 4, 8, and 15 processor systems. Static and dynamic load balancing strategies are shown in the upper and lower portions of the

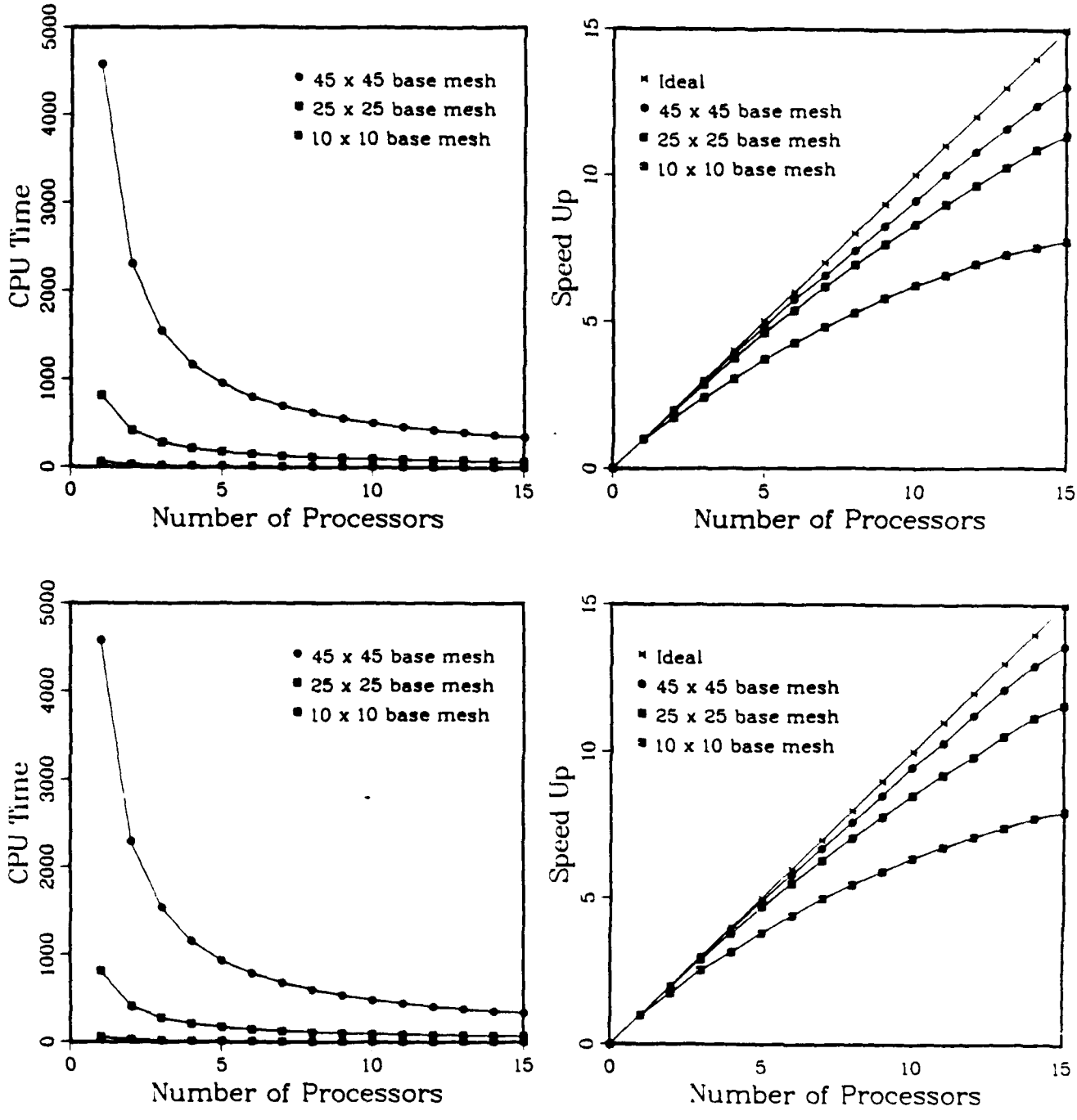


Figure 2. CPU time (left) and parallel speed up (right) for Example 1 on uniform meshes without adaptivity using static (top) and dynamic (bottom) load balancing.

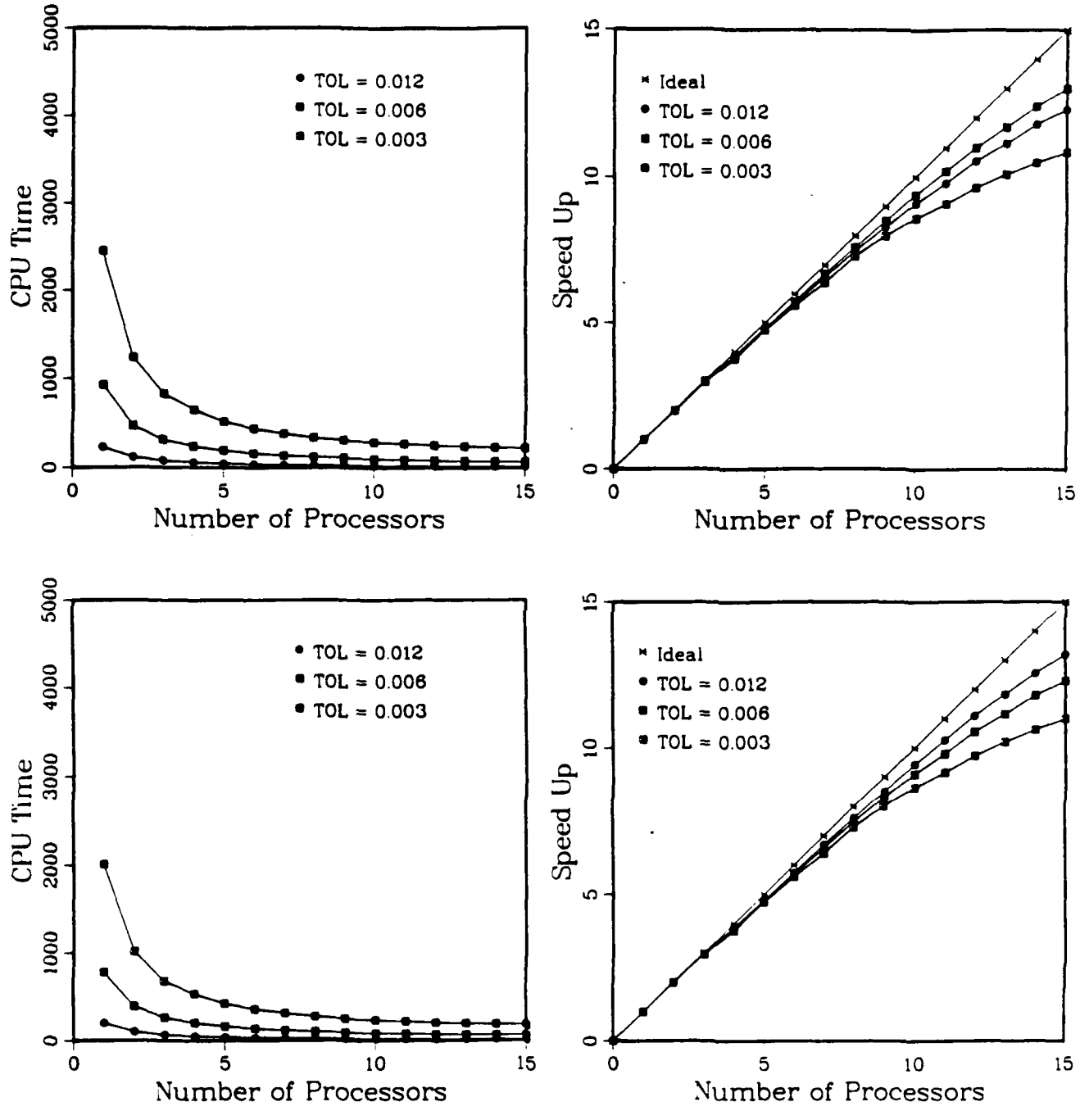


Figure 3. CPU time (left) and parallel speed up (right) for Example 1 using dynamic load balancing and adaptive h-refinement with local binary refinement (top) and M-ary followed by 2-ary refinement (bottom).

figure, respectively. For each strategy, the upper set of curves, displaying non-adaptive results, are much less efficient and converging at a much slower rate than the adaptive solutions shown in the lower set of curves. The adaptive solutions are converging at a rate of approximately 1.4 relative to CPU time while the non-adaptive solutions are converging at a rate of approximately 0.4. These results demonstrate a strong preference for adaptive methods for all but the largest tolerances. Note that the CPU times are identical for the two load balancing strategies when only one processor is used for both non-adaptive and adaptive solutions because the configuration reduces to that of a uniprocessor system. Also note that the global L^1 error for a particular choice of base mesh (for non-adaptive methods) or local refinement tolerance (for adaptive methods) is independent of the number of processors used.

3. ELLIPTIC SYSTEMS. With the goal of describing a strategy for solving linear algebraic systems resulting from the finite element discretization of elliptic systems, let us consider a two-dimensional linear elliptic problem in m variables having the form

$$-[\mathbf{D}(x,y)\mathbf{u}_x]_x - [\mathbf{D}(x,y)\mathbf{u}_y]_y + \mathbf{Q}(x,y)\mathbf{u} = \mathbf{f}(x,y), \quad (x,y) \in \Omega, \tag{7a}$$

$$u_i = c_i^E(x,y), \quad (x,y) \in \partial\Omega_i^E, \quad (\mathbf{D}\mathbf{u}_\nu)_i = c_i^N(x,y), \quad (x,y) \in \partial\Omega_i^N,$$
$$i = 1, 2, \cdots, m. \tag{7b,c}$$

The diffusion $\mathbf{D}$ and source $\mathbf{Q}$ are positive definite and positive semi-definite $m \times m$ matrices, respectively, $\partial\Omega = \partial\Omega_i^E \cup \partial\Omega_i^N$, $i = 1, 2, \cdots, m$, and ν is a unit outer normal to $\partial\Omega$.

The Galerkin form of (7) consists of determining $\mathbf{u} \in H_E^1$ satisfying

$$A(\mathbf{v},\mathbf{u}) + (\mathbf{v},\mathbf{f}) = \sum_{i=1}^{m} \int_{\partial\Omega_i^N} v_i c_i^N \, ds, \quad \text{for } \textit{all } \mathbf{v} \in H_0^1, \tag{8a}$$

where

$$A(\mathbf{v},\mathbf{u}) = \int_\Omega [\mathbf{v}_x^T \mathbf{D}\mathbf{u}_x + \mathbf{v}_y^T \mathbf{D}\mathbf{u}_y + \mathbf{v}^T \mathbf{Q}\mathbf{u}] \, dxdy, \quad (\mathbf{v},\mathbf{u}) = \int_\Omega \mathbf{v}^T \mathbf{u} \, dxdy. \tag{8b,c}$$

As usual, the Sobolev space H^1 consists of functions having first partial derivatives in L^2. The subscripts E and 0 further restrict functions to satisfy the essential boundary conditions (7b) and trivial versions of (7b), respectively. Finite element solutions of (8) are constructed by approximating H^1 by a finite-dimensional subspace $S^{N,p}$ and determining $\mathbf{U} \in S_E^{N,p}$ such that

$$A(\mathbf{V},\mathbf{U}) + (\mathbf{V},\mathbf{f}) = \sum_{i=1}^{m} \int_{\partial\Omega_i^N} V_i c_i^N \, ds, \quad \text{for } \textit{all } \mathbf{V} \in S_0^{N,p}. \tag{9}$$

Selecting $S^{N,p}$ as a space of continuous piecewise p *th*-degree hierarchical polynomials [24] with respect to the partition of Ω into triangular finite elements, substituting these approximations into (9), and evaluating the integrals by quadrature rules yields a

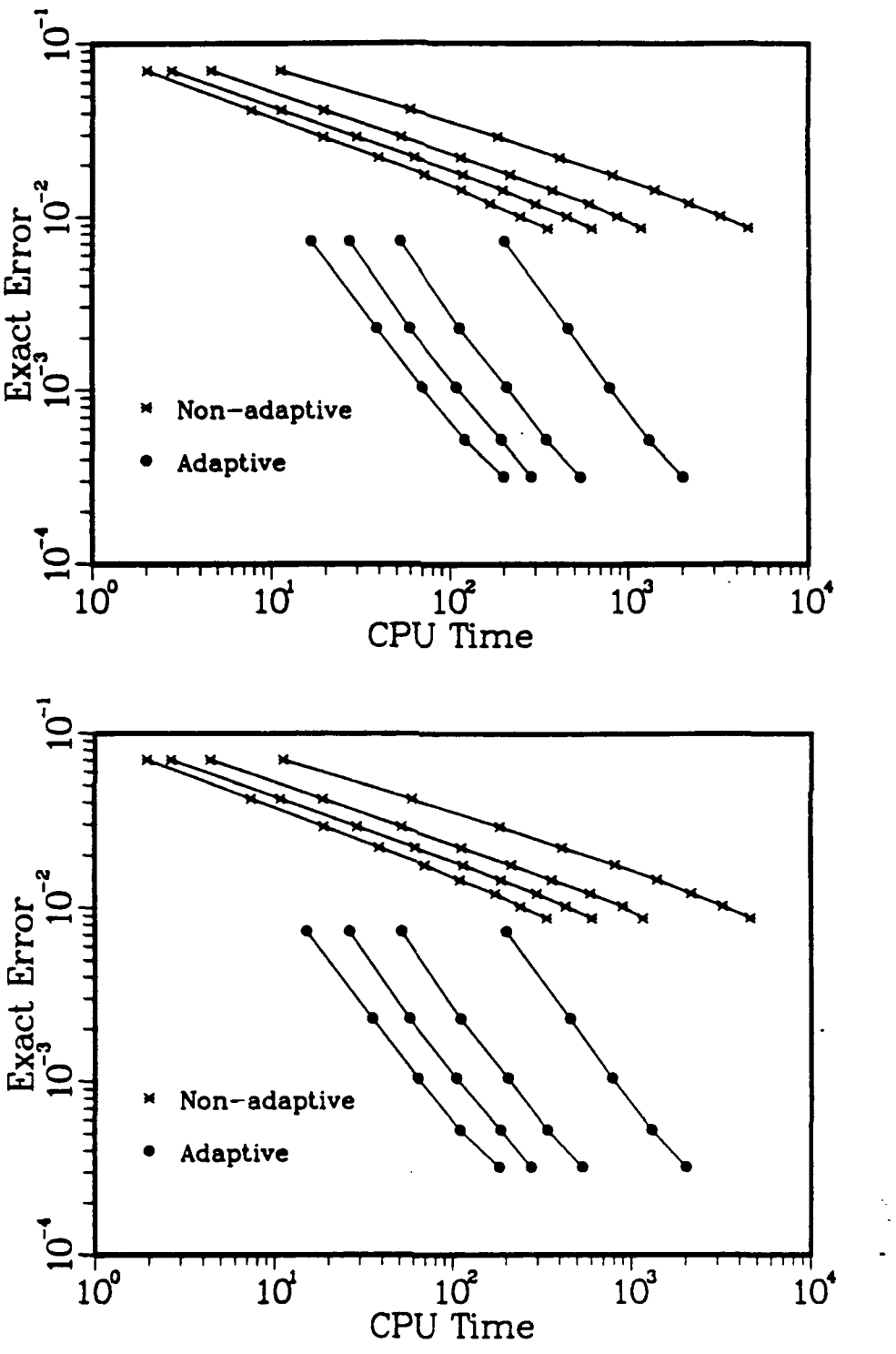


Figure 4. Global L^1 error as a function of CPU time for Example 1 using non-adaptive methods (upper set of curves) and adaptive h-refinement methods (lower set of curves) with static (top) and dynamic (bottom) load balancing. Computations are shown for systems using 1, 4, 8, and 15 processors (right to left for each set of curves).

sparse, symmetric, positive definite, N-dimensional linear system of the form

$$\mathbf{KX} = \mathbf{b}, \tag{10}$$

where $\mathbf{X}$ is an N-vector of Galerkin coordinates.

Meshes of triangular or quadrilateral elements are created automatically on Ω by using the *finite quadtree* procedure [11]. This structure is somewhat different than the tree of grids described in Section 2. With this technique, Ω is embedded in a square "universe" that may be recursively quartered to create a set of disjoint squares called *quadrants*. Data associated with these quadrants is managed by using a hierarchical tree structure with the original square universe regarded as the root and with smaller quadrants created by subdivision regarded as offspring of larger ones. Quadrants intersecting $\partial\Omega$ are recursively quartered until a prescribed spatial resolution of Ω has been obtained. At this stage, quadrants that are leaf nodes of the tree and intersect $\Omega \cup \partial\Omega$ are further divided into small sets of triangular or quadrilateral elements. Severe mesh gradation is avoided by imposing a maximal one-level difference between quadrants sharing a common edge. This implies a maximal two-level difference between quadrants sharing a common vertex. A final "smoothing" of the triangular or quadrilateral mesh improves element shapes and further reduces mesh gradation near $\partial\Omega$.

A simple example involving a domain consisting of a rectangle and a quarter circle, as shown in Figure 5, will illustrate the finite quadtree process. In the upper left portion of the figure, the square universe containing the problem domain is quartered creating the one-level tree structure shown at the upper right. Were this deemed to be a satisfactory geometrical resolution, a mesh of five triangles could be created. As shown, the triangular elements are associated with quadrants of the tree structure. In the lower portion of Figure 5, the quadrant containing the circular arc is quartered and the resulting quadrant that intersects the circular arc is quartered again to create the three-level tree shown in the lower right portion of the figure. A triangular mesh generated on this tree structure is also shown.

Arbitrarily complex two-dimensional domains may be discretized in this manner and generally produce unstructured grids; however, the underlying tree of quadrants remains regular. Adaptive mesh refinement is easily accomplished by subdividing appropriate leaf-node quadrants and generating a new mesh of triangular or quadrilateral elements locally; thus, unifying the mesh generation and adaptive solution phases of the problem under a common tree data structure.

Preconditioned conjugate gradient (PCG) iteration is an efficient means of solving the linear algebraic systems (10) that result from the finite element discretization of self-adjoint elliptic partial differential systems [8]. The key steps in the PCG procedure [22] involve (i) matrix-vector multiplication of the form

$$\mathbf{q} = \mathbf{Kp} \tag{11a}$$

and (ii) solving linear systems of the form

$$\overline{\mathbf{K}}\mathbf{d} = \mathbf{r}, \tag{11b}$$

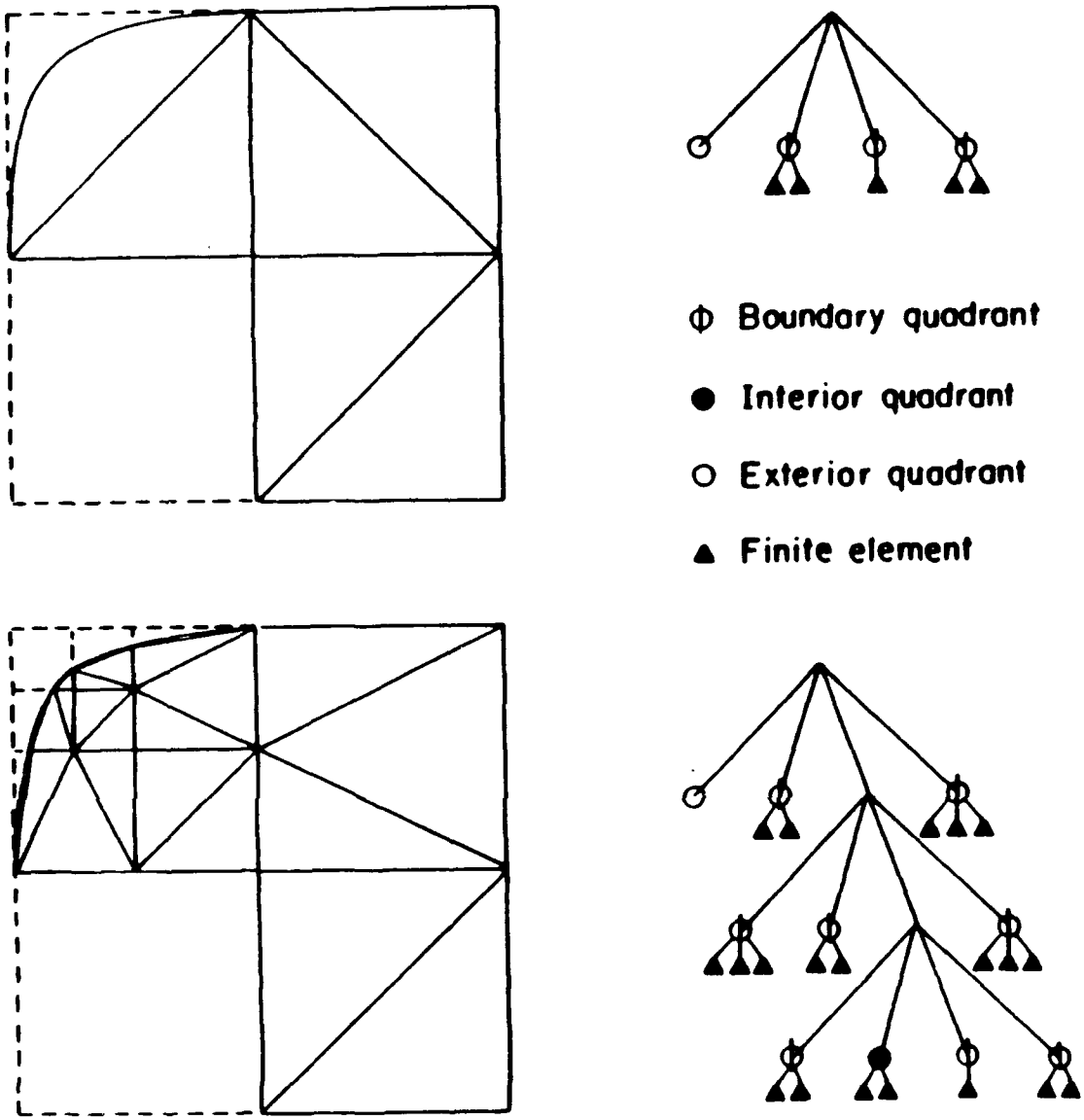


Figure 5. Finite quadtree mesh generation for a domain consisting of a rectangle and a quarter circle. One-level and three-level tree structures and their associated meshes of triangular elements are shown at the top and bottom of the figure, respectively.

where **r** and **p** are the residual vector and conjugate search direction, respectively. The preconditioning matrix $\overline{\mathbf{K}}$ may be selected to reduce computational cost. The element-by-element (EBE) and symmetric successive over-relaxation (SSOR) preconditionings are in common use and seem appropriate for use with quadtree-structured grids. The EBE preconditioning is an approximate factorization of the stiffness matrix **K** into a product of elemental matrices. If the grid has been "colored" so as to

segregate non-contiguous elements, then (11b) can be solved in parallel on elements having the same color. Since the matrix-vector multiplication (11a) can also be performed in an element-by-element fashion, the entire PCG solution can be done in parallel on non-contiguous elements. While this simple approach has been used in several applications [18, 19, 21], we found the SSOR preconditioning to be more efficient in every instance [13] and, therefore, shall not discuss EBE preconditionings any further.

SOR and SSOR iteration have been used for the parallel solution of the five-point difference approximation of Poisson's equation on rectangular meshes by numbering the discrete equations and unknowns in "checkerboard" order [1]. With this ordering, unknowns at "red" mesh points are only coupled to those at "black" mesh points and vice versa; thus, solutions at all red points can proceed in parallel that may be followed by a similar solution at all black points. Preserving symmetry, as with the SSOR iteration, will make the SOR method a suitable preconditioning for the PCG method. Adams and Ortega [1] describe multicolor orderings on rectangular grids using several finite element and finite difference stencils. However, multicolor orderings for unstructured meshes are more difficult since nodal connectivity and difference stencils for high-degree polynomial approximations can be quite complex. The computational effort can be reduced when using quadtree-structured grids by considering multicolor orderings for block SSOR preconditionings at the quadrant level. To be specific, partition the stiffness matrix $\mathbf{K}$ by quadrants as

$$\mathbf{K} = \mathbf{D} - \mathbf{L} - \mathbf{L}^T \tag{12a}$$

where

$$\mathbf{D} = \begin{bmatrix} \mathbf{K}_{1,1} & & & \\ & \mathbf{K}_{2,2} & & \\ & & \cdots & \\ & & & \mathbf{K}_{Q,Q} \end{bmatrix}, \quad \mathbf{L} = -\begin{bmatrix} 0 & & & \\ \mathbf{K}_{2,1} & 0 & & \\ & & \cdots & \\ \mathbf{K}_{Q,1} & \mathbf{K}_{Q,2} & & 0 \end{bmatrix}. \tag{12b,c}$$

Nontrivial entries in a diagonal block $\mathbf{K}_{i,i}$ arise from Galerkin coordinates that are connected through the finite element basis to other unknowns in quadrant i. Nontrivial contributions to block $\mathbf{K}_{i,j}$ of the lower triangular matrix $\mathbf{L}$ arise when the support of the basis associated with a Galerkin coordinate in quadrant i intersects quadrant j.

Using an SSOR preconditioning, the solution of (11b) would be computed according to the two-step procedure

$$\mathbf{X}^{n+\frac{1}{2}} = \omega(\mathbf{L}\mathbf{X}^{n+\frac{1}{2}} + \mathbf{L}^T\mathbf{X}^n + \mathbf{r}) + (1 - \omega)\mathbf{X}^n, \tag{13a}$$

$$\mathbf{X}^{n+1} = \omega(\mathbf{L}^T\mathbf{X}^{n+1} + \mathbf{L}\mathbf{X}^{n+\frac{1}{2}} + \mathbf{r}) + (1 - \omega)\mathbf{X}^{n+\frac{1}{2}}, \quad n = 1, 2, \cdots, M. \tag{13b}$$

Thus, each block SSOR iteration consists of two block SOR steps; one having the reverse ordering of the other. Typically, $M = 3$ SSOR steps are performed between each PCG step.

Suppose that the Q quadrants of a finite quadtree structure are separated into γ disjoint sets. Then, using the symmetric γ-color block SSOR ordering, we would sweep the quadrants in the order $C_1, C_2, \cdots, C_\gamma, C_\gamma, C_{\gamma-1}, \cdots, C_1$, where C_i is the set of quadrants having color i. Because quadrants rather than nodes are colored, a node can be connected to other nodes having the same color. Thus, the forward and backward SOR sweeps may differ for a color C_i, $i = 1, 2, \cdots, \gamma$. During an SOR sweep, unknowns lying on quadrant boundaries are updated as many times as the number of quadrants containing them.

Coloring the regular quadrants of a finite quadtree is far simpler than coloring the elements of a mesh. Differences in the small number of elements within quadrants having the same color may cause some load imbalance and this effect will have to be investigated. Naturally, coloring procedures that use the fewest colors increase data granularity and reduce the need for process synchronization. At the same time, the cost of the coloring algorithm should not be the dominant computational cost. With these views in mind, we developed an eight-color procedure that has linear time complexity [13]. This procedure only required a simple breadth-first traversal of the quadtree, but performance never exceeded that of the six-color procedure which is described in the following paragraphs. Four-color procedures are undoubtedly possible, but we have not formulated any. Their complexity, unlike the eight- and six-color procedures, may be nonlinear.

With the aim of constructing a quadtree coloring procedure using a maximum of six colors, let us define a binary directed graph called a "quasi-binary tree" from the finite quadtree by using the following recursive assertive algorithm.

i. The root of the quadtree corresponds to the root of the quasi-binary tree.

ii. Every terminal quadrant is associated with a node in the quasi-binary tree; however, not every quasi-binary tree node must correspond to a quadrant.

iii. In the planar representation of the quadtree, nodes across a common horizontal edge are connected in the quasi-binary tree.

iv. When a quadrant is divided, its parent node in the quasi-binary tree becomes the root of a subtree.

Planar representations of simple quadtrees and their quasi-binary tree representations are illustrated in Figure 6. The leftmost quadtree illustrates root-node and offspring construction of the quasi-binary tree. Connection of nodes across horizontal edges is shown with and without quadrant division in all three illustrations. Subtree definitions according to assertion (iv) are shown in the center and rightmost quadtrees.

From Figure 6, we see that the column-order traversal of a finite quadtree is the depth-first traversal of its associated quasi-binary tree. Let us define six colors divided into three sets a, b, and c of two disjoint colors that alternate through the columns in a column-order traversal of the quadtree. Whenever left and right quasi-binary tree branches merge, column-order traversal continues using the color set associated with the left branch. Two of the three color streams, say a and b, are passed to a node of the quasi-binary tree. At each branching, the color stream a and the third color stream

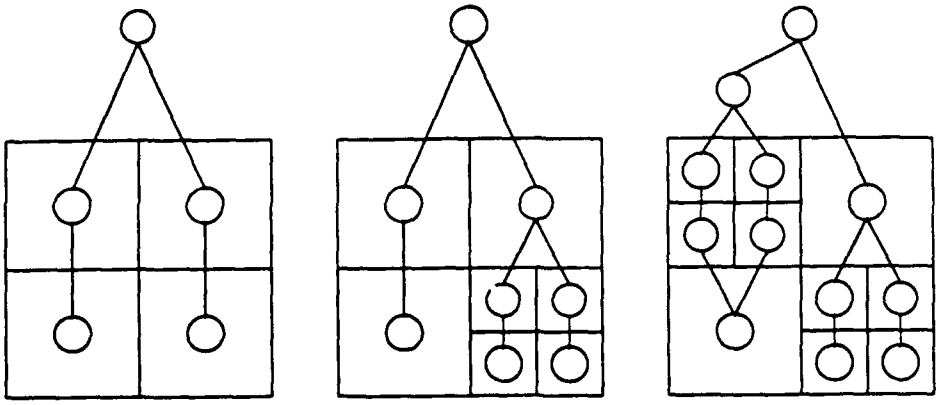


Figure 6. Planar representations of three quadtrees and their associated quasi-binary trees.

c are passed to the left offspring while the streams a and b are passed in reverse order to the right offspring. Additional details and a correctness proof of this algorithm will appear [15].

Computational experiments of Benantar et al. [13] demonstrate the excellent parallelism that may be obtained by the six-color SSOR PCG procedure with piecewise linear finite element approximations. However, higher-order polynomial bases create additional possibilities for processor load imbalance with coloring at the quadrant level. Let us illustrate this with a simple problem. As in Section 2, a 16-processor Sequent Balance 21000 computer was used for the experiment.

Example 2. Consider the Dirichlet problem

$$u_{xx} + u_{yy} = f(x,y), \quad (x,y) \in \Omega, \tag{14a}$$

$$u = 0, \quad (x,y) \in \partial\Omega, \tag{14b}$$

with $\Omega = \{ (x,y) \mid -3 < x, y < 3 \}$. We solved this problem on a 400-element mesh using piecewise linear, quadratic, and cubic approximations. Adaptive p-refinement with the polynomial degree p restricted to be 1, 2, or 3 was also performed. Parallel speed up and processor idle time resulting from the need to synchronize at the completion of each color are shown in Figure 7.

Parallel performance degrades as polynomial degree increases, with the adaptive strategy having the poorest performance. Adaptive algorithms typically have serial logic which limits speed up. Of course, speed up is not the only measure of complexity and an adaptive solution strategy could require less CPU time to solve the problem to a given level of accuracy. Nevertheless, additional research is necessary to improve

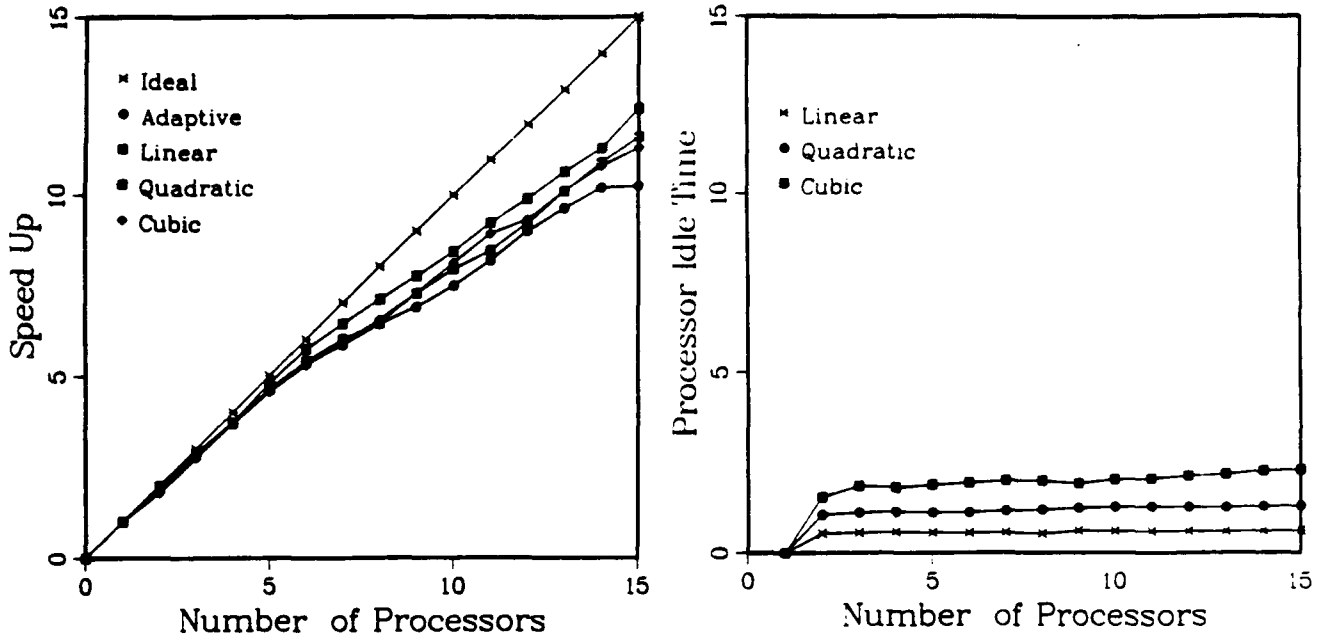


Figure 7. Parallel speed up (left) and processor idle time (right) for the finite element solution of Example 2 using piecewise linear, quadratic, and cubic approximations as well as adaptive p-refinement.

performance with high-order and adaptive strategies.

Using a hierarchical basis, all Galerkin coordinates for polynomial degrees higher than one are associated with mesh points that are either along element edges or within elements. Thus, the Galerkin coordinates for continuous piecewise linear approximations are the only ones associated with element vertices. Parallel performance could, therefore, be improved by coloring element edges rather than quadrants and we have designed a three-color procedure having linear time complexity to do this [15]. Since hierarchical bases add incremental corrections as the polynomial degree is increased, one could conceive an algorithm where quadrant coloring is used with the piecewise linear portion of the approximation and edge coloring is used for higher-degree approximations.

4. DISCUSSION. High-order and hp-refinement strategies have the highest convergence rates on serial processors. Successful use of adaptive strategies in parallel environments depends heavily on the efficient implementation of these procedures on shared- and distributed-memory computers. The edge coloring procedure alluded to in Section 3 should provide some improvement over existing strategies on shared-memory systems, but no procedure is available for using hp-refinement on data-parallel computers. High-order and hp-refinement techniques are being added to our collection of

methods for solving hyperbolic systems using the finite element methods of Cockburn and Shu [14]. The p-hierarchical Legendre polynomial basis embedded in these methods should also furnish error estimates similar to those that we have developed for parabolic systems [3]. These techniques are far more efficient than Richardson's extrapolation.

Our h-refinement procedure for hyperbolic systems could be improved by beginning each base-mesh time step with an adaptively chosen mesh that utilizes known nonuniformities in the solution discovered during the previous base-mesh time step. Processors would still have to be scheduled to balance loads in this case and procedures for doing this are unavailable. Finally, parallel procedures for distributed memory systems and procedures for three-dimensional problems are of great interest.

REFERENCES.

1. L. Adams and J. Ortega, A multi-color SOR method for parallel computation, in K. E. Batcher, W. C. Meilander, and J. L. Potter, Eds., *Proceedings of the International Conference on Parallel Processing*, Computer Society Press, Silver Spring, 1982, pp. 53-56.

2. S. Adjerid and J. E. Flaherty, A moving mesh finite element method with local refinement for parabolic partial differential equations, *Comput. Meths. Appl. Mech. Engr.* **56** (1986), pp. 3-26.

3. S. Adjerid, J. E. Flaherty, and Y. Wang, A posteriori error estimation with finite element methods of lines for one-dimensional parabolic systems, in preparation, 1990.

4. D. C. Arney, R. Biswas, and J. E. Flaherty, An adaptive mesh moving and refinement procedure for one-dimensional conservation laws, Tech. Rep. 90-6, Department of Computer Science, Rensselaer Polytechnic Institute, Troy, 1990.

5. D. C. Arney and J. E. Flaherty, A two-dimensional mesh moving technique for time-dependent partial differential equations, *J. Comput. Phys.* **67** (1986), pp. 124-144.

6. D. C. Arney and J. E. Flaherty, An adaptive mesh-moving and local refinement method for time-dependent partial differential equations, *ACM Trans. Math. Softw.* **16** (1990), pp. 48-71.

7. U. M. Ascher, R. M. Mattheij, and R. D. Russell, *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*, Prentice-Hall, Englewood Cliffs, 1988.

8. O. Axelsson and V. A. Barker, *Finite Element Solution of Boundary Value Problems: Theory and Computation*, Academic Press, Orlando, 1984.

9. I. Babuska and E. Rank, An expert-system like approach in the hp-version of the finite element method, Tech. Note BN-1084, Institute for Physical Science and

Technology, University of Maryland, College Park, 1986.

10. I. Babuska, B. A. Szabo, and I. N. Katz, The p-version of the finite element method, *SIAM J. Numer. Anal.* **18** (1981), pp. 515-545.

11. P. L. Baehmann, S. L. Wittchen, M. S. Shephard, K. R. Grice, and M. A. Yerry, Robust, geometrically based, automatic two-dimensional mesh generation, *Int. J. Numer. Meths. Engrg.* **24** (1987), pp. 1043-1078.

12. R. E. Bank, *PLTMG: A Software Package for Solving Elliptic Partial Differential Equations. Users' Guide 6.0*, Frontiers in Applied Mathematics 7, SIAM, Philadelphia, 1990.

13. M. Benantar, R. Biswas, J. E. Flaherty, and M. S. Shephard, Parallel computation with adaptive methods for elliptic and hyperbolic systems, *Comput. Meths. Appl. Mech. Engr.*, to appear, 1990.

14. B. Cockburn and C.-W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework, *Maths. Comp.* **52** (1989), pp. 411-435.

15. J. E. Flaherty and M. Benantar, Parallel element-by-element techniques for elliptic systems using finite quadtree meshes, in preparation, 1990.

16. J. E. Flaherty, P. J. Paslow, M. S. Shephard, and J. D. Vasilakis, *Adaptive Methods for Partial Differential Equations*, SIAM, Philadelphia, 1989.

17. C. W. Gear, *Numerical Initial Value Problems in Ordinary Differential Equations*, Prentice-Hall, Englewood Cliffs, 1971.

18. I. Gustafsson and G. Lindskog, A preconditioning technique based on element matrix factorizations, *Comput. Meths. Appl. Mech. Engr.* **55** (1986), pp. 201-220.

19. R. B. King and V. Sonnad, Implementation of an element-by-element solution algorithm for the finite element method on a coarse-grained parallel computer, *Comput. Meths. Appl. Mech Engr.* **65** (1987), pp. 47-59.

20. P. K. Moore and J. E. Flaherty, Adaptive local overlapping grid methods for parabolic systems in two space dimensions, Tech. Rep. 90-5, Department of Computer Science, Rensselaer Polytechnic Institute, Troy, 1990.

21. B. Nour-Omid and B. N. Parlett, Element preconditioning using splitting techniques, *SIAM J. Sci. Stat. Comput.* **6** (1985), pp. 761-770.

22. J. M. Ortega, *Introduction to Parallel and Vector Solution of Linear Systems*, Plenum Press, New York, 1988.

23. R. D. Richtmyer and K. W. Morton, *Difference Methods for Initial-Value Problems*, Interscience, New York, 1967.

24. B. Szabo and I. Babuska, *Introduction to Finite Element Analysis*, John Wiley and Sons, to appear, 1990.

SYNTHESIS OF REAL TIME ACCEPTORS

Amr Fawzy Fahmy and Alan W. Biermann

Duke University

Durham, N.C. 27706

ABSTRACT

This paper gives a methodology for constructing real time acceptors from examples of their behavior. The technique involves constructing a behavior graph and factoring it using Hartmanis-Stearns decomposition theory. It provides a learning mechanism that can find a general representation of a class given examples of that class.

INTRODUCTION

This paper describes a method for automatically creating a real time acceptor P from examples of its behavior B. As an illustration, suppose the desired acceptor is to accept the strings a!a, aa!aa ab!ba, ba!ab, ... and reject all others. The methodology begins with these examples and creates a general P which will enter a final state if and only if the input is a string from the general set $\{w!w^R \mid w \in \{a,b\}^*\}$.

This research is a project in learning theory and attempts to find methodologies that generalize from examples to find a representation of the total class. The paper defines the concepts of *data structure* and *control structure* for a real time acceptor and then shows that the *behavior graph* for the acceptor is the product of the data structure and control structure graphs. Synthesis is done by creating the behavior graph from the examples and factoring it into two graphs, the data and control structures for the acceptor.

REAL TIME ACCEPTORS

A *data structure* is defined to be a Moore type deterministic state machine.

$$D = \{S_D, \Sigma_D, O_D, \delta_D, d_O, \lambda_D\}$$

where

- S_D is a finite or infinite set of *states*.
- Σ_D is a finite set of symbols called the *input alphabet of D,* or the *instruction set* of *D.*

- $\delta_D : S_D \times \Sigma_D \rightarrow S_D$ is a mapping by which D changes states, called the transition function of D.

- $d_O \in S_D$ is the *initial state* of D.

- O_D is a finite set of *output symbols* that we interpret as the *test values* on the states of D.

- $\lambda_D : S_D \rightarrow O_D$, is a mapping that assigns to each state of D a test value from O_D.

Conceptually, a data structure has a state and receives instructions from the outside world which modify its state. It is also capable of yielding values which depend on its state. For example, a pushdown stack might begin in the empty state and then receive sequentially the inputs a and b. It's output would then be the item at the top of the stack, b in this case. The state graph of Figure 1 shows the initial part of the infinite graph for a stack.

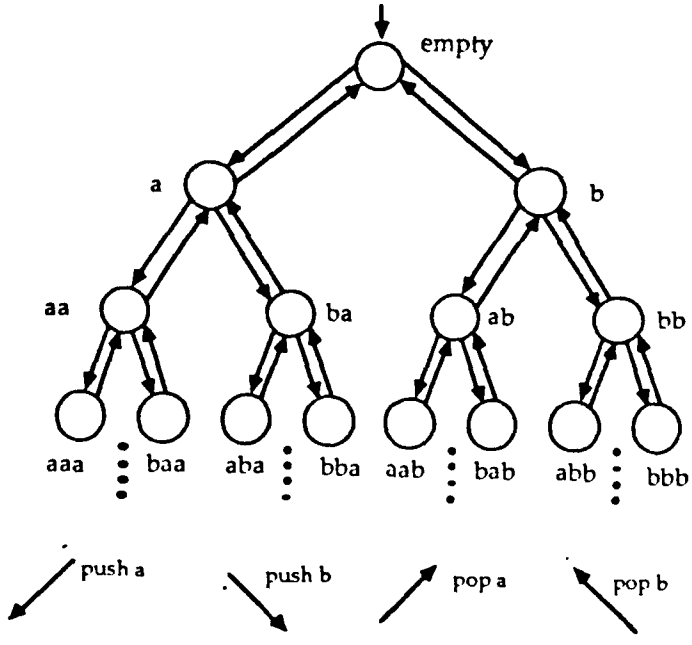


Figure 1. The stack data structure.

A *control structure* is a finite deterministic state machine

$$C = (S_C, \Sigma_C, O_C, \delta_C, c_O, \lambda_C, F_C)$$

where

- S_C is a finite nonempty set of states.

- Σ_C is a finite set of symbols called the *input alphabet* to C. Σ_C is the cross product of two alphabets; the finite nonempty *input alphabet* from the outside world, Σ, and the output alphabet of the DS. So we have

$$\Sigma_C = \Sigma \times O_D$$

- O_C is a set of instructions that can be issued to D. So $O_C = \Sigma_D$.

- $\delta_C : S_C \times \Sigma_C \rightarrow S_C$ is a mapping by which C changes states called the *transition function* of C.

- c_O is the *initial state* of C.

- $\lambda_C : S_C \times \Sigma_C \rightarrow \Sigma_D$ is a mapping that assigns to each transition of C an instruction to D called the *instruction assignment function.*

- $F_C \subseteq S_C$ is a set of final states of C.

The task of the control structure is to receive inputs from the outside world, make modifications to the data structure, and yield an output indicating acceptance or rejection of the string. A *real time acceptor* P operates as shown in Figure 2 and follows three rules. Suppose the acceptor P has its control structure C in state c, its data structure D in state d, and receives input α.

1. C changes states from c to some state c' using its transition function δ_c. The next state depends on α and $\lambda_D(d)$, so we have

$$\delta_c(c,(\alpha, \lambda_D(d))) = c'$$

2. While C is changing states, it sends an instruction to the DS. The instruction to be sent depends on the current state of C, the input alphabet and the test value of the current state of D and is given by

$$\lambda_c(c,(\alpha,\lambda_D(d))) = I$$

3. On receiving the instruction, D will change states from the current state, d, to some state d' using its transition function

$$\delta_D(d,I) = d'$$

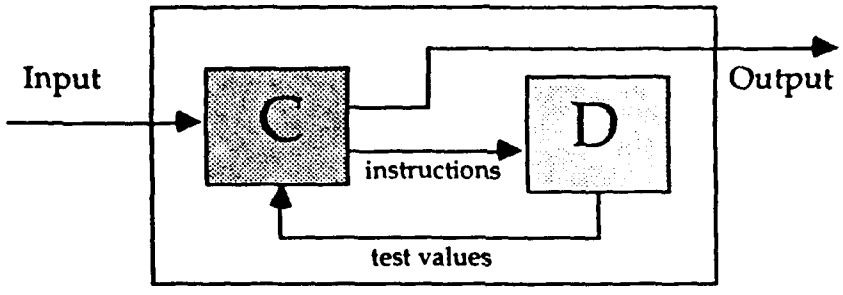


FIGURE 2. The acceptor with its control structure C and data structure D.

An example control structure which employs the stack data structure and accepts the set described above, w!w^R, is shown in Figure 3. Its operation is easy to follow. It begins in the top center state with its associated stack empty and moves left or right depending on whether the first symbol is an a or b. For example, on the string ab!ba, it moves left without changing the stack after the first symbol a. When the b arrives, it pushes that on the stack without changing its control state. On input !, it transitions down one state, and on b and a, it pops the stack and then (with the stack empty) moves to the final state at the center.

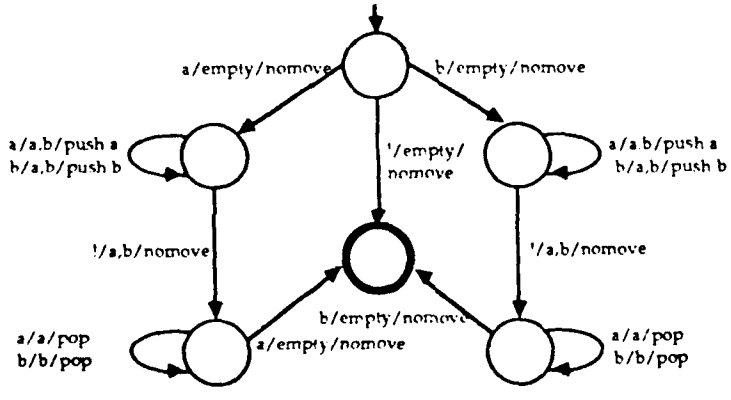


FIGURE 3. Control structure.

SYNTHESIS

A graph cross product operation will be defined in this section for the control structure and data structure graphs. Then we will observe that the behavior graph has the same structure as this cross product graph. The synthesis procedure thus involves creating the behavior graph and then factoring it into the appropriate control and data structures.

Let control structure C and data structure D be

$$C = \{S_C, \Sigma_C, O_C, \delta_C, c_O, \lambda_C, F_C\}$$

and

$$D = \{S_D, \Sigma_D, O_D, \delta_D, d_O, \lambda_D\}$$

The product CxD is defined to be the Moore type state machine

$$M_{CxD} = \{S_{CxD}, \Sigma, \delta_{CxD}, p_O, F_{CxD}\}$$

where

- $S_{CxD} \subseteq S_C \times S_D$.
- $\forall \alpha \in \Sigma$

$$\delta_{CxD}((c,d),\alpha) = (\delta_c(c(\alpha, \lambda_D(d))), \delta_D(d,I))$$

- where

$$I = \lambda_C(c,(\alpha, \lambda_D(d)))$$

- $p_O=(c_O,d_O)$.
- $(c,d) \in F_{CxD}$ *if* $c \in F_C$.

For example, if the graphs of Figures 1 and 3 are combined using this cross product operation, the graph of Figure 4 results. Figure 4 contains no control or data structure information; it is simply a graph of all possible behaviors for the automaton with Figures 1 and 3 as its data and control structures.

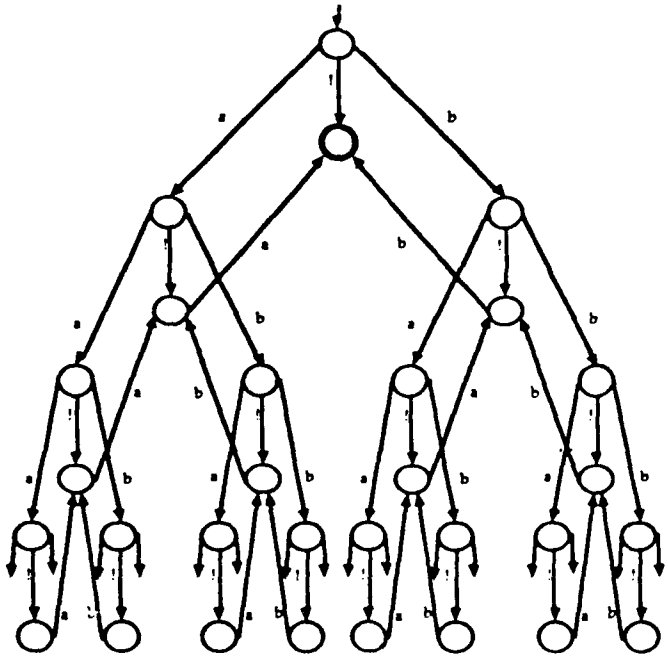


Figure 4. The cross product graph M_{CxD}.

All of this is summarized in the following theorem where L(X) is defined to be the language of X.

Theorem 1. $L(B) = L(M_{CxD})$

The proof appears in Fahmy [88].

The next question asks under what conditions such a decomposition of the behavior graph can occur. The answer is given by an extension to Hartmanis-Stearns decomposition theory (Hartmanis and Stearns [66]).

First some definitions must be made. Let b=(c,d) represent the behavior graph node b that comes from composing control structure state c and data structure state d in the cross product operation of CxD. If $b_1=(c_1,d_1)$ and $b_2=(c_2,d_2)$, then four partitions π_C,π_D,ρ_{CD}, and ρ_{DC} over the set of states of B can be defined.

$b_1 \equiv b_2\ (\pi_C)$ if and only if $c_1 = c_2$.

$b_1 \equiv b_2\ (\pi_D)$ if and only if $d_1 = d_2$.

$b_1 \equiv b_2\ (\rho_{CD})$ if and only if $\forall \alpha \in \Sigma,\ \forall v \in O_D, \lambda_C(c_1,(\alpha,v)) = \lambda_C(c_2,(\alpha,v))$.

$b_1 \equiv b_2(\rho_{DC})$ if and only if $\lambda_D(d_1) = \lambda_D(d_2)$.

The decomposition theorem is as follows:

Theorem 2. Given a Behavior Graph, B, four partitions π_C, π_D, ρ_{CD} and ρ_{DC} over the set of states of B, and a data structure D', there exists a program $P = \{C,D'\}$ such that $L(P) = L(B)$ and the four partitions are the associated partitions of program P over the BG B if and only if

1. $\pi_C \cdot \rho_{DC} \leq M(\pi_C)$

2. $\pi_D \cdot \rho_{CD} \leq M(\pi_D)$

3. $\pi_C \leq \rho_{CD}$

4. $\pi_D \leq \rho_{DC}$

5. $\pi_C \cdot \pi_D = 0$

6. The number of blocks of π_C is finite.

7. $\pi_C \leq \pi_F$.

8. The machine

$$D = \{S_D, \Sigma_D, O_D, \delta_D, d_O, \lambda_D\}$$

defined by

- $S_D = \pi_D$.
- $\Sigma_D = \rho_{CD} \times \rho_{DC} \times \Sigma$.
- $O_D = \rho_{DC}$.
- $d_O = \beta_{\pi D}(b_O)$.
- $\lambda_D(\beta_{\pi D}) = \beta_{\rho DC}[\beta_{\pi D}]$.
- $\delta_D(\beta_{\pi D}(b), (\beta_{\rho CD}(b), \beta_{\rho DC}(b), \alpha)) = \beta_{\pi D}(\delta(b,\alpha))$.

is isomorphic to a submachine of D'.

The $M(\pi)$ operator is the Hartmanis-Stearns maximum sum operation over the partitions π' such that (π',π) is a partition pair. The proof of the theorem and considerable elaboration appears in Fahmy [88].

The synthesis procedure is derived from the decomposition theorem and follows these steps:

1. Construct as much of the behavior graph as possible from the given examples.

2. Select a proposed data structure for the computation.

3. Construct a partition π_C on the states of B that will define the state transitions for the target controller.

4. Attempt to complete the factorization by building π_D which meets the constraints of Theorem 2.

5. Repeat 4 and 5 until a well-formed factorization is found. The partitions π_C and π_D then define the state assignments and transitions for the controller and data structure.

Fahmy [88] gives a convergence theorem for this learning methodology and a number of examples of its usage. Thus a programmed version of this approach was able to construct controllers of complexity shown in Figure 3 from behaviors as illustrated in Figure 4 in a few seconds.

ACKNOWLEDGEMENTS

This research was supported by ARO grant DAAG-29-84-K-0072.

REFERENCES

A.F.Fahmy (1988), "Synthesis of Real Time Programs," Doctoral dissertation, Duke University, Durham, N.C.

J. Hartmanis and R.E.Stearns (1966), *Algebraic Structure Theory of Sequential Machines, Prentice-Hall.*

A Deductive System for Theories of NonMonotonic Reasoning

Frank M. Brown and Carlos L. Araya
Artificial Intelligence Laboratory
University of Kansas
Lawrence Kansas, 66045
(913)864-4482
ai-lab@csvax.cs.ukans.edu

Abstract
An Automatic Deduction System for NonMonotonic Reasoning in the Modal Quantificational Logic Z is discussed. This system can deduce nonmonotonic consequences specified by several other theories of nonmonotonic reasoning, and in the variable free case provides a decision procedure for each of these theories. These theories include Moore's autoepistemic logic, nonconstructive default logic, Reiter's default logic, McCarthy's Parallel Circumscription with both fixed and variable predicates, and the closed world assumption. The computational properties of these theories are compared.

1. Introduction

A computer program which can prove theorems in several different theories of nonmonotonic reasoning is described. This is done by implementing a deduction system for the modal quantificational logic Z [Brown86a], by showing how every (finite) set of nonlogical axioms and defaults in any of these theories is represented by a sentence of Z which has essentially the same meaning [Brown89a], and by using an automatic deduction system to deduce the appropriate consequences of the representation in Z of such axiom sets of these theories of nonmonotonic reasoning. The proof procedures are briefly introduced in section 2. Discussions of the solution methods and traces of example proofs are given in section 3. The results obtained are given in section 4. The implementation is discussed in section 5 and some conclusions are drawn in section 6.

2. The Proof Procedure

The modal quantificational logic Z consists of the the following symbols: falsity: F, truth: T, and: $\wedge$, or: $\vee$, forall: $\forall$, for some: $\exists$, not: $\neg$, necessary: [], synonymous: $\equiv$, and the following defined symbols:

$(\alpha \rightarrow \beta)$	=df $((\neg\alpha)\vee\beta)$	α implies β
$(\alpha \leftrightarrow \beta)$	=df $((\alpha\rightarrow\beta)\wedge(\beta\rightarrow\alpha))$	α iff β
$(<> \alpha)$	=df $(\neg[](\neg \alpha))$	α is logically possible
$(\alpha \equiv \beta)$	=df $([](\alpha \leftrightarrow \beta))$	α is synonymous to β
$([\beta]\alpha)$	=df $([](\beta \rightarrow \alpha))$	β entails α
$(<\beta>\alpha)$	=df $(<>(\beta \wedge \alpha))$	α is possible with β
(WORLD α)	=df $((<>\alpha)\wedge(\forall\gamma(([\alpha]\gamma)\vee([\alpha](\neg\gamma)))))$ for γ not in α	α is a world
(GEN α)	=df $(\vee i=1,n\ (\exists x1i...xmi(\alpha\equiv(\pi i\ x1i...xmi))))$ where all the predicates are $\pi1...\pi n$	α is a generator
$(\delta1[=]\delta2)$	=df $[](\delta1=\delta2)$	$\delta1$ necessarily equals $\delta2$

The axioms and inference rules of Z include those of first order logic [Mendelson] plus the following inference rules and axioms about necessity []:

R0: from α infer ([] α)
A1: (([]p)→p)
A2: (([p]q) → (([]p) → ([]q)))
A3: (([]p) ∨ ([]¬[]p))
A4: ((∀w((WORLD w) → ([w]p))) → ([]p))
A5: (WORLD(∀p((GEN p)→(p↔[]α)))) for every expression α
A6: (¬((π1 x1...xn)≡(π2 y1...ym)))
where π1 and π2 are different predicates.
A7: (((π x1...xn)≡(π y1...yn))↔(T∧(x1[=]y1)∧...∧(xn[=]yn)))
A8: (¬((∞1 x1...xn)[=](∞2 y1...ym)))
where ∞1 and ∞2 are different functions
A9: (((∞ x1...xn)[=](∞ y1...yn))↔(T∧(x1[=]y1)∧...∧(xn[=]yn)))

The laws R0, A1, A2 and A3 constitute an S5 modal logic which with the nonmodal laws is similar to [Carnap46,Bressan]. A4 says that a proposition is logically necessary if it is entailed by every world proposition. Laws A5, A6, and A7 axiomatize the predicates. A5 is the key axiom which says that any exhaustive conjunction of negated or unnegated distinct generators is a world if there is a sentence α expressible in the formal language of Z which holds when p is an unnegated generator of that conjunction. It extends the A5 axiom used in [Brown86a] to handling quantifiers over arbitrary domains. Laws A8 and A9 axiomatize the functions. The axiom scheme A5 which has a recursively enumerable number of instances, expresses what is logically possible to the extent that it can be so expressed. What is possible with respect to a knowledgebase K is expressed by the defined symbol <K>p which means K and p together is logically possible. For example, the sentence (<>(¬((∃x(P x))→(∀x(P x))))) can be derived from axiom A5 and A6 as follows. Assuming P is one of the predicates and letting α be (p≡(P A)) axiom A5 of Z becomes: (WORLD(∀p(((∃x(p≡(P x)))∨β)→(p↔[](p≡(P A)))))) where β is the rest of the GEN definition. This gives: (WORLD((∀x((P x)↔([]((P x)≡(P A)))))∧(∀p((β→(p↔[](p≡(P A))))))))) which by axiom A6 and since worlds are logically possible implies (<>(∀x((P x)↔[](x[=]A)))) which implies: (<>((P A)∧(¬(P B)))) for a function B and hence (<>(¬((∃x(P x))→(∀x(P x)))))).

Nonmonotonic reasoning is performed in Z by solving equivalences. Generally, some variable such as K represents the meaning of the knowledgebase which is then axiomatized by asserting that K is synonymous to the conjunction of all the axioms of the theory and all the reflective statements such as any defaults. Thus the reflective equivalence: K≡α where K may occur within α, axiomatizes the knowledgebase K. Any equation of the form K≡β where K does not occur in β and which implies K≡α, is a solution to the original equivalence.

3. Examples

Two partial traces of example deductions performed by our automatic deduction system are listed below. The traces are intermixed with a discussion of some of the derived rules of inference of Z which were used. The deductions are carried out by Symeval-like techniques [Brown86b] of replacing expressions by logically equivalent expressions in an inner to outer manner similar to LISP evaluation. Each application shows the name of the rule, the lexical level, the input to the rule: >, the intermediate output of the rule: -, and the output to the rule after evaluation: <. The number before the lexical level in some rules is useful only when more than one outputs are produced and could be

ignored in this example as these rules when successfully applied use the cut! symbol to delete any alternative rules that might also be applicable. The first example is a propositional reflective equation involving two defaults which states that if a is possible then not b and that if b is possible then not a:

```
INPUT:    (≡ k(∧(→(< k > a)(¬ b))(→(< k > b)(¬ a))))
```

The first implication is rewritten into a disjunction as follows:

```
 <k>def       1>. (< k > a)
          1   1<. (<> (∧ k a))
→def          1>. (→ (<> (∧ k a)) (¬ b))
¬<>           2>.. (¬ (<> (∧ k a)))
¬∧demorgan    3>... (¬ (∧ k a))
          5   3<... (∨ (¬ k) (¬ a))
          4   2<.. ([] (∨ (¬ k) (¬ a)))
          3   1<. (∨ ([] (∨ (¬ k) (¬ a))) (¬ b))
```

The second implication is also written as a disjunction and then the rules about ≡ are applied which causes case analysis on the expression ([](∨(¬ k)(¬ a))):

```
≡nm[]0        1>.(≡ k(∧(∨([](∨(¬ k)(¬ a)))(¬ b))(∨([](∨(¬ k)(¬ b)))(¬ a))))
          1   1-..(cut! (let((+exp(∧(∨ $t(¬ b))(∨([](∨(¬ k)(¬ b)))(¬ a))))
                             (-exp(∧(∨ nil(¬ b))(∨([](∨(¬ k)(¬ b)))(¬ a))))
                             (r(¬(∨(¬ a)))))
                        (∨ (if(free? k +exp)(∧(≡ k +exp)(¬(<>(∧ k r))))
                             (∧(≡ k +exp)(¬(<>(∧ +exp r)))))
                          (if(free? k -exp)(∧(≡ k -exp)(<>(∧ k r)))
                             (∧(≡ k -exp)(<>(∧ -exp r)))))))
```

+exp and -exp are the right-hand side of the equation when the default is or is not the case respectively. The other modal expression ([](∨(¬ k)(¬ b))) is now analyzed in each of the first two cases. The +exp branch is analyzed first:

```
≡nm[]0        2>..(≡ k(∨([](∨(¬ k)(¬ b)))(¬ a)))
          2   2-...(cut!(let((+exp(∨ $t(¬ a)))(-exp(∨ nil(¬ a)))(r(¬(∨(¬ b)))))
                    (∨(if(free? k +exp)(∧(≡ k +exp)(¬(<>(∧ k r))))
                         (∧(≡ k +exp)(¬(<>(∧ +exp r)))))
                      (if(free? k -exp)(∧(≡ k -exp)(<>(∧ k r)))
                         (∧(≡ k -exp)(<>(∧ -exp r)))))))
          2   2<..(≡ k(¬ a))
∧≡sub         2>..(∧(≡ k(¬ a))(¬(<>(∧ k a))))
          3   2-...(cut!(∧(≡ k(¬ a))(¬(<>(∧(¬ a) a)))))
          3   2<..(≡ k(¬ a))
```

The -exp branch of is processed similarly contributing another alternative to the final two solutions of the original equation:

```
≡nm[]0        2>..(≡ k(∧(¬ b)(∨([](∨(¬ k)(¬ b)))(¬ a))))
          4   2<..(≡ k(¬ b))
```

```
∧≡sub          2>..(∧(≡ k(¬ b))(<>(∧ k a)))
           5   2<..(≡ k(¬ b))
           1   1<.(∨(≡ k(¬ a))(≡ k(¬ b)))
OUTPUT:    (∨(≡ k(¬ a))(≡ k(¬ b)))
TIME: 17.294 secs., CLOSURES:  61 cls.
```

The case analysis technique used in the above example can be generalized to provide a decision procedure for computing all the fixed points of any variable free nonmonotonic equation in Z as given by Theorem Z5 [Brown89a] below. We define a group of modalized subexpressions to be a propositional combination of expressions each beginning with a modal symbol. For example, ([]A) and (([]A)∧((<>B)∨(<>C))) are both groups whereas ((<>A)→B) is not.

Theorem Z5: Decidability of variable free nonmonotonic equations of Z:
The fixed points of any nonmonotonic equation: K≡(f K) containing no variables (except K) are decidable if every K in (f K) occurs within the scope of a modal symbol. There are at most 2↑n solutions to such an equation where n is the number of groups of modalized subexpressions in (f K).

Many equations with universal and existential quantifiers can also be solved by the equation solver. This is done by splitting the quantified expression into those instances of it which are true in the theory, and those instances of it which are false in the theory. This is done in the following example which states that Tweety (i.e. tw) is a bird and that if it is possible for birds to fly then they do so:

```
INPUT:    (≡ k(∧(bird tw)(∀(x)(→(∧(bird x)(< k >(fly x)))(fly x)))))
∧division     1>.(∧(bird tw)(∀(x)(∨([](∨(¬ k)(¬(fly x))))(¬(bird x))(fly x))))
           1  1-..(cut!(let((y(make-symbol "Y")))
                        (∧(bird tw)
                          (∀ nil(∨([](∨(¬ k)(¬(fly tw))))(¬(bird tw))(fly tw)))
                          (∀(y)(→(¬([=] y tw))
                                 (∨([](∨(¬ k)(¬(fly y))))
                                   (¬(bird y))(fly y)))))))
```

This rule factors out of the quantified default an instance of it dealing with the particular case of Tweety whose being a bird appears in the conjunction.

```
           1  1<.(∧(bird tw)(∨([](∨(¬ k)(¬(fly tw))))(fly tw))
                         (∀(y)(∨([=] y tw)([](∨(¬ k)(¬(fly y))))
                                (¬(bird y))(fly y))))
```

Next, the case analysis rule is applied over the unquantified default:

```
≡nm[]0         1>.(≡ k(∧(bird tw)(∨([](∨(¬ k)(¬(fly tw))))(fly tw))
                       (∀(y)(∨([=] y tw)([](∨(¬ k)(¬(fly y))))
                              (¬(bird y))(fly y)))))
           2   1-..(cut!(let((+exp(∧(bird tw)(∨ $t(fly tw))
                                  (∀(y)(∨([=] y tw)([](∨(¬ k)(¬(fly y))))
                                         (¬(bird y))(fly y)))))
```

```
                    (-exp(∧(bird tw)(∨ nil(fly tw))
                         (∀(y)(∨([=] y tw)([](∨(¬ k)(¬(fly y))))
                              (¬(bird y))(fly y)))))
                    (r(¬(∨(¬(fly tw))))))
               (∨ (if(free? k +exp)(∧(≡ k +exp)(¬(<>(∧ k r))))
                    (∧(≡ k +exp)(¬(<>(∧ +exp r)))))
                  (if(free? k -exp)(∧(≡ k -exp)(<>(∧ k r)))
                    (∧(≡ k -exp)(<>(∧ -exp r)))))))
```

Working over the branch corresponding to +exp, the system tries to eliminate the quantified default:

```
≡k[]elim     2>..(≡ k(∧(bird tw)(∀(y)(∨([=] y tw)([](∨(¬ k)(¬(fly y))))
                                  (¬(bird y))(fly y)))))
          3  2-...(define qexp(∀(y)(∨([=] y tw)(¬(bird y))(fly y))))
          3  2-...(if(eq?(k-elim k
                     (∧(bird tw) qexp
                       (∀(y)(∨@(extract modalized?
                                  (([=] y tw)(¬(bird y))(fly y)))
                                (¬(∨(¬(fly y))))))))) $t)
                (cut!(≡ k(∧(bird tw) qexp)))
                $fail)
```

The elimination rule works in the following way: first it assumes that negation is factored out of ([](∨(¬ K)(¬(fly y)))) giving (¬(<>(∧ K (fly y)))). Then K is replaced by an expression stronger than itself and its possibility is tested. If it is possible, as in the present case, then the conjunct is eliminated, leaving an expression free of defaults, otherwise it fails:

```
          3  2<..(≡ k(∧(bird tw)(∀(y)(∨([=] y tw)(¬(bird y))(fly y)))))
```

Since this is a solution, it is substituted back into the original possibility test:

```
∧≡sub        2>..(∧(≡ k(∧(bird tw)(∀(y)(∨([=] y tw)(¬(bird y))(fly y)))))
                (¬(<>(∧ k(fly tw)))))
```

The new possibility expression is sent to a heuristic procedure that implements the axiom A5 of Z Modal Logic based on theorem ZP1 of [Brown89a] which only succeeds if it can build a world in which the body is the case:

```
<>axiom5     3>...(<>(∧(bird tw)(fly tw)(∀(y)(∨([=] y tw)(¬(bird y))(fly y)))))
          5  3-....(<a5>(∧(bird tw)(fly tw)(∀(y)(∨([=] y tw)(¬(bird y))(fly y)))))
          5  3<... $t
```

Since it is possible, its negation produces $f which in turn prunes the branch corresponding to this case.

```
          4  2<.. nil
```

A similar treatment of the default is successfully applied over the other branch:

```
≡k[]elim      2>..(≡ k(∧(bird tw)(fly tw)
                      (∀(y)(∨([=] y tw)([](∨(¬ k)(¬(fly y))))
                            (¬(bird y))(fly y)))))
              6  2<..(≡ k(∧(bird tw)(fly tw)(∀(y)(∨([=] y tw)(¬(bird y))(fly y)))))
```

Substituting over the possibility test qualifying this case we get:

```
                 2>..(∧(≡ k(∧(bird tw)(fly tw)(∀(y)(∨([=] y tw)(¬(bird y))(fly y)))))
                     (<>(∧ k(fly tw))))
<>axiom5         3>...(<>(∧(bird tw)(fly tw)(∀(y)(∨([=] y tw)(¬(bird y))(fly y)))))
              8  3<... $t
              7  2<..(≡ k(∧(bird tw)(fly tw)(∀(y)(∨([=] y tw)(¬(bird y))(fly y)))))
```

This branch is then returned as the only solution of the original problem:

```
              2  1<.(≡ k(∧(bird tw)(fly tw)(∀(y)(∨([=] y tw)(¬(bird y))(fly y)))))
OUTPUT:       (≡ k(∧(bird tw)(fly tw)(∀(y)(∨([=] y tw)(¬(bird y))(fly y)))))
TIME: 27.117 secs., CLOSURES: 556 cls.
```

4. Results

The deduction system has been used to produce all the fixed points of sets of axioms in [Moore]'s Autoepistemic Logic, our Nonconstructive Default Logic, and [Reiter]'s Default Logic. It also has been used to compute the result of [McCarthy's] parallel Circumscription with both fixed and variable predicates and the Closed World Assumption. Results obtained for each of these systems are described in the following subsections.

4.1 Autoepistemic Logic and NonConstructive Default Logic

The fixed point equation of a theory *g* in Autoepistemic Logic [Moore,Konolige] is:

k=(classical-theorems-of(g∪{(L α):αεk}∪{¬(L α):¬(αεk)}))

If the number of groups of modal statements in *g* is finite then the meaning of the intersection of those fixed points entails the meanings of the same sentences (not containing *L*) as does: (∃K(K∧(K≡(G2∧(∧i=1,n((([K]Ai)∧(∧j=1,mi(<K>Bij)))→Ci))))))
where (G2∧(∧i=1,n((([K]Ai)∧(∧j=1,mi(<K>Bij)))→Ci))) is the meaning of *g* written in a normal form with *L* replaced by [K], and [K] does not occuring in G2, Ai, Bij, nor Ci.

Autoepistemic fixed points with all sentences containing *L* eliminated are equivalent to the fixed points of our Nonconstructive Default Logic. The fixed point equation of our Nonconstructive Default Logic, called NCD, is:

k=(classical-theorems-of(g∪{(ci):((ai)εk)∧(∧j=1,mi(¬((¬bij)εk)))}))

where for each i, the sentences *ai,bij*, and *ci* constitute a default. If the number of defaults n is finite then the meaning of the intersection of the fixed points entails the meanings of the same sentences as does:

(∃K(K∧(K≡(G∧(∧i=1,n((([K]Ai)∧(∧j=1,mi(<K>Bij)))→Ci))))))

where G is the meaning of the sentences in *g*, and Ai,Bij,Ci respectively are the meaning of the sentences *ai,bij,ci* in the defaults. This Z representation can be generalized to the case where quantified variables cross modal scopes with the equation: (∃K(K∧(K≡(G∧(∧i=1,n(∀x1...xmi((([K]Ai)∧(∧j=1,mi(<K>Bij)))→Ci)))))))

Figure 1 gives examples in the Z representation of Autoepistemic Logic and

Nonconstructive Default Logic. An interesting example given at the bottom of this figure is the Gelfond-Przymusinska example which has 2 fixed points in Autoepistemic Logic, as is computed by our deduction system. It is interesting because it only has 1 fixed point in Reiter's default logic as is computed in figure 3, contradicting the claim in [Konolige] that the two systems were identical.

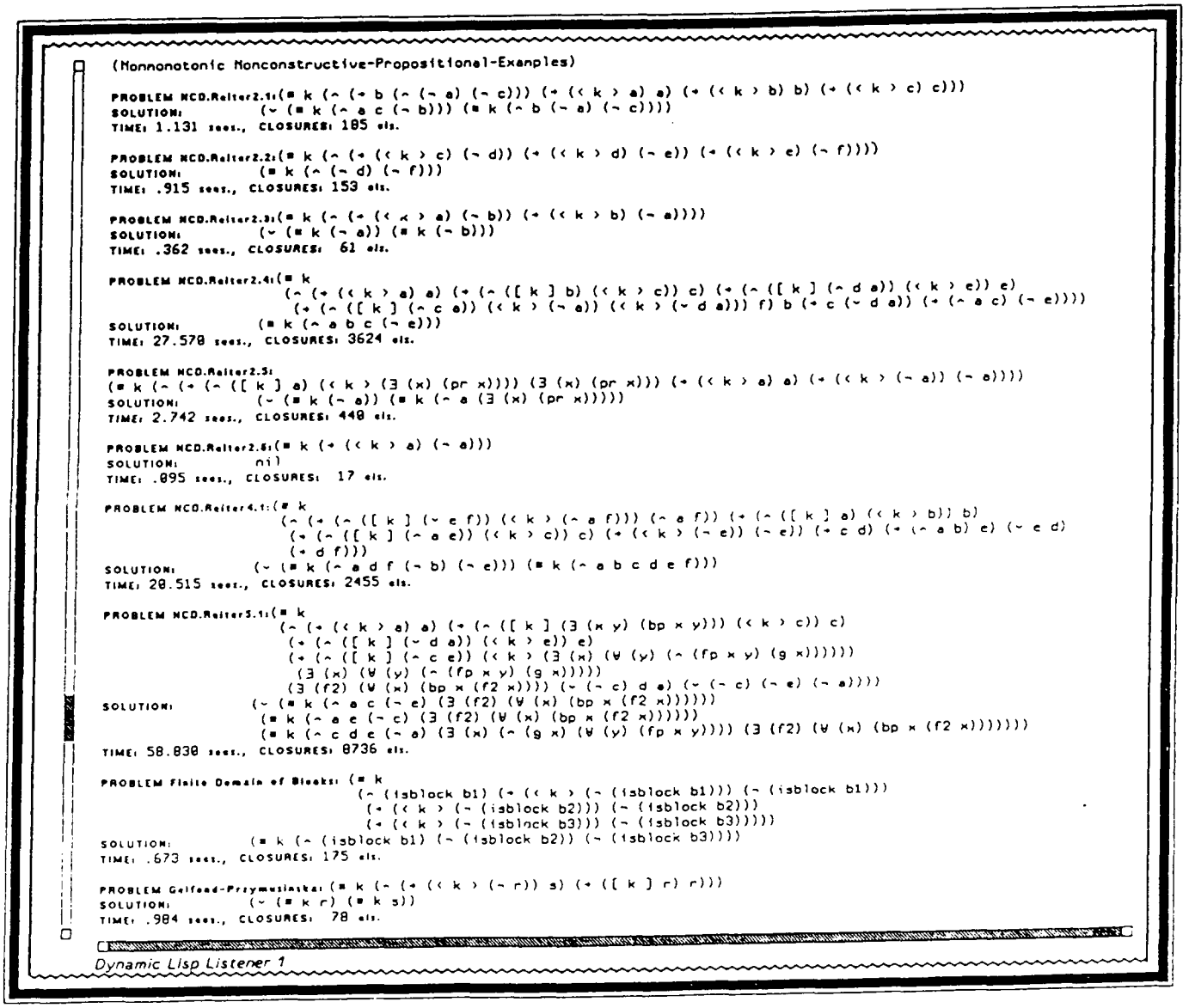


Figure 1

Figure 2 gives examples which are not representable in Autoepistemic Logic or NonConstructive Default logic, but which can be viewed as extensions to these two systems to the case where variables are allowed to cross modal scopes.

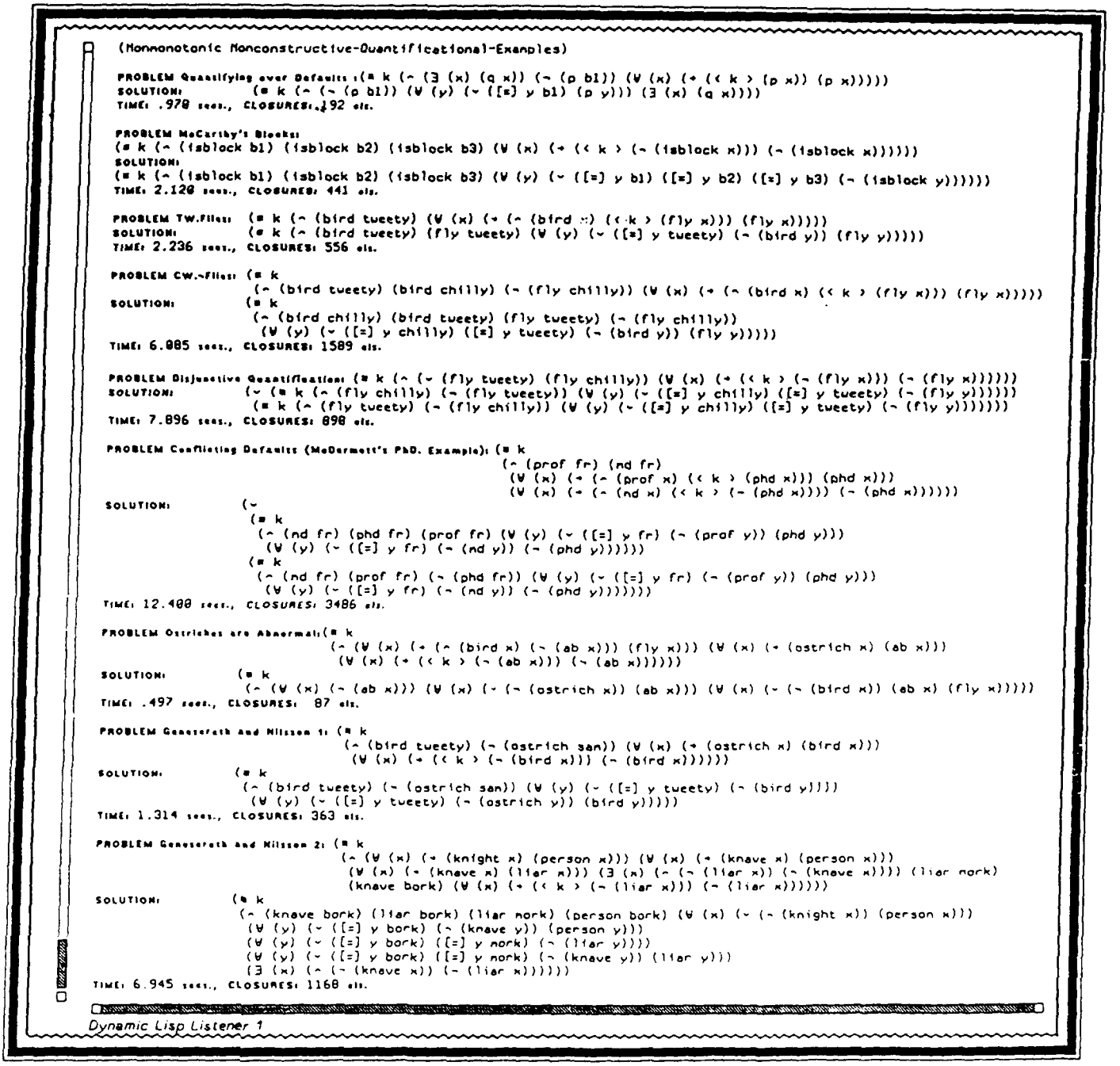


Figure 2

4.2 Constructive Default Logic

The fixed point equation for [Reiter]'s default logic, which we call Constructive Default Logic to distinguish it from other default logics, is defined as:

$$k=(\cap\{p:((\textit{classical-theorems-of } p)\subset p)\wedge(g\subset p)$$
$$\wedge(\wedge i=1,n((((ai)\varepsilon p)\wedge(\wedge j=1,mi(\neg((\neg bij)\varepsilon k))))\rightarrow(ci\varepsilon p)))\})$$

For a finite number of defaults, the meaning of the intersection of the fixed points entails the meanings of the same sentences as does:

$(\exists K(K\wedge(K\equiv(\exists P(P\wedge([P]G)\wedge(\wedge i=1,n((([P]Ai)\wedge(\wedge j=1,mi(<K>Bij)))\rightarrow([P]Ci))))))))$

where G is the meaning of the sentences in *g*, and Ai,Bij,Ci are respectively the meaning of the sentences *ai,bij,ci* in the defaults.

Figure 3 gives examples in the Z representation of [Reiter]'s Default Logic. It will be noted that example CD.Reiter2.4 has only 1 fixed point, contrary to the claim in [Reiter] that it has 3 fixed points in his default logic.

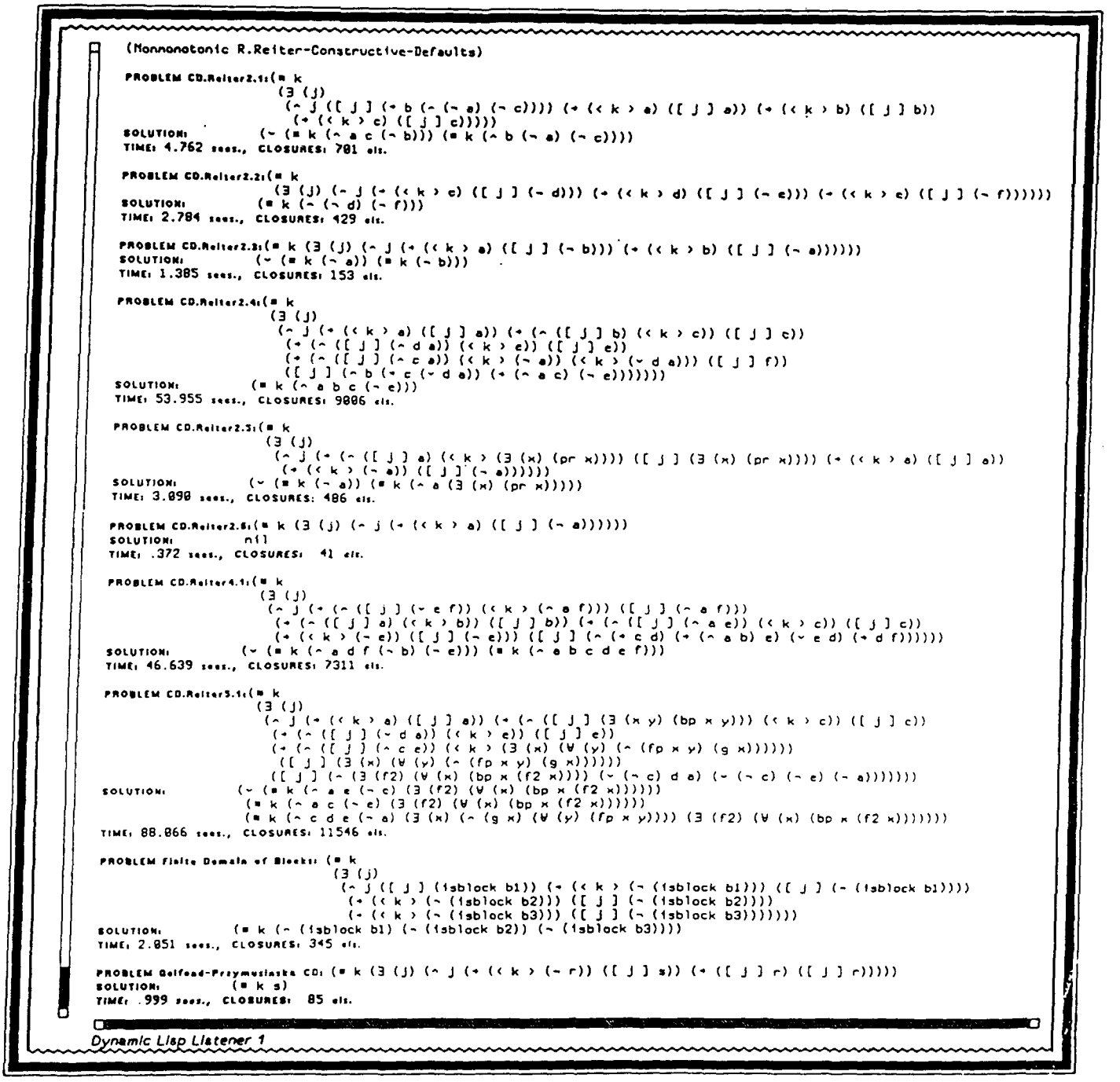


Figure 3

Figure 4 gives examples which are not representable in [Reiter]'s default logic but which can be viewed as an extension of this system to the case where variables are allowed to cross modal scopes. Such equations are of the form:
(∃K(K∧(K≡(∃P(P∧([P]G)∧(∧i=1,n(∀x1...xmi((([P]Ai)∧(∧j=1,mi(<K>Bij)))→([P]Ci)))))))))
The "Quantifying over defaults" example is interesting because it contradicts the suggested solution in [Reiter] for an extension of his logic to allowing quantified variables across modal scopes.

```
(Nonmonotonic Constructive-Quantificational-Defaults)

PROBLEM Quantifying over Defaults :
(= k (3 (j) (^ j ([ j ] (^ (3 (x) (q x)) (~ (p b1)))) (V (x) (-> (< k >) (p x)) ([ j ] (p x)))))))
SOLUTION:          (= k (^ (~ (p b1)) (V (y) (~ ([=] y b1) (p y))) (3 (x) (q x))))
TIME: 1.694 secs., CLOSURES: 225 elt.

PROBLEM McCarthy's Blocks:(= k
                           (3 (j)
                            (^ j ([ j ] (^ (isblock b1) (isblock b2) (isblock b3)))
                             (V (x) (-> (< k >) (~ (isblock x))) ([ j ] (~ (isblock x))))))))
SOLUTION:
(= k (^ (isblock b1) (isblock b2) (isblock b3) (V (y) (~ ([=] y b1) ([=] y b2) ([=] y b3) (~ (isblock y))))))
TIME: 1.989 secs., CLOSURES: 491 elt.

PROBLEM Disjunctive Quantification:
(= k (3 (j) (^ j ([ j ] (~ (fly tweety) (fly chilly))) (V (x) (-> (< k >) (~ (fly x))) ([ j ] (~ (fly x))))))))
SOLUTION:          (~ (= k (^ (fly chilly) (~ (fly tweety)) (V (y) (~ ([=] y chilly) ([=] y tweety) (~ (fly y))))))
                    (= k (~ (fly tweety) (~ (fly chilly)) (V (y) (~ ([=] y chilly) ([=] y tweety) (~ (fly y))))))))
TIME: 5.497 secs., CLOSURES: 907 elt.

PROBLEM Ostriches are Abnormal:(= k
                                (3 (j)
                                 (^ j
                                  ([ j ] (^ (V (x) (-> (^ (bird x) (~ (ab x))) (fly x))) (V (x) (-> (ostrich x) (ab x)))))
                                  (V (x) (-> (< k >) (~ (ab x))) ([ j ] (~ (ab x))))))))
SOLUTION:          (= k
                    (^ (V (x) (~ (ab x))) (V (x) (~ (^ (ostrich x)) (ab x))) (V (x) (~ (~ (bird x)) (ab x) (fly x)))))
TIME: .755 secs., CLOSURES: 116 elt.

PROBLEM Genesereth and Nilsson 1: (= k
                                   (3 (j)
                                    (^ j ([ j ] (^ (bird tweety) (~ (ostrich sam))))
                                     ([ j ] (V (x) (-> (ostrich x) (bird x))))
                                     (V (x) (-> (< k >) (~ (bird x))) ([ j ] (~ (bird x))))))))
SOLUTION:          (= k
                    (^ (bird tweety) (~ (ostrich sam)) (V (y) (~ ([=] y tweety) (~ (bird y))))
                     (V (y) (~ ([=] y tweety) (~ (ostrich y)) (bird y))))))
TIME: 2.358 secs., CLOSURES: 409 elt.

PROBLEM Genesereth and Nilsson 2: (= k
                                   (3 (j)
                                    (^ j
                                     ([ j ]
                                      (^ (V (x) (-> (knight x) (person x))) (V (x) (-> (knave x) (person x)))
                                       (V (x) (-> (knave x) (liar x))) (3 (x) (^ (~ (liar x)) (~ (knave x)))) (liar nork)
                                       (knave bork)))
                                     (V (x) (-> (< k >) (~ (liar x))) ([ j ] (~ (liar x))))))))
SOLUTION:          (= k
                    (^ (knave bork) (liar bork) (liar nork) (person bork) (V (x) (~ (~ (knight x)) (person x)))
                     (V (y) (~ ([=] y bork) (~ (knave y)) (person y)))
                     (V (y) (~ ([=] y bork) ([=] y nork) (~ (liar y))))
                     (V (y) (~ ([=] y bork) ([=] y nork) (~ (knave y)) (liar y)))
                     (3 (x) (^ (~ (knave x)) (~ (liar x)))))))
TIME: 7.049 secs., CLOSURES: 1380 elt.

Dynamic Lisp Listener 1
```

Figure 4

4.3 Parallel Circumscription

The Parallel Circumscription [McCarthy80] of a theory G of first order logic without equality by π with ∂ variable is defined as follows:

$(Circ(G\ \pi\ \partial)\pi\ \partial) =df ((G\ \pi\ \partial)\wedge(\forall p,z((G\ p\ z)\wedge(\forall i,x((pi\ x)\rightarrow(\pi i\ x))))\rightarrow(\forall i,x((\pi i\ x)\rightarrow(pi\ x)))))$
where π and ∂ are finite sequences of predicates, p and z are sequences of predicate variables of the same length as π and ∂ respectively, the arity of corresponding predicates and predicate variables is the same, x following a predicate or predicate variable represents a sequence of variables of that arity, and (G p z) is the replacement of all unmodalized occurrences of π and ∂ in $(G\ \pi\ \partial)$. Parallel Circumscription, with both variable and fixed predicates, is the disjunction of the solutions of the fixed point equation containing the initial axioms G and defaults specifying that every proposition formed from a circumscribed predicate is false by default and every proposition formed from a fixed predicate is both true and false by default [Brown89a].

$(CIRC(G\ \pi\ \partial)\pi\ \partial)\equiv(\exists K(K\wedge(K\equiv((G\ \pi\ \partial)\wedge(\forall i(\forall x((<K>(\neg(\pi i\ x)))\rightarrow(\neg(\pi i\ x)))))$
$\wedge(\forall i(\forall x((<K>(\neg(\ulcorner\pi\partial i\ x)))\rightarrow(\neg(\ulcorner\pi\partial i\ x)))))$
$\wedge(\forall i(\forall x((<K>(\ulcorner\pi\partial i\ x))\rightarrow(\ulcorner\pi\partial i\ x)))))\ ))))$

where $\ulcorner\pi\partial$ is the sequence of all predicates not in π or ∂. Finite parallel circumscription is circumscription where $(\forall x(x[=]\delta 1)\vee\ldots\vee(x[=]\delta n))$ follows from $(G\ \pi\ \partial)$.

Figure 5 gives examples of Circumscription in the Z representation.

```
(Nonmonotonic Circumscription)

PROBLEM Quantifying over Defaults :(∃ (k) (^ k (■ k (^ (∃ (x) (q x)) (¬ (p b1)) (∀ (x) (→ (< k > (p x)) (p x))))))))
SOLUTION:          (^ (¬ (p b1)) (∀ (y) (~ ([=] y b1) (p y))) (∃ (x) (q x)))
TIME: 1.208 secs., CLOSURES: 260 cls.

PROBLEM McCarthy's Blocks:
(∃ (k) (^ k (■ k (^ (isblock b1) (isblock b2) (isblock b3) (∀ (x) (→ (< k > (¬ (isblock x))) (¬ (isblock x)))))))))
SOLUTION:          (^ (isblock b1) (isblock b2) (isblock b3)
                    (∀ (y) (~ ([=] y b1) ([=] y b2) ([=] y b3) (¬ (isblock y)))))
TIME: 2.023 secs., CLOSURES: 587 cls.

PROBLEM Disjunctive Quantification:
(∃ (k) (^ k (■ k (^ (~ (fly tweety) (fly chilly)) (∀ (x) (→ (< k > (¬ (fly x))) (¬ (fly x))))))))
SOLUTION:          (~ (^ (fly tweety) (¬ (fly chilly)) (∀ (y) (~ ([=] y chilly) ([=] y tweety) (¬ (fly y)))))
                    (^ (fly chilly) (¬ (fly tweety)) (∀ (y) (~ ([=] y chilly) ([=] y tweety) (¬ (fly y))))))
TIME: 4.791 secs., CLOSURES: 1055 cls.

PROBLEM Ostriches are Abnormal:(∃ (k)
                                (^ k
                                 (■ k
                                  (^ (∀ (x) (→ (^ (bird x) (¬ (ab x))) (fly x))) (∀ (x) (→ (ostrich x) (ab x)))
                                   (∀ (x) (→ (< k > (¬ (ab x))) (¬ (ab x))))))))))
SOLUTION:          (^ (∀ (x) (→ (^ (bird x) (¬ (ab x))) (fly x))) (∀ (x) (~ (^ (ostrich x)) (ab x) (fly x))))
TIME: .889 secs., CLOSURES: 141 cls.

PROBLEM Genesereth and Nilsson 1: (∃ (k)
                                   (^ k
                                    (■ k
                                     (^ (bird tweety) (¬ (ostrich sam)) (∀ (x) (→ (ostrich x) (bird x)))
                                      (∀ (x) (→ (< k > (¬ (bird x))) (¬ (bird x))))))))))
SOLUTION:          (^ (bird tweety) (¬ (ostrich sam)) (∀ (y) (~ ([=] y tweety) (¬ (bird y)))))
                    (∀ (y) (~ ([=] y tweety) (¬ (ostrich y)) (bird y))))
TIME: 1.772 secs., CLOSURES: 474 cls.

PROBLEM Genesereth and Nilsson 2: (∃ (k)
                                   (^ k
                                    (■ k
                                     (^ (∀ (x) (→ (knight x) (person x))) (∀ (x) (→ (knave x) (person x)))
                                      (∀ (x) (→ (knave x) (liar x))) (∃ (x) (^ (¬ (liar x)) (¬ (knave x)))) (liar nork)
                                      (knave bork) (∀ (x) (→ (< k > (¬ (liar x))) (¬ (liar x))))))))))
SOLUTION:          (^ (knave bork) (liar bork) (liar nork) (person bork) (∀ (x) (~ (¬ (knight x)) (person x)))
                    (∀ (y) (~ ([=] y bork) (¬ (knave y)) (person y)))
                    (∀ (y) (~ ([=] y bork) ([=] y nork) (¬ (liar y))))
                    (∀ (y) (~ ([=] y bork) ([=] y nork) (¬ (knave y)) (liar y))) (∃ (x) (^ (¬ (knave x)) (¬ (liar x))))
                    )
TIME: 8.523 secs., CLOSURES: 1599 cls.

Dynamic Lisp Listener 1
```

Figure 5

Figure 6 gives examples which are not representable in Circumscription but which can be viewed as an extensions to this system to the case where more complex defaults are used, such as in Autoepistemic Logic or in [Reiter]s default logic.

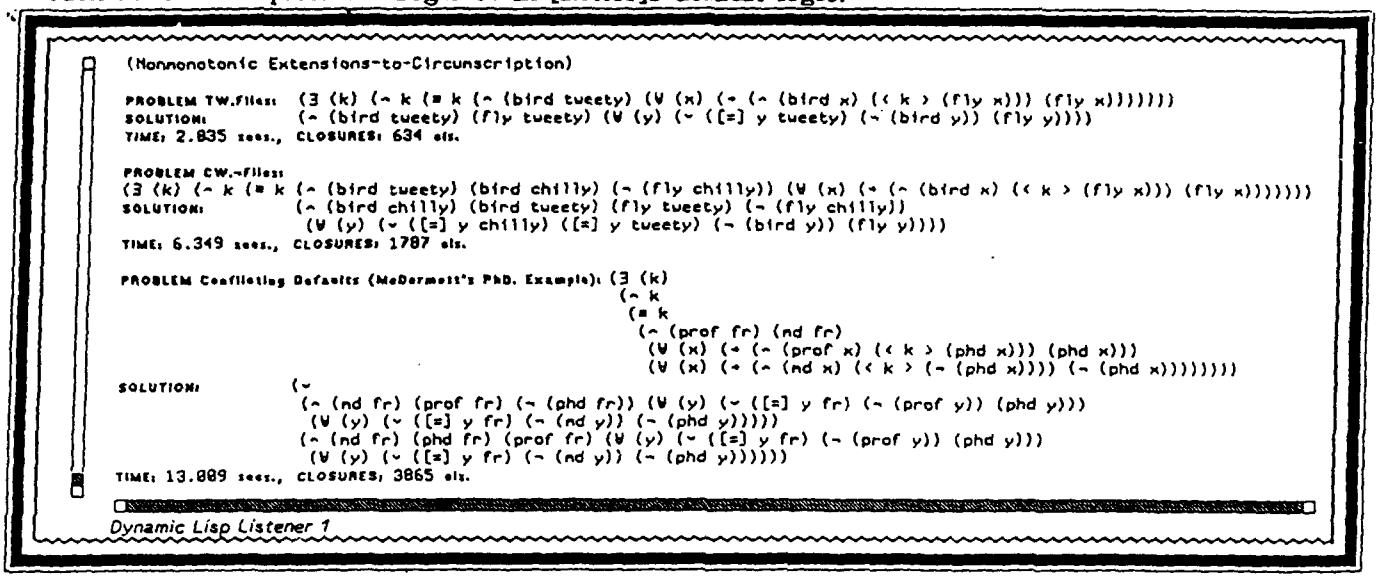


Figure 6

4.4 The Closed World Assumption

The closed world assumption on a theory g is the union of g and the set of negations of a set of simple sentences formed from predicates πi and sequences of variable free terms δ of the appropriate arity which are not deducible from g:

$$(g \cup \{(list\ '\neg(list\ \pi i\ \delta)) : (\neg((list\ \pi i\ \delta) \varepsilon (cl\text{-}th\ g)))\})$$

The closed world assumption on a set of sentences g with respect to π is the meaning of g anded to the defaults which state that if the negation of a sentence formed from π and a sequence of variable free terms $\delta=\delta 1 \ldots \delta ni$ is possible then that negation is the case: $(G \wedge (\forall i(\forall \delta((<G>(\neg(\pi i\ \delta))) \rightarrow (\neg(\pi i\ \delta))))))$ where G is the meaning of g. The Generalized Closed World Assumption is represented by the expression:

$$(CWA^*\ G\ \pi) =df\ G \wedge (\forall i(\forall x((<G>(\neg(\pi i\ x))) \rightarrow (\neg(\pi i\ x)))))$$

where x=x1...xni is a sequence of variables whose length is the arity of πi.

Figure 7 gives examples in the Z representation of the closed world assumption.

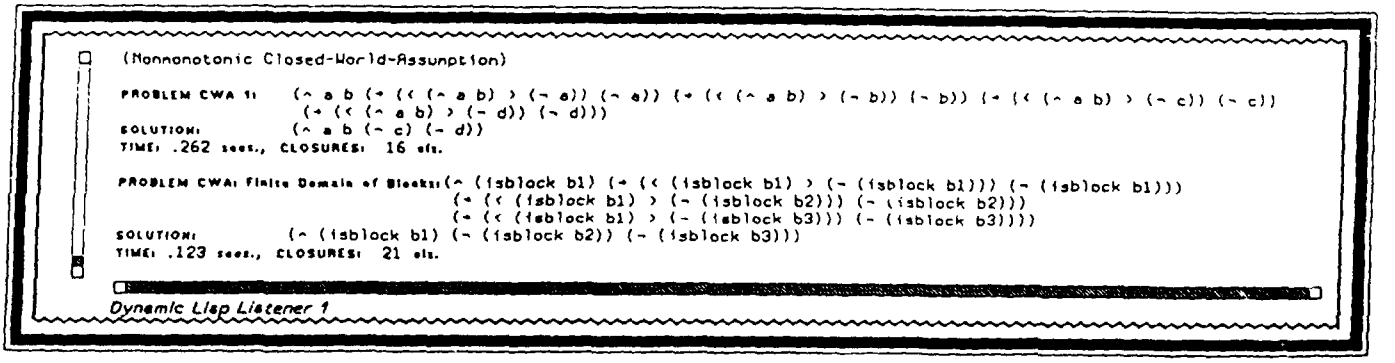


Figure 7

Figure 8 gives examples which are not representable in CWA but which can be viewed as an extensions to this system to the case where variables are allowed to cross modal scopes.

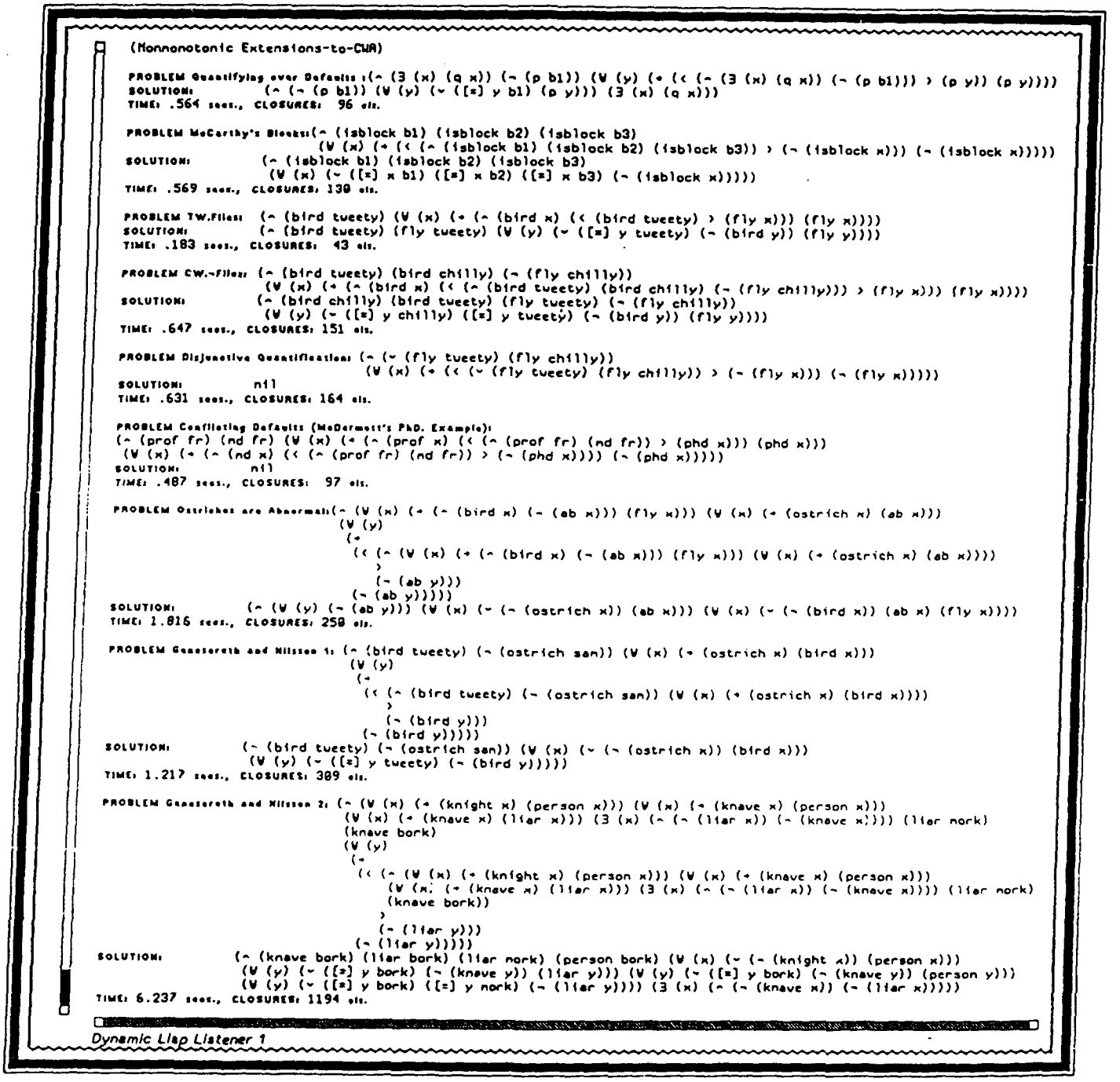


Figure 8

5. Implementation

The derived rules of inference of Z used for nonmonotonic reasoning are implemented

as definitions in the programming language Schemata, which was specifically designed to express arbitrary deductive operations. Listed below are the Schemata definitions which implement derived rules of inference needed to manipulate the disjunction symbol:∨.

```
(define ∨ _∨)
(defaxiom! ∨0        (∨) $f)
(defaxiom  ∨-Sort    (∨ . r)(if(null?(set! r(sort<< r ∨order)))(cut!(∨ . r))(cut!(∨ . r))))
(defaxiom! ∨[]pp     (∨ •••1(_[] p)•••2 p •••3)(∨ •••1 •••2 p •••3))
(defaxiom! ∨∧subsump(∨ •••1(_∧ . r1)•••2(!test(_∧ . r2)(subsume? r1 r2))•••5)
                     (∨ •••1(_∧ . r1)•••2 •••5))
(defaxiom! ∨∧abs1d   (∨ •••1(_∧ •••2 x •••3)•••4 x •••5)(∨ •••1 •••4 x •••5))
(defaxiom! ∨abs¬     (∨ •••1(_¬ x)•••2(!both(π x)(!test e(unmodalized-in? x e)))•••3)
                     (∨ •••1(_¬ x)•••2(π $t)•••3))
(defaxiom! ∨abs      (∨ •••1 x •••2(!both(π x)(!test e(unmodalized-in? x e)))•••3)
                     (∨ •••1 x •••2(π $f)•••3))
(defaxiom! ∨assoc    (∨ •••1(_∨ •••2)•••3)(∨ •••1 •••2 •••3))
(defaxiom! ∨f        (∨ •••1 $f •••2)(∨ •••1 •••2))
(defaxiom! ∨t        (∨ •••1 $t •••2) $t)
(defaxiom! ∨1        (∨ x) x)
```

Defaxiom defines a derived rule of inference which causes the subexpressions which match its second argument to be replaced by the corresponding binding of its third argument. Defaxiom! is like defaxiom except that it cuts alternative applications from the search space. The underscore symbol _ in the second argument indicates that the succeeding symbol is a constant to the matcher and is not a local variable that may be bound by matching. The •••n variables are segment variables which match 0 or more expressions and π is an schemator variable [Morse].

6. Conclusion

Since, in general, each of these 4 nonmonotonic theories compute different things, it might at first appear that their computational properties cannot be compared. However, this is not entirely the case, as there are a number of metatheorems which show that various nonmonotonic theories give identical results when applied to axiom sets of certain forms.

There are two results comparing Autoepistemic Logic and Nonconstructive Default Logic to Reiter's Default Logic. The first result [Brown89a] states that these logics give essentially the same answer if no default in the axiom set has a necessary hypothesis. This result is not a necessary condition as examples NCD.Reiter2.4 in figure 1 and CD.Reiter2.4 in figure 3 show. In fact, all the analogous examples except the Gelfond-Przymusinska example in these two figures have the same result. Since the deduction times for Nonconstructive Default Logic are more than 2 times faster than the analogous examples for Reiter's Default Logic, there is some evidence in favor of the nonconstructive default logic. The difference in speed is caused by the more complex nature of Reiter's fixed point equation which involves an extra quantifier ∃j inside. The second result relating these systems [Konolige] says essentially that the solutions of Reiters Default Logic are a subset of those of nonconstructive default logic. Given the relative deduction speeds obtained, the possibility arises of computing the fixed

points of Reiter's Default logic, by solving for the Z solutions of Nonconstructive Default logic, and then plugging each such solution back into the Z equation for Reiter's default logic and eliminating those which do not simplify to true.

One fairly general result [Brown89a] relating Parallel Circumscription to the quantificationally generalized closed world assumption states that if the generalized closed world assumption of any set of predicates is logically possible then it is the same as the circumscription of that set with those predicates regardless of which other predicates are fixed or variable. Since the generalized closed world assumption of any set of predicates of a set of Horn clauses is logically possible (i.e. because the intersection of all the models of such theories is itself a model) this is a fairly useful relationship. The disjunctive quantification example (figures 5 and 8) show that it cannot be extended to sets of non-Horn clauses. For Horn clause theories, deducing the generalized closed world assumption always took less time than deducing the corresponding circumscription. Thus this provides some evidence in favor of the former theory.

Since all these theories have been represented in the modal quantificational logic Z and since it can represent extensions to all these theories (see figures 2, 4, 6, 8), and since all these problems have been solved with a deduction system consisting of derived rules of inference of Z, it may be concluded that Z, even though it is a monotonic logic, is an interesting theory of nonmonotonic reasoning.

Acknowledgments
We thank Sal Hundal, Dr. Seung Park and Dr. Peter Woodruff for their help on the definition of the system.

References

Bressan,Aldo,A GENERAL INTERPRETED MODAL CALCULAS, Yale University Press, 1972.
Brown86a,F.M., "A Comparison of the Commonsense and Fixed Point Theories of Nonmonotonicity", PROCEEDINGS AAAI-86, Morgan Kaufmann, 1986.
Brown86b,F.M., "An Experimental Logic based on the Fundamental Deduction Principal", ARTIFICIAL INTELLIGENCE vol. 30 no. 2, North Holland, 1986.
Brown89a "The Modal Quantificational Logic Z, A Monotonic Theory of Nonmootonic Reasoning", University of Kansas technical report 89-1, 83 pages June, 1989.
Carnap, Rudolf, "Modalities and Quantification" JOURNAL OF SYMBOLIC LOGIC, volume 11, number 2, 1946.
Konolige,Kurt, "On the Relation Between Default Theories and Autoepistemic Logic", IJCAI87, Morgan Kaufmann, 1987.
McCarthy,J "Circumscription -- A Form of Nonmonotonic Reasoning" ARTIFICIAL INTELLIGENCE, volume 13, North Holland, 1980.
McDermott,Drew, "Nonmonotonic Logic II: Nonmonotonic Modal Theories" , JACM 29, 1982.
Mendelson,E. INTRODUCTION TO MATHEMATICAL LOGIC, Van Norstrand, 1964.
Moore,R.C. "Semantical Considerations on Nonmonotonic Logic" ARTIFICIAL INTELLIGENCE 25, North Holland, 1985.
Morse, A.P. A THEORY OF SETS, Academic Press, 1965.
Reiter,R., "A Logic for Default Reasoning" ARTIFICIAL INTELLIGENCE, 13, 1980.

A Logic Programming Approach to Network Flow Algorithms.

Andrew W. Harrell

U.S. Army Engineer Waterways Experiment Station,

Vicksburg, Mississippi

Abstract

The well-known Ford-Fulkerson algorithm and most of the more recent approaches to solving the network maximal flow and minimal-cost flow problems use a labeling procedure. Labeling involves using nodes in the network which have values and are updated by adding a series of augmenting flows or edges until the optimal solution is reached. In this paper, an alternative approach is examined using the full ordered list of flow paths without cycles. This list is generated by a Prolog-based depth-first search with backtracking of the type described by Winston [13],[14]. This approach keeps track of the full queue of partial search paths and is easier to use to examine the solution for weak-links or critical nodes. If a modeler is creating a network to represent a real situation it is reasonable to assume that the number of ingoing and outgoing edges to a vertex are limited. Time bounds are presented to demonstrate that the above approach is under these conditions as efficient as the n-cubed algorithms explained in Tarjan [12] which use a method of labeled preflows.

Key words - network algorithm, depth-first search, maximal flow, min-cost flow, logic programming, backtracking.

1. Introduction.

With the development of computer graphics techniques for displaying digital map information, new ways of representing unit movement and aircraft or ship routing have been developed. However, the use of digital map data presents problems as for example representing the effects of various types of on- and off-road obstacles, underwater mines, bridge interdiction on the movement rates, and routing possibilities. Programs must be written to define and store route movement networks and arrays of obstacles.

This research was supported in part by Headquarters, US Army Corps of Engineers, Washington, DC 20314-1000.
Author's Address: Dr. Andrew Harrell, US Army Engineer Waterways Experiment Station, 3909 Halls Ferry Road, Vicksburg, MS 39181-0631 Internet: h4enaah0@vicksbrg.army.mil

For this paper, we will assume this already exists along with avenues of approach or movement corridors and their corresponding traverse speeds across the map. Harrell [7],[8],[9] gives a partial description of some current techniques of doing this. The following short glossary defines some of the basic terms that will be used:

Backtracking - An algorithmic search scheme which in order to compute all the ways to satisfy a given goal computes one solution through following a series of branching points and then retraces its steps to the last previous decision in order to compute another possible solution.

Cycle - A path with the same starting and ending node.

Dead end - A node in a network from which no edges proceed.

Edge - The line connecting two nodes in a network. Each edge in a network usually has associated with it a traverse time, vehicular speed, or a flow-rate, and represents a given portion of the overall map.

Flow- An assignment of of flow-rates to some or all of the edges in a given network. Each flow-rate has to be less than or equal to the flow-capacity of its edge.

Flow-capacity- The largest allowable flow-rate for a particular edge.

Flow-rate - The number of vehicles per hour that can pass over a given edge in the network. As explained in the text this can be calculated as [1/(time it takes a group of vehicles to traverse the edge)]*number of vehicles in the group.

Maximal flow problem - The problem of determining what is the greatest number of vehicles/hour that can travel through a network at a given time. It is computed by designing an algorithm to optimize the assignments of flows to edges in the network.

Maximal flow value - The value which is a solution to a Maximal flow problem. Note, that it is possible for there to be several different network flows which realize a given maximal flow value.

Min-cost flow network- A flow network with costs (times to traverse) as well as flows associated with its edges. In this paper in order to determine the cost associated with a flow the following procedure is followed: 1) The flow rate on each flow path solution through the network is multiplied by its time of traversal and the result summed over all paths in order to obtain a total cost associated with a given maximum flow solution. This is the measure of effectiveness which determines the optimality of the solution. The total cost of the flow can then be divided by the total maximum

flow to obtain an average cost per vehicle to travel through the network.

Min-cost flow problem- The problem of determining from all the possible flows which realize a given network maximal flow value, those which do it with minimal cost.

Network - A collection of nodes and edges that represent movement possibilities over a given terrain area.

Node - A point of reference in a network from which edges are drawn from and into.

Path - An ordered list of edges each of which has the same starting node as the preceeding edge's ending node.

An artificial intelligence network algorithm is methodology based on searches for paths from start nodes through a network to ending goal nodes using the methods of logic programming. The search mechanism proceeds in an orderly fashion unifying the variables in the search predicates from one level of search in the network to another. The algorithm used must save the partial solutions *in an environment* list so it can backtrack its way through the previous variable bindings in order to generate all possible ways of reaching the goal state. This differs from many network algorithms that use labels (instead of a list of partial search paths) at the nodes to store information as the steps in the algorithm proceed. Thus, after the labeling algorithms are through generating solutions, information is not kept on "how" the solutions were reached.

The search algorithm discussed in Section 2 below will print out ordered lists of shortest paths with and without the presence of obstacles. These lists reveal the critical nodes or weak links that most affect the optimal paths in the network. In order to do this and compute movement possibilities across cross-corridors an algorithm has to keep track of more information than can be stored on just a single label per node in the network or on a single search tree. One needs to store the same kind of list of partial solutions that a logic programming unification algorithm does when it tries to satisfy goal predicates.

Similarly, in developing programs to compute network flow rates which identify the critical nodes in the solution, it is important to compute the maximal or min-cost flows in terms of an ordered list of paths from the start node (or set of nodes) to the goal node (or set of nodes). The solution can be displayed just as a logic programming interpreter displays in turn the list of predicate variable identifications which satisfy the specified goal. The question then becomes whether this approach is feasible in terms of search time bounds and how it is implemented.

These questions are answered in this paper which contains five sections. In Section 2 the main search algorithm used to compute shortest paths or maximal flow paths and give the

derivation of the number of search steps required to generate all these solutions is presented. In Section 3 an explanation of how this algorithm can be used to solve the maximum flow problems associated with certain types of networks is discussed. In section 4 simple modifications to this algorithm are presented that can be used to solve the corresponding min-cost flow problem. Section 5 contains a short discussion of the appropriateness of these algorithms for the transportation problem and the assignment problem, and the next and final section contains the conclusions.

ACKNOWLEDGEMENT

The tests described and the resulting data presented herein, unless otherwise noted were obtained from research conducted under the MILITARY RESEARCH AND DEVELOPMENT TEST AND EVALUATION PROGRAM of the United States Army Corps of Engineers by the US Army Waterways Experiment Station. Permission was granted by the Chief of Engineers to publish this information. Unlimited Distribution/Public Release.

The Main Search Algorithm

As mentioned above, outputting the full search path allows the user to determine the effect of weak links or choke points on the solution. For example, in on- or off-road movement networks based upons digitized maps, it is important to know the effects of minefields, anti-tank ditches, abatis, and road craters on the overall possible vehicular flow rate vehicles traverse across the terrain. Network path-generating algorithms based upon dynamic programming, dynamic tree structures, or node labeling do not save the information on the movement possibilities through cross-corridors in the terrain. This increases computational speed in many cases, but important information about the vulnerability or sensitivity of the solution to degrading factors is lost. The best algorithm for these purposes is one that provides a way to measure the effect of changing flow rates and times in certain parts of the network on the overall solution.

An example of such an algorithm is given below. The search procedure presented keeps track of the next best choices in a sorted priority queue. This is necessary so the algorithm can backtrack quickly to find another solution after it has determined the shortest path or failed to reach a goal in a given direction. In order to do this, it was convenient to write the program in Prolog. An algorithm that does this is described in the book by Winston [13] . The description of the algorithm is as follows :

> Step 1 Form a queue of partial paths. Let the initial queue consist of the zero-length, zero-step path from the start node to nowhere.

Step 2 Until the queue is empty or the goal has been reached determine if the first path in the queue reaches the goal node.

Step 2a If the first path reaches the goal node, do nothing.

Step 2b If the first path does not reach the goal node:

Step 2b1 Remove the first path from the queue

Step 2b2 Form new paths from the removed path by extending them one step

Step 2b3 Add the new paths to the queue

Step 2b4 Sort the queue by cost accumulated so far, with least cost paths placed in front.

Step 3 If the goal node has been found, announce success; otherwise, announce failure.

The algorithm as given terminates when the shortest incomplete path is longer than the shortest complete path. In this situation there are no further paths needing to be investigated for optimality. Since the paths which could never be optimal have been pruned out at an earlier stage, the queue remaining (which has been sorted at each stage) contains at its head the optimal path.

The Prolog source code and Pascal source code for one particular implementation of the algorithm is given in Harrell's report [9] and it can be implemented in the C language using essentially the same code. There is a way to implement the algorithm using a dynamic tree structure to keep the environment of partial solutions which it is able to backtrack through (see the book by Bratke). However, as mentioned above, a tree can store information about only one partial path from its root to each leaf or subtree node.

The question then becomes whether the list of all partial paths accumulated using the search algorithm becomes so large that it is impractical to manipulate. The theorems and the lemmas listed below prove that under certain restrictions, such as: 1) no dead ends in the network, 2) the maximum numbers of nodes going in and out of a vertex bounded above, and 3) the maximum number of nodes which are critical in the sense defined below is bounded, the time it takes to finish this type of algorithm is not longer than for the algorithms which compute shortest paths to create maximal and min-cost flows according to the approaches of Edmonds and Karp [4].

Let:

ni = the number of nodes with i edges proceeding from them,

nji = the number of nodes with j edges entering them and i edges proceeding from them,

maxe = the maximum number of edges proceeding from any node in the network,

emax = the maximum number of edges entering any node in the network.

Call a node a critical backtracking node if it has more than 1 edge proceeding from it and more than one edge entering it.

Let ncrit = the number of critical bactracking nodes in a network.

Call a node of the network a q stage 1th critical path backtracking node if it is a critical path backtracking node and it is preceeded in the network by q levels of backtracking nodes, each having more than 1 edge entering them. Moreover, there must be 1 of these backtracking nodes with more than 1 edge at the preceeding search level to the given node.

Let $n_{q1}ji$ = the number of q stage ,1th critical path backtracking nodes with j edges proceeding into them and i edges leaving them.

n1crit = the number of critical path backtracking nodes which are not q level 1th critical for q or 1 > 1.

Examples of these definitions will be given in the course of the following discussion.

Theorem 1 Given a connected directed graph with a starting node and a goal node and no dead ends other than the goal node. Moreover, if there are at most ncrit critical path backtracking nodes with at most a q level instance of prior influence, then the number of different paths (containing no cycles) from the starting node to the ending node is bounded by the expression

$$1 + (emax - 1)*(n - n1 - ncrit) + (maxe*emax - 1)*n1crit + ((maxe*emax)^q - 1)*(ncrit - n1crit)$$

Proof: This theorem is proved by following through the steps of the above algorithm and counting the number of ways new paths are generated. Step 2b4 which insures the solutions will be generated in order of shortest length is not necessary if the algorithm is only being used to generate all possible paths. At Step 2b, new paths are added to the queue of partial paths each time the search predicate finds a node following the current node which does not form a cyclic path. Since 1) the graph is finite, 2) there are no dead ends, 3) the graph is connected, 4)no cycles are permitted, then each new path will eventually reach the goal

node.

During the generation of the list of partial paths, nodes with only one edge proceeding into them and one edge leaving them expand the current path but do not add any additional combinatorial search possibilities to keep track of during the backtracking process.

Then, the number of paths which the non-critical backtracking nodes enter into is:

$$1 + 1*n2 + 2*n3 + 3*n4 + (j-1)*nj + \ldots (emax - 1)*nemax \quad (1)$$

Using the fact that $n = n1 + n2 + \ldots nemax$ we note that the above number is bounded by:

$$(emax - 1)*(n - n1) + 1$$

The example below (Figure 1) illustrates how equation (1) counts paths in a network without any critical backtracking nodes.

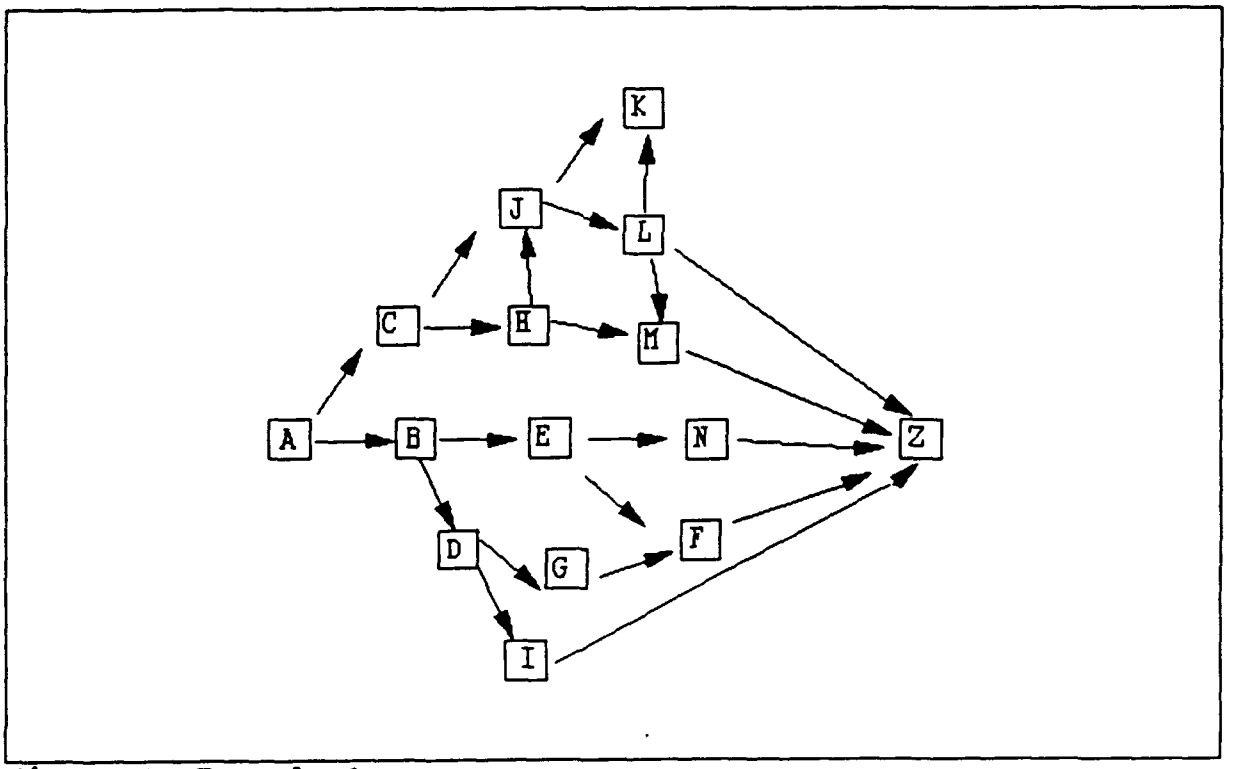


Figure 1. Example 1

There are 14 nodes,
not counting Z.

n1 = 6 , n2 = 7

n3 = 1

emax = 3

number of paths = (7 + 1)*1 + 2*(1) = 10

path	search steps used to generate path[1]
ABDIZ	4
ABDGZ	1
ABDGFZ	1
ABEFZ	2
ABENZ	1
ACHMZ	3
ACJLMZ	3
ACJLZ	0
ACJLKZ	1
ACJKZ	1

If the network we are considering has q stage 1th critical path bactracking nodes but none with q or 1 greater than 1 then the search algorithm will generate an additional:

$3*n_{11}22 + 5*n_{11}23 + \ldots 5*n_{11}32 + \ldots (j*i - 1)*n_{11}ji$
$+\ldots$ (maxe*emax - 1)*n1maxeemax paths.

This number is bounded by:
(maxe*emax - 1)*(n1crit).

Example 2 - consider the following movement network, having two starting nodes A1 and A2 and three goal nodes E1, E2, and E3:

Solution:

---

[1] A search step is defined to be one cycle of search through the database of edges to determine which nodes are connected to a given edge. It is assumed the network information is stored in a vector structure in which each edge along with its starting and ending node and value are kept. Since in generating the queue of search paths a new path uses the nodes from the prior search paths, it is not necessary to search through the database for all the prior nodes in creating the new paths.

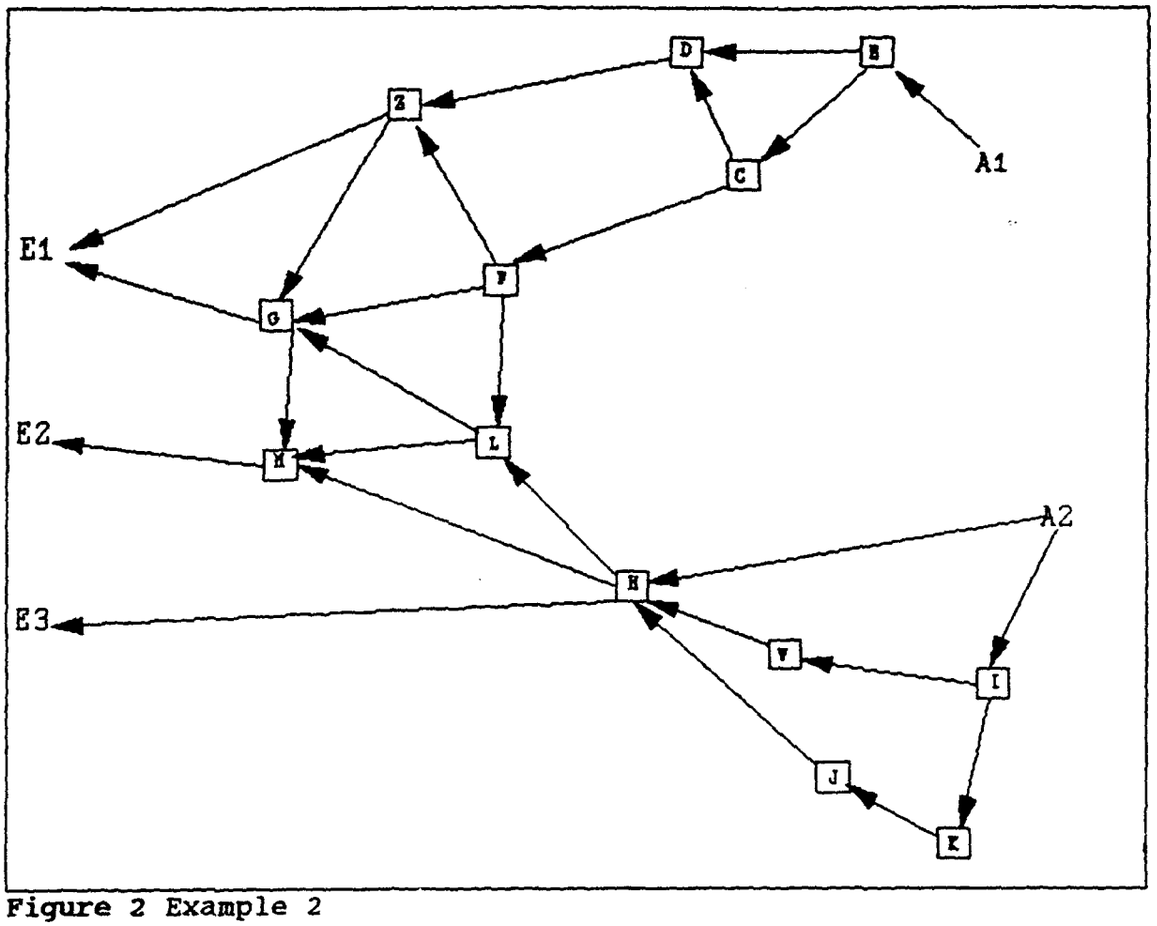


Figure 2 Example 2

To compute all the paths for this network break it up into six connected components corresponding to each possible combination of starting node and ending node. Figures 3 through 7 show this.

1) A1 - E1

ABDZE
ABCFGE
ABCDZE
ABCFLGE
ABDZGE
ABCFZE
ABCDZGE
ABCFZGE

8 solutions Z is a 1-stage 1th critical path backtracking node

$n2 = 2 \quad n3 = 1 \quad n_{11}22 = 1$

$8 = 1 + n2*1 + n3*2 + n_{11}22*(2*2 - 1)$

$= 1 + 2 + 2 + 3$

2) A1 - E2

ABCFGME
ABCFLME
ABCFLGME
ABDZGME
ABCDZGME
ABCFZGME

6 solutions n3 = 1 n2 = 3

6 = 1 + 1*n2 + 2*n3 = 1 + 3 + 2

3) A1 - E3 no solutions

4) A2 - E1

AHLGE
AIWHLGE
AIKJHLGE

3 solutions n2 = 2

3 = 1 + 1*n2 = 1 + 2

5) A2 - E2

AHME
AIWHME
AIKJHME
AHLME
AHLGME
AIWHLME
AIWHLGME
AIKJHLME
AIKJHLGME

9 solutions n2 = 3 $n_{11}^{32} = 1$

9 = 1 + n2*1 + n_{11}^{32}*(3*2 - 1)
= 1 + 3 + 5

6) A2 - E3

AHE
AIWHE
AIKJHE

3 solutions n2 = 2

3 = 1 + n2*1 = 1 + 2

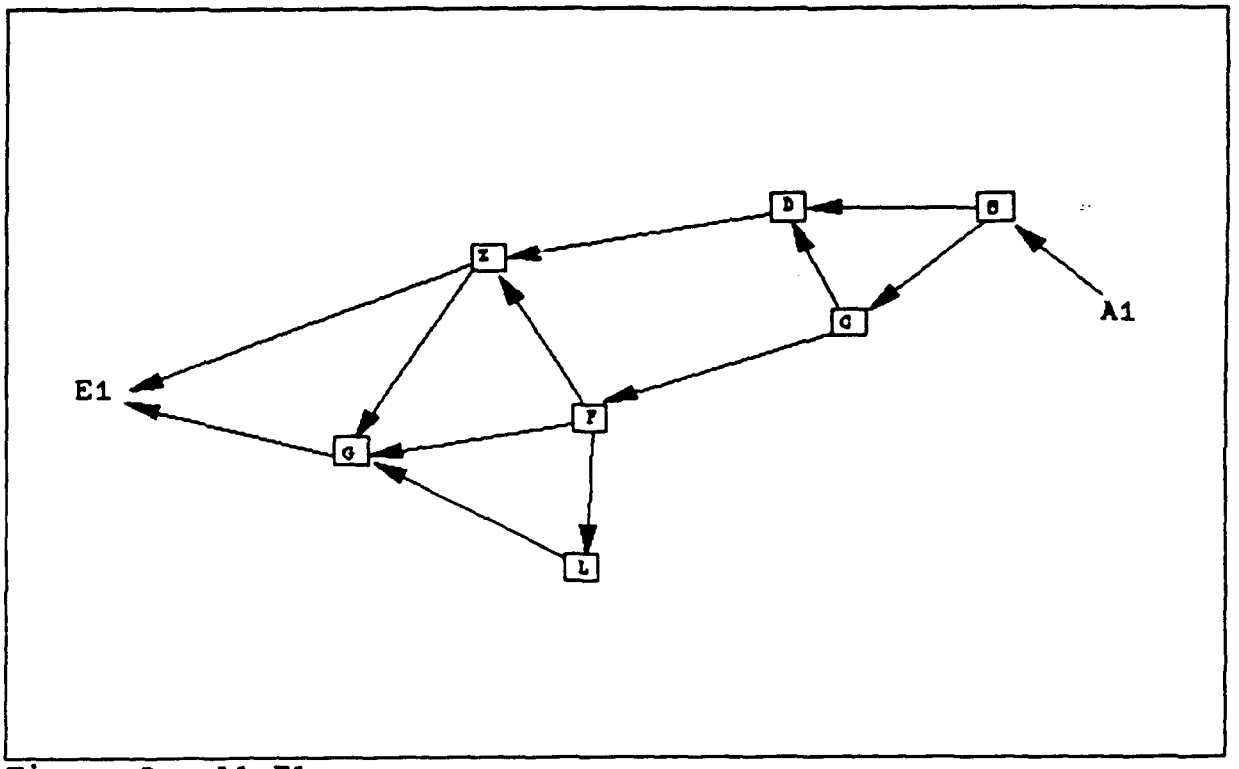


Figure 3 A1-E1

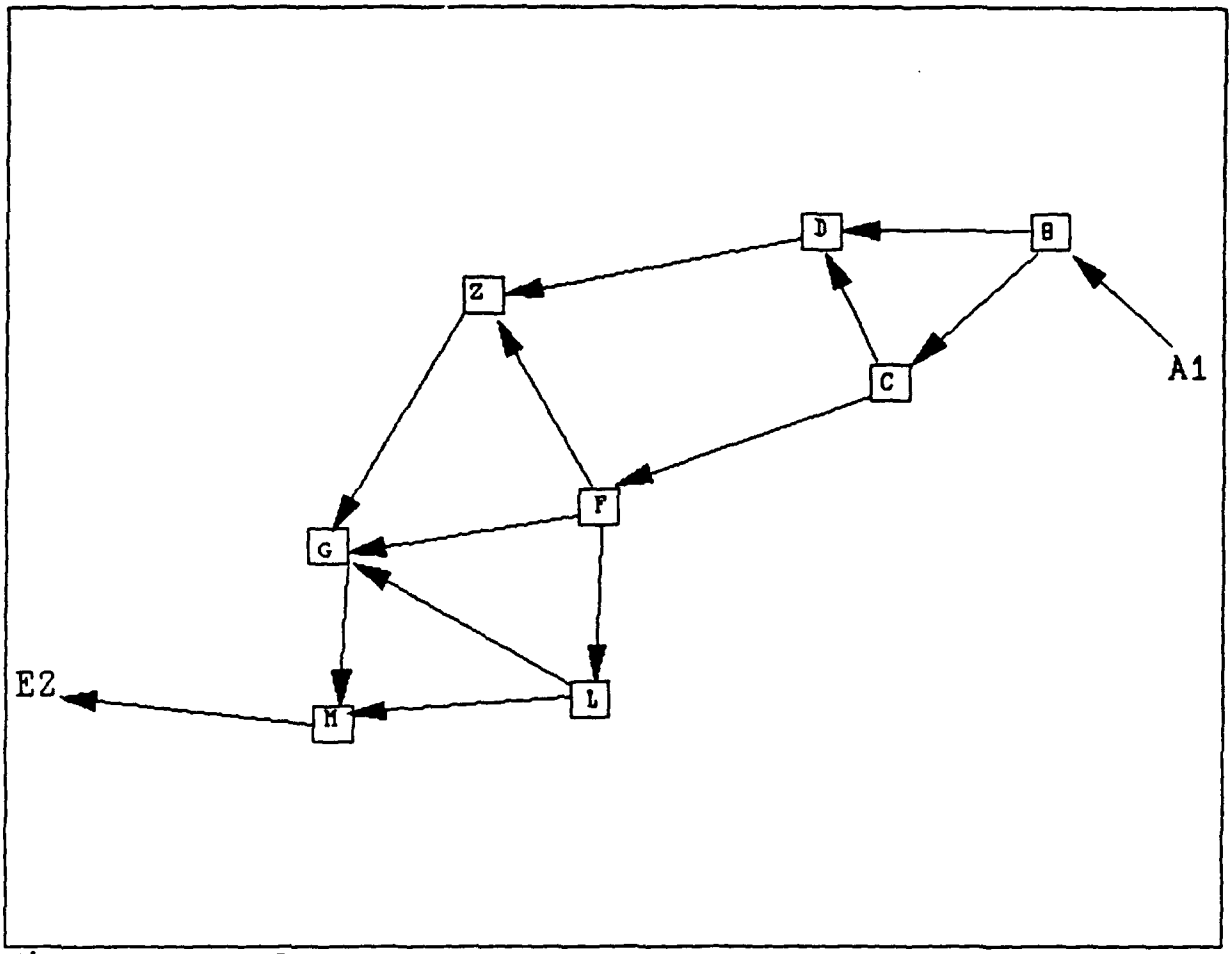


Figure 4 A1 - E2

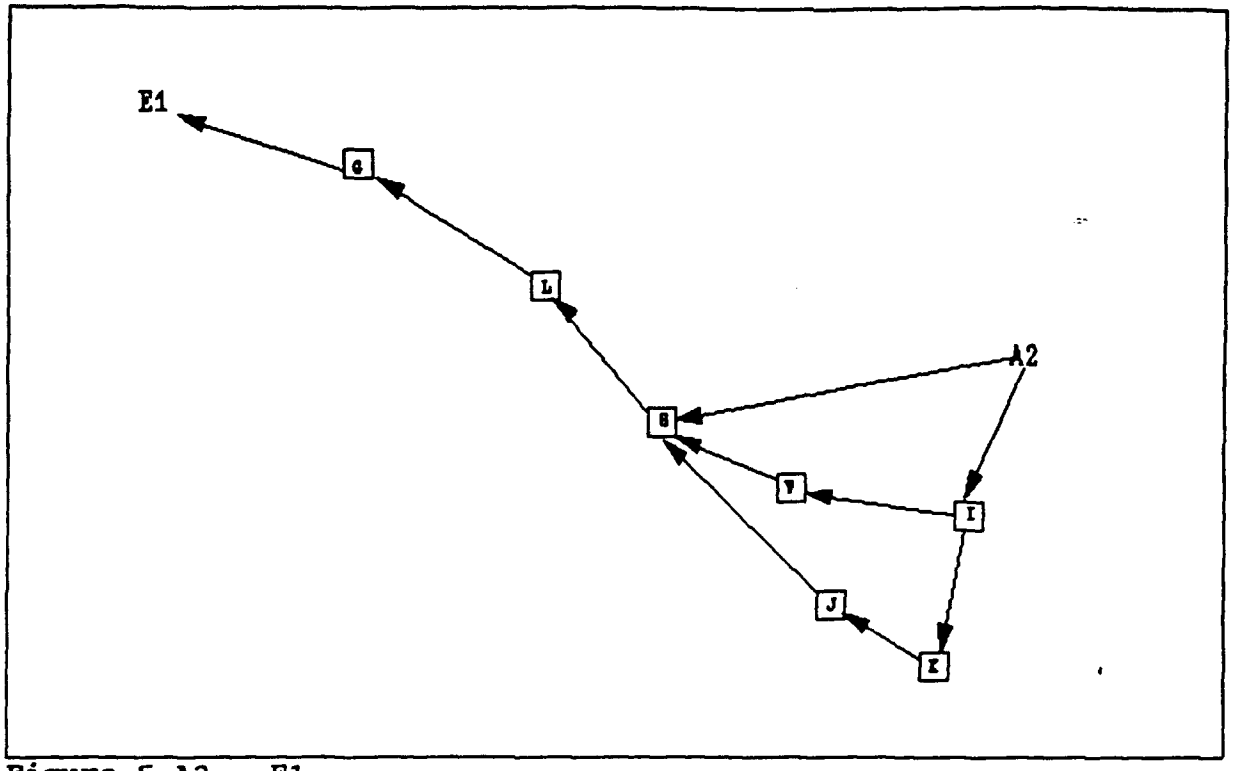


Figure 5 A2 - E1

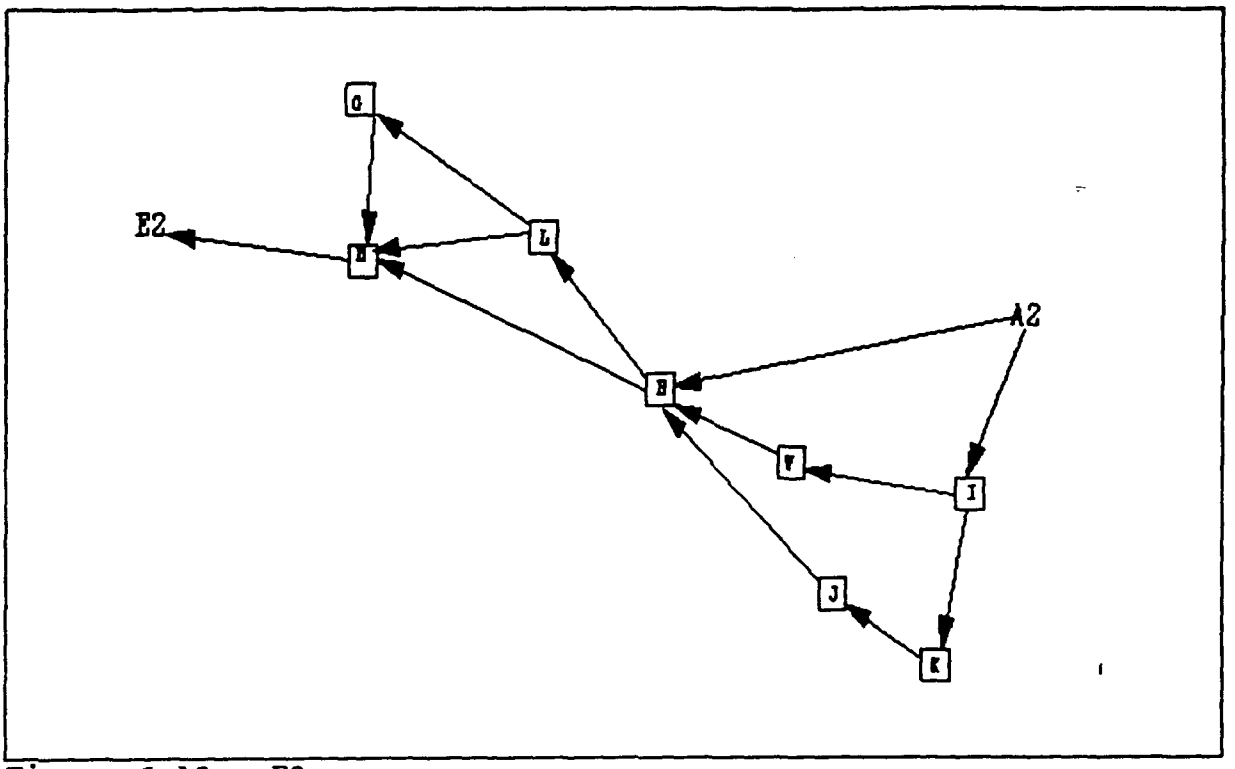


Figure 6 A2 - E2

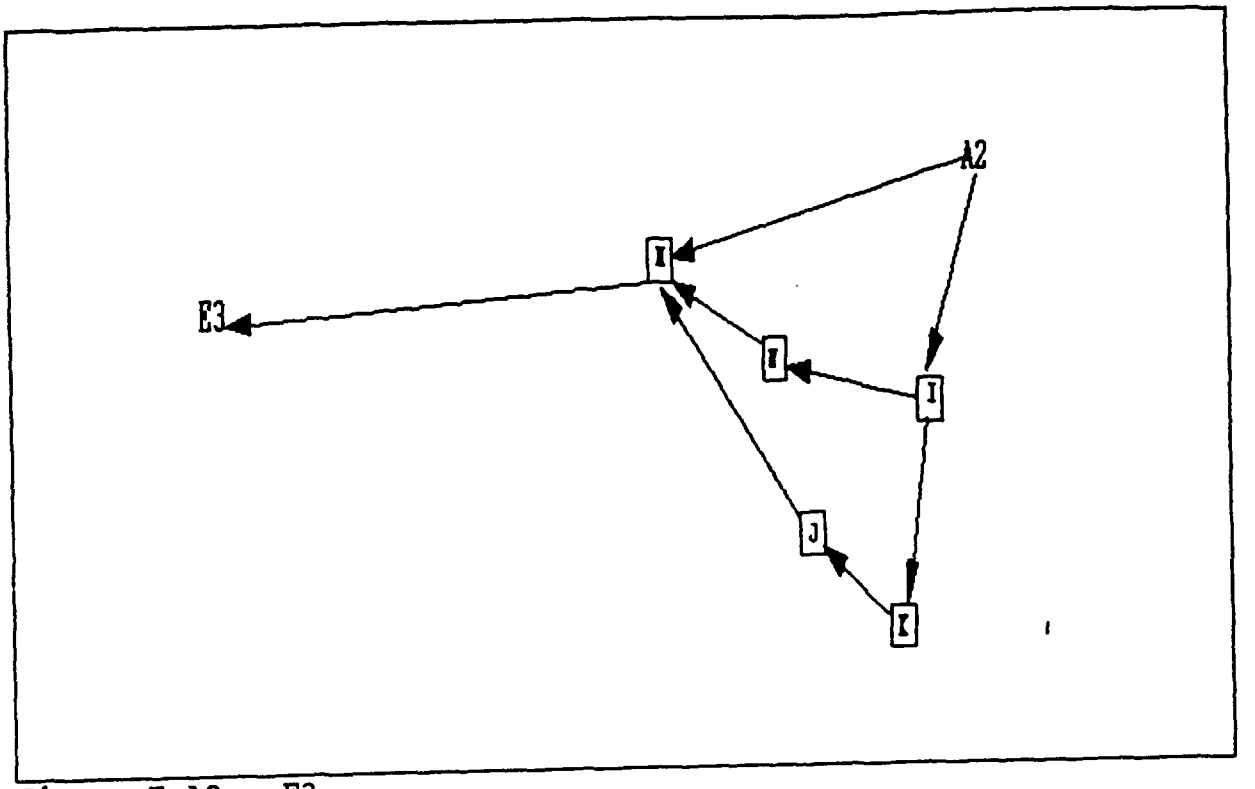


Figure 7 A2 - E3

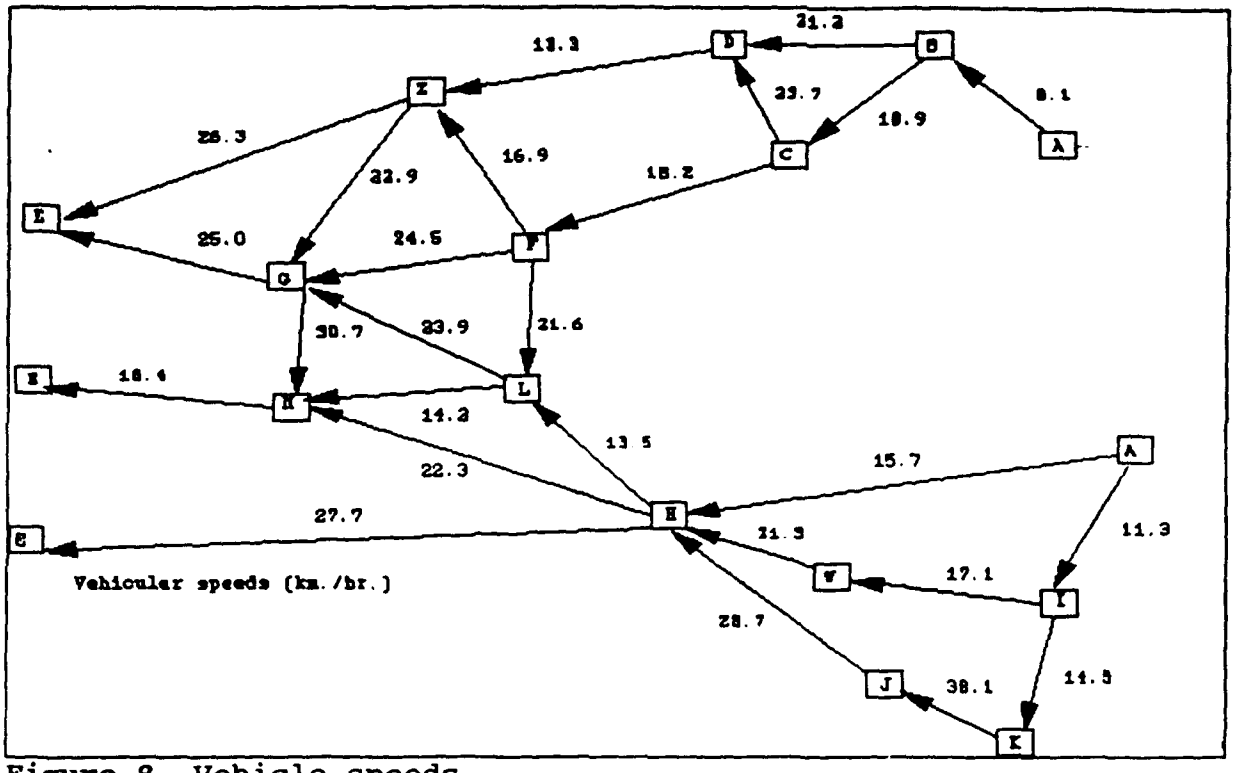


Figure 8. Vehicle speeds

If vehicular speeds are added to the edges in the network of example 2 as shown in Figure 8, the shortest path search algorithm may be used to produce the full list of non-cyclic paths ordered by their length in time (minutes to traverse). Figure 9 shows the one shortest path and the full ordered list is shown below.

SHORTEST PATHS

AHE time(min)= 48.7
AIWHE time(min)= 56.8
AHME time(min)= 63.0
AIKJHE time(min)= 66.2
ABDZE time(min)= 67.2
ABCFGE time(min)= 67.2
AIWHME time(min)= 71.0
ABCFGME time(min)= 74.1

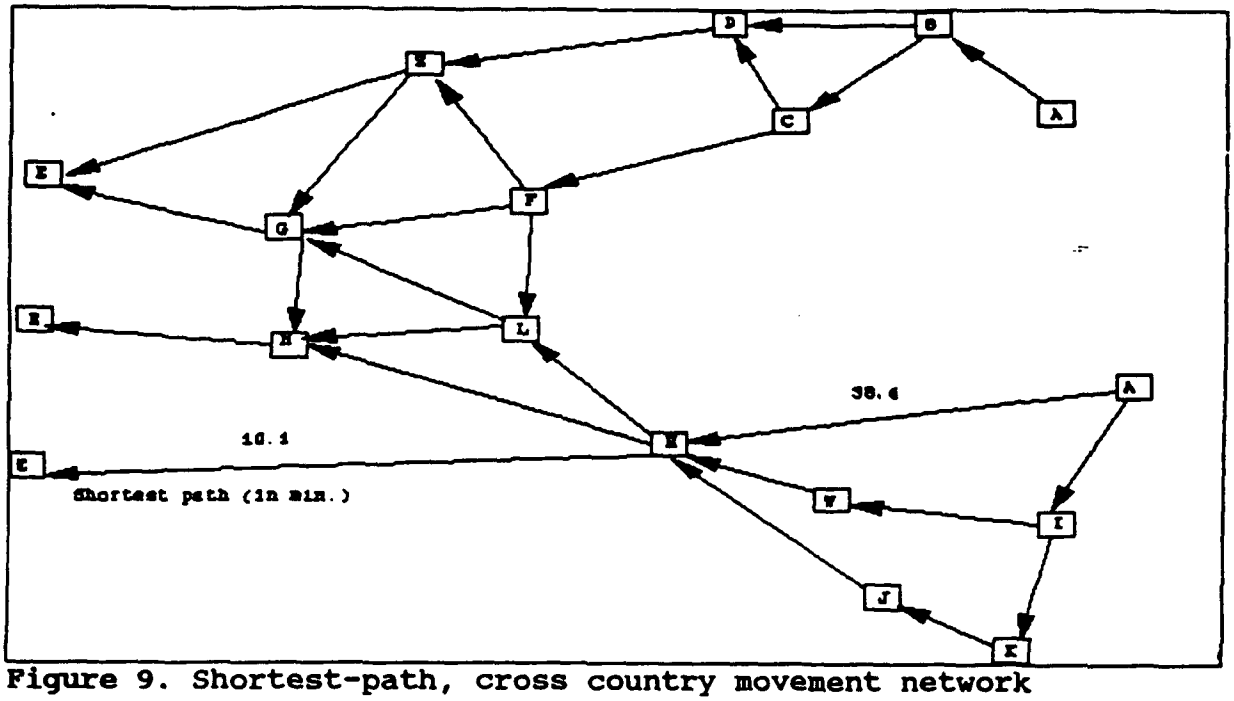


Figure 9. Shortest-path, cross country movement network

```
ABCDZE time(min)=      74.9
ABCFLGE time(min)=      76.2
ABCFLME time(min)=      78.9
AHLGE time(min)=      79.8
AIKJHME time(min)=      80.5
ABDZGE time(min)=      80.6
ABCFZE time(min)=      81.4
AHLME time(min)=      82.5
ABCFLGME time(min)=      83.0
AHLGME time(min)=      86.7
ABDZGME time(min)=      87.4
AIWHLGE time(min)=      87.9
ABCDZGE time(min)=      88.4
AIWHLME time(min)=      90.6
AIWHLGME time(min)=      94.7
ABCFZGE time(min)=      94.8
ABCDZGME time(min)=      95.2
AIKJHLGE time(min)=      97.4
AIKJHLME time(min)=     100.0
ABCFZGME time(min)=     101.7
AIKJHLGME time(min)=     104.2
```

If the network contains q level 1th critical path backtracking nodes with q or 1 greater than 1, then the computation of the number of possible paths becomes more complex. Given $n_{q1}ji$ q level

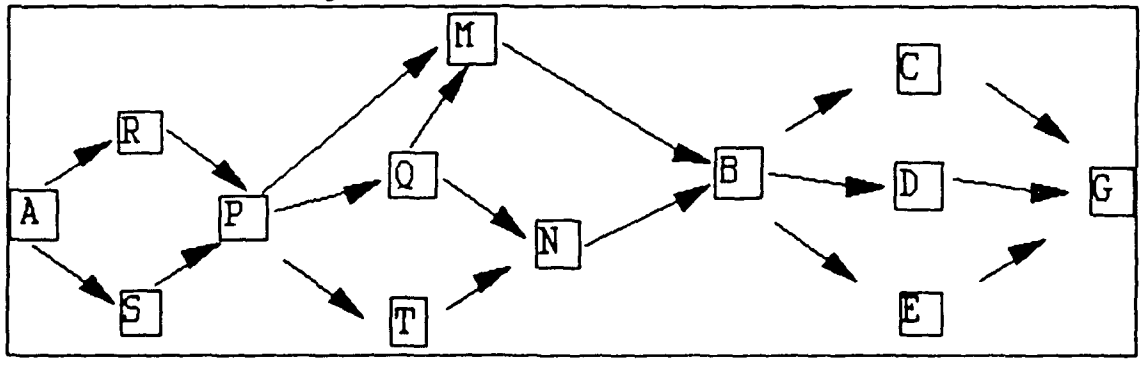
the mth among the q levels of that sth node which are preceeded by backtracking nodes with more than 1 edge entering them and the rth among the l edges entering the node, let $PRIOR_{m.r.s}$ be the number of edges entering that prior critical backtracking node. Then, the number of new paths generated by these nqlji q level lth critical backtracking nodes is bounded by:

$$\sum_{s=1}^{n_{qlji}} \left(\prod_{m=1}^{q-1} \left(\prod_{r=1}^{l} PRIOR_{m.r.s}\right)\right) * i_s - 1$$

i_s = number of paths leaving the sth critical backtracking node (which is bounded by maxe)

So, the total number of such additional paths will be bounded by (ncrit - n1crit)*(emax^q *maxe) - 1. Note, in this computation, a critical backtracking node may preceed (by occuring closer to the start node in the network) more than one other such node a certain number of times. In this case this formula will overcount the number of paths by that same factor. See the example below for an illustration of how this can occur. But whatever the case, these additional paths exhaust all the ways in which the algorithm can possibly backtrack to produce solutions. Thus, adding this bound to the previous one we have the expression given in the statement of the theorem. Q.E.D.

Example 3) In the network below the node B is a 2 level 2th critical bactracking node.



ARPMBCG	ARPNBCG
ASPMBCG	ASPNBCG
ARPQMBCG	ARPTNBCG
ASPQMBCG	ASPTNBCG
ARPMBDG	ARPNBDG
ASPMBDG	ASPNBDG
ARPQMBDG	ARPTNBDG
ASPQMBDG	ASPTNBDG
ARPMBEG	ARPNBEG
ARPQMBEG	ASPNBEG

24 solutions s = 1 , q = 2, l = 2, i_1 = 3

$48 = p_{3.1.1}\{p_{2.1.1}*p_{1.1.1} + p_{2.2.1}*p_{1.2.1}\}*3$

$= 2\{2*2 + 2*2\}*3 > 24$

$PRIOR_{m.r.s} = p_{m.r.s}$

ARPMBEG	ARPNBEG	$PRIOR_{m.r.s} = p_{m.r.s}$
ARPQMBEG	ASPNBEG	
ASPNBEG	ARPTNBEG	
ASPQMBEG	ASPTNBEG	

In this example B is a two stage 2th critical backtracking node. This is because the 2 backtracking nodes M and N which precede B in the network have more than 2 edges entering them and and there are 2 levels of critical backtracking nodes P and then M,N which preceed B. Because the node P preceeds the nodes M and N, the formula overcounts by a factor of 2.

In the applications that these theorems are used we have some freedom in how many nodes and edges are included in the network. The network will be a representation or model of some physical situation or process. In practice it becomes more and more unlikely in a random physical situation that planar graphs containing many q level 1th critical backtracking nodes where q or 1 >> 1 will occur. In movement networks based upon digitized maps such nodes correspond to critical choke points at which the routes diverge to go around an obstacle and then reconverge.

Theorem 2 Considering the same type of graph as in Theorem 1, now allow paths to travel in directions opposed to the way the edges are directed. Then the bound for the number of possible paths is increased to :

$$1 + emax*(n - n1 - ncrit)_q + ((emax + maxe)*maxe - 1)*n1crit + (((emax + maxe)*maxe)^q - 1)*(ncrit - n1crit)$$

Proof: Count the number of paths using the algorithm as in theorem 1. For those nodes with only one edge entering them, the new number of possible edges which proceed outward has been increased by one. These edges will create :

$$1 + 2*n2 + 3*n3 + i*ni \ldots emax*nemax \text{ paths.}$$

This number is bounded by 1 +emax*(n - n1 -ncrit). For the nodes with j > 1 edges entering them, the new maximum number of edges leaving the node is emax + maxe. The rest of the formula follows from the previous calculations.

Theorem 3 The number of search steps required to compute all the paths of the type of graphs mentioned in theorem's 1 and 2 is bounded by:

$$1 + emax*(n - ncrit) + [(emax)^{(q+1)}]* ncrit$$

where q = the maximum level of any q level 1th critical path backtracking nodes in the network,

Proof: This can be verified by tracing through the algorithm and adding a marker to a node's count each time a search step is performed. In following through the algorithm note that since the network is connected each node is encountered during the searching and backtracking exactly the sum of the number of times that there are different edges proceeding into it multiplied by the number of times the preceding nodes to that edge have already been encountered. For the nodes which are not critical path backtracking nodes we have:

1*(1n +1) + 2*(2n) + .. emax*(emaxn) search steps,

where in = the number of nodes witn exactly i edges entering into them. This number is bounded by the number 1 + emax(n - ncrit). For a node which is a q level 1th critical backtracking node, the sum of the number of times that there are different edges proceeding into it multiplied by the number of times the preceeding nodes to that edge have already been encountered is bounded by $(emax)^{(q+1)}$. Q.E.D.

Example 2 (continued)

In this graph, which has no critical path backtracking nodes, we have 1n = 10 and 2n = 3 so the number of search steps should be (10 +1) + 2*3 = 17. This can be verified by tracing through the algorithm and keeping a count as below:

node	search encounters	node	search encounters
A	I	H	I
B	I	M	II
D	I	L	I
C	I	J	I
I	I	K	II
G	I	N	I
F	II		
E	I		

total number of encounters = 17

example 3 (continued)

This graph has a two level 2nd critical backtracking node B. The number of search encounters for the various nodes is as below:

node	search encounters	node	search encounters
P	II	B	IIIIIIII
Q	II	C	IIIIIIII
T	II	D	IIIIIIII
M	IIII	E	IIIIIIII
N	IIII		

total number of encounters = 46

The following theorem holds for the same reasons as the

preceeding discussions.

Theorem 4 The number of search steps required to compute all the paths in the case where the paths are allowed to travel in opposite directions is bounded by:

$$1 + (emax + maxe)*(n - ncrit) + \{(emax + maxe)^{(q+1)}\}*(ncrit - n1crit)$$

Methodology for Solving Maximal Flow problems

Given that there is a method of generating all the shortest paths through the network, it is easy to modify the predicates to generate a list of maximal flow paths through the network. We now assume that each route segment has an associated maximal flow capacity. At each stage, simply choose the direction of maximal flow to expand the paths, and perform a sort on the maximal flow of the route segments instead of minimum length. Then, write a predicate that each time we reach the goal node with a route, go back and subtract that route's flow values from the network capacities. When the Prolog backtracking search does not generate any more solutions, all directed paths through the network have at least one edge which is already filled to capacity. One other way exists to increase the flow in the network. The paths which contain edges that point backward along the allowed route segments can be considered. Then, when the goal node is reached, proceed back to the source and modify the flow capacities, and add flow capacity along those segments, instead of subtracting it. As explained by Sedgewick [10] when the above procedure reaches a situation in which all paths have either full forward edges or nonempty back edges, then the Ford Fulkerson theorems says the maximal flow of the network has been reached. Many of the algorithms presently in use Goldberg [6] and Tarjan [12] for solving the maximal flow problem do not save lists of partial paths but instead use a labeling process to update information at each node. This increases computational speed, but makes it difficult to pick in order the main routes that contribute to the optimal solution. Since it may be desirable to do sensitivity analyses which locate the points in the network which most affect all the possible solutions, the above approach gives more information after the process is completed. Thus, it is possible to print out in order of flow the paths which contributed to the max-min cut situation. This then can be used to plan barriers for the defense or attack routes for the offense. The description of the algorithm is as given below:

Step 1. Search for the best(maximal flow) route from the starting to the ending node. Only search in directions in which there is either a forward edge with positive unused flow capacity, or a backward edge with positive existing flow. If no route exists, terminate the algorithm.

Step 2. Subtract the value of that flow from the capacity slots in the route's definition and add the flows (or subtract the flows if headed in a backward direction along an edge). Go to step 1.

Further explanation of the algorithm and examples of its use are given in Harrell's paper [6]. The source code of its implementation is in the technical report [9]. If flow rates are added to the edges in Example 2 the above procedure will generate the maximum vehicular flow across the network and compute the sensitivity of the solution to changes in flow rates at critical nodes.

Off-road vehicle flow capacities may be estimated from vehicular movement formations and speeds in the terrain corridor that each edge corresponds with. Suppose that the terrain will support a certain number of units as shown in Figure 11 and the movement speeds are as in Figure 8. Figure 12 shows the number of vehicles per square kilometer that correspond to a particular movement formation. Multiplying [1/(time it takes a group of vehicles to traverse the edge)]* number of vehicles in the group determines the flow rate associated with an edge. Then Figure 13 shows a maximum flow solution and Figure 14 shows the changes in the solution caused by changing the speeds on three edges.

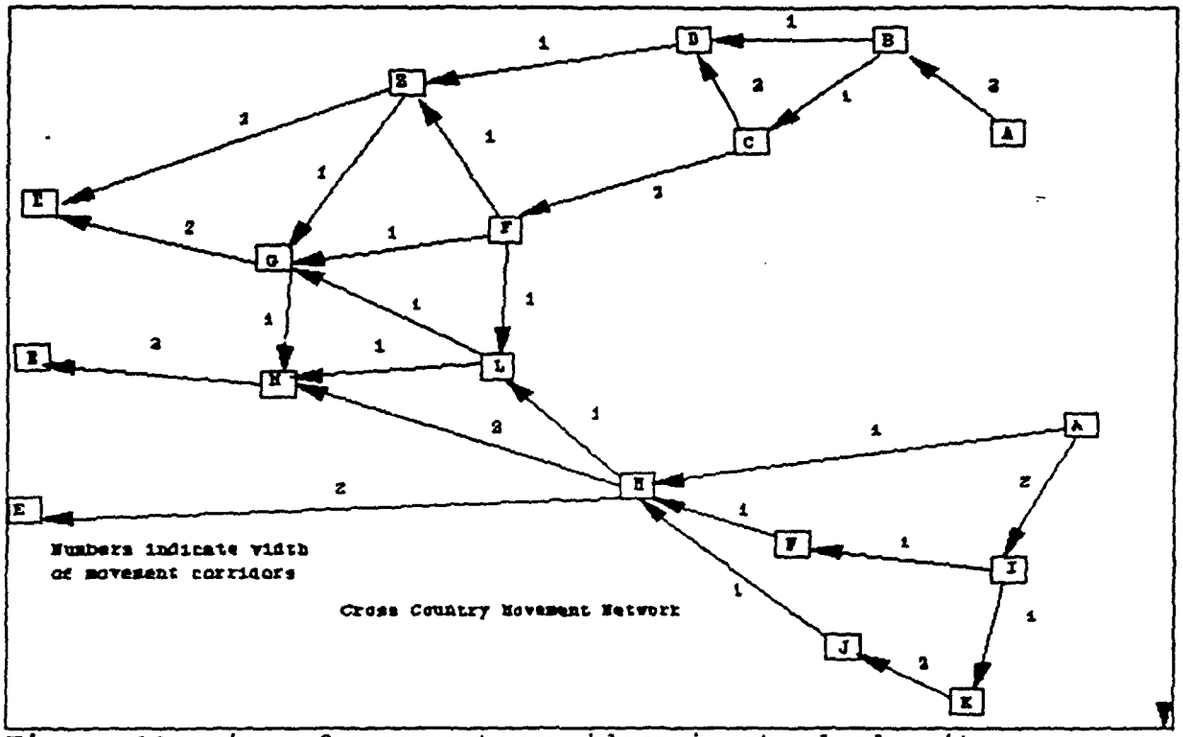


Figure 11. size of movement corridors in standard units

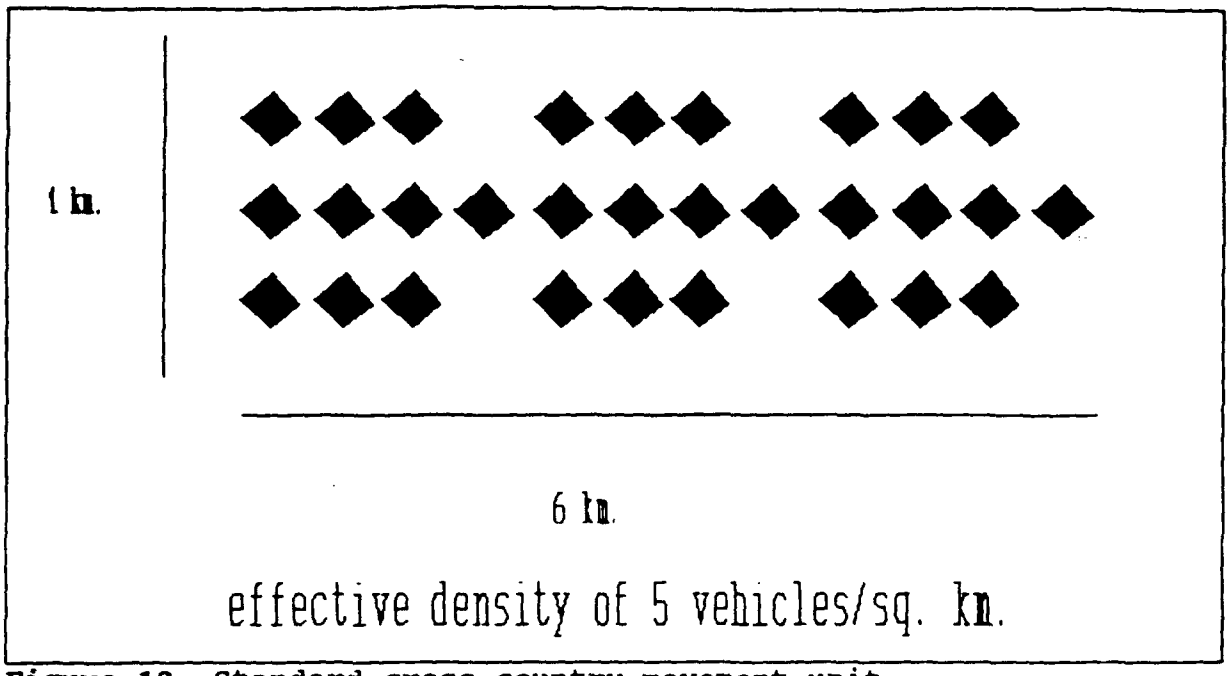


Figure 12. Standard cross country movement unit

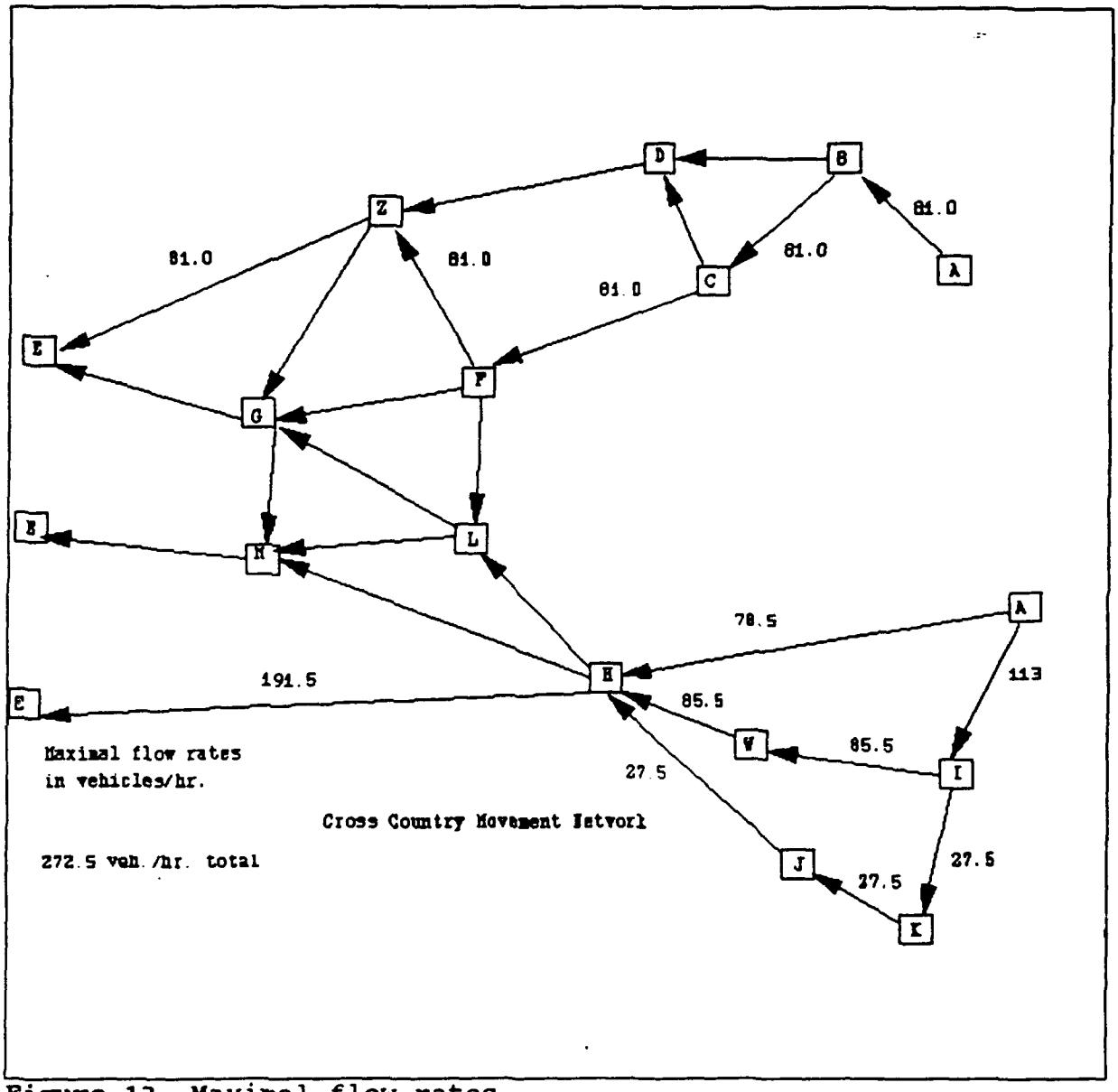


Figure 13. Maximal flow rates

The flow paths and their maximum capacities that are associated with the solutions of Figures 13 and 14 are :

Maximal flows without obstacles

MAXIMAL FLOWS

AIWHE flow veh/hr = 85.5
ABCFZE flow veh/hr = 81.0
AHE flow veh/hr = 78.5
AIKJHE flow veh/hr = 27.5
total flow 272.5 veh./hr.

If:

a. Four anti-tank ditches with corresponding parallel tank bumps,
b. One standard(conventional) rectangular minefield Maximal Flows with obstacles,
c. Four scatterable minefields,

are emplaced the maximal flows are reduced.

Maximal flows with obstacles

MAXIMAL FLOWS

ABCFZE flow veh/hr = 81.0
AHE flow veh/hr = 78.5
AIWHE flow veh/hr = 15.5
AIKJHE flow veh/hr = 6.5
total flow 181.5 veh./hr

Theorems 1 through 4 solve the problem of determining how many steps it takes this algorithm to generate all the paths that create a max-min cut. As outlined in Example 3, after each search step the new edges must be placed in a sorted priority queue of current partial search paths. The maximum number of insertions required after each search step is emax + maxe. Each insertion requires the checking of the lengths of at most:

$$1 + emax*(n - n1 - ncrit) + ((emax + maxe)*maxe - 1)*n1crit + (((emax + maxe)*maxe)^{q} - 1)*(ncrit - n1crit)$$

partial paths {according to Theorem 2} against the lengths of the new paths created by adding an edge onto the active search path. Let e* = maxe + emax, and assume e* <= some constant C1,then the above expression is less than or equal to:

$$1 + ncrit* C1^{2q} + n*2C1^{2}$$

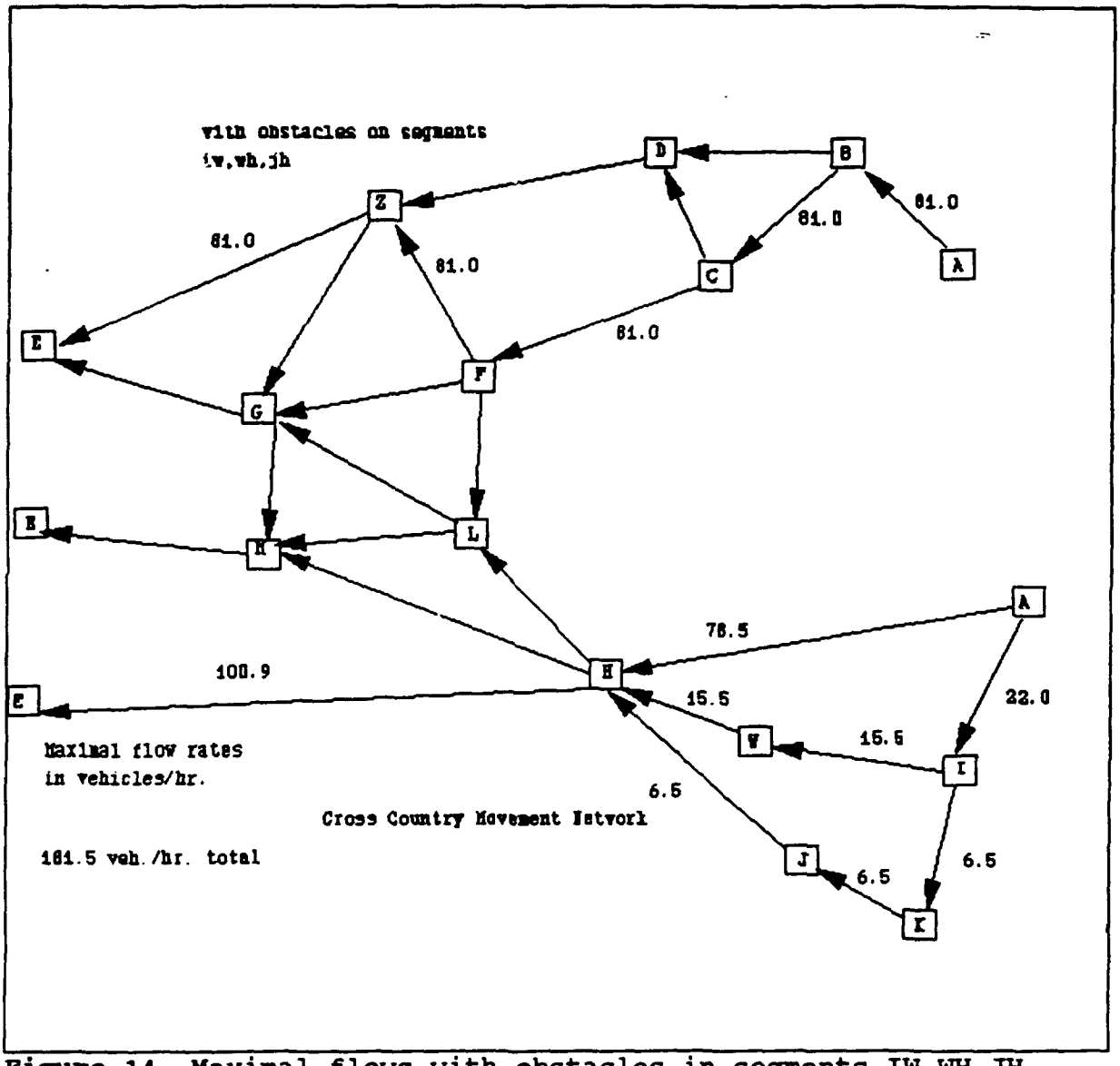


Figure 14. Maximal flows with obstacles in segments IW,WH,JH

The total number of steps is then bounded by:

$$\{1 + (emax + maxe)*(n - ncrit) + (emax + maxe)^{(q+1)}*(ncrit - n1crit)\} * [emax + maxe] * \{1 + ncrit*C1^{2q} + n*2C1^{2}\}.$$

This is less than or equal to:

$$\{C1 + n*C1^{2} + ncrit*C1^{(q+2)}\}\{1 + ncrit*C1^{2q} + n*2C1^{2}\}.$$

And this expression is of the order:

$$\{n^{2}\}*2*C1^{4} + ncrit * C1^{(3q+2)}\}$$

Theorem 5 If in designing the networks which represent the movement possibilities, we limit ourselves to the case:

$$(emax + maxe)^{4} <= n \text{ and } (emax + maxe)^{(3q+2)} <= n^{2}, \; q<=2$$

then the number of steps needed to solve the maximal flow problem is of the order of n^{3}, the number of vertices cubed.

Also, if we are not interested in generating the saturating flow paths in priority order of increasing flow, then the partial paths do not need to be sorted after each search step. By examining the above expressions we see that this reduces significantly the time the algorithm takes to compute a maximal flow.

4. Methodology for Solving the Minimal Cost Network Flow Problem

As a further benefit of the above approach, the procedures developed can be used to solve the minimal cost network flow problem. The minimal cost network flow problem is a generalization of the transportation network problem in operations research. In its formulation each edge is assumed to have a cost as well as a flow capacity associated. In this paper in order to determine the cost associated with a flow the following procedure is used: 1) The flow rate on each flow path solution through the network is multiplied by its time of traversal and the result summed over all paths in order to obtain a total cost associated with a given maximum flow solution. This is the measure of effectiveness which determines the optimality of the solution. The total cost of the flow can then be divided by the total maximum flow to obtain an average cost per vehicle to travel through the network. The min-cost flow problem is then the problem of determining from all the possible flows which realize a given network maximal flow value,those which do it with minimal cost. This measure of effectiveness is important in wargaming because it represents the

amount of target exposure which is required for an offensive force to reach its objectives. As in the shortest path algorithm presented earlier, the algorithm that will be used to compute total cost of the flow expands the paths at each stage in the direction of shortest time. The resulting expanded pathlist will be sorted on time of the routes (in the maximal flow algorithm the maximal flow possibilities were sorted on). Only those directions in which the maximum flow algorithm says there is either a forward edge with positive unused flow capacity or a backward edge with positive existing flow should be chosen. As in the maximum flow algorithm, when the goal node is reached we proceed back to the source and subtract the maximum flow that least cost incremental flow route can handle(in the maximal flow algorithm, the route is not necesarily the least cost incrementing flow).

The following theorem from Ford and Fulkerson [5], which is also noted in Deo [3], insures that this process will generate the optimum solution.

Theorem 6 Let f be the minimal cost flow pattern of value w from start to finish. The flow pattern f' obtained by adding delta <= 0 to the flow in the forward edges of a minimal cost unsaturated path, and subtracting delta from the flow in the backward edges of the path is a minimal cost flow of value w + delta for the original network.

The same source code predicates can be used to implement this algorithm:

Step 1: Search for the best(minimal cost) route from the starting to the ending node. Only search in directions in which there is either a forward edge with positive unused flow capacity, or a backward edge with positive existing flow. If no such route exists, then terminate the algorithm.

Step 2: Subtract the value of that flow from the capacity slots in the route's definition and add the flows (or subtract the flows if going in a backward direction) along the edges. Go to Step 1.

If we again consider the network in Example 2, it is now possible to solve the problem of determining which of the several maximum flow solutions costs less in the above sense.
Minimal cost flows for the same network and same vehicle/weather conditions are shown below in Figure 15. The maximum throughput for the minimal cost flows is the same as that which results from the maximal flow algorithm. The paths followed to acheive this throughput is, however, different in the minimal cost flows from those which result from running the maximal flow algorithm. This is to be expected since in the minimal cost case the algorithm chooses the direction of shortest time to expand the search path. In the maximal flow case, the algorithm chooses the direction of

maximum flow to expand the search paths. In the list of the minimum cost flow paths, we have included an average cost for each flow. This is defined to be the average of the sum of the amount of each flow path (vehicles) times the cost of it (minutes). Note that this average does not change much with and without the presence of obstacles. This is because what the obstacles affect (being employed over only a part of the network) is primarily the maximum throughput, and not the time through the network.

```
MIN_COST FLOWS
     Minimum cost flows without obstacles


AHE     flow veh/hr =   78.5 cost time(min)=   48.7
AIWHE   flow veh/hr =   85.5 cost time(min)=   56.8
AIKJHE  flow veh/hr =   27.5 cost time(min)=   66.2
ABDZE   flow veh/hr =   66.5 cost time(min)=   67.2
ABCFGE  flow veh/hr =   14.5 cost time(min)=   67.2
```

total flow = 272.5 veh\hr. total cost = 78.5*48.7 + 85.5*56.8 + 27.5*66.2 + 66.5*67.2 +14.5*67.2. average cost = 58.5 minutes per vehicle

Explanation of How this Algorithm Can Be Used To Solve the Transportation Network Problem

By the transportation network problem the following is meant: Consider n points located on a map as origins of logistical material. Each point has associated with it a supply of a[i] units of the material. In addition, there are m destination points, with each destination point requiring b[i] unit of the material. Associated with each link in a network between the sources and the destinations there is a unit cost of transportation and a flow capacity. The problem is to determine the shipping pattern from origins to destinations that minimizes the total cost under the constraints imposed by the flow capacities on each link. By defining n paths each with a flow capacity equal to a[i] from a notional starting point, and m paths each with a flow capacity equal to b[i] from the destinations to a notional ending points, this problem can be considered as a special case of the minimal cost network flow problem discussed in the previous section. The algorithm given to solve that problem will in the process of computing the maximal flow in the network just defined, produce the minimal shipping cost solution which satisfies most of the total requirements at the destinations. With simple modifications to the starting requirements for the search routines the algorithm

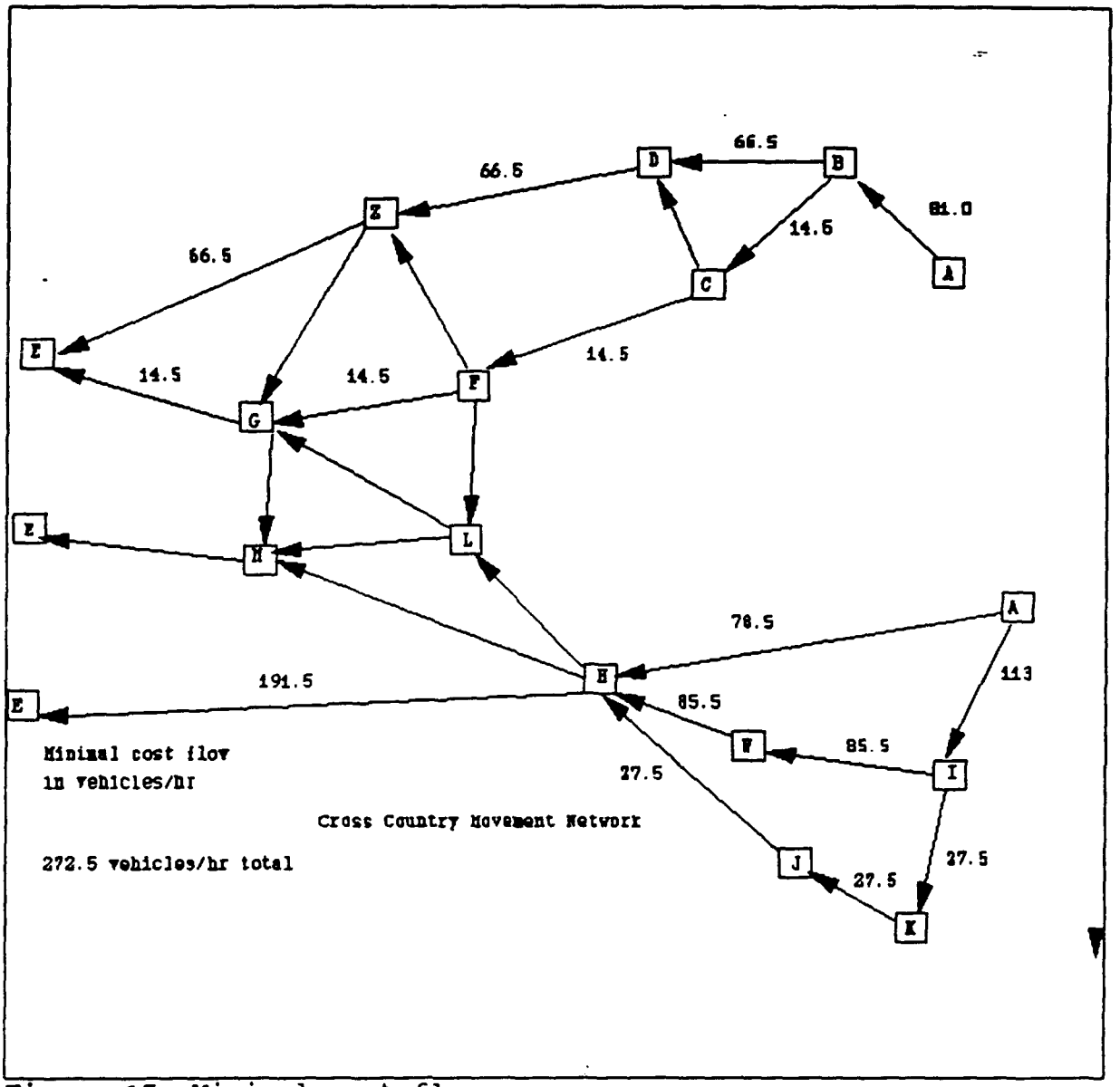


Figure 15. Minimal cost flows

will produce solutions which satisfy the list of destinations in any given prioritized sequence.

Conclusions

6. We have given definitions and examples of some artificial intelligence network terminology and discussed several ways in which sorted priority queue depth-first searches can be used to solve shortest path, network maximal flow, and min-cost network flow problems. We have shown that the time bounds for these algorithms depend on the number of critical path backtracking nodes in the following sense: If there are no critical backtracking nodes, and the network is directed, then there are at most (emax -1)*(n - n1) + 1 paths in the whole search space. If there are critical path backtracking nodes then the number of paths is bounded by:

$$1 + (emax - 1)*(n - n1 - ncrit) + (maxe*emax - 1)*n1crit + ((maxe*emax)^{q} - 1)*(ncrit - n1crit)$$

We obtained similiar expressions for the case in which the paths can go either forward or backward along edges in the network. We used these expressions to obtain time bounds for the total number of steps to solve the maximal flow and min-cost flow problems.

BIBLIOGRAPHY

[1] Borlund Int., Turbo Prolog,Version 2.0,User's Guide, Scotts Valley, CA,1988.

[2] Bratko, Ivan, Prolog Programming for Artificial Intelligence, Addison Wesley Publishing Co, Reading, MA,1986.

[3] Deo,Narsingh, Graph Theory with Applications to Engineering and Computer Science., Prentice Hall, Englewood Cliffs, NJ, 1974.

[4] Edmonds, J. and Karp, R. M.,"Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems," Journal for the Association of Computing Machinery, Vol. 19, no. 2, pp.248-264, New York, N.Y.

[5] Ford, L. R. and Fulkerson, D. R., Flows in Networks, Princeton University Press, Princeton, N.J.,1962.

[6] Goldberg, A. E. and Tarjan, R. E., "A New Approach to the Maximum-Flow Problem", Journal for the Association of Computing Machinery, vol.35,no. 4,pp. 921-940, New York, N.Y.,1988.

[7] Harrell, A. W., "The Concept of Individual Vehicular and Unit Mobility and its Effect on Wargaming", Proceedings 27th Army Operations Research Symposium,Vol. II,pp. 3-993 to 3-1006,1988.

[8] Harrell, A. W., "The Concept of Individual Vehicular and Unit Mobility and its Effect on Wargaming", Proceedings 56th Military Operations Research Symposium, 1989.

[9] Harrell, A. W., "Evaluating the Effect of Off-Road Obstacles on Unit Movement",Technical Report GL-89-4, Waterways Experiment Station,Vicksburg, Ms., 1989.

[10] Sedgewick, Robert, Algorithms, Addison-Wesley, Reading, MA. ,1983.

[11] Sterling, Leon and Shapiro, Ehud, The Art of Prolog, The MIT Press, Cambridge, MA.,1986.

[12] Tarjan, R. E., Data Structures and Network Algorithms, Society for Industrial and Applied Mathematics, Philadephia, PA,1983.

[13] Winston, Patrick Henry, Artificial Intelligence 2nd ed., Addison Wesley, Reading, MA.,1984.

[13] ....., Lisp 2nd ed., Reading, MA.,1984.

Derive as a Research Tool

David C. Arney
Jeffrey L. Misner

Department of Mathematical Sciences
United States Military Academy
West Point, NY 10996-1786

ABSTRACT. Computer Algebra Systems (CAS) are powerful and efficient tools that can help with problem solving in research environments. In this paper, we outline some of the capabilities of one particular CAS, *Derive*, which runs on IBM personal computers and compatibles. The discussion includes analysis of symbolic and numeric computation and graphic displays. Emphasis is placed on the special functions and computational modes most beneficial to researchers. We present some of the general algorithms used by this software package and discuss the role of CAS in solving research problems. The limitations of *Derive* are presented.

INTRODUCTION. A Computer Algebra System, capable of symbolic manipulation, is a powerful and efficient tool in a research environment. At the United States Military Academy (USMA), every student is required to purchase *Derive* (distributed by Soft Warehouse, Inc) for their use in all mathematics courses. Also, all the mathematics, science and engineering faculty have *Derive*. USMA is an undergraduate institution with a student population of approximately 4400 cadets, each of whom possesses an IBM-compatible personal computer. Every graduate fulfills the requirements for a bachelor of science degree, with about 50% obtaining a major or field of study in a math, science, or engineering discipline. USMA is certainly not a research institution, but a number of both students and faculty are involved in a myriad of research areas, often involving interaction with other Department of Defense agencies. *Derive* is proving to be an extremely useful tool for both the student or faculty researcher.

In a research environment, the most popular uses of a computer algebra system are in the following areas:

- Solving differential equations
- Fourier transforms
- Group theory

- Laplace transforms
- Number theory
- Calculus (including multivariable and vector)
- Linear algebra

CAPABILITIES AND EXAMPLES. Let us examine a few of these areas to see how *Derive* can assist the researcher. Once the program is executed, it is possible to load one or more utility files into the *Derive* window. These utility files can contain user built functions, which can then be used for repetitive calculations. These functions supplement the functions executed through the *Derive* menu or the standard in-line functions always available. *Derive* itself comes with several of these files, including files for first- and second-order differential equations, recurrence equations, probability functions and special functions such as Bessel and Airy functions. Also included is a file containing many common physical constants and unit conversion factors. Users can create their own files containing functions from the above files or their own unique functions. For example, the two tables below provide a partial listing of pre-defined functions available to solve first-order differential equations symbolically.

Function Name	Purpose/Form of Equation to be Solved
FUN_LIN_CCF (q,a,b,c,x,y,x0,y0)	solves $y' = q(x,y)(ax + by + k), y(x0) = y0$
LIN_FRAC (r,a,b,c,s,t,d,x,y,x0,y0)	solves $y' = r(x,y)((ax + by + c)/(sx + ty + d))$, y(x0)=y0, sa-eb $\neq 0, cd \neq 0$
INTEG_FCTR_FREE_OF_X (r,p,q,x,y,x0,y0)	solves $p(x,y) + q(x,y)y' = 0$, when the integration test is free of x and gives r
INTEG_FCTR_FREE_OF_Y (r,p,q,x,y,x0,y0)	solves $p(x,y) + q(x,y)y' = 0$, when the integration test is free of y (similar to above)
GEN_HOM(r,k,x,y,x0,y0)	solves general homogeneous equation $y' = r(x,y) = h(yx^k)y/x$
ALMOST_LIN (r,h,p,q,x,y,x0,y0)	solves $r(x,y)y' + p(x)h(y) = q(x), y(x0) = y0$
CLAIRAUT	helps solve the Clairaut equation
TAY_ODE1(r,x,y,x0,y0)	finds 4th degree Taylor-series solution to $y' = r(x,y), y(x0) = y0$
PICARD(r,y_prev,x,y,x0,y0)	given approximate solution y_{prev} to $y' = r(x,y)$, $y(x0) = y0$, and finds an improved iterate

The basic commands for solving first-order differential equations available in the utility file ODE1 and their purpose.

Function Name	Purpose
SEPARABLE(p,q,x,y,x0,y0)	solves a separable differential equation, $y' = p(x)q(x), y(x0) = y0$
EXACT_IF_0(p,q,x,y)	checks if equation $p + qy' = 0$ is exact
EXACT(p,q,x,y,x0,y0)	solves the exact equation (above), with y(x0) = y0
USE_INTEG_FCTR (m,p,q,x,y,x0,y0)	solves equation with known integrating factor m
LINEAR1(p,q,x,y,x0,y0)	solves linear equation $y' + py = q$, with y(x0) = y0
BERNOULLI (p,q,k,x,y,x0,y0)	solves the Bernoulli equation $y' + py = qy^k$, with y(x0)=y0
HOMOGENEOUS_IF_FREE_OF_X (r,x,y)	if this returns a 0, the equation $y' = r(x,y)$ is homogeneous
HOMOGENEOUS(r,x,y,x0,y0)	solves the homogeneous equation (shown above)

Other commands available in ODE1 and the form of the equation they solve.

A few practical examples should provide the necessary illumination on how these kinds of functions can be used. Suppose we wished to solve the initial-value problem

$$y'' - 2y' + y = 10e^{-2x}\cos(x), \qquad \text{subject to } y(0) = 1 \text{ and } y'(0) = 2.$$

This is recognized as a second-order, constant coefficient, nonhomogeneous differential equation. After loading the ODE2 file, we proceed as follows:
First, classify the nature of the homogeneous solution using the command

LIN2_RED_CCF_DISC(−2,1)

The discriminant will be either positive (real, distinct roots), negative (imaginary roots), or zero (repeated roots). In our example, the discriminant is zero, indicating repeated roots. Our next command would then be

LIN2_RED_CCF_0(−2,x)

This would return the homogeneous solution, complete with arbitrary constants, as shown below

$$@1\, e^x + @2\, x e^x.$$

The particular solution can then be found using the command

$$\text{LIN2_COMPLETE}(e^x, xe^x, 10e^{-2x}\cos x, x)$$

The particular and homogeneous solutions can then be added together, and initial conditions applied by using the command

$$\text{IMPOSE_IC2}(x, f(x), 0, 1, 2)$$

which would then yield the solution

$$y = \tfrac{1}{5}e^x + 4xe^x + (\tfrac{4}{5}\cos x - \tfrac{3}{5}\sin x)\, e^{-2x}.$$

Derive can also plot this solution using either a full or split screen window.

Some other quick examples of the capabilities of ***Derive*** include the following:
The command

$$\text{LAPLACE}(f(t), t, s)$$

yields the Laplace transform of f(t). A piecewise continuous function can be constructed using built-in functions of ***Derive***, and then its Fourier series found by using the command

$$\text{FOURIER}(f(x), x, a, b, n)$$

where *a* and *b* specify the interval and *n* is the number of terms desired. ***Derive*** can solve multiple integrals (or derivatives of any order) symbolically, and can approximate definite integrals with an adaptive quadrature routine. ***Derive*** also has built-in commands for such vector operations as finding the Laplacian, divergence, curl, and potential of a given vector. Many matrix algebra functions are also included in ***Derive***, such as computing determinants or inverses, finding eigenvalues, or row-reducing a matrix. In short, the capabilities of ***Derive*** can deliver the researcher from much of the tedium of "number crunching," thereby allowing

greater time for actual research.

LIMITATIONS. Lest we come to think of ***Derive*** as being capable of performing all of our mathematical manipulations, it is important to realize its limitations. First and foremost, it has no programming language to do branching or iteration. There is no way to input superscripted or subscripted variables. ***Derive*** is relatively slow at plotting graphs, especially when plotting "out-of-range." Its three-dimensional plotting is adequate, but would be much enhanced if contour plots could be generated. Also, there is no way to edit graphs to add such amenities as axis labels. Perhaps the most significant shortcoming occurs when an impossible operation is attempted. Sometimes, instead of providing some type of warning or error message, ***Derive*** simply "does nothing," leaving the user wondering why the command was not executed.

CONCLUSION. Despite ***Derive***'s limitations, its user-friendly interface and relatively low cost make it an excellent aid to any researcher. The researcher with limited computation hardware can especially appreciate its small size – the entire program, to include utility files, will fit on a standard $5\frac{1}{4}$ inch floppy disk and executes on standard IBM PCs and compatibles. This is extremely convenient for those researchers that have computers without a hard disk drive. Finally, for students and faculty here at USMA, the availability of ***Derive*** makes it an extremely effective tool for conducting group research projects.

COMPUTER ALGEBRA SYSTEMS: CAPABILITIES, APPLICATIONS, AND IMPACT ON EDUCATION

David C. Arney

Department of Mathematical Sciences
United States Military Academy
West Point, NY 10990-1786

James P. Cummings

Department of Mathematical Sciences
United States Military Academy
West Point, NY 10990-1786

Lee S. Dewald

Department of Decision Sciences
Army Logistics Management College
Fort Lee, VA 23801

ABSTRACT. The United States Military Academy (USMA) is fortunate to have a computer-rich educational environment. Every student purchases a personal computer, every faculty member has one at his or her desk, and all the departments have mobile computers equipped with an overhead projection device to bring the computer experience into the classroom. Additionally, each student purchases standard software consisting of word processing, spreadsheet, and computer algebra systems (CAS). The challenge to the faculty is to effectively use the available resources to enrich the students and increase their understanding of the concepts presented. The Department of Mathematical Sciences (D/MS) relies heavily on CAS to take the drudgery out of the computations and to put excitement into its courses.

1. INTRODUCTION. Prior to its foray into using CAS, D/MS had already decided on some of its software requirements. Minitab and Quattro would be used as statistical and spreadsheet packages respectively, and some other software packages, like Calculus Toolkit and the Mathematics Plotting Program (MPP) would be available as demonstration packages. While these last two programs performed well, they were limited in their scope.

In the spring of 1989, D/MS began a search for a reliable CAS. It wanted something that was easy to use and could integrate numerical, graphical, and analytical procedures. Since all the cadets would be buying a copy for their own computer, cost was important.

Derive, a commercial product from The Soft Warehouse was selected as the software that best met the requirements. The D/MS faculty began the following program to integrate CAS into the classwork.

The first 100 copies arrived in May of 1989 and were distributed to the math and science faculty. In June, members of the D/MS presented a demonstration to the math, science, and engineering faculty to let them know the state of CAS and to prepare them to integrate it into their courses. As the first purchase was going to all the freshman and sophomore classes, most of the engineering faculty would have one or two years to validate our experience and integrate it into their own disciplines.

The students received their copies of Derive in October, two months into the academic year. Unfortunately, this was too late in the semester to properly integrate it into the instruction, but plans were made to do so in the second semester. During that fall semester, the instructors were strongly urged to utilize Derive in their classroom sessions and many did so.

During the second semester, students in the Calculus I course used Derive extensively in the classroom and on homework. This was accomplished by utilizing the school's computer labs in conjunction with pre-planned laboratory worksheets. Other math courses used Derive on a weekly basis and included Derive worksheets in some of their lesson material. The Calculus I students, while not unanimous in their opinion of CAS, gave the following comments: "Often in the past, I would give up on homework. Derive is another alternative." "I finally figured out that Derive could help me explain things, not just give me results." "I feel with time I will be able to integrate Derive into my homework work schedule."

A new core math curriculum was implemented in the fall of 1990. This caused the department to change many course priorities and objectives to meet the new requirements of the curriculum. The summer of 1990 was used to restructure courses not only so that they would complement the new curriculum, but with the specific purpose of integrating the CAS into daily lessons.

2. OUR APPROACH. Currently, nine math courses at USMA utilize CAS as a part of their instruction: Precalculus, Discrete Dynamical Systems, Calculus I & II, Differential Equations, Probability and Statistics, Multi-Variable Calculus, Mathematical Modeling, and Numerical Analysis. Additionally, Derive is powerful enough to be used by the students in a variety of other mathematics, science, and engineering courses. Its ability to evaluate integrals and

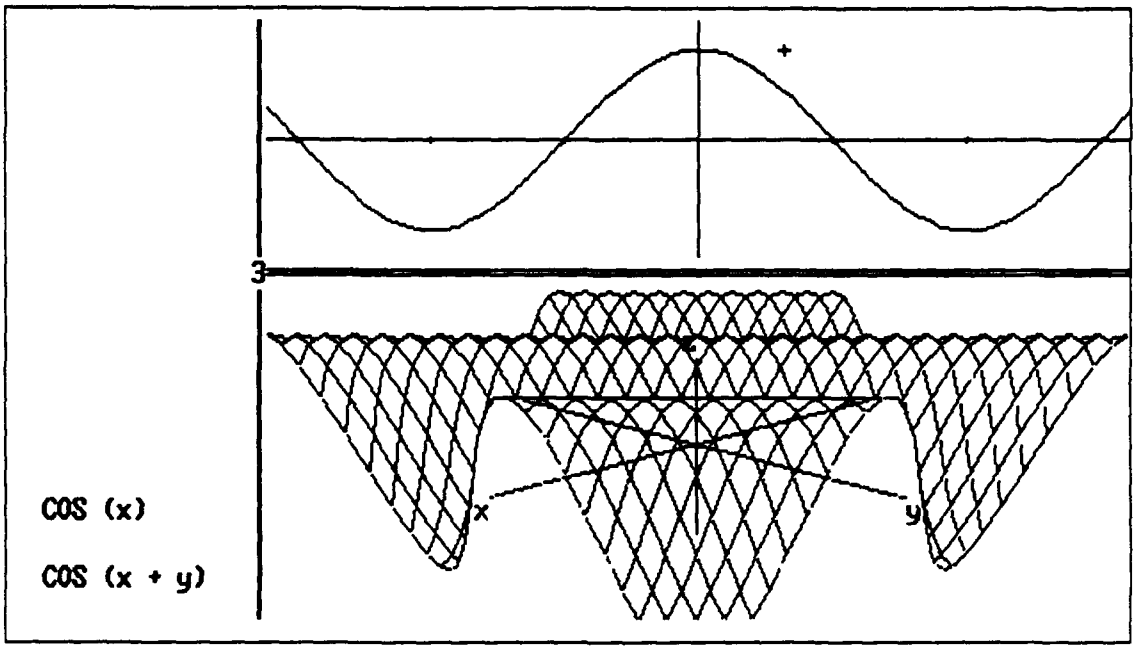
combinations makes it ideal for checking solutions to probability problems. Multiple integration is a challenging problem for our students, but Derive enables students to solve these problems without becoming mired in their intricacies. It is also useful for checking any kind of numerical work that involves complex calculations. The students can do their homework assignment and then rapidly check their work.

In the classroom, Derive is used to demonstrate ideas, solve problems, conduct experiments, and explain results. Many times, this is done as an introduction to the next lesson, whereby the instructor takes the current subject and conducts a "What if?" experiment to introduce the next topic.

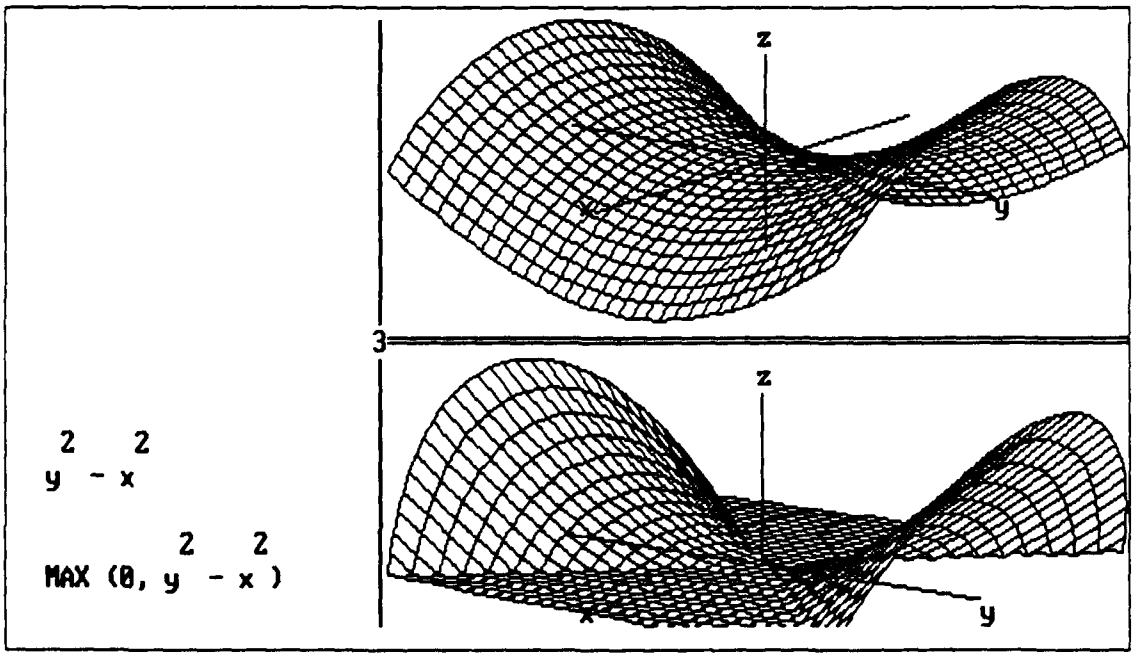
In the larger picture, CAS can help cover the entire spectrum of mathematics; discrete-continuous, linear-nonlinear, and deterministic-stochastic. In this way, the CAS allows the student to do mathematical modeling and work on more realistic problems instead of spending his time memorizing algorithms. Thus, the student is better able to internalize the algorithm through repeated use, instead of memorizing it for a test and then forgetting it through lack of use. The students are encouraged to conduct their own numerical experiments. Changing a parameter, introducing more terms, and exploring limits are some of the things that a student can readily do to experience the excitement of mathematics instead of the drudgery of computational exercises.

Derive's graphing capability helps the student accomplish these things and many others. Changing an expression and seeing a different table of numbers is nice, but watching this change occur on a graph lets the student actually see how the changes influence the behavior of the result. Of equal importance is Derive's capability to use multiple windows. This allows the student to explore different areas while keeping the same screen setup.

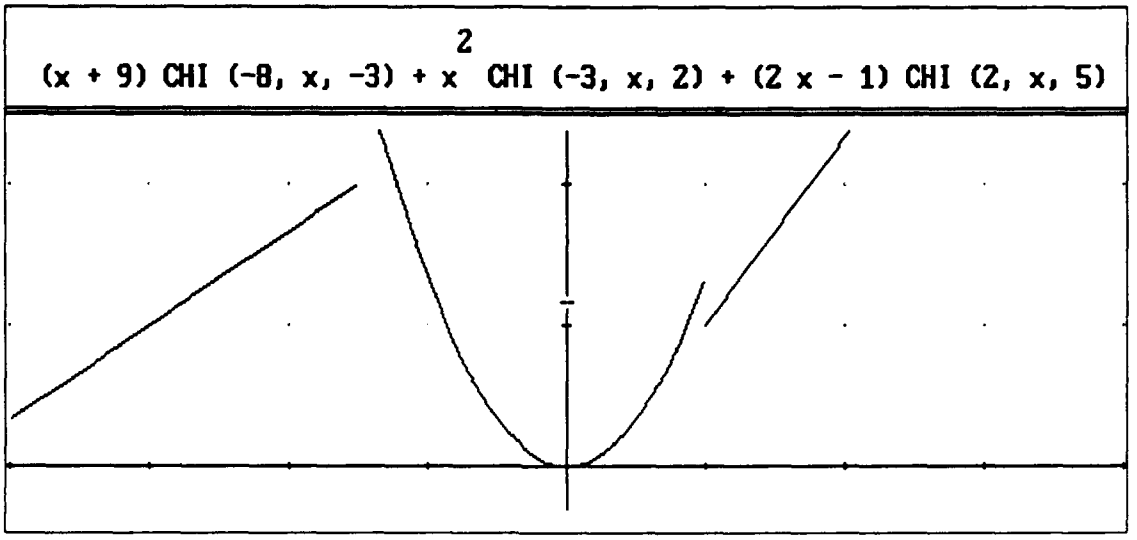
The following figure shows a screen display of three different windows; algebra, 2-D graph, and 3-D graph. If a student wished to explore the cosine function, he or she would do so merely by entering a new expression in Window 1 and having Derive plot the new expression. The original cos(x) can be deleted or left on the graph for easy comparison with the new expression.



Derive is a very capable teaching tool. Students historically have problems dealing with functions in two variables. Using Derive, it can be shown that these functions are accessible. The following graph illustrates the function $f(x,y) = y^2 - x^2$ in the upper right and after cutting it with the plane z = 0 using the MAX function in Window 3.



Similarly, the CHI function is used to introduce piecewise functions and demonstrate discontinuities. Derive allows the student to visualize the effects of discontinuities. This allows the student to see what is taking place when an algorithm indicates that a function has a discontinuity. Derive's expression and plot for a piecewise continuous function is shown in the following figure.

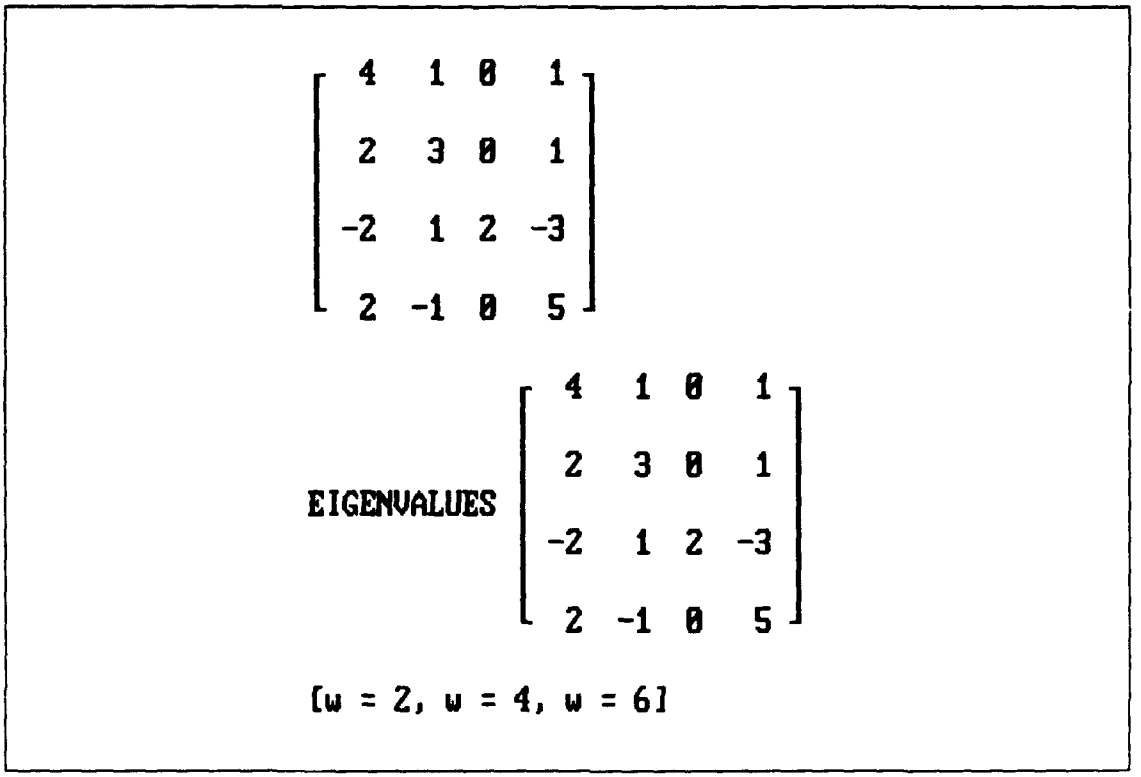


Calculus functions, as are most of Derive's functions, are reached through the menu system. Besides finding derivatives and integrals, Derive gives the student ready access to limits, summations, and Taylor Series. These are useful, not only to determine numerical results, but also to explore the functions and to discover the various attributes of each.

Vector and matrix operations are other types of operations that Derive will do for the student. Scalar multiples, dot products and cross products are all at the student's fingertips. Matrix inverses, addition, multiplication, and finding eigenvalues are all standard operations in Derive.

The student is encouraged to use Derive in two fashions. First, the student can perform many operations sequentially. This helps in understanding the algorithms. Then, the student can simply use the appropriate Derive function to determine if the process was correct. In this way, the students can do larger, more realistic, problems without worrying about getting lost in the quagmire of the individual calculations. Students are also encouraged to use these functions any time that they have homework assignments.

The following figure shows the process of finding the eigenvalues of a 4x4 matrix.



No software package is perfect, and we would be remiss if we did not point out some of Derive's limitations. First and foremost is its lack of programming. However, this may actually benefit students who need to perform step-by-step operations to understand the concept of the algorithm. Another area in which it falls short is its lack of a mouse interface. While most of the commands are easy to get to, editing a previously entered command or moving the cursor around the 2-D graph is tedious when compared to the rest of the program's user-friendly implementation. Another shortcoming is its limited text formatting. All keyboard entries, unless enclosed by quotes, are treated as variables and therefore, entering and printing technical notation is difficult.

3. IMPACT AND LESSONS LEARNED: D/MS is in the process of changing from text and algorithm oriented courses to laboratory and problem solving courses. As these changes are made, there is no longer an emphasis on the techniques of mathematics, but on its concepts and applications. Just as the advent of hand-held calculators removed interpolation and finding square roots from our curriculum, we see the rise of CAS replacing the classroom hours spent mastering algorithms with time spent on the applications and concepts of mathematics.

From the students' point of view, 55% of those who used it during the second semester MA101 course felt that Derive was beneficial to their understanding of the curriculum. Interestingly, this group of students did better than the previous year's students on a standard calculus test administered towards the end of the semester. It takes a conscious effort on the instructor's part to bring a CAS into the classroom and, more importantly, to have their students use it outside the classroom. Failure to do so magnifies any misgivings that the student has about using the computer as a learning tool. The solution to this problem is to include Derive in the testing and evaluation phase of the course.

4. CONCLUSIONS: The D/MS has made significant progress integrating CAS in the classroom, but there is still much to be done. The new course, Discrete Dynamical Systems, which now begins our core math program, was designed specifically to employ CAS to the fullest possible extent. The large number of students taking this course, about 1150, provides a large student population to test our new teaching philosophy and the use of CAS in undergraduate mathematics.

Domain Decomposition Methods for Problems with Uniform Local Refinement in Two Dimensions

JAMES H. BRAMBLE, RICHARD E. EWING, ROSSEN R. PARASHKEVOV, AND JOSEPH E. PASCIAK

Abstract. In this talk, we first present a flexible mesh refinement strategy for the approximation of solutions of elliptic boundary value problems in two dimensional domains. Coupled with this approximation scheme, we shall describe preconditioners for the resulting discrete system of algebraic equations. These techniques lead to efficient computational procedures in serial as well as parallel computing environments. The preconditioners are based on overlapping domain decomposition and involve solving (or preconditioning) subproblems on regular subregions. These techniques are analyzed in a forthcoming paper [2]. We present the results of numerical experiments illustrating the preconditioning algorithms.

INTRODUCTION

To provide the required accuracy in many applications involving large scale scientific computation, it becomes necessary to use local mesh refinement techniques. These techniques allow the use of finer meshes in regions of the computational domain where the solution exhibits large gradients. This remains practical only if efficient techniques for the solution of the resulting discrete systems are available. In this talk, we will give a flexible scheme for refinement as well as develop effective iterative methods for the solution of the resulting systems of discrete equations. This was also presented at the Fourth Iternational Symposium on Domain Decomposition Methods, Moscow, USSR, May, 1990. The analysis for the methods discussed in this talk is given in [2].

We shall be interested in techniques for problems with refinements which are not quite local. As an example, one might consider a front passing through a two dimensional domain. In this case, it might be necessary to refine in the neighborhood of the front.

There are a number of ways of developing preconditioned iterative schemes for the discrete systems resulting from local mesh refinement in the literature. Techniques based on nested multilevel spaces are given in [1],[10],[11]. Techniques based on domain decomposition are given in [3],[14],[15]. The analysis presented there implicitly depends on the shape of the the refinement domain, and hence the resulting algorithms may not be as effective with irregularly shaped refinement regions. These algorithms also require the solution of a subproblem or preconditioner on the refinement regions. This talk will provide alternative preconditioned iterative techniques for these problems based on overlapping domain decomposition. Our algorithms are simpler and possibly more effective when implemented

This manuscript has been authored under contract number DE-AC02-76CH00016 with the U.S. Department of Energy. Accordingly, the U.S. Government retains a non-exclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes. This work was also supported in part under the National Science Foundation Grant No. DMS84-05352 and by the U.S. Army Research Office through the Mathematical Science Institute, Cornell University. Additional supporters of this work include the Office of Naval Research under contract No. 0014-88-K-0370 and by the Institute for Scientific Computation at the University of Wyoming though National Science Foundation grant No. RII-8610680.

Typeset by $\mathcal{A}_{\mathcal{M}}\mathcal{S}$-TEX

since they often lead to preconditioning subproblems defined on either regular subregions or topologically 'nice' meshes. The refinement region is the union of the subregions.

The proposed mesh refinement strategy is important in that it provides a basic approach for implementing dynamic local grid refinement. An example of a refinement strategy involves starting with a uniform coarse-grid and refining in small subregions associated with a selected set of coarse-grid vertices. These subregions are allowed to overlap and there are no theoretical restrictions on the resulting refinement region (the union of the subregions). Dynamic refinement is achieved by simply dynamically changing the selected set of coarse-grid vertices.

In addition, the technique can be integrated into existing large scale simulators without a complete redesign of the code. This is because most of the computation involves tasks on either the global coarse grid or the refinement grids associated with the refinement subregions. Choosing the coarse and refinement grid structure to be that already used in the code saves considerable development costs. For example, if one uses regularly structured meshes in the coarse and refinement grids, a substantial part of the resulting algorithm only requires operations on regular grids even though the resulting final approximation space is not regular.

The outline of the remainder of the talk is as follows. In Section 2, we define some preliminaries and describe the second-order elliptic problems which will be considered. The overlapping domain decomposition algorithms for grids with partial refinement is defined in Section 3. The theoretical estimates for the resulting preconditioned systems (from [2]) are also given there. Finally, computational aspects and the results of numerical experiments using these preconditioning techniques are discussed in Section 4.

2. THE ELLIPTIC PROBLEM AND PRELIMINARIES

We shall be concerned with the efficient solution of discrete equations resulting from approximation of second-order elliptic boundary value problems in a polygonal domain Ω contained in two dimensional Euclidean space R^2. We consider the problem of approximating the solution u of

$$\begin{aligned} Lu &= f \qquad \text{in } \Omega, \\ u &= 0 \qquad \text{on } \partial\Omega. \end{aligned} \tag{2.1}$$

Here L is given by

$$Lv = -\sum_{i,j=1}^{2} \frac{\partial}{\partial x_i} a_{ij} \frac{\partial v}{\partial x_j},$$

and $\{a_{ij}(x)\}$ is a uniformly positive definite, bounded, piecewise smooth coefficient matrix on Ω. The corresponding bilinear form is denoted by $A(\cdot,\cdot)$ and is given by

$$A(v,w) = \sum_{i,j=1}^{2} \int_{\Omega} a_{ij} \frac{\partial v}{\partial x_i} \frac{\partial w}{\partial x_j}\, dx, \tag{2.2}$$

and is defined for functions v,$w \in H^1(\Omega)$. Here $H^1(\Omega)$ is the Sobolev space of order one on Ω. We denote the $L^2(\Omega)$ inner product by $(\cdot,\cdot)$. The weak solution u of (2.1) is the

function $u \in H_0^1(\Omega)$ satisfying

$$A(u,\varphi) = (f,\varphi) \qquad \text{for all } \varphi \in H_0^1(\Omega).$$

Here, $H_0^1(\Omega)$ is the subspace of functions in $H^1(\Omega)$ whose traces vanish on $\partial\Omega$.

We consider the above model problem for convenience. Many extensions of the techniques to be presented are possible; for example, one could consider equations with lower-order terms and different boundary conditions.

In this talk, we shall deal with various domains. These domains will always be open.

3. THE OVERLAPPING ALGORITHMS

In this section, we shall define iterative methods for problems with partial refinement based on overlapping domain decomposition. We start with a coarse mesh $\cup\tau_H^i$ consisting of triangles of quasi-uniform size H. The associated finite element space M_0 is defined to be the set of continuous piecewise linear functions on the coarse mesh which vanish on $\partial\Omega$. The interior nodes of this mesh will be denoted $\{x_i\}$, for $i = 1,\ldots,N_c$. The mesh refinement is defined in terms of a number of coarse grid subdomains $\{\Omega_i\}$ for $i = 1,\ldots,K$. By convention, Ω_i is defined to be the interior of the union of the closures of the coarse grid triangles. The refinement regions will also be referred to as "the subdomains." We assume that they have limited overlap in that any point of Ω is contained in at most a fixed number (not depending on H) of the subdomains. We define the domain of refinement Ω^r to be the union of the subdomains, $\Omega^r = \cup_{i=1}^K \Omega_i$. There are no theoretical restrictions concerning the definition of the refinement subregions except that they are defined in terms of the coarse grid triangles and satisfy the overlap property as described above.

We provide two examples of this construction. For both examples, the subregions are associated with coarse grid nodes. For the first example, we define the region associated with a coarse-grid node x_i as the subdomain Ω_i which contains the coarse-grid triangles having x_i as a vertex. For the second example, we consider a mesh which is topologically equivalent to a regular rectangular mesh (see Figure 3.1). In this case, we define Ω_i to be the four quadrilaterals which share the vertex x_i. Some reasons for such a choice will be explained later. In either case, an index set $I \subseteq [1,\ldots,N_c]$ is selected and the domains $\{\Omega_i\}$ with $i \in I$ are used to define the refinement region. By possibly changing the numbering of the coarse grid nodes, we assume, without loss of generality, that $I = [1,2,\ldots,K]$. There are no additional restrictions concerning this set I and hence rather complex refinement regions are possible.

The composite space is defined in terms of a quasi-uniform mesh $\{\tau_h^i\}$ on Ω of size $h < H$ which satisfies

$$\cup_i \partial\tau_H^i \subseteq \cup_i \partial\tau_h^i.$$

The space of continuous piecewise linear functions with respect to this triangulation (which vanish on $\partial\Omega$) will be denoted by $\tilde{M}$. Note that this space is introduced for the construction and analysis of the composite grid space. It is not used in actual computation since it has too many degrees of freedom in Ω/Ω^r. The subspace M_i associated with the subdomain Ω_i is defined by

$$M_i = \{\phi \in \tilde{M} \mid \text{support } \phi \subseteq \Omega_i\}. \tag{3.1}$$

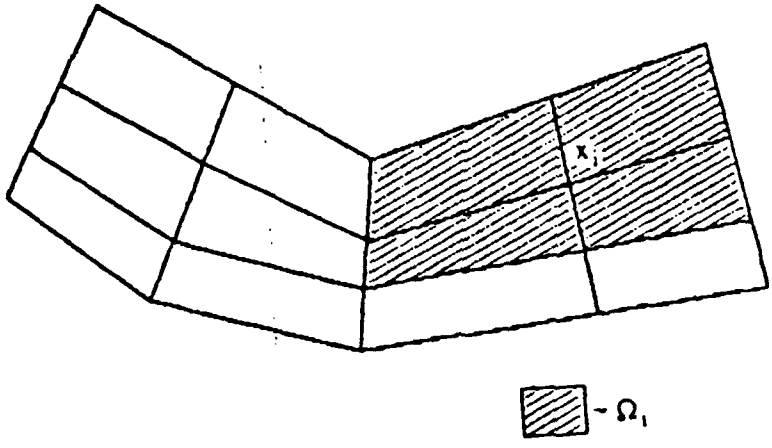


Figure 3.1
A distorted rectangular mesh.

The composite finite element space is then defined to be

$$M = \sum_{i=0}^{K} M_i.$$

Note that the space M provides finer grid approximation in the refinement region Ω^r. An illustrative example of a mesh so generated is given in Figure 3.2. The nodes on the boundary of the refinement region which are not coarse-grid nodes are slave nodes since, by continuity, the values of functions in M on these points are completely determined by their values on neighboring coarse-grid nodes. The operator $A_i : M_i \mapsto M_i$ is defined for $v \in M_i$ by

$$(A_i v, \phi) = A(v, \phi) \qquad \text{for all } \phi \in M_i.$$

Our goal is to efficiently solve the composite grid problem: Given a function $f \in L^2(\Omega)$, find $U \in M$ satisfying

$$A(U, \phi) = (f, \phi) \qquad \text{for all } \phi \in M. \tag{3.2}$$

As above, we define $A : M \mapsto M$ by

$$(Av, \phi) = A(v, \phi) \qquad \text{for all } \phi \in M.$$

Problem (3.2) can then be rewritten as

$$AU = F, \tag{3.3}$$

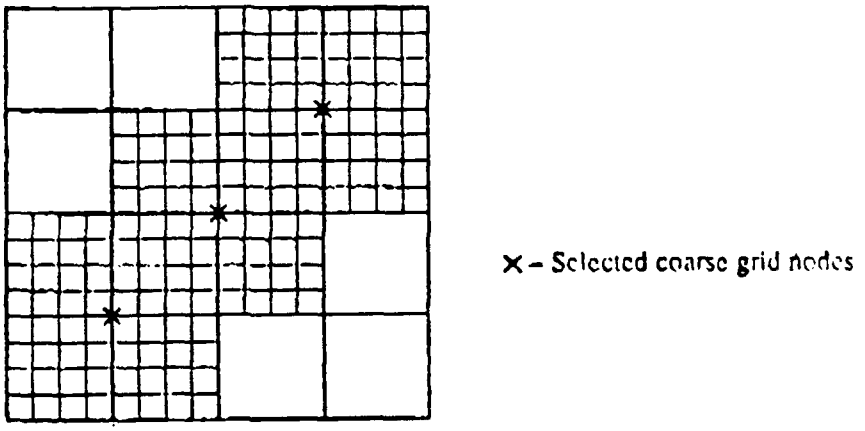


Figure 3.2
A composite grid.

for appropriate $F \in M$. We will develop preconditioners for (3.3) by using overlapping domain decomposition.

There are basically two classes of these preconditioners, the additive and multiplicative. The additive version defines the preconditioner B_a for A of (3.3) by

$$B_a = \sum_{i=0}^{K} R_i Q_i.$$

Here, Q_i denotes the $L^2(\Omega)$ projection operator onto M_i and R_i is a symmetric positive definite operator on M_i. Explicit choices for R_i will be discussed later; however, we note that it suffices to take R_i to be a preconditioner for A_i.

The multiplicative version is defined by applying the R_i consecutively. The multiplicative preconditioner B_m applied to a function $W \in M$ is defined as follows:

(1) Set $Y_0 = 0$.

(2) For $i = 1, \ldots, K+1$, define Y_i by

$$Y_i = Y_{i-1} + R_{i-1} Q_{i-1} (W - A Y_{i-1}). \tag{3.4}$$

(3) For $i = K+2, \ldots, 2K+2$, define Y_i by

$$Y_i = Y_{i-1} + R_{2K+2-i} Q_{2K+2-i} (W - A Y_{i-1}). \tag{3.5}$$

(4) Set $B_m W = Y_{2K+2}$.

It is not difficult to see that B_m is a symmetric linear operator on M.

The operators B_a and B_m defined above will be effective as preconditioners A if they satisfy the following:

(1) They are relatively inexpensive to evaluate.
(2) They lead to well conditioned linear systems.

The first criterion involves implementation issues and will be discussed later in more detail. The second criterion requires that the condition numbers $K(B_aA)$ and $K(B_mA)$ be small. In the case of the additive algorithms, this is equivalent to the existence of positive constants c_0, c_1 satisfying

$$c_0A(v,v) \leq A(B_aAv,v) \leq c_1A(v,v) \qquad \text{for all } v \in M, \tag{3.6}$$

with c_1/c_0 small. A similar statement holds for the product algorithm.

The analysis presented in [2] requires the following hypotheses. It is assumed that there are positive constants C_0 and ω which do not depend on h, H or the subdomains and satisfy

$$C_0A(w,w) \leq A(R_iA_iw,w) \leq \omega A(w,w) \qquad \text{for all } w \in M_i. \tag{3.7}$$

This means that the operators R_i are spectrally good preconditioners for A_i. For the product algorithm, we also assume that $0 < \omega < 2$. The following theorem is proved in [2].

THEOREM 3.1. *Assume that there are no isolated points on the boundary of Ω^r. Then the condition numbers $K(B_aA)$ and $K(B_mA)$ remain bounded independently of h, H and the choice of subdomains $\{\Omega_i\}$.*

REMARK 3.1: The analysis given in [2] uses techniques from both the theory of overlapping domain decomposition [12],[13] as well as the standard domain decomposition theory [5]-[8] to provide the result for the additive algorithms. The result for the multiplicative version follows from that for the additive and the application of a general theory given in [9].

REMARK 3.2: The hypothesis concerning isolated points on the boundary of Ω^r is included to provide a uniform estimate for the preconditioned systems. If $\partial\Omega^r$ contains isolated points then it is possible to show (cf. Remark 4.2 of [2]) that the condition number grows at most on the order of $\ln^2(H/h)$. This sort of decay is actually seen in the last numerical example in Section 6.

REMARK 3.3: There is very little restriction concerning the way that the domains Ω_i are defined. Note that if only one refinement domain is used, then Theorem 3.1 provides a result for the imbedded space case proposed in [3]. Alternatively, one can consider the case where Ω^r is all of Ω and hence $M = \tilde{M}$. In this case, Theorem 3.1 guarantees uniform bounds for the condition numbers without putting restrictions on the shapes of the subdomains $\{\Omega_i\}$. Thus, for example, the subdomains can be taken to be strips as long as the coarse problem is included.

4. Computational aspects and Numerical Examples

In this section, we discuss some of the computational properties associated with the method. In particular, we shall consider its feasibility for use in dynamic refinement strategies. We shall also see that with this type of method, it is possible to develop highly vectorizable and parallelizable code. Finally, we provide the results of numerical examples illustrating the condition numbers for the preconditioned systems described earlier.

We consider the earlier discussed examples where the domain of refinement is defined by simply selecting coarse-grid nodes and a rule for defining the refinement region associated with a coarse node. Specifically, we consider the example where the coarse mesh is defined from quadrilaterals and the refinement region associated with a coarse-grid vertex is defined to be the four quadrilaterals which share the vertex. An easy way to implement this refinement involves using vectors of unknowns with some redundancy. Associated with each quadrilateral, we associate a vector which contains the fine-grid unknowns in the quadrilateral and its boundary. The program is designed to operate on a data structure which contains a coarse-grid vector and a list of fine-grid vectors corresponding to the quadrilaterals appearing in the refinement regions. This process is controlled by a list of pointers which connect the location of quadrilateral fine-grid vectors in this data structure to the coarse grid node refinement regions in which they appear. A simple control structure is also developed to handle the redundancy in the data vectors. These control structures can be easily derived from the list of coarse-grid refinement nodes and the coarse mesh geometry. Thus, a dynamic change in the refinement region only requires a simple (and of negligible cost) computation of some control pointers associated with the coarse grid.

An advantage of the proposed approach is that it can be used to invoke refinement without the use of the general data structures associated with meshes which are not regular. One assigns a regular mesh topology to the coarse mesh and to the meshes in the refinement subregions. This means that even though the composite mesh is highly irregular, all of the problems (on M_i, $i \in I_0$) which need to be solved or preconditioned will be on regular rectangular meshes. Similarly, it is possible to decompose the evaluation of the composite grid operator into pieces which involve operator evaluation on the topologically rectangular mesh parts. For these topologically rectangular meshes, highly efficient modules for preconditioning and operator evaluation are available for both vector and parallel computing architectures.

We shall consider the model problem

$$\begin{aligned} -\Delta u &= f \qquad \text{in } \Omega, \\ u &= 0 \qquad \text{on } \partial\Omega, \end{aligned} \tag{4.1}$$

where Δ denote the Laplacian and Ω is the unit square $[0,1] \times [0,1]$. To define the coarse mesh, the domain Ω is first partitioned into $m \times m$ square subdomains of side length $H = 1/m$. Each smaller square is then divided into two triangles by one of the diagonals (e.g. the diagonal which goes from the bottom left to the upper right hand corner of the square). The coarse-grid approximation space M_0 is defined to be the set of functions which are continuous on Ω, are piecewise linear with respect to the triangulation, and vanish on $\partial\Omega$. The space $\tilde{M}$ is defined from a similar finer mesh of size $h = H/l$ for some integer $l > 1$.

For our first two examples, we consider an application where it is required to refine along the diagonal connecting the origin with the point $(1,1)$. Such a refinement might be necessary if the function f has large gradients near this diagonal but is well behaved in the remainder of Ω. Accordingly, we select the coarse-grid nodes on the diagonal for refinement. We define the refinement region associated with a refinement node to be the four coarse mesh squares which have that node as a corner. Note that the refinement region is highly irregular even though the coarse problem and the refinement subproblems involve regular rectangular meshes.

We will illustrate the rate of convergence of preconditioned algorithms for solving (3.2) where $A(\cdot,\cdot)$ is given by the Dirichlet form. To do this, we shall numerically compute the largest and smallest eigenvalue (λ_1 and λ_0 respectively) of the preconditioned operator $B_a A$. As is well known, the rate of convergence of the resulting preconditioned algorithms can be bounded in terms of the condition number $K(B_a A) = \lambda_1/\lambda_0$. We shall not report results for preconditioning with the product operator B_m, although our previous experience [9] suggests that the product version will converge somewhat faster than the additive.

Table 4.1 gives the largest and smallest eigenvalue and the condition number of the system $B_a A$ as a function of h. In this example, we took $R_i = A_i^{-1}$; i.e., we solved exactly on the subspaces $\{M_i\}$. For Table 4.1, $m = 4$ and there are three refinement subdomains $(0,1/2) \times (0,1/2)$, $(1/4,3/4) \times (1/4,3/4)$, and $(1/2,1) \times (1/2,1)$. Note that both the upper and lower eigenvalues appear to be tending to a limit as the ratio $h/H \to 0$. Similar behavior is seen in Table 4.2, which corresponds to $m = 8$ and uses seven smaller refinement subregions.

Table 4.1
Condition numbers for 3 overlapping subregions

h	λ_1	λ_0	$K(B_a A)$
1/8	2.44	0.50	4.9
1/16	2.50	0.41	6.1
1/32	2.51	0.38	6.6
1/64	2.52	0.36	6.9
1/128	2.52	0.35	7.1

In almost all realistic applications, the direct solution of subproblems is much more expensive than the evaluation of a suitable preconditioner. To illustrate the effect on the convergence rate of the preconditioned iteration, we next consider the previous example but with the direct solves on the subspaces replaced by multigrid preconditioners. Specifically, we employ the V-cycle multigrid algorithm (cf. [4]) using one pre- and post-smoothing Jacobi iteration on each grid level. This leads to a preconditioning operator $R_i : M_i \to M_i$ which satisfies

$$0.4A(v,v) \le A(R_i A_i v, v) \le A(v,v) \qquad \text{for all } v \in M_i. \tag{4.2}$$

Table 4.2
Condition numbers for 7 overlapping subregions

h	λ_1	λ_0	$K(B_a A)$
1/16	2.46	0.47	5.2
1/32	2.52	0.39	6.5
1/64	2.54	0.35	7.2
1/128	2.54	0.34	7.5

The constant 0.4 above was computed numerically and holds for all of the subspace problems which are required for this application, including M_0.

Tables 4.3 and 4.4 provide the eigenvalues and condition numbers for the above examples when direct solves were replaced by multigrid preconditioners. Note that in all of the reported runs, the condition number with multigrid preconditioners was at most 5/4 times as large as that corresponding to exact solves. Such an increase in condition number is negligible in a preconditioned iteration. In contrast, the computational time required for the multigrid sweep is considerably less that that needed for a direct solve (especially in more general problems with variable coefficients).

Table 4.3
Preconditioned subproblems, 9 overlapping subregions

h	λ_1	λ_0	$K(B_a A)$
1/8	2.37	0.53	4.5
1/16	2.12	0.33	6.4
1/32	2.07	0.27	7.6
1/64	2.04	0.25	8.2
1/128	2.02	0.24	8.4

Table 4.4
Preconditioned subproblems, 7 overlapping subregions

h	λ_1	λ_0	$K(B_a A)$
1/16	2.36	0.40	5.9
1/32	2.11	0.28	7.5
1/64	2.06	0.24	8.8
1/128	2.03	0.22	9.4

As a final example, we consider a case where the isolated point hypothesis of Theorem 3.1 is not satisfied. Specifically, we consider a coarse mesh of size $H = 1/4$ and select the four nodes with (x, y) values (1/4,1/2), (3/4,1/2), (1/2,1/4), and (1/2,3/4). The refinement region is everything but the subsquares $[0, 1/4] \times [0, 1/4]$, $[0, 1/4] \times [3/4, 1]$, $[3/4, 1] \times [0, 1/4]$, and $[3/4, 1] \times [3/4, 1]$. Note that, to satisfy the hypotheses of the theorem, it would be necessary to include a refinement region centered at the coarse-grid node $(1/2, 1/2)$. Table 4.5 gives the smallest eigenvalue for the operator $B_a A$ as a function of h. The function $(.32 + .36 \log_2(h^{-1}))^{-2}$ is also provided for comparison. These results suggest that smallest eigenvalue λ_0 decays as predicted by the theoretical bound $C/\ln(H/h)^2$ (see Remark 3.2).

Table 4.5
A "bad" example in two dimensions.

h	λ_0	$(.32 + .36 \log_2(h^{-1}))^{-2}$
1/8	.50	.51
1/16	.32	.32
1/32	.22	.22
1/64	.16	.16
1/128	.12	.12

REFERENCES

1. R.E. Bank, T. Dupont, and H. Yserantant, *The hierarchical basis multigrid method*, Num. Math. **52** (1988), 427–458.
2. J.H. Bramble, R.E. Ewing, R.R. Parashkevov, and J.E. Pasciak, *Domain decomposition methods for problems with partial refinement*, Proceedings of the Copper Mountain Meeting on Iterative Methods, April, 1990 (submitted).
3. J.H. Bramble, R.E. Ewing, J.E. Pasciak and A.H. Schatz, *A preconditioning technique for the efficient solution of problems with local grid refinement*, Comp. Meth. Appl. Mech. Eng. **67** (1988), 149–159.
4. J.H. Bramble and J.E. Pasciak, *New convergence estimates for multigrid algorithms*, Math. Comp. **49** (1987), 311–329.
5. J.H. Bramble, J.E. Pasciak and A.H. Schatz, *The construction of preconditioners for elliptic problems by substructuring, I*, Math. Comp. **47** (1986), 103–134.
6. J.H. Bramble, J.E. Pasciak and A.H. Schatz, *The construction of preconditioners for elliptic problems by substructuring, II*, Math. Comp. **49** (1987), 1–16.
7. J.H. Bramble, J.E. Pasciak and A.H. Schatz, *The construction of preconditioners for elliptic problems by substructuring, III*, Math. Comp. **51** (1988), 415–430.
8. J.H. Bramble, J.E. Pasciak and A.H. Schatz, *The construction of preconditioners for elliptic problems by substructuring, IV*, Math. Comp. **53** (1989), 1–24.
9. J.H. Bramble, J.E. Pasciak, J. Wang, and J. Xu, *Convergence estimates for product iterative methods with applications to domain decomposition and multigrid*, (preprint).
10. J.H. Bramble, J.E. Pasciak, J. Wang, and J. Xu, *Multigrid results which do not depend upon elliptic regularity assumptions*, (in preparation).
11. J.H. Bramble, J.E. Pasciak and J. Xu, *Parallel multilevel preconditioners*, Math. Comp., (in print).
12. M. Dryja and O. Widlund, *An additive variant of the Schwarz alternating method for the case of many subregions*, Technical Report, Courant Institute of Mathematical Sciences **339** (1987).

13. P.L. Lions, *On the Schwarz alternating method*, In the Proceedings of the First International Symposium on Domain Decomposition Methods for Partial Differential Equations, R. Glowinski, G.H. Golub, G.A. Meurant, and J. Periaux (eds.), 1987.
14. S. McCormick and J. Thomas, *The fast adaptive composite grid (FAC) method for elliptic equations*, Math. Comp. **46** (1986), 439–456.
15. O. Widlund, *Optimal iterative refinement methods*, Technical Report, Courant Institute of Mathematical Sciences **391** (1988).

Keywords. second-order elliptic equation, domain decomposition, overlapping domain decomposition, local mesh refinement
1980 *Mathematics subject classifications*: 65N30, 65F10

Department of Mathematics
Cornell University
Ithaca, NY 14853

Mathematics Department
The University of Wyoming
Laramie, WY 82071

Mathematics Department
The University of Wyoming
Laramie, WY 82071

Department of Applied Science
Brookhaven National Laboratory
Upton, NY 11973

APPLICATIONS OF ALGEBRAIC LOGIC TO RECURSIVE QUERY OPTIMIZATION

PAUL BROOME
U.S. ARMY BALLISTIC RESEARCH LABORATORY
ABERDEEN PROVING GROUND, MD
21005-5066
(*BROOME@BRL.MIL*)

Abstract. The book by Tarski and Givant, *A Formalization of Set Theory Without Variables*, describes a powerful system of equational logic based on proper binary relations. This simple system is rich enough to serve either as a specification language for describing interacting systems, a concise and flexible query language, or a notation for program synthesis.

We consider the problem of program synthesis and introduce an operator to specify single linear recursions. Equations about such recursions are chained together to give a method of query transformation that collapses loops by rewriting. The equations hold under all interpretations and are appropriate support for a distributed database in which each node has its own snapshot of the data. These results are part of an effort to find a mechanism for quickly expressing and efficiently solving complex logical queries.

1. Introduction. One of the most attractive program de' 'lopment methods is transformational programming. This report describes the synthesis : .d transformation of logical queries. The assumptions are that

1. Queries are to be initially written as straightforward, high level specifications. More efficient, but probably less readable, queries are derived through correctness preserving transformations.
2. Functional and relational operators provide an appropriate context for specifications. They have the algebraic properties required for the transformations.

The central contribution is the application of the relation algebras and Q-systems of Tarski[1]to query optimization. The notation is attractive, as it is a convenient combination of logic programming and functional programming. This middle ground has the expressibility of the former and the manipulability of the latter. Specifically, a key contribution is an operator for linear recursions over regularly structured objects along with equations for merging loops and propagating constraints. The objects are terms of a single binary function symbol.

2. Motivation. Relational queries are typically straightforward, declarative expressions. The nonprocedural quality of such queries is intuitive and leads to easy optimization. However, query languages are limited in their ability to compute results. The usual repair is to combine the query language with a procedural language resulting in a large increase in complexity and error prone programming.

Because of their declarative nature, relational systems are appropriate for quickly and naively formulating queries. This is important for time critical activities in which machine efficiency is secondary to overall completion time. On the other hand, the limited power of relational query languages has meant an inability to handle unpredictable situations or plan for the unexpected. In other words, all the planning must have been done in the laboratory instead of in the field.

In order to allow the customer some flexible planning ability, such a system needs some extensibility and much safety. This extensibility is independent of concerns about space and time requirements or accounts of storage allocation and reclamation. Freedom of control over storage is a primary characteristic of declarative programming. Declarative programming is a problem oriented style of programming.

Finally, software is often developed in an ad hoc fashion. The design of computational

Symbol	Meaning
$A - Z$	variables
$a - z$	constants or function symbols
$\mathcal{T}$	set of terms
$\mathcal{S}$	set of ground terms
$\mathcal{B}(\mathcal{S})$	set of binary relations on $\mathcal{S}$
$\wedge$	logical and
$\vee$	logical or
$\neg$	logical negation
$\exists$	existential quantifier
$\forall$	universal quantifier
$\leftrightarrow$	if and only if
$\equiv$	equivalent to
$\stackrel{def}{\equiv}$	defined as
$\langle \cdot \mid \cdot \rangle$	ordered couple

TABLE 1
Summary of notation

tools usually begins with the hardware, followed by the construction of a generally useful collection of operations, ending with (often inadequate) attempts at optimization. The alternative approach taken here is to consider optimization first, that is, to choose the notation so that optimization is easy. This suggests that program operations will have many algebraic properties.

3. Preliminaries. The notation is to be interpreted both mathematically and as program constructs with the clear understanding that the two are distinct. Within definitions and arguments for correctness we follow an extended predicate logic $\mathcal{L}^+$ [1]. This system has two important levels: a logic for program specification and an algebra for program optimization.

Every program is specified as a relation between inputs and outputs and includes subprograms specifying predicates on pairs, or as generators of pairs. Program forming operations are operations on relations and optimization rules are equations between relations. Control is assumed to proceed from left to right, top to bottom. Efficiency of query solution is heavily dependent on both data and control. Some operations, such as negations, are delayed until arguments are available. Transformations collapse redundant computations or reorder to avoid delaying. The objective is to schedule relations to uniquely and efficiently generate answers.

The theory of relations is one of the most thoroughly developed branches of logic[1-3]. The algebraic terminology categorizes a portion of the large collection of relation equations. It also helps describe algorithms abstractly, independently of a model. The most specific structures are relation algebras and Q-relation algebras as described in [1]. Table 1 is a summary of notation.

In particular, the calculus of binary relations follows the well understood laws of Boolean algebra. A Boolean algebra is defined as a structure $\langle \mathcal{U}, +, \bar{} \rangle$ satisfying a set of equations as axioms. Although there are no implicit assumptions about the underlying universe $\mathcal{U}$, we are most interested in Boolean algebras defined on $\mathcal{B}(\mathcal{S})$, the set of binary relations between

Relation Symbol	Meaning
$\mathbf{0}$	empty relation
$\mathbf{1}$	universal relation
$R+S$	sum of relations
$R \cdot S$	product of relations
$R \square S$	ordered coupler of relations
$R \odot S$	relative product or composition
$\overline{R}$	complement of the relation R
$R^{\smile}$	converse of a relation
R^{*}	arbitrary exponentiation of R
$[R\|S]$	relational constructor of couples
$pi(D,S,C)$	linear recursion
id	identity relation, (also written $\mathring{\mathbf{1}}$)
di	diversity relation, $\mathring{\mathbf{0}}$
hd	first projection, a
tl	second projection, b

TABLE 2
Summary of Relations and Relation Operators

terms. The following is a sample axiomatization of a Boolean Algebra.

$$(X+Y)+Z \equiv X+(Y+Z), \tag{1}$$

$$X+Y \equiv Y+X, \tag{2}$$

$$X \equiv \overline{(\overline{X}+Y)}+\overline{\overline{X}+\overline{Y}}. \tag{3}$$

A relation algebra (an RA) is a structure $\langle \mathcal{U}, +, \overline{\ }, \odot, {}^{\smile}, id \rangle$. The common language describes these operations as sum, complementation, composition, converse, and the identity. A summary of the relation operators appears in Table 2. The following is an axiomatization for an RA [1,4].

$$(F+G)+H \equiv F+(G+H), \tag{4}$$

$$F+G \equiv G+F, \tag{5}$$

$$F \equiv \overline{(\overline{F}+G)}+\overline{(\overline{F}+\overline{G})}, \tag{6}$$

$$F \odot (G \odot H) \equiv (F \odot G) \odot H, \tag{7}$$

$$(F+G) \odot H \equiv F \odot H + G \odot H, \tag{8}$$

$$F \odot id \equiv F, \tag{9}$$

$$F^{\smile\smile} \equiv F, \tag{10}$$

$$(F+G)^{\smile} \equiv F^{\smile}+G^{\smile}, \tag{11}$$

$$(F \odot G)^{\smile} \equiv G^{\smile} \odot F^{\smile}, \tag{12}$$

$$F^{\smile} \odot \overline{F \odot G}+\overline{G} \equiv \overline{G}. \tag{13}$$

From these fundamental operations we can abstractly define other relations and relation operators. Positive, logical definitions of product $F \cdot G$ and the universal relation $\mathbf{1}$ are

preferred. They are more easily implemented and constructive proofs are more intuitive. Implementation considerations for negations such as complement and diversity *di* are discussed in the next section. The following are further definitions in terms of the fundamental operations.

$$F \cdot G \stackrel{def}{\equiv} \overline{\overline{F} + \overline{G}}, \tag{14}$$

$$\mathbf{1} \stackrel{def}{\equiv} id + \overline{id}, \tag{15}$$

$$\mathbf{0} \stackrel{def}{\equiv} \overline{\mathbf{1}}, \tag{16}$$

$$di \stackrel{def}{\equiv} \overline{id}. \tag{17}$$

It is easy, but tedious, to check that the following logical definitions for these relation operators satisfy the properties of an RA.

$$\forall XY\{X\ (F \cdot G)\ Y \leftrightarrow X\ F\ Y \wedge X\ G\ Y\}, \tag{18}$$

$$\forall XY\{X\ (F + G)\ Y \leftrightarrow X\ F\ Y \vee X\ G\ Y\}, \tag{19}$$

$$\forall XY\{X\ (F \odot G)\ Y \leftrightarrow \exists Z\{X\ F\ Z \wedge Z\ G\ Y\}\}, \tag{20}$$

$$\forall XY\{X\ F^{\smile}\ Y \leftrightarrow Y\ F\ X\}, \tag{21}$$

$$\forall X\{X\ id\ X\}, \tag{22}$$

$$\forall XY\{X\ \mathbf{1}\ Y\}, \tag{23}$$

$$\forall XY\{X\ \overline{F}\ Y \leftrightarrow \neg(X\ F\ Y\}, \tag{24}$$

$$\forall XY\{X\ di\ Y \leftrightarrow \neg(X\ id\ Y\}. \tag{25}$$

These operators satisfy many equations. For example, some simple properties of sum and product are

$$A + \mathbf{0} \equiv A, \tag{26}$$

$$\mathbf{0} + A \equiv A, \tag{27}$$

$$(A \cdot B) \cdot C \equiv A \cdot (B \cdot C), \tag{28}$$

$$A \cdot \mathbf{1} \equiv A, \tag{29}$$

$$\mathbf{1} \cdot A \equiv A. \tag{30}$$

For any given $RA = \langle U, \cup, \cap, \overline{\ }, \mathbf{0}, \mathbf{1}\rangle$, two elements $\mathbf{a}, \mathbf{b}$ are called *quasiprojections* if $\mathbf{a}^{\smile} \odot \mathbf{a} \subseteq id$, $\mathbf{b}^{\smile} \odot \mathbf{b} \subseteq id$, and $\mathbf{a}^{\smile} \odot \mathbf{b} \equiv \mathbf{1}$. An RA is called a *Q-relation algebra* if its operators include quasiprojections.

These definitions say nothing about the intended realization, and their abstractness is what makes them appropriate for describing broadly applicable program operations. In [1] the intended interpretation is encapsulated into a membership relation $\mathbf{E}$. The laws of an RA and a Q-relation algebra hold for any definition of $\mathbf{E}$, and $\mathbf{E}$ connects the relation algebras with a particular universe over which the relations are to range. In an application, the query language is the set of abstract operations and the database is $\mathbf{E}$.

In particular, a Q-relation algebra defined over a nontrivial universe contains ordered pairs or couples and the quasiprojections suggest that there are operations for selecting components of these couples. Two such selectors over ordered couples called *conjugated*

projections (or just projections) are defined and represented by *hd* and *tl*.

(31) $$\forall XY\{\langle X|Y\rangle\ hd\ X\},$$

(32) $$\forall XY\{\langle X|Y\rangle\ tl\ Y\}.$$

Note that they satisfy $hd^{\smile} \odot hd \equiv id, tl^{\smile} \odot tl \equiv id$, and $hd^{\smile} \odot tl \equiv \mathbf{1}$. Thus they qualify as quasiprojections. These projections select components of ordered couples. The following operators construct or perform related functions on couples.

(33) $$[F|G] \stackrel{def}{\equiv} (F \odot hd^{\smile}) \cdot (G \odot tl^{\smile}),$$

(34) $$F \square G \stackrel{def}{\equiv} [hd \odot F | tl \odot G].$$

The following properties hold for construction and coupling operators. They all have similar proofs.

(35) $$[F + G|H] \equiv [F|H] + [G|H],$$

(36) $$[F|G + H] \equiv [F|G] + [F|H],$$

(37) $$[F|G] \cdot [H|K] \equiv [(F \cdot H)|(G \cdot K)],$$

(38) $$(F \square G) \odot (H \square K) \equiv (F \odot H) \square (G \odot K),$$

(39) $$(F \square (G \odot W)) \odot (R \square S) \equiv (F \square G) \odot (R \square (W \odot S)),$$

(40) $$(F \odot W \square G) \odot (R \square S) \equiv (F \square G) \odot (W \odot R \square S),$$

(41) $$(F \square G)^{\smile} \equiv (F^{\smile} \square G^{\smile}),$$

(42) $$(F + G) \square H \equiv (F \square H + G \square H),$$

(43) $$F \square (G + H) \equiv (F \square G + F \square H),$$

(44) $$(F \square G) \cdot (H \square K) \equiv (F \cdot H) \square (G \cdot K).$$

These prelimaries from the theory of relations have been studied elsewhere. Closely related are the FP systems of Backus [5] as they also depend upon operators, particularly composition, to form more complex programs out of simpler ones. Berghammer and Zierer [6] have given a relation algebra semantics for FP-like languages. Mili, Desharnais, and Mili [7] have given heuristics for the design of deterministic programs from relational equations. This paper describes a transformation system based on the operators, relations, and equations of a Q-relation algebra.

4. A facility for definitions. The logical definitions of $+, \cdot, \odot, {}^{\smile}, \overline{\ }, hd, tl, \square, [\ |\ ], id$, and $\mathbf{1}$ (18-25) suggest an implementation with an SLDCNF-resolution[8]logic programming system in the logic of predicates of three variables. For example, the goal $\exists X, Y(X\ R\ Y)$ would be represented as $: - p(X, R, Y)$. Goals are solved in left-to-right, top-to-bottom order. SLDCNF is briefly discussed in the section on negations.

We insist that no definition introduce variables into the relation argument. Thus every literal in the body of a program clause of the form $p(X, R, Y)$ will contain only variables R that were named in the clause head. Therefore if a goal is $p(X, R, Y)$ where R is ground, then every relation argument in every subgoal will be ground.

The expression $R \stackrel{def}{\equiv} S$ is to be read as 'R is equivalent to, but should be rewritten as, S.' But this is represented in a logic program as the clause $p(X, R, Y) : - p(X, S, Y)$ which is the only clause defining R. Conjunctions are solved from left to right. The intent is to match control with the order in which variables are bound. Therefore we partition the arguments of a relation into two structured components, an 'input' and an 'output.' Predicates of single arguments are conceptually extended to two arguments, both of which are the same value.

4.1. Negations. The identity X *id* Y unifies terms X and Y. The diversity relation depends on an antiunification algorithm[9]and is denoted A *di* B. If A and B are both ground constants then A *di* B will fail or succeed depending on whether they are the same or different constants. On the other hand, if they contain unbound variables then they are delayed until arguments are known or output as answers if they are never known.

Terms with structure are solved recursively in a manner similar to unification. If the two terms A and B have either different principal functors or different numbers of arguments then antiunification succeeds without new inequality constraints. On the other hand, with the same principal functor and, say N, arguments, diversity in any argument is enough for antiunification to succeed. Thus possibly N new choice points are created by recursing the algorithm on these arguments. If one of the corresponding pairs of arguments can be determined to be different then no other alternatives need be considered as the algorithm terminates successfully.

For example, consider the problem $f(X, g(a))$ *di* $f(u, g(Y))$. The two solutions (inequality constraints) are $X \neq u$ and $Y \neq a$. On the other hand, the subgoal $f(X, a)$ *di* $f(Y, b)$ succeeds with no new inequality constraints.

Negation as finite failure cannot compute the complement of a relation. The goal $\exists XY, X\ \overline{R}\ Y$ is equivalent to $\exists XY, \neg X\ R\ Y$ but negation by failure instead determines $\neg(\exists XY, X\ R\ Y)$. Although negation as finite failure has the usual logical interpretation for ground goals [10], in the face of transformation we cannot be assured when a variable will be ground.

"Constructive negation" is an extension to negation by failure [8] The elements of $\overline{R}$ are not necessarily constructed, but instead the resolution procedure is extended with inequality constraints which are delayed until the arguments are known. If arguments are never known then these constraints are returned as answers. Thus, *solutions to nonground, negative* subgoals are constructed as a set of equations and inequations. Constructive negation has both a clean semantics and the advantage of speeding up some computations although a set of solutions must be finite to be complemented.

4.2. Sequences. A convenient way of sharing an input to more than one function is with Backus' constructor functional[5]. A similar constructor operation on relations can be defined. The ordered coupling operator creates an ordered couple as an output from a pair of inputs in a one to one fashion. Both operations are strict. While complex structures are made with constructors, they are taken apart with *hd* and *tl*. These are called *selectors* by Backus. Selectors disassemble what the constructors and ordered couplers build.

There is a constant available to represent end of lists. Since terms are finite then *hd* cannot always succeed. This implies an indivisiable constant. The symbol $\langle\rangle$ will represent this constant. A sequence of one element A will be represented as $\langle A\rangle$, that is $\langle A|B\rangle$ where $B = \langle\rangle$. The function *null* tests sequences for emptiness and is an identity on $\langle\rangle$. That is, *null* is just the single pair $\langle\rangle, \langle\rangle$.

An associated relation to deposit this symbol into a list construction can be defined. This empty sequencer is a constant function that ignores other domain elements, returning the object $\langle\rangle$. This function is indicated as $[\,]$. It is defined as $[\,] \stackrel{def}{\equiv} \mathbf{1} \odot null$. The following are some simple properties of *null* and $[\,]$.

$$[A|B] \odot null \equiv \mathbf{0}, \tag{45}$$

$$(A \square B) \odot null \equiv \mathbf{0}, \tag{46}$$

$$null \odot (A \square B) \equiv \mathbf{0}, \tag{47}$$

(48) $$[\,] \odot null \equiv [\,],$$

(49) $$null \odot [A|B] \equiv [null \odot A | null \odot B].$$

5. Linear recursion. The *exponentiation* or *closure* of a relation R is R repeatedly composed with itself and is defined as $R^* \equiv id + R \odot R^*$. As an illustration consider finding the greatest common divisor of two natural numbers using *subtract* as defined only on that set. Thus $\langle 3|5\rangle$ *subtract* Y would fail but $\langle 5|3\rangle$ *subtract* 2 succeeds.

EXAMPLE 1 *Greatest common divisor*

$$gcd \stackrel{def}{\equiv} ((id + [tl|hd]) \odot [subtract|tl])^* \odot (hd \cdot tl).$$

A simple program can be envisioned to start with two numbers and continually subtract the second number from the first until the numbers are the same. If necessary, it reorders the numbers so that the largest is first. A trace of the computation on the couple $\langle 6|18\rangle$ is $\langle 6|18\rangle \Rightarrow \langle 18|6\rangle \Rightarrow \langle 12|6\rangle \Rightarrow \langle 6|6\rangle \Rightarrow 6$. Simple exponentiation is not expressive enough and is naturally oriented to elements without structure. Instead a linear recursion operator is defined to extend the effect of the ordered coupler to lists and list–like structures. This operator divides the structure into a couple with D, solves for base cases with S, then combines the elements of a couple with C.

(50) $$pi(D, S, C) \stackrel{def}{\equiv} S + D \odot (id \square pi(D, S, C)) \odot C.$$

Arbitrary exponentiation can be included with pi as $R^* \equiv pi([1|R], id, tl)$. Also, we can define *map* to apply the effect of a relation to each item of a sequence. Thus for example, the goal $\langle 2, 3\rangle$ $map(id + sub1)$ Y has four solutions for Y. The solutions would be $Y = \{\langle 2,3\rangle, \langle 2,2\rangle, \langle 1,3\rangle, \langle 1,2\rangle\}$. The definition is

(51) $$map(R) \stackrel{def}{\equiv} pi(id, null, R \square id).$$

In our relations we collect inputs into a single structured input. For example, the Prolog goal $append(A, B, C)$ is written as $\langle A, B\rangle$ *append* C. On the other hand, predicates of a single argument become two argument, subrelations of the identity. An example is 3 *odd* 3. These extensions are important because they allow us to use higher order operators with Boolean valued relations to define new predicate operators.

All recursions that follow will be given in terms of pi. They are preliminary to the example transformations. The next definition is the function that cumulatively concatenates a sequence of sequences.

(52) $$conc \stackrel{def}{\equiv} pi(id, null, append).$$

(53) $$append \stackrel{def}{\equiv} pi([hd \odot hd | tl \square id], (null \square id) \odot tl, id).$$

An item can be coupled to each element of a sequence with either *distr* or *distl* as in [5]. These functions can be simply defined in terms of pi and the definitions expose their inherent symmetry.

(54) $$distr \stackrel{def}{\equiv} pi([hd \square id | tl \square id], hd \odot null, id).$$

(55) $$distl \stackrel{def}{\equiv} pi([id \square hd | id \square tl], tl \odot null, id).$$

To see what is happening here consider, for example, the function *distr*. It inputs an ordered couple, a sequence and an item. The result is a sequence of couples each of which has the given item as a second component. Schematically this is

$$\langle\langle A_1, \cdots, A_N\rangle, C\rangle \; distr \; \langle\langle A_1, C\rangle, \langle A_2, C\rangle, \cdots, \langle A_N, C\rangle\rangle.$$

Many relations can now be concisely defined in a single expression. For example, we can define *member* on a couple, an element and a sequence. Thus *member* is simply a predicate (an identity) that tests the element for membership in the sequence. For example both $(\langle 3, \langle 2,3,4\rangle\rangle \; member \; \langle 3, \langle 2,3,4\rangle\rangle)$ and $(\langle 3, \langle 5\rangle\rangle \; nonmember \; \langle 3, \langle 5\rangle\rangle)$ are true.

(56) $$select \stackrel{def}{\equiv} pi(id, hd, tl).$$

(57) $$member \stackrel{def}{\equiv} [hd \cdot (tl \odot select)|tl].$$

(58) $$nonmember \stackrel{def}{\equiv} [hd|distl \odot map((hd \odot di) \cdot tl)].$$

The following equations characterize not only some forms of loop merging but can also propagate constraints. Since control is left to right, constraints on search are best performed as soon as possible. The following equation enables further propagation of constraints.

PROPOSITION 1 Merging recursions. If $(id\Box id) \odot C_1 \odot S_2 \equiv \mathbf{0} \wedge S_1 \odot D_2 \odot (id\Box id) \equiv \mathbf{0} \wedge (C_1 \odot D_2 \equiv id\Box id \vee C_1 \odot D_2 \equiv id)$

$$pi(D_1, S_1, C_1) \odot pi(D_2, S_2, C_2) \equiv pi(D_1, S_1 \odot S_2, C_2).$$

The proof is by induction over arguments of relations. The arguments are defined on a well founded set, ordered by D_1 and C_2. Each pi-term in the expression $pi(D_1, S_1, C_1) \odot pi(D_2, S_2, C_2)$ is unfolded to arrive at

$$(S_1 + D_1 \odot (id\Box pi(D_1, S_1, C_1)) \odot C_1) \odot (S_2 + D_2 \odot (id\Box pi(D_2, S_2, C_2)) \odot C_2).$$

Equations in the hypothesis eliminate the cross terms to give

$$S_1 \odot S_2 + D_1 \odot (id\Box pi(D_1, S_1, C_1)) \odot (id\Box pi(D_2, S_2, C_2)) \odot C_2.$$

Now we can apply 38 to bring the two pi-terms together.

$$S_1 \odot S_2 + D_1 \odot (id\Box pi(D_1, S_1, C_1) \odot pi(D_2, S_2, C_2)) \odot C_2.$$

Applying the induction hypothesis we now have

$$S_1 \odot S_2 + D_1 \odot (id\Box pi(D_1, S_1 \odot S_2, C_2)) \odot C_2.$$

Folding, we finally conclude that $pi(D_1, S_1, C_1) \odot pi(D_2, S_2, C_2) \equiv pi(D_1, S_1 \odot S_2, C_2)$. Using equations 46 and 47 we can state a useful corollary as

$$pi(D, S \odot null, id) \odot pi(id, null \odot T, C) \equiv pi(D, S \odot null \odot T, C).$$

PROPOSITION 2 Propagation of constraints.

$$pi(D, S, (R \square id) \odot C) \equiv pi(D \odot (R \square id), S, C).$$

Again D must map a well founded set into lists. By induction and equation 34

$$\begin{aligned} pi(D, S, (R \square ia) \odot C) &\equiv S + D \odot (id \square pi(D, S, (R \square id) \odot C)) \odot (R \square id) \odot C \\ &\equiv S + D \odot (R \square id) \odot (id \square pi(D, S, (R \square id) \odot C)) \odot C \\ &\equiv S + D \odot (R \square id) \odot (id \square pi(D \odot (R \square id), S, C)) \odot C \\ &\equiv pi(D \odot (R \square id), S, C). \end{aligned}$$

The relation R has changed positions. Now $R \square id$ tests on the way down, before any large structure has been built, instead of on the way back up. A simple consequence is the equation $map(F) \odot map(G) \equiv map(F \odot G)$. This follows from the previous two propositions and the fact that $map(R) \equiv pi(id, null, R \square id)$.

If a concatenation of sequences is required to be empty, we can avoid the concatenation by requiring that each subsequence of the sequence be empty.

(59) $$conc \odot null \equiv map(null) \odot [\,].$$

This rule follows by induction from the definitions of *conc*, *null*, and *map* with equations 8, 49, 39, and 40.

6. Transformations. The equations developed form the foundation for a nontrivial program synthesis that is interesting for two reasons. Primarily, it relies less on heuristics than, for example, the Burstall and Darlington fold/unfold technique [11]. Instead it is directed by the operator definitions and equations between relations. Therefore the method is easily mechanized. Secondly, unlike Hogger's techniques [12], it is carried out using relation level reasoning without resorting to object level variables. This saves symbols and makes the derivation more concise and broadly applicable.

The sample problem has two parts. The first part finds the list intersection of two lists. This might be an example of a library program. Library programs may well be written efficiently for the immediate application but combinations are often inefficient. If two programs are written so that constraints are applied as early as possible, before alternatives are created, the composition may have some constraints that are applied too late.

A programmer should not expect to be penalized with the inefficiencies of library programs. The second program exhibits this problem. It is a clear, easy to understand program that simply tests to see if two sequences are disjoint. Using the rules developed earlier, we remove the inefficiencies in that program.

6.1. Disjointedness as empty intersection. The first part of this program finds the intersections of two lists by distributing one list over the elements of the other, then finding those other elements that are members of the list.

(60) $$list_intersect \stackrel{def}{\equiv} distr \odot map(member \odot [hd] + nonmember \odot [\,]) \odot conc.$$

The program *disjoint* determines if two lists are disjoint by simply testing for an empty intersection. If it returns any answer at all then the two lists are disjoint. As defined, this program is unnecessarily inefficient although it is understandable in terms of its parts.

(61) $$disjoint \stackrel{def}{\equiv} list_intersect \odot null.$$

The transformation strategy applies relation equations outwards in, from left to right. After each successful rewrite the enclosing expression is attempted before deeper optimization is done to the subexpressions[13]. The first two recursions are within the definition of *list_intersect*. Let *buildone* represent the expression $member \odot [hd] + nonmember \odot [\,]$. Then,

$$list_intersect \odot null \equiv distr \odot map(buildone) \odot conc \odot null.$$

Attention is directed to the two recursions in $distr \odot map(buildone)$ Both *distr* and *map* are defined with *pi* so they are merged and simplified.

$$\begin{aligned} distr \odot map(buildone) &\equiv pi([hd\Box id|tl\Box id], hd \odot null, id) \odot map(buildone), \\ &\equiv pi([hd\Box id|tl\Box id], hd \odot null, id) \odot pi(id, null, buildone\Box id), \\ &\equiv pi([hd\Box id|tl\Box id], hd \odot null, buildone\Box id), \\ &\equiv pi([hd\Box id|tl\Box id] \odot (buildone\Box id), hd \odot null, id), \\ &\equiv pi([hd\Box id \odot buildone|tl\Box id \odot id], hd \odot null, id), \\ &\equiv pi([hd\Box id \odot buildone|tl\Box id], hd \odot null, id). \end{aligned}$$

Next, $conc \odot null$ is expanded to $map(null) \odot [\,]$ which is $pi(id, null, null\Box id) \odot [\,]$. Thus $list_intersect \odot null$ is now two recursions followed by $[\,]$. This is

$$pi([hd\Box id \odot buildone|tl\Box id], hd \odot null, id) \odot pi(id, null, null\Box id) \odot [\,].$$

Once again we merge recursions and propagate constraints to obtain

$$\begin{aligned} list_intersect \odot null &\equiv pi([hd\Box id \odot buildone|tl\Box id], hd \odot null, null\Box id) \odot [\,], \\ &\equiv pi([hd\Box id \odot buildone|tl\Box id] \odot null\Box id, hd \odot null, id) \odot [\,], \\ &\equiv pi([hd\Box id \odot buildone \odot null|tl\Box id \odot id], hd \odot null, id) \odot [\,]. \end{aligned}$$

Now *buildone* is a sum, over which we can distribute *null* to obtain $buildone \odot null \equiv member \odot [hd] \odot null + nonmember \odot [\,] \odot null$. In addition, one branch of the sum goes away as $[hd] \odot null$ always fails. Therefore, we have

$$\begin{aligned} disjoint &\equiv pi([hd\Box id \odot (nonmember \odot [\,] \odot null)|tl\Box id \odot id], hd \odot null, id), \\ &\equiv pi([hd\Box id \odot (nonmember \odot [\,])|tl\Box id], hd \odot null, id) \odot . \end{aligned}$$

This is the order in which the implemented transformation system rewrites the original program. The mechanically derived program for *disjoint* tests for nonmembership before creating large structures. When the two lists differ at their first components, the remaining components need not be tested. For two lists of 30 elements, this program is approximately 100 times faster than the original.

This is a significant speedup, but we should note that an arbitrary amount of speedup is possible with the converse operator. For example with sequential, left-to-right AND, the cost of solving $X\ (parent^*)^\smile\ Y$ is much greater than the cost of solving $X\ (parent^\smile)^*\ Y$ for a large family tree.

6.2. Eliminating intermediate lists. Wadler describes a *deforestation* method for avoiding intermediate lists [14]. He applies it to a program to compute the sum of squares of numbers between 1 and N. The method uses the unfold-fold method on recursive equations and applies to deterministic programs. Transformations based on relation algebras extend these techniques to nondeterministic computations in a verifiable way.

The program performs just three main steps. The program constructs a sequence from N to 1, squares every number, and sums the sequence. The first interesting observation is that all of these operations are described by the *pi* operator.

$$pi([id|sub1], eq0 \odot [\,], id) \odot$$
$$pi(id, null, sqr \square id) \odot$$
$$pi(id, null \odot \mathbf{1} \odot eq0, plus).$$

The realization for a Q-relation algebra requires the extra relations $eq0$, $sub1$, sqr, and $plus$. These have the obvious definitions except that $eq0$ is just the single pair $\langle 0|0 \rangle$.

The first two recursions can be merged, with proposition 1, and the sqr operation can be brought forward, with proposition 2, to obtain

$$pi([sqr|sub1], eq0 \odot [\,], id) \odot pi(id, null \odot \mathbf{1} \odot eq0, plus).$$

Once again the two recursions can be merged to obtain the result $pi([sqr|sub1], eq0, plus)$.

7. Conclusion. Programs can be specified as proper binary relations with projections. Binary relations have an associated algebraic structure useful for program synthesis, verification, and optimization. In particular, Tarski's Q-relation algebras offer a concise notation and a firm foundation for transformational programming. The abstract relation operators are appropriate for describing the generic constructs that often arise in programming.

This work includes the definition of a general operator pi that describes linear recursions and gives two broadly applicable equations that merge recursions and propagate constraints. These equations provide an equational method for reasoning and scheduling specifications for computation.

The system based on these equations transforms program specifications so that the result often specifies a different algorithm. This implementation shows that we can in some cases build new programs on previously constructed ones without the usual efficiency penalty from the combination.

Acknowledgements. I am grateful to Rich Kaste for his suggestions and attention to detail. I also thank Roger Maddux, Jim Lipton, and Anil Nerode for discussions and background on relation algebras and operators. Thanks also go to Barbara Broome for her comments, to Morton Hirschberg for his support, and to Jossie Kirst for her assistance.

References.

[1] A. Tarski & S. Givant, "A formalization of set theory without variables," in *Colloquium publications* #41, American Mathematical Society, Providence, Rhode Island, 1987.

[2] C. S. Peirce, "On the Algebra of Logic: A Contribution to the Philosophy of Notation," *American Journal of Mathematics* VII (1885).

[3] A. Tarski, "On the calculus of relations," *The Journal of Symbolic Logic* 6 (1941), 73–89.

[4] Roger D. Maddux, "Introductory course on relation algebras, finite-dimensional cylindric algebras, and their interconnections," 1988, Notes.

[5] John Backus, "Can Programming Be Liberated from the von Neumann Style?; A Functional Style and Its Algebra of Programs," *Comm. ACM* 21 (1978), 613–641.

[6] R. Berghammer & H. Zierer, "Relational Algebraic Semantics of Deterministic and Nondeterministic Programs," *Theoretical Computer Science* 43 (1986), 123–147.

[7] A. Mili, J. Desharnais & F. Mili, "Relational Heuristics for the Design of Deterministic Programs," *Acta Informatica* 24 (1987), 239–276.

[8] D. Chan, "Constructive Negation Based On the Completed Database," in *Logic Programming: Proceedings of the Fifth International Conference and Symposium*, R. A. Kowalski and K. A. Bowen, ed., MIT Press, Cambridge, MA, 1988, 111–125.

[9] A. Colmerauer, "Equations and inequations on finite and infinite trees," in *FGCS'84 Proceedings*, 1984, 85–99.

[10] Joxan Jaffar, Jean-Louis Lassez & John Lloyd, "Completeness of the Negation as Failure Rule," in *Proceedings of the 1983 IJCAI*, 1983, 500–506.

[11] R.M. Burstall & John Darlington, "A Transformation System for Developing Recursive Programs," *J. Assoc. Comput. Mach.* 24 (Jan., 1977), 44–67.

[12] C.J. Hogger, "Derivation of Logic Programs," *J. Assoc. Comput. Mach.* 28 (Apr., 1981), 372–392.

[13] Noor Islam, Tom Myers & Paul Broome, "A Simple Optimizer for FP-like Languages," in *Proc. ACM Conf. on Functional Programming Languages and Computer Architecture*, 1981, 33–39.

[14] Philip Wadler, "Deforestation: Transforming programs to eliminate trees," in *LNCS*, H. Ganzinger, ed. #300, Springer-Verlag, Berlin, 1988, 344–358.

TIMES OF THE SIGNS

Moss E. Sweedler*
Mathematical Sciences Institute
Cornell University
Ithaca NY 14853

ABSTRACT: Evaluate a real polynomial f and its derivatives at a real number r and consider the signs { -, **0**, + } of the results. Herein lies new techniques for obtaining information from the sequence of signs which arise. One techniques assigns a (magic) number to a sign sequence. Another assigns a path to a sign sequence.

INTRODUCTION: Say f is a degree n polynomial with real coefficients. We have the derivative sequence $D(f) = (f, f', f'', ..., f^{(n)})$ for f. For a real number r, $D(f)(r)$ stands for the sequence: $(f(r), f'(r), f''(r), ..., f^{(n)}(r))$ and sign $D(f)(r)$ stands for the sign sequence: (sign f(r), sign f'(r), sign f''(r),..., sign $f^{(n)}(r)$) where sign 0 is **0**. Such a sign sequence is said to be derived from f at r. The following will be discussed in greater detail:

1 Given two real numbers r and s, how to tell from the sign sequences: sign $D(f)(r)$ and sign $D(f)(s)$ if $r < s$. **APPROXIMATE-ANSWER:** Multiply adjacent signs and dot product the result with (1,2,...) to obtain the Magic Numbers #sign $D(f)(r)$ and #sign $D(f)(s)$. Then #sign $D(f)(r)$ < #sign $D(f)(s)$ if and only if $r < s$.
2 How to form a path through a sign triangle corresponding to a sign sequence.
3 How to tell from the paths of sign $D(f)(r)$ and sign $D(f)(s)$ if $r < s$?
4 How the path condition relates to the [Coste-Roy] algorithm to determine from sign $D(f)(r)$ and sign $D(f)(s)$ if $r < s$?
5 How paths relate to Magic Numbers?
6 How the property of paths not crossing helps determine when sign sequences arise as sign $D(f)(r)$ and sign $D(f)(s)$, i.e. from a single polynomial and its derivatives?
7 Generalizations to non-polynomial functions f.
8 Unrealizability of certain sign sequence patterns.

Proofs, further details and additional related results will be published later. Discussions with and between Dexter Kozen and Jim Renegar got me interested in this area. The excellent article [Coste-Roy] finished getting me hooked.

MAGIC NUMBERS: If S is a finite sequence of -'s, **0**'s and +'s we calculate the Magic Number #S for S as follows. Say $S = (s_0, s_1, s_2, ..., s_n)$ where each $s_i \in \{ -, 0, + \}$. As with counting sign changes in Sturm Sequences, the number #S depends on the sign change transitions from s_{i-1} to s_i as i varies from 0 to n. However, how far along the transition occurs, is also taken into account. #S is defined in two stages. First μS is defined as the vec-

* Supported in part by U.S. Army Research Office and the National Science Foundation

tor: (s_0s_1 , s_1s_2 , s_2s_3 , ... , $s_{n-1}s_n$) . For multiplication purposes, - should be considered -1 and + should be considered +1 . The definition of #S is concluded as the dot product: #S = $\mu S \cdot (1,2,3, \ldots n)$. Notice that the dot product weights later transitions more heavily than early transitions.

EXAMPLE: Suppose S(r) = sign D(f)(r) where f is the polynomial $X^2 - 1$. The sign sequences derived from f at r and their Magic Numbers are given by:

range of r	sign D(f)(r)	#sign D(f)(r)
r < -1	(+, -, +)	-3
r = -1	(0, -, +)	-2
-1 < r < 0	(-, -, +)	-1
r = 0	(-, 0, +)	0
0 < r < +1	(-, +, +)	+1
r = +1	(0, +, +)	+2
+1 < r	(+, +, +)	+3

The reader might consider what sign sequences are derived from $f = X^2$ and $f = X^2 + 1$.

EXAMPLE: The last sign. For a polynomial f with positive leading coefficient, all sign sequences derived from f have + as their last (right most) sign. Similarly, all sign sequences derived from polynomials with negative leading coefficient have - as their last sign.

EXAMPLE: Extreme left and right. For a polynomial f let sign D(f)(-∞) denote the unique sign sequence sign D(f)(r) for r << 0 and let sign D(f)(+∞) denote the unique sign sequence sign D(f)(r) for r >> 0 . sign D(f)(-∞) consists of alternating - and + signs. Hence every transition is (+,-) or (-,+) and #sign D(f)(-∞) reduces to the dot product (-1, -1, ... -1) · (1,2,...,degree f). Thus #sign D(f)(-∞) = - (degree f) (1 + degree f) / 2 . sign D(f)(+∞) consists of all + signs and #sign D(f)(+∞) = (degree f) (1 + degree f) / 2 .

Suppose r and s are real numbers where neither f nor any of its derivatives have a zero between r and s. Then f and all its derivatives have the same sign at r and s. I.e. sign D(f)(r) = sign D(f)(s) and so #sign D(f)(r) = #sign D(f)(s) . This shows why the hypothesis about r and s being separated by a zero of f or one of its derivatives is needed in the following theorem. It is also what was missing from the **APPROXIMATE ANSWER** at (1).

9 THEOREM: Suppose f is a real polynomial and r < s are real numbers where f or one of its derivatives has a zero in the closed interval [r, s]. Then #sign D(f)(r) < #sign D(f)(s) .

The easy proof will be published later. Impatient readers are encouraged to prove it on their own. The theorem has a number of immediate corollaries. For flavor, here are three:

* If r < s and f or one of its derivatives has a zero in the closed interval [r, s] then #sign D(f)(r) cannot equal #sign D(f)(s) .

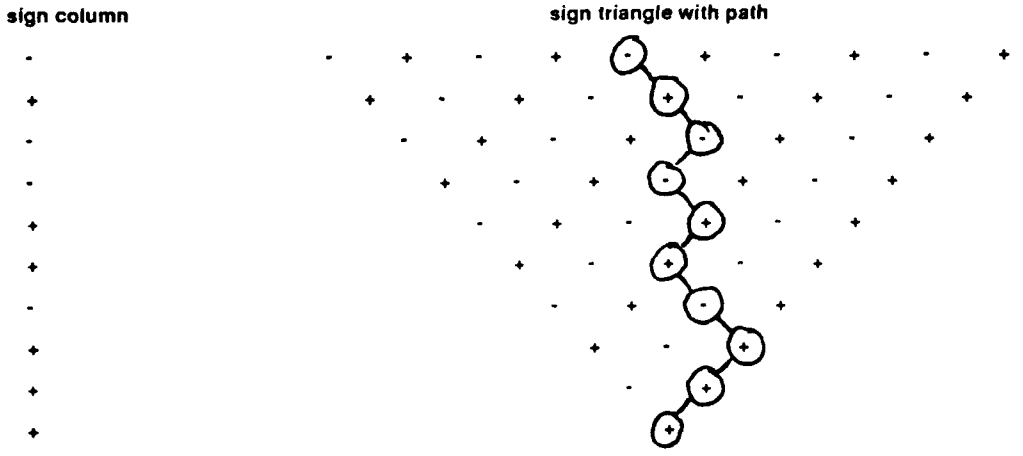
* There is no real polynomial f and real numbers r and s where sign D(f)(r) does not equal sign D(f)(s) but #sign D(f)(r) = #sign D(f)(s). (Because "sign D(f)(r) does not equal sign D(f)(s)", implies that f or one of its derivatives must have a zero in [r,s] or [s,r] depending whether r < s or s < r and the previous corollary applies.

* If r < s and f or one of its derivatives has a zero in the closed interval [r, s] then sign D(f)(r) cannot equal sign D(f)(s). (This special case of Thom's theorem follows immediately from the first corollary.)

10 DERIVABILITY: One aspect of the corollaries is to describe behavior which cannot occur among sign sequences which arise as sign D(f)(r) and sign D(f)(s). For example, the second corollary shows that for the sign sequences (0, 0, +) and (-, 0, +) there is no quadratic real polynomial f and real numbers r and s where: sign D(f)(r) = (0, 0, +) and sign D(f)(s) = (-, 0, +). An interesting problem is to develop necessary and sufficient compatibility conditions for m sign sequences to arise as sign D(f)(r_1) ,..., sign D(f)(r_m) from a single polynomial f and its derivatives evaluated at m points. Sets of sign sequences which arise in this fashion are called **derivable**.[3] We have more to say about derivability in the next section.

SIGN TRIANGLES AND PATHS: To simplify the discussion, we confine ourselves to sign sequences without zeros. These will give rise to piecewise linear **paths** in a sign triangle.[4] The sign triangle and the path corresponding to a sign sequence (rather sign column) is illustrated below. **Start at the bottom** and draw a path passing through the signs coming from the sign column. This is illustrated in:

11 EXAMPLE



---

[3] They are derived from the polynomial f at the points r_1, ... r_m.

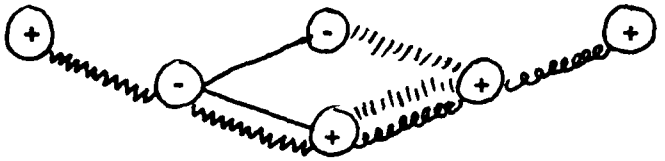
[4] Had zeros been present, the paths would have "width". In fact, when zeros are present, the paths tend to look like the wakes of ships and are called wakes instead of paths.

The illustration not only assumed that the sign column had no zeros but also that the bottom sign was a + . Had the bottom sign been a - , use a sign triangle with +'s and -'s interchanged.[5] We are interested in paths arising from sign columns which are the transpose of sign sequences sign D(f)(r). Such a path is simply called the path of sign D(f)(r) , glossing over the transpose. Modulo the simplifying assumption of sign sequences not containing zero we have:

12 THEOREM: Suppose f is a real polynomial and $r < s$ are real numbers where f or one of its derivatives has a zero in the closed interval [r, s]. The path of sign D(f)(r) does not cross the path of sign D(f)(s) and at some row of the sign triangle the path of sign D(f)(r) lies strictly to the left of the path of sign D(f)(s) .[6]

EXPLANATION: What does "paths not crossing" mean? All paths **touch** at the bottom of a sign triangle and may overlap. We say that two paths **cross** if at some row $Path_1$ is strictly to the left of $Path_2$ and at another row $Path_1$ is strictly to the right of $Path_2$. It is easy to produce examples of a real polynomial f together with real numbers r and s where the path of sign D(f)(r) and the path of sign D(f)(s) touch and diverge at several places.

EXAMPLE: Here are the paths of sign $D(X^2 - 1)(r)$ for $r < -1$, $-1 < r < 0$, $0 < r < 1$, $1 < r$:



(12) has stronger but similar corollaries to (9). For a while it was hoped that "paths not crossing" gives a necessary and sufficient derivability condition (10). This is true for sign sequences of length four or less. I.e. coming from polynomials of degree three or less. Bruce Anderson produced an example of a set of sign sequences of length five whose paths do not cross but are not derived from any degree four real polynomial. He is working on an example of sign sequences of length six whose paths do not cross and yet are not derived from any real valued function together with five successive derivatives. Carl de Boor has suggested the name "higher-order Rolle's theorem" for certain theorems which describe collections of sign sequences whose paths do not cross - i.e. are allowed by Rolle's Theorem - but are not derivable.

---

5 Actually, the +'s and -'s in the triangle are not really needed to construct the path once one observes that the rule is: diagonally up to the right when the sign stays the same and diagonally up to the left when the sign changes.

6 Carl DeBoor views this as a graphic way to express Rolle's theorem.

The first algorithm to determine the relative size of r and s from the sign sequences sign D(f)(r) and sign D(f)(s) appears in [Coste-Roy]. Translated into our path-language their algorithm[7] can be described as follows:

> All paths touch at the bottom vertex of the sign triangle and never cross.
> To determine which of two paths lies to the left of the other
>
> 1. Start from the bottom of both paths and proceed upward row by row of the sign triangle.
> 2. At the row where one path diverges to the left, that path corresponds to the smaller of r and s.

GENERIC BEHAVIOR: Conceptually one may think of zeros between the '-'s and '+'s in each row of the sign triangle. If rows of sign triangles are numbered from bottom to top the rows are the generic sign behavior of polynomial functions - with positive leading coefficient - of the corresponding degree:

13 EXAMPLE

row	generic behavior of function
5	- 0 + 0 - 0 + 0 - 0 +
4	+ 0 - 0 + 0 - 0 +
3	- 0 + 0 - 0 +
2	+ 0 - 0 +
1	- 0 +
0	+

Not only are we seeing the generic sign behavior of functions of each degree, assuming each function is the derivative of the one above, the signs are appropriately located left to right. For functions or derivatives with multiple roots, it is useful to collapse parts of rows of the sign triangle corresponding to the multiple roots.

SURPRISING APPLICATION OF PATHS: One may sometimes determine relative magnitude of functional values from sign sequences.[8] This is true (on the rare occasions) when the paths of two distinct sign sequences coincide in the top row of the sign triangle and one row below.[9] For example, suppose f is a real polynomial, r and s are real numbers and:

sign D(f)(r) = (+, +, +, +, +, +, -, +, -, +) sign D(f)(s) = (+, +, -, +, -, +, +, +, +, +)

---

[7] In the special case where neither sign sequence has a zero.

[8] We ignore the trivial case where two sign sequences begin with different signs.

[9] Presumably this is part of a more encompassing theory or technique. Unfortunately, the *big picture* is a mystery.

From paths or Magic Numbers we see that r lies to the left of s. Since both sign sequences begin with + , we see that both f(r) and f(s) are positive. Surprisingly, one can conclude from the sign sequences that $f(r) < f(s)$. The reasoning goes as follows.

> Since sign D(f)(r) contains no zeros, one may move r a small amount to the right or left and keep the same sign sequence but vary the height of f(r) slightly. Thus we may assume that f(r) does not equal f(s). Suppose $f(r) > f(s)$. Choose a constant c where: $f(r) > c > f(s)$ and let $g = f - c$. The successive derivatives of g are the same as the successive derivatives of f. By choice of c, $g(r) > 0$ and $g(s) < 0$. Thus
>
> sign D(g)(r) = (+, +, +, +, +, +, -, +, -, +) sign D(g)(s) = (-, +, -, +, -, +, +, +, +, +)
>
> I.e. the sign sequences for f and g agree except the initial + in sign D(f)(s) has been changed to a - . This is a contradiction, because these paths of sign D(g)(r) and sign D(g)(s) cross. Hence, $f(r) < f(s)$.

By considering similar but much lower degree examples, one can easily reason - without using paths - that $f(r) < f(s)$ or vice-verse. Presumably the same type of reasoning on a larger scale could be applied to the example above.

PATHS and MAGIC NUMBERS: The Magic Number results may be proved directly or from sign triangles by introducing numbers to accompany the signs within the sign triangle. A path will give rise to a column or vector of numbers which allows one to translate geometric path results into Magic Number results. The key idea is numbering sign triangles as indicated in:

```
+:-6     -:-4     +:-2     -:0      +:2      -:4      +:6
    -:-5     +:-3     -:-1     +:1      -:3      +:5
        +:-4     -:-2     +:0      -:2      +:4
            -:-3     +:-1     -:1      +:3
                +:-2     -:0      +:2
                    -:-1     +:1
                        +:0
```

BEYOND POLYNOMIALS: Lawrence D. Brown has shown sign sequence and path results may be applied to non-polynomial functions. See the examples below. This led to an abstract formulation of the functions which may be used in place of successive derivatives and opened up the field to rational functions, fractional exponents and other functions. Here is an informal sketch.

Instead of the sequence of functions (f, f', f", ...) , consider a sequence of real functions $(f_0, f_1, ..., f_t)$ which are continuous on an open interval (a,b) where:

14 DEFINITION:

1. Each f_i only has a finite number of zeros in (a,b).
2. f_t has no zeros in (a,b).
3. For p in (a,b), if $f_i(p) = 0$ then
   a. $f_i(p+)$ and $f_{i+1}(p+)$ have the same sign and
   b. $f_i(p-)$ and $f_{i+1}(p-)$ have opposite signs.

If you wish to try an example, draw a random f_t with no zeros in (a,b). You will see that f_{t-1} can have at most one zero in (a,b). By induction, f_{t-i} can have at most i zeros on (a,b).

For p in (a,b) the sign sequence at p is: (sign $f_0(p)$, sign $f_1(p)$, ..., $f_t(p)$) and is denoted **SS**(p) . Let S be the union of all the zeros of the f_i's in (a,b). S is finite and splits up (a,b) into open intervals. For r and s in the same open interval: **SS**(r) = **SS**(s) , as before. In fact, when you substitute **SS**(r) for sign D(f)(r) and **#SS**(r) for #sign D(f)(r), the theory - as pertains to relative magnitude of r and s - is the same *Magic Number-wise* and *path through sign triangle-wise*. For example, paths do not cross and for r and s which are **not** in the same open interval: $r < s$ if and only if the path of **SS**(r) is to the left of the path of **SS**(s).

15 EXAMPLE: Let $f_0 = f / g$ where f is a degree n polynomial and g is continuous. Let (a,b) be an open interval where g has no roots. Assume g is positive on (a,b). The sequence of functions: $f_0 = f/g$, $f_1 = f'$, $f_2 = f''$, ... $f_n = f^{(n)}$ satisfy (14).

16 EXAMPLE: On the interval $(0, \infty)$ the sequence: $f_0 = X^{1/2} - X^{1/3}$, $f_1 = 3X^{1/3} - 2X^{1/6}$, $f_2 = 6X^{1/6} - 2$, $f_3 = 6$ satisfies (14).

The key to (16) is the following. Use "o" for composition of functions. Suppose $f_0 = fog$ where f is a degree n polynomial and g is continuous and **increasing** on (a,b). The sequence of functions is: $f_0 = fog$, $f_1 = f'og$, $f_2 = f''og$, ... $f_n = f^{(n)}og$. Using $f = Z^3 - Z^2$ and $g = X^{1/6}$ gives (16).

FINAL REMARKS: The following issues are under consideration.

Generalization of the above work to complex polynomials and generalization to functions from R^n to R^n. In the complex case, **argument** replaces sign.

Certain functional transformations do not change the set of sign sequences derived from a polynomial. More specifically, suppose f(X) is a real polynomial. Replace f by $g(X) = af(bX - c)$ for real numbers a, b and c with $a, b > 0$. Since sign D(g)((p+c)/b) = sign D(f)(p) , the same sign sequences are derived from f and g but with different transition points from one sign sequence to the next.[10] What other functional transformations preserve sign sequences in this sense? The full set of such transformations forms a group under composition.

---

[10] The Magic Numbers of the sign sequences show that if the same sign sequences are derived from f and g, they occur in the same order.

It would be interesting to find normal form representatives for sets of sign sequences which can be derived from a function. Here one is looking for a collection of polynomials (the h_f's) where for each polynomial f there is an h_f such that the same sign sequences are derived from f and h_f. Moreover if the same sign sequences are derived from f and g then $h_f = h_g$. h_f is the normal form of (the sign sequences derived from) f. One wants an algorithm to find h_f from f or from the set of sign sequences derived from f.

In the formulation of the Magic Number, (1,2,3, ... n) is dotted with μS . If (1,2,3, ... n) is replaced by a different increasing sequence of positive numbers, #S is changed, but (9) still holds. This easily comes out of the numbered sign triangles mentioned at (**PATHS and MAGIC NUMBERS**) above. Given a derivable set of sign sequences **S**, can one derive **S** from a polynomial f and find an increasing sequence of positive numbers to replace (1,2,3, ... n) so that for each sign sequence S in **S**, S is derived from f at #S , i.e.: S = sign D(f)(#S) ?

REFERENCE: M.Coste, M.F.Roy 1988 Thom's lemma, the coding of real algebraic numbers and the computation of the topology of semi-algebraic sets. J. Sym. Comp. **5** 121-129.

GEOMETRIC MODELS FOR DEVELOPABLE AND MINIMAL SURFACES

T. F. Chen, G. J. Fix, R. Kannan
Department of Mathematics
University of Texas at Arlington
Arlington, Texas 76019

Abstract.

Computational geometry is a relatively new and unusual subject in the sense that its orientation is quite applied in nature, yet it makes heavy use of pure mathematics, notably algebraic and differential geometry. The technological importance of geometric models has increased in recent years. Indeed, the major bottleneck in developing effective computer aided design (CAD) software has centered around geometry and a range of issues associated with computer vision. In addition, computational geometry is finding wide applications in the computer graphics industry. In this paper we summarize the results for selected problems in this area associated with developable and minimal surfaces.

I. The Two Curve Problem for Developable Surfaces.

Developable surfaces are defined as regular surfaces with zero Gaussian curvature ([1]-[2]). As such they are locally isometric to planar regions; i.e., they can be obtained from planes by bending (which preserves arc length and angles). Their technological importance arises from this property. Indeed, more surfaces constructed using composite materials will fall into this category ([3]).

A problem of interest in design is in the following. Given two spatial curves $\vec{\alpha}(\cdot)$, $\vec{\beta}(\cdot)$ find a developable surface $\mathcal{S}$ connecting these curves. We shall consider solutions within a subclass of all developable surfaces. In particular, we shall consider surfaces $\mathcal{S}$ which consist exclusively of parabolic points, i.e., surfaces where only one of the two principal curvatures vanish.

It is known ([1], [2], [4]) that such surfaces are ruled, and admit a parameterization of the form

$$\vec{x}(u, v)=\vec{\alpha}(u)+u\vec{\omega}(u) \tag{1}$$

as u, v vary over open sets of the reals **R**. For a surface of the form (1) to be a developable (i.e., have

zero Gaussian curvature) it is necessary and sufficient that the tangent $\dot{\vec{\alpha}}$ to the generating curve $\vec{\alpha}$, the tangent $\dot{\vec{\omega}}$ to the rulings $\vec{\omega}(\cdot)$, and the ruling itself be coplanar. This condition can be written

$$0=\left[\dot{\vec{\alpha}}, \dot{\vec{\omega}}, \vec{\omega}\right], \tag{2}$$

where $[\cdot, \cdot, \cdot]$ is the standard box product. The two curve problems can be stated in this context as seeking a function v(u) for which

$$\vec{\beta}(v(u))\vec{\alpha}(u)+v(u)\vec{\omega}(u) \tag{3}$$

as u varies over its parameter range.

Weiss and Furtner [5] has proposed an interactive searching algorithm to solve this problem. Their idea was to construct the developable by rulings (lines). The criteria is to connect a point $\vec{\alpha}_0=\dot{\vec{\alpha}}(u_0)$ on the $\vec{\alpha}-$ curve to a point $\vec{\beta}_0=\vec{\beta}(v_0)$ on the $\vec{\beta}-$curve if the two tangents $\dot{\vec{\beta}}_0=\dot{\vec{\beta}}(v_0)$, $\dot{\vec{\alpha}}_0\dot{\vec{\alpha}}(u_0)$ and the displacement vector $\vec{\beta}_0-\vec{\alpha}_0$ are coplanar; i.e.,

$$\left[\dot{\vec{\alpha}}_0, \dot{\vec{\beta}}, \vec{\alpha}_0-\vec{\beta}_0\right]=0. \tag{4}$$

Using a graphics terminal, one traces say the $\underline{\alpha}$-curve searching at each step for an appropriate connection to the $\vec{\beta}$-curve.

There are cases where (4) may not hold any point pairs $\vec{\alpha}(u_0)$, $\vec{\beta}(v_0)$. This is for example the case when $\vec{\alpha}(\cdot)$ is a circle and $\vec{\beta}(\cdot)$ is a line through the center of the circle. This example also shows that the two circle problem itself may not have a solution. In other cases, (4) may be very hard to verify at a large number of points.

To develop an alternative to this approach a dynamical algorithm was developed in reference [6] for the function v(u). In particular, it was shown that this function satisfied the nonlinear ordinary differential equation

$$\left[\dot{\vec{\beta}}, \dot{\vec{\alpha}}, \vec{\beta}-\vec{\alpha}\right]\frac{dv}{du}=\left[\dot{\vec{\beta}}, \ddot{\vec{\alpha}}, \vec{\beta}-\vec{\alpha}\right], \tag{5}$$

where $\vec{\beta}$ and its derivatives are evaluated at v(u) and $\vec{\alpha}$ and its derivatives are evaluated at u. From this system a number of conclusions can be drawn. First, a condition like (4) is needed at least at one point pair $\underline{\alpha}_0$, $\underline{\beta}_0$. This serves are initial conditions for the system. If (4) holds nowhere, then

obviously a connecting developable does not exist as the line, circle cited above indicates. Also since (5) is nonlinear local existence and uniqueness is assured only if (4) holds at some point.

To illustrate the algorithm for a few cases consider first the problem of connecting a circle and an ellipse $\vec{\alpha}$, $\vec{\beta}$ with a developable. In this case u is taken as the arc length of α, and an elementary analysis shows that a global solution v(u) exists for (4), (5)($u_o = 0$), and that

$$\theta(L) = \theta(0), \tag{6}$$

where L is the total arc length of $\vec{\alpha}$. To construct the associated developable surface, the ordinary differential equation (5) was integrated from t = 0 to t = L with fourth order explicit Runga-Kutta rule. The results are shown in Figure 1.

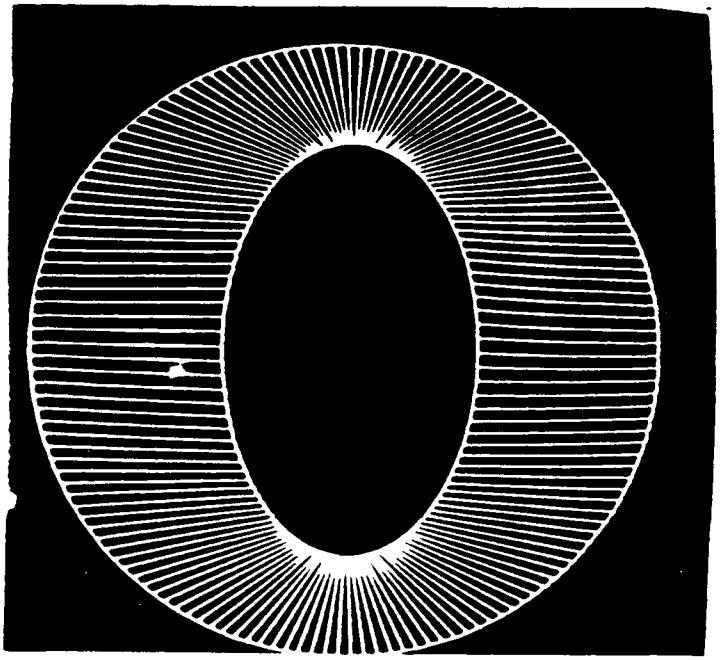


FIGURE 1

The next example consists of a circle $\vec{\alpha}$ and a three leaf curve $\vec{\beta}$. Again a periodic solution v(u) is obtained with the period equal to the total arc length of the circle $\vec{\alpha}$. The results are shown in Figure 2.

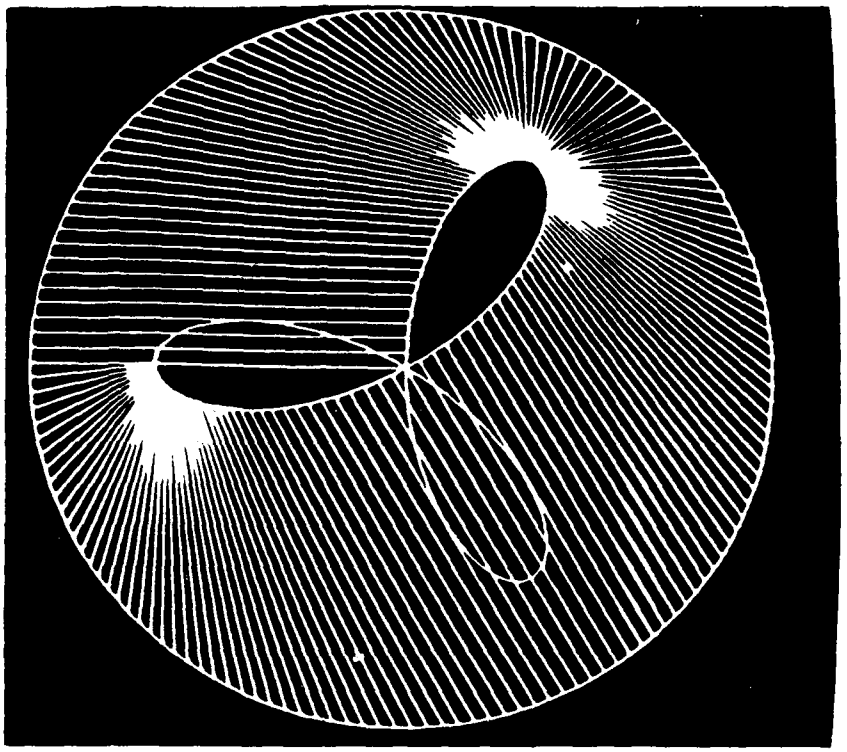


FIGURE 2

The developable joining $\underset{\sim}{\alpha}$ and $\underset{\sim}{\beta}$ may not in general connect each point of $\vec{\beta}$ to a point on $\vec{\alpha}$. Such a situation is shown in Figure 3. In this case the global solution v(u) does not take on the full parameter range for $\vec{\beta}(v)$. Results are shown in Figure 3.

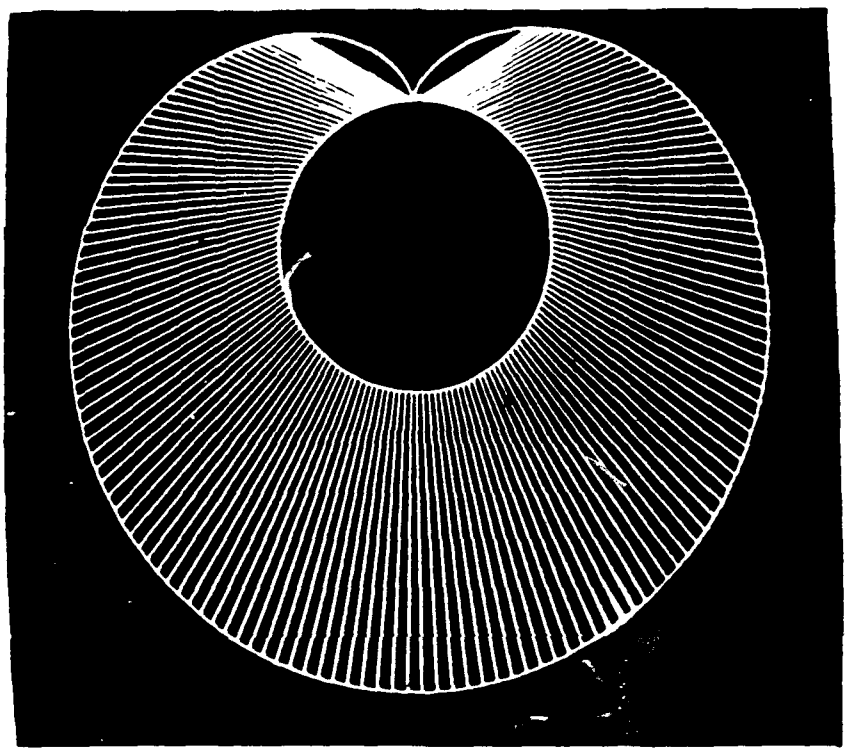


FIGURE 3

A finite time blow up in (5) could occur. This will happen for example as an inflection point $\vec{\beta}(v(u_1))$ of the $\vec{\beta}$-curve (i.e. $\ddot{\vec{\beta}}=\underline{0}$ at this point). If $\vec{\alpha}(u_1)$ is not an inflection point for the $\vec{\alpha}$-curve then the dynamical system (5) can be reformulated with u as a function of v, and continued past the inflection point in the $\vec{\beta}$-curve.

Current research is dealing with a rational treatment of singularities which occur in either the $\vec{\alpha}$-curve or the $\vec{\beta}$-curve. Such situations are common in practice and correspond to edges or corners of the object to be designed. In reference [6] preliminary results using conic lofting techniques (see also [7]-[8]) are presented.

REFERENCES

1. M. P. DiCarmo, *Differential geometry of curves and surfaces*, Prentice Hall, 1976.
2. J. J. Stoker, *Differential Geometry*, Wiley-Interscience, New York, 1969.
3. J. R. Vinson and R. L. Sierakowski, *The behaviour of structures composed of composite materials*, Martinus Nijhoff, 1986.
4. W. C. Graustein, *Differential Geometry*, Macmillan Co., 1935.
5. G. Weiss and P. Furtner, "Computer aided treatment of developable surfaces", *Computers and Graphics*, 12 (1988), 39-51.
6. *G. Fix, R. Johnson, R. Kannan*, "A dynamical algorithm for the developable surface problem", *Computer Aided Design* (to appear).
7. I. D. Faux and M. J. Pratt, *Computational Geometry for design and manufacture*, Ellis Harwood Publ., 1978.
8. M. E. Mortensen, *Geometric Modelling*, John Wiley, New York, 1985.
9. E. T. Y. Lee, "The rational Bezier representation for conics", *Geometric Modelling*, SIAM, 1987.

II. Grid Refinement and Nonlinear SOR Techniques Applied to the Minimal Surface Problem.

Numerical methods for the Plateau problem have been dealt with by either a combination of variational and finite difference methods or by a finite element method with appropriate restrictions on the existence of double points when projected on a plane. In all of these approaches the basic idea is always to take advantage of the strict convexity of the associated variational formulation in solving the discrete problem. Some references to these ideas may be seen in [1-4]. However one of the basic problems that remains to be formally studied is in proper choice of the iterative process to solve the discrete system of equations and/or proper selection of the grid points. As an example, one of the common features of all the above referred papers is to use a SOR-type iterative process with an arbitrary choice of the relaxation parameter. This leads to a trial and error approach to the choice of a relaxation parameter for the entire iterative process. The second remark that concerns the above mentioned papers is that not much emphasis is placed on locating the grid points. We have attempted with considerable success both of these aspects in this paper.

In order to illustrate the ideas we restrict ourselves to a typical example studied by most of the authors, namely, the catenoid. Thus let $Y_{(x)} = \frac{1}{a}\cosh\left[a(x-c)\right]$, $0 \leq x \leq 1$. The constants a and c

are to be determined by the associated boundary conditions. If one rotates the graph of $Y=f_{(x)}$ we get a catenoid. Let us set $Y_{(0)}=Y_{(1)}=\lambda$, where λ is a constant. Then $c=0.5$ and the constant "a" is determined by the equation $\cosh(\frac{a}{2})=a\lambda$. The critical value of λ such that the equation has no real solutions for λ below this value is 0.75444. In fact, one can see that as λ goes from 0.7 to 0.755 in the approach of [2], the number of iterations goes from 462 to 3448.

Before we present some of the numerical results we outline the iterative process used. Thus, if $\phi:R^N \rightarrow R$ be the strictly convex functional corresponding to the discrete problem the iterative sequence is defined by:

Given $\{x^j\}$, $j=1, 2, \cdots, k$. Let $\bar{x}^k$ be such that

i) $\phi(\bar{x}^k)=\phi(x^k)$

ii) $x_j^k=\bar{x}_j^k$, $j\neq i_k$, i_k is any integer between 1 and N.

iii) $x_{i_k}^k \neq \bar{x}_{i_k}^k$ unless $\phi_{i_k}(x^k)=0$.

Then set

$$x_{i_k}^{k+1}=x_{i_k}^k+r^k(\bar{x}_{i_k}^k-x_{i_k}^k)$$

where $r^k \in (0, 1)$.

Some of the numerical results are presented below. The advantage of grid refinement can be easily seen by the performance of the algorithm even when we approach the critical value of λ.

REFERENCES

1. M. Hinata, M. Shimasaki and T. Kiyono, "Numerical solution of Plateau's problem by a finite-element method, *Math. Comp.* 28 (1974) 45-59.
2. T. Tsuchiya, "On two methods for approximating minimal surfaces in parametric form", *Math. Comp.* 46(1986) 517-529.
3. H. J. Wagner, "A contribution to the numerical approximation of minimal surfaces", *Computing*, 19 (1977) 35-58.
4. T. F. Chen and R. Kannan, Grid refinement and Nonlinear SOR, to appear.

TABLE FOR NONLINEAR SOR AND GRID REFINEMENT

Table 1. Numerical solution vs. exact solution with relaxation parameter r=0.85 and boundary data y(0)=y(1)=1.

x	0.20	0.30	0.40	0.50	0.60	0.70	0.80
finite difference(25)	0.90305	0.87294	0.85473	0.84828	0.85363	0.87097	0.90070
finite element(25)	0.90208	0.87222	0.85445	0.84856	0.85445	0.87221	0.90208
exact	0.90194	0.87202	0.85424	0.84834	0.85424	0.87202	0.90194

Table 2. Finite difference solution with relaxation parameter r=0.85, and boundary data y(0)=y(1)=0.78.

a) uniform grid

x	0.20	0.30	0.40	0.50	0.60	0.70	0.80
r_a=0.78(53)	0.61075	0.55527	0.51888	0.50159	0.50392	0.52723	0.57440
exact	0.60955	0.56005	0.53083	0.52121	0.53083	0.56005	0.60995

b) nonuniform grid

x	0.16216	0.27477	0.38739	0.50000	0.61261	0.72523	0.83784
r_a=0.78(25)	0.63754	0.57542	0.53633	0.52042	0.52864	0.56336	0.62986
exact	0.63458	0.57063	0.53342	0.52121	0.53342	0.57063	0.63456

Table 3. Finite difference solution with relaxation parameter r=0.85, nonuniform grid, and boundary data y(0)=y(1)=r_a.

x	0.11789	0.21695	0.35848	0.50000	0.64152	0.78305	0.88211
r_a=0.756(50)	0.61498	0.53693	0.47169	0.44699	0.46675	0.54367	0.62832
r_a=0.755(52)	0.61152	0.53226	0.46594	0.44061	0.46034	0.53846	0.62486
exact(r_a=0.756)	0.61658	0.53417	0.46289	0.43993	0.46289	0.53417	0.61658
exact(r_a=0.755)	0.61149	0.52692	0.45393	0.43046	0.45393	0.52692	0.61149

Table 4. Finite element solution with relaxation parameter r=0.85 and boundary data $y(0)=y(1)=r_a$.

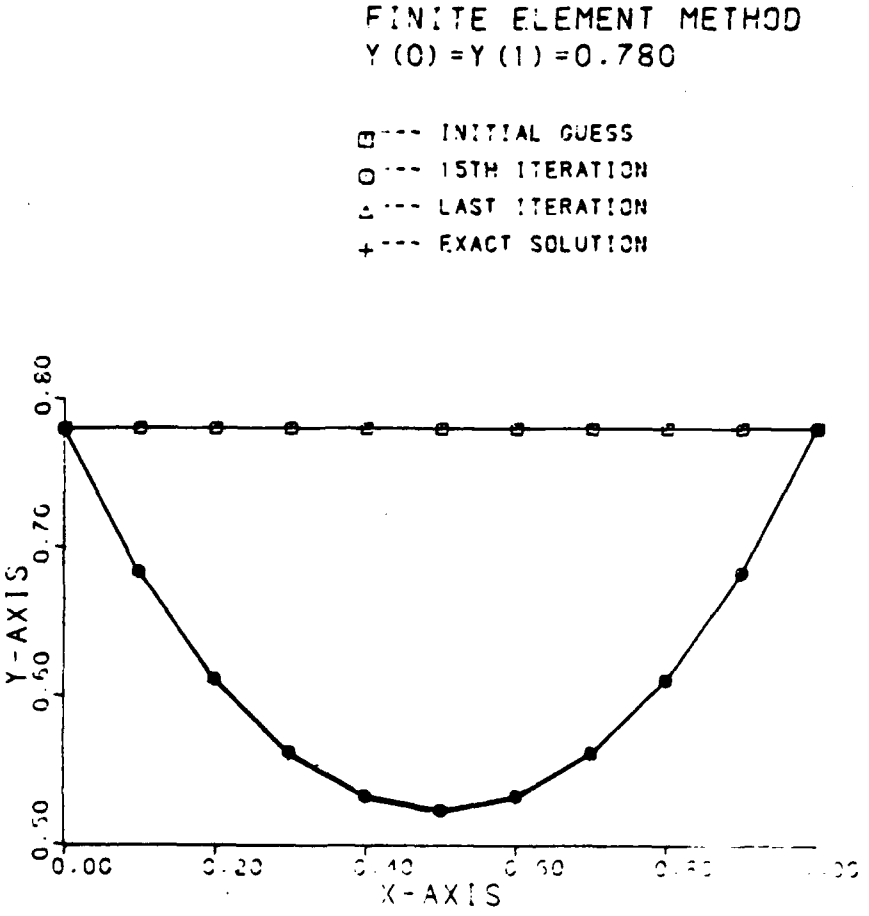
x	0.20	0.30	0.40	0.50	0.60	0.70	0.80
$r_a=0.78_{(21)}$	0.61121	0.56166	0.53261	0.52305	0.53262	0.56165	0.61123
$r_a=0.77_{(27)}$	0.59141	0.53926	0.50880	0.49878	0.50879	0.53923	0.59135
$r_a=0.76_{(42)}$	0.56672	0.51084	0.47829	0.46757	0.47827	0.51085	0.56670
$r_a=0.756_{(49)}$	0.55356	0.49532	0.46149	0.45039	0.46145	0.49527	0.55350
$r_a=0.755_{(55)}$	0.54941	0.49035	0.45610	0.44487	0.45608	0.49033	0.54935

Table 5. Exact solution for boundary data $y(0)=y(1)=r_a$.

x	0.20	0.30	0.40	0.50	0.60	0.70	0.80
$r_a=0.78$	0.60995	0.56005	0.53083	0.52120	0.53083	0.56005	0.60995
$r_a=0.77$	0.58931	0.53663	0.50586	0.49574	0.50586	0.53663	0.58931
$r_a=0.76$	0.56279	0.50588	0.47276	0.46190	0.47276	0.50588	0.56279
$r_a=0.756$	0.54625	0.48618	0.45135	0.43993	0.45135	0.48618	0.54625
$r_a=0.755$	0.53930	0.47776	0.44212	0.43046	0.44212	0.47776	0.53930

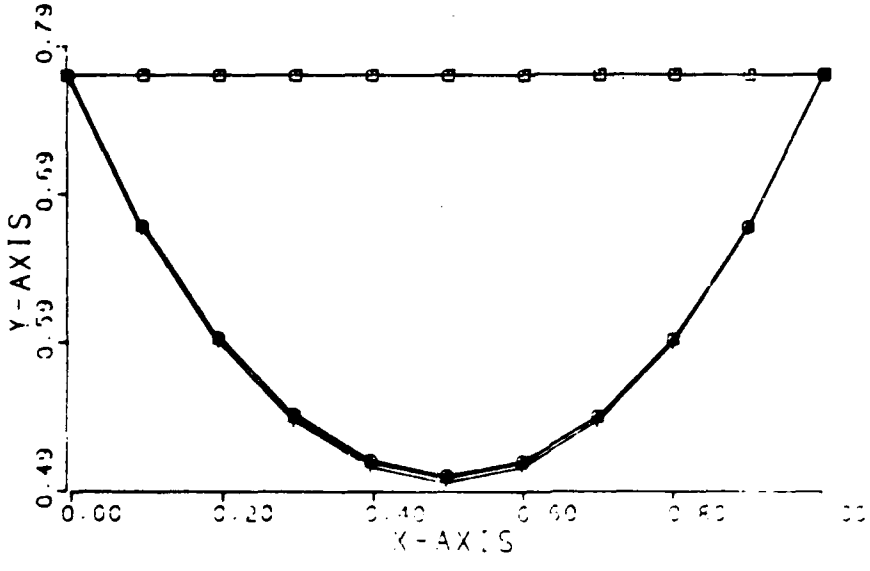
Table 6. Finite element solution for boundary data y(0)=y(1)=0.755, r=0.85, nonuniform grid.

x	0.18153	0.28769	0.39384	0.50000	0.60616	0.71231	0.81847
finite element$_{(52)}$	0.56315	0.49624	0.45751	0.44481	0.45745	0.49614	0.56307
exact	0.55374	0.48389	0.44361	0.43046	0.44361	0.48389	0.55374


FINITE ELEMENT METHOD
Y(0) = Y(1) = 0.780
--- INITIAL GUESS
--- 15TH ITERATION
--- LAST ITERATION
--- EXACT SOLUTION
0.80
0.70
0.60
0.50
Y-AXIS
0.00
0.20
X-AXIS

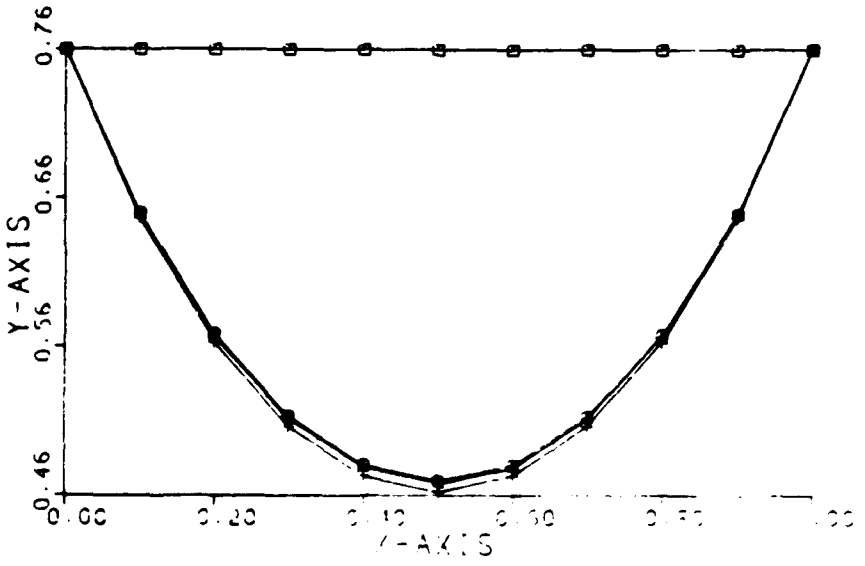
FINITE ELEMENT METHOD
Y(0) = Y(1) = 0.770

□--- INITIAL GUESS
○--- 15TH ITERATION
△--- LAST ITERATION
+--- EXACT SOLUTION



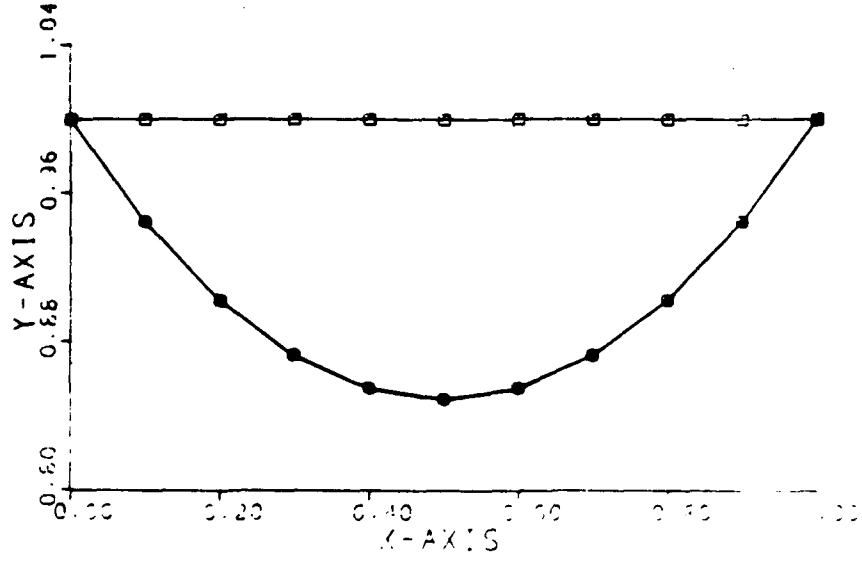
FINITE ELEMENT METHOD
Y(0) = Y(1) = 0.760

□ --- INITIAL GUESS
○ --- 15TH ITERATION
△ --- LAST ITERATION
\+ --- EXACT SOLUTION



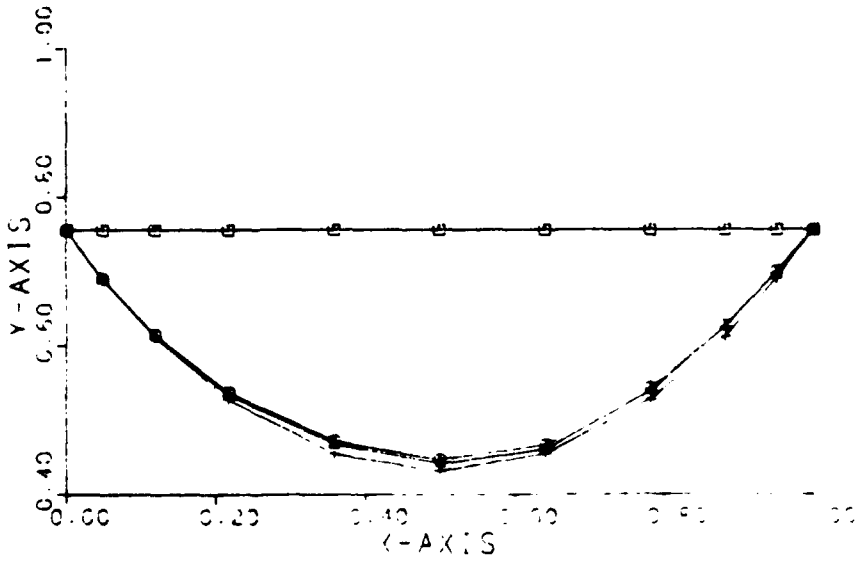
FINITE ELEMENT METHOD
Y(0) = Y(1) = 1.000

□--- INITIAL GUESS
○--- 15TH ITERATION
△--- LAST ITERATION
+--- EXACT SOLUTION



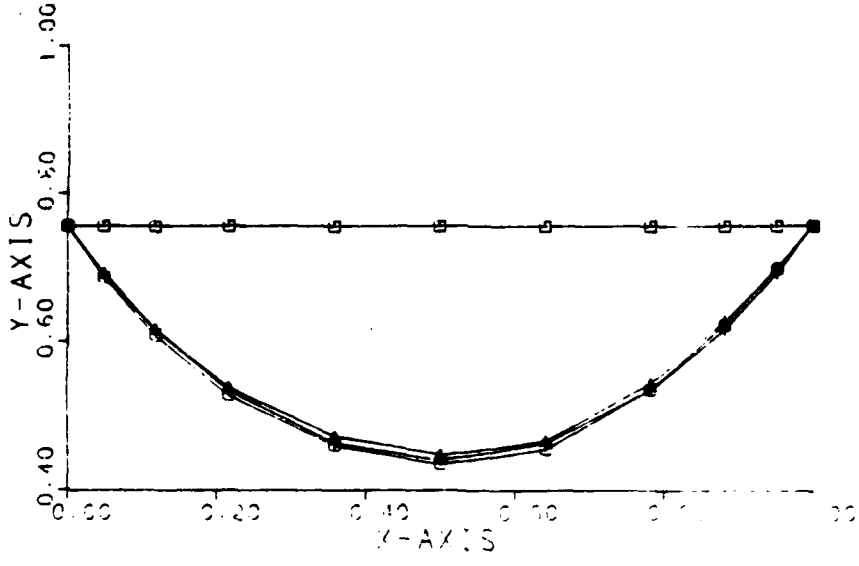
FINITE DIFFERENCE METHOD
Y(0)=Y(1)=0.755

□ --- INITIAL GUESS
○ --- 15TH ITERATION
△ --- LAST ITERATION
+ --- EXACT SOLUTION



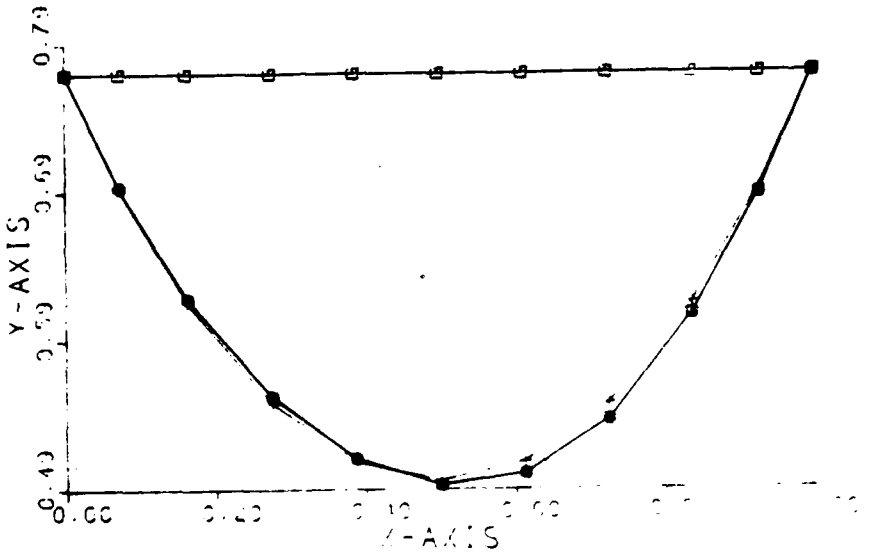
FINITE DIFFERENCE METHOD
Y(0)=Y(1)-0.756

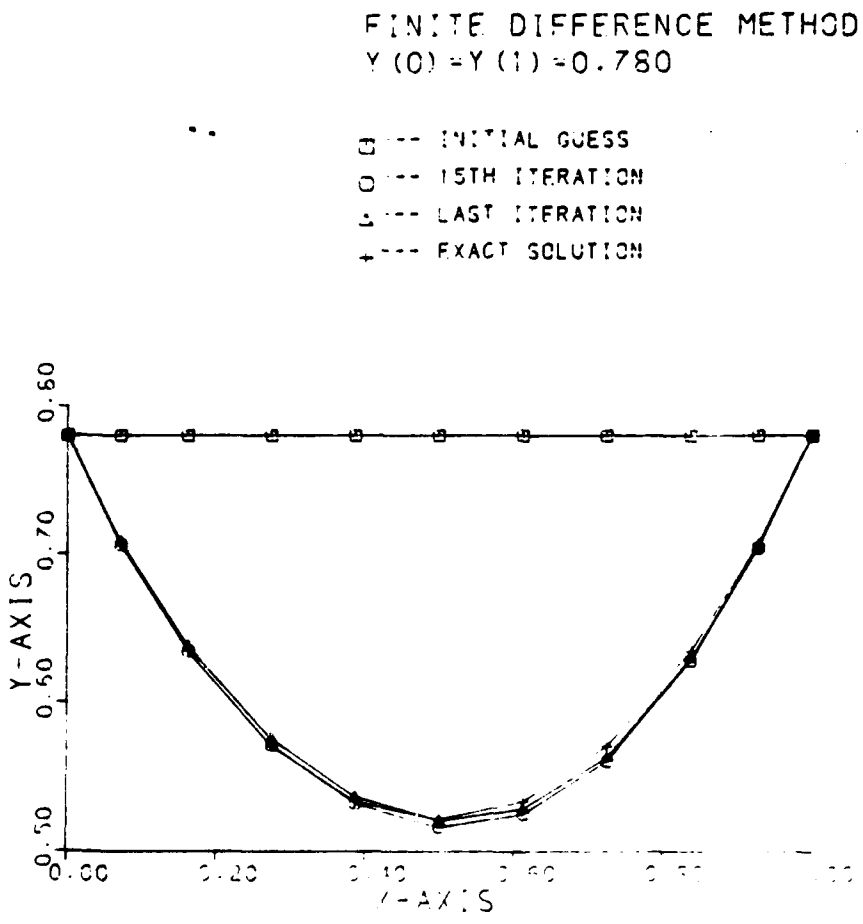
□ --- INITIAL GUESS
○ --- 15TH ITERATION
△ --- LAST ITERATION
+ --- EXACT SOLUTION



FINITE DIFFERENCE METHOD
Y(0) =Y(1) =0.770

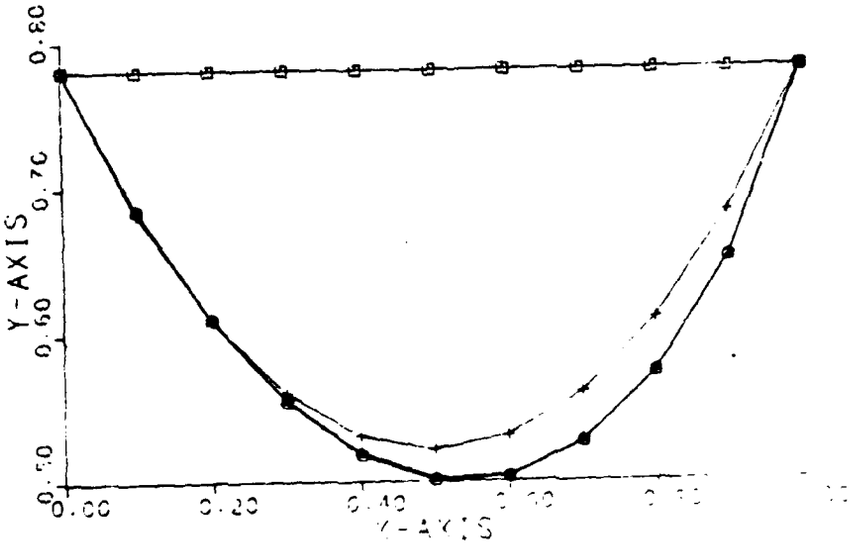
B -- INITIAL GUESS
C --- 15TH ITERATION
L --- LAST ITERATION
+ --- EXACT SOLUTION




FINITE DIFFERENCE METHOD
Y(0)=Y(1)=0.780
--- INITIAL GUESS
--- 15TH ITERATION
--- LAST ITERATION
--- EXACT SOLUTION
0.60
0.70
0.50
Y-AXIS
0.00
X-AXIS

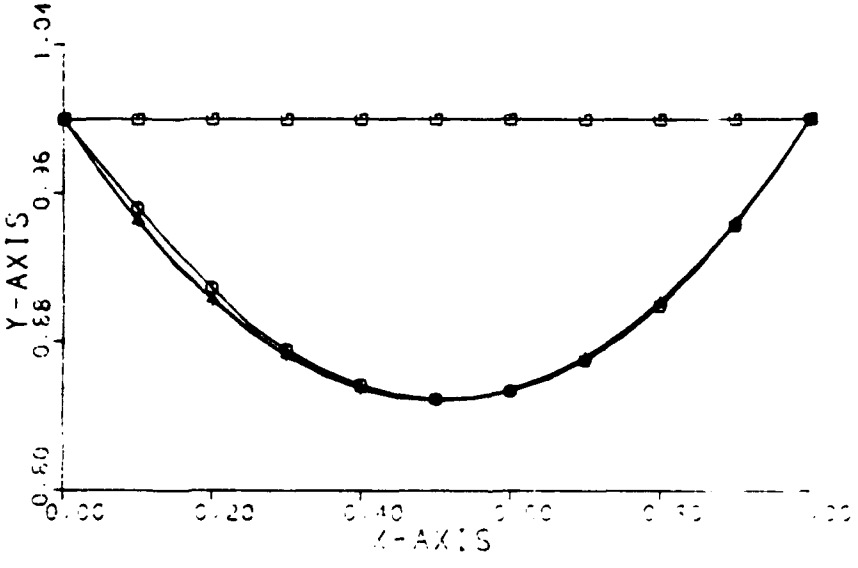
FINITE DIFFERENCE METHOD
Y(0) =Y(1) =0.780

□ --- INITIAL GUESS
○ --- 15TH ITERATION
△ --- LAST ITERATION
+ --- EXACT SOLUTION



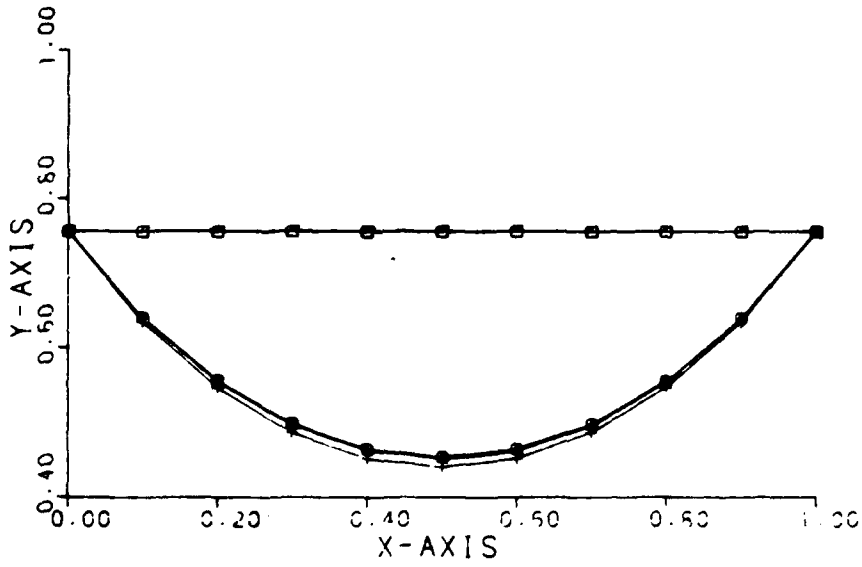
FINITE DIFFERENCE METHOD
Y(0)=Y(1)=1.000

B -- INITIAL GUESS
O --- 15TH ITERATION
* --- LAST ITERATION
+ --- EXACT SOLUTION



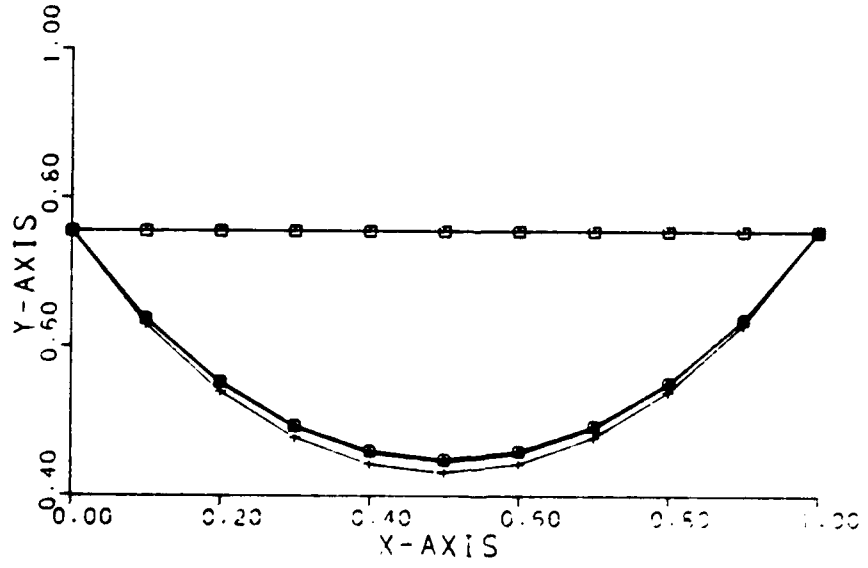
FINITE ELEMENT METHOD
Y(0) = Y(1) = 0.756

□ --- INITIAL GUESS
○ --- 15TH ITERATION
△ --- LAST ITERATION
\+ --- EXACT SOLUTION



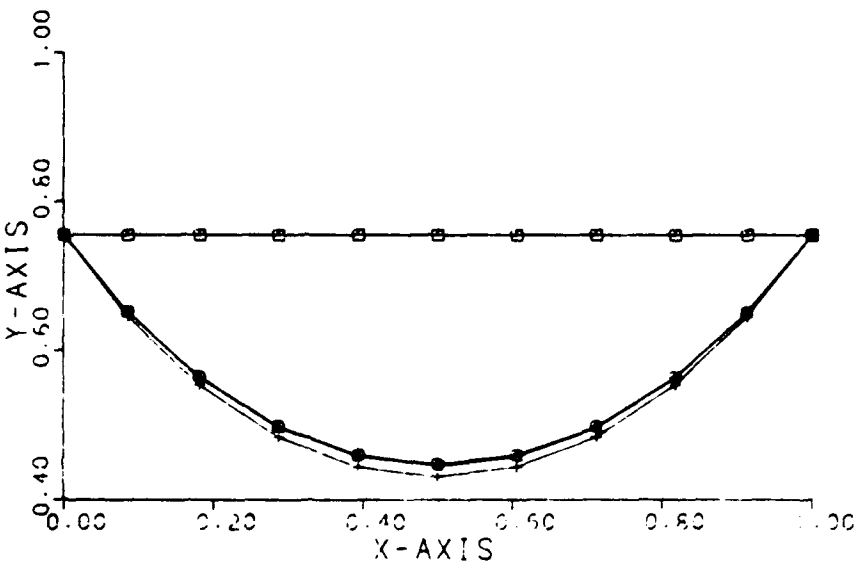
FINITE ELEMENT METHOD
Y(0) = Y(1) = 0.755

□ --- INITIAL GUESS
○ --- 15TH ITERATION
△ --- LAST ITERATION
\+ --- EXACT SOLUTION



FINITE ELEMENT METHOD
Y(0) = Y(1) = 0.755

□--- INITIAL GUESS
○--- 15TH ITERATION
△--- LAST ITERATION
+--- EXACT SOLUTION



CONSTRAINT-BASED SPATIAL PROBLEM SOLVING OF TACTICAL FORCE INTERACTIONS USING THE FUNCTIONAL BINARY DECOMPOSITION SPATIAL REPRESENTATION

Douglas Walter J. Chubb
USA CECOM Center for Signals Warfare
ATTN:AMSEL-RD-SW-TRI
Vint Hill Farms Station
Warrenton, Virginia
22186-5100

Abstract

In 1985 the author introduced the Functional Binary Decomposition (FBD) algorithm, $FBD_R \circ FBD_T$. Applied to a raster formatted spatial representation, these functions generate a set of 8-connected, 1-element regions which are provably robust. In 1987 the author developed the straight line path algorithm which uses this representation to solve spatial problems. Recently the author developed a temporal problem solver which also uses this representation. These algorithms, and others, have been developed in-house at the CECOM Center for Signals Warfare (C2SW) as part of an ongoing research effort to develop an automated system which will assist the U.S. Army Division/CORPS Intelligence Officer (G2) during his preparation of a tactical situation assessment (TSA).

During the TSA process the G2 attempts to *recognize* observed, ongoing enemy tactical plans. Fortunately, finite domain plan theory is well formulated. David Chapman, however, has shown (1987) that finite domain planning which permits action-domain modification is, although domain independent and logically consistent, undecidable. For real-world (infinite) domains, Chapman has relaxed the domain independence requirement and has conjectured that planners must instantiate local truth criteria to achieve (piece-wise) global logical consistency. Chubb (1989), however, has shown that such criteria can not exist and that furthermore the real-world planner is incapable of reliably recognizing such criteria if it did exist. Chubb has conjectured that real-world planners make special circumscription and invariance assumptions while constructing real-world plans. It is shown that these assumptions form the basis for the $FBD_R \circ FBD_T$ representation. This representation is then generalized to an N-dimensional form capable of representing conjunctive plans. A conjunctive form (CFBD) of the N-dimensional representation is next developed and spatial and temporal problem solutions generated using both the N-dimensional and CFBD representations are shown to be equivalent. A simple Army logistics problem is developed using two CFBD representations. Preliminary experimental results suggest that optimal tactical plans are ones which are maximally robust with minimal associated plan execution cost.

Introduction

Since 1984 the U.S. Army CECOM Center for Signals Warfare (C2SW) has been actively involved in the research and development of a body of mathematics which would be used to provide automated assistance to the Army division/CORPS Intelligence Officer (G2) during his preparation of a tactical situation assessment (TSA) [4,9,11,12]. The Army TSA consists of two major components: an *Order of Battle* (OB) and a *threat assessment*, or threat; the two are closely related. The OB is a description of WHAT enemy forces and equipment are present within the tactical domain of interest and WHERE they are located. The threat assessment describes WHY the enemy has his OB so configured. Early on in this research effort we noted that during the preparation of the TSA the G2 is attempting to recognize ongoing enemy plans. Since accurate plan generation and plan recognition capabilities form the basis for effective command and control, albeit military or otherwise, the development of accurate and efficient planning algorithms has become a research topic of considerable interest [1,2,3,4,8,10,13,14,15,16].

Plan generation algorithms may be classified by the rank of the planning domain, *i.e.*, finite or infinite. Several finite domain plan generation algorithms exist [1,2,8,10,14,15,16] and finite domain planning appears to be well understood. However, the extensibility of such techniques to infinite domains continues to be a research topic. Recently, D. Chapman [2] proved that nonlinear planning within finite domains which includes representations for conditional actions, dependency of effects upon input situations, or derived side effects is *undecidable*. A paradox central to any formal theory of infinite domain plan generation and recognition is how humans generate or recognize real-world plans. Chapman and Agre [1,2] have conjectured that the plan domain independence criteria be relaxed and that plan logical consistency be assured locally by subsumption of domain specific truth criteria, a type of circumscription. Chubb, however, has shown [3,4] that, with the possible exception of finite domains, the existence of such criteria can not be verified *a priori* by the planner. Chapman and others [1,2] have suggested that real-world planners improvise, doing something easy and observing results. This heuristic, however, begs the question; what discriminator is used by the planner to discern "easy" from "difficult" plans since the criteria necessary to demonstrate plan admissibility within infinite domains neither exists nor can be recognized? As a result, Chubb [3,4] has conjectured that real-world planners *must assume* the following,

C-1 (circumscription): *prior to plan action execution* the planner is aware of *all* of the domain features and associated values which may effect the desired plan action execution, and,

C-2 (invariance): *during the plan action execution* the domain will remain essentially invariant for a period of time sufficient to realize the expected value of the planner's executed plan action.

Conjectures **C-1** and **C-2** are logically equivalent to Rao and Foo's [14] *Axiom of Simple Actions*. We will show that **C-1** and **C-2** form the basis for a semantic spatial representation called a Functional Binary Decomposition (**FBD**) which is used to solve spatial and temporal problems similar to those considered by the G2 during his preparation of the TSA. The mathematical foundations for the FBD are introduced in Section I. A generalized N-dimensional **FBD** called a Conjunctive **FBD** (**CFBD**) is described in Section II, and an example of its usage is given in Section III.

I. A Planning Paradigm

A plan, **P**, is defined as a time-ordered sequence of plan actions, $\mathbf{A}_i$, $i = 1, \ldots, n$, where each action is executed by a plan actor, **AC**, within some planning domain context, $\mathbf{C}_i$. Each actor-action-context expression, $\{\mathbf{AC}, \mathbf{A}, \mathbf{C}\}_i$, is called a *plan tuple*. That is,

$$\mathbf{P} = \{\{\mathbf{AC}, \mathbf{A}, \mathbf{C}\}_1, \ldots, \{\mathbf{AC}, \mathbf{A}, \mathbf{C}\}_n\} \quad (1)$$

where $n > 0$. Each plan tuple is a description of the actor/action/context *prior to the initiation of the action by the actor*. Actions are *executed* by the plan actor within domain contexts which contain features which makes the initiation of the action possible. Action execution transforms the domain context expression into a new context expression wherein a new plan action may be executed.

The plan tuple domain context expression, $\mathbf{C}_i$, is represented by a finite set of feature tuples, $\{\{f_{1,x,y}, vf_{1,x,y}\}, \ldots, \{f_{q,r,t}, vf_{q,r,t}\}\}$. Each tuple expression contains a feature type designator, $f_{q,r,t}$ and an associated feature value, $vf_{q,r,t}$. Feature subscripts represent the *feature type* and *spatial location within* $\mathbf{C}_i$. For the purposes of this paper, we assume that $\mathbf{C}_i$ is syntactically represented as a raster-formatted matrix of pixels, c_{xy}, $x,y = 1, \ldots, m$. The *spatial location* portion of $f_{q,r,t}$ refers to the physical location within $\mathbf{C}_i$ represented by the pixel c_{rt}. Successful plans result in the development of an $\mathbf{C}_{n+1}$ context expression which contains the desired plan goal, **G**, *i.e.*, $\mathbf{G} \subseteq \mathbf{C}_{n+1}$.

Plan Action Execution

Plan action execution is the actor's reasoned manipulation of domain context and is described by the plan tuple execution function, $\boldsymbol{E}$. Let $C = \{\{f_g, vf_k\} | \forall\ g,k\}$, the set of all possible context feature types and values.

Definition: If **P** is a plan and $\{\mathbf{AC},\mathbf{A},\mathbf{C}\}_i \in \mathbf{P}$, for $i = 1, \ldots, n$, then $\boldsymbol{E}$ is a mapping from $\{\mathbf{AC},\mathbf{A},\mathbf{C}\}_i \in \mathbf{P}$ to the elements of $P(C)$, the power set of C.

By **C-1**, for each $\{\mathbf{AC}, \mathbf{A}, \mathbf{C}\}_i \in \mathbf{P}$, there exists some subset, $\beta_i \in P(C)$, where (presumably) $\beta_i \subseteq \mathbf{C}_i$, such that if $\{f_{q,r,t}, vf_{q,r,t}\} \in \beta_i$, then $\{f_{q,r,t}, vf_{q,r,t}\}$ effects the admissibility of $\boldsymbol{E}\{\mathbf{AC}, \mathbf{A}, \mathbf{C}\}_i$. Note that by **C-1**, $\boldsymbol{E}\{\mathbf{AC}, \mathbf{A}, \mathbf{C}\}_i = \boldsymbol{E}\{\mathbf{AC}, \mathbf{A}, \beta\}_i$. Fortunately, the tuple $\{\mathbf{AC}, \mathbf{A}, \beta\}_i$ is known and understood for a variety of military contexts. In particular, for military equipment (e.g., **AC** = military tank) the elements of β_i are known *a priori* since during the production and testing of the actor (tank), a variety of actor actions (movement, target location, firing, etc) are tested under strictly controlled (β_i) environmental conditions. For most military equipment, the relationship between **AC**, $\mathbf{A}_i$ and β_i has been documented and is readily available[7]. Then, given a $\{\mathbf{AC}, \mathbf{A}\}_i$, the value of β_i can be predicted a priori and may be written as a Boolean expression in terms of the feature tuples, $\{f_r, vf_r\}$ belonging to each $c_{xy} \in \mathbf{C}_i$, *i.e.*, $\{\{f_j, vf_j\} \wedge \sim\{f_g, vf_g\} \wedge \{\{f_k, vf_k\} \vee \{f_r, vf_r\}\} \ldots \vee \sim \{f_p, vf_p\}\}$.

Functional Binary Decomposition

This Boolean expression is called a **Functional Binary Decomposition** *transformation* ($\mathbf{FBD}_T$). If $\mathbf{FBD}_T$ is used as a criteria for actor action execution

admissibility, it is a mapping from $\mathbf{C}_i$ to the binary set {0,1}. That is, given some $\{\mathbf{AC}, \mathbf{A}\}_i$,

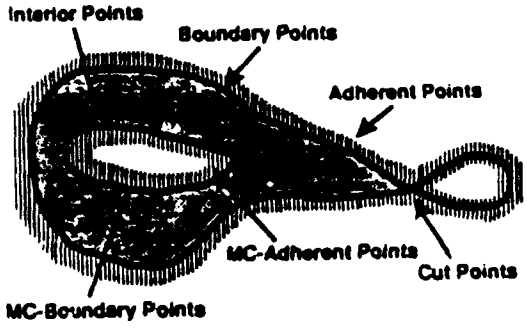
$$\mathbf{FBD}_T(c_{xy}) \rightarrow \{0,1\} \tag{2}$$

for every $c_{xy} \in \mathbf{C}_i$.

For spatial and temporal problem solving the **FBD** transformation of $\mathbf{C}_i$ is given the following properties:

1) $\mathbf{FBD}_T(c_{xy}) = 0 \Rightarrow$ action execution admissible,
2) $\mathbf{FBD}_T(c_{xy}) = 1 \Rightarrow$ action execution not admissible,
3) 0-element pixels are *4-connected*, and,
4) 1-element pixels are *8-connected*.

An **FBD** transformed $\mathbf{C}_i$ may be represented in terms of it's 1-elements, *i.e.*, $\mathbf{FBD}_T(c_{xy}) = 1$. Every path-connected set of 1-elements is called an *obstacle region* or *region* and is characterized by the **FBD** *representation* ($\mathbf{FBD}_R$). Region characterization consists of the region's *boundary list*, *adherent list*, *boundary cut points*, and *interior points* for simply connected regions. Multiply-connected (MC) regions are also characterized by their *MC-boundary lists*, *MC-adherent lists*, and *MC-boundary cut points*, *i.e.*, see Figure 1.



A multiply-connected region and it's associated **FBD** characterization.

Figure 1.

The set of domain 1-elements so characterized,

$$\mathbf{FBD}_R \circ \mathbf{FBD}_T(\mathbf{C}_i) \rightarrow \{\boldsymbol{fbd}_1, \boldsymbol{fbd}_2, ..., \boldsymbol{fbd}_k\} \tag{3}$$

where $\boldsymbol{fbd}_i$ is the *ith* 1-element region, is provably robust [6]. Representational robustness is especially important during tactical situations when spatial features may change quickly and abruptly.

The $\mathbf{FBD}_T(\mathbf{C}_i)$ binary representation is a *semantic* representation of action execution admissibility. This representation, in keeping with **C-1** and **C-2**, represents the coupling between the execution of an action and the action execution domain for a (possibly short) period of time.

II. Problem Solving Using $FBD_R \circ FBD_T(C_i)$

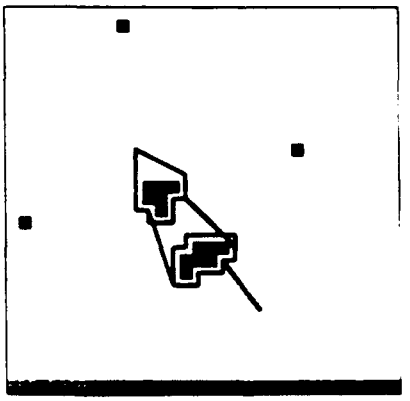
We will assume that for those actions of interest to the G2, $E\{AC, A, C\}_i$ will involve spatial movement or change. Obviously, *temporal* change occurs during action execution. Even the execution of the "wait" or "no-action" action involves temporal change. The **FBD** representation in (3) will be used to solve spatial problems, while a variant of this representation is used to solve temporal problems.

A spatial problem is a hypothesis about the spatial movement required during $E\{AC, A, C\}_i$. Action execution begins spatially and temporally at the actor's spatial location prior to action execution, i.e., some $c_{xy} \in C_i$. At the conclusion of the action execution the actor is hypothesized to be spatially located at one or more *goal* positions, $\{g_{rs}, ..., g_{kh}\}$.

Definition: A *spatial problem*, (c_{xy}, g_{rs}), is *satisfied* or *solved* if and only if there exists a 4-connected, 0-element *path*, $P(c_{xy}, g_{rs})$, between the 0-element pixels c_{xy} and g_{rs} such that every element of $P(c_{xy}, g_{rs})$ is an element of $FBD_T(C_i)$. Pixels c_{xy}, and g_{rs} are *path-connected* in $FBD_T(C_i)$.

Problem goals may be specified as either *disjunctive* or *conjunctive*, *e.g.*, $P(c_{xy}, \{g_{kh} \wedge g_{rs}\})$ requires that *both* $P(c_{xy}, g_{kh})$ and $P(c_{xy}, g_{rs})$ be satisfied.

If we assume that the time required for action execution is minimal (*i.e.*, **C-2**), then a type of spatial problem solving technique is to find one or more paths between 0-element points. Chubb's *Straight Line Path Algorithm* (SLPA) [5] makes use of the fbd_i adherent list information to efficiently search for such 0-element, 4-connected



An SLPA solution where two *fbd*, regions occlude the straight line projection between start and goal points. Note that four possible paths are generated.

Figure 2.

paths (see Figure 2). This technique, however, only provides satisfied spatial solutions and does not take into consideration the *time* required to traverse the path. A satisfied spatial problem implies that action execution is globally admissible. How-

ever, action execution may result in temporal requirements which are neither practical nor possible.

Definition: A *temporal problem*, $((c_{xy}, g_{rs}), K_{xy\text{-}rs})$ is satisfied or solved if and only if there exists a spatial solution, $P(c_{xy}, g_{rs})$, such that the time required to traverse $P(c_{xy}, g_{rs}) \leq K_{xy\text{-}rs}$, where $K_{xy\text{-}rs}$ is the problem's temporal criteria.

Since, by definition, a temporal solution is a spatial solution but the converse is not necessarily true, a more general and powerful problem solving methodology is to first develop a temporal problem solution which is then recursively constrained to produce a tenable spatial solution. For example, assume that the set A represents a temporal solution for some $((c_{xy}, g_{rs}), K_{xy\text{-}rs})$. Then there exists some $\delta \geq 0$ such that $K_{xy\text{-}rs} - \delta$ represents a temporal solution where (rank A) is *minimal*. The value of δ may be developed by recursively solving for *min*(rank A) with monotonically increasing values of δ. The algorithm which develops the temporal solution is now described.

Temporal Problem Solving

The time needed to traverse a 4-connected pixel, $c_{xy} \in \mathbf{C}_i$, is,

$$t_{xy} = R_{xy} \div (w_{xy} * V_{max}) \tag{4}$$

where R_{xy} is the side *length* of the (square) location represented by c_{xy}, V_{max} is the *maximum velocity* which the actor can move while executing this action, and w_{xy} is the *coefficient of compliance* between the domain and actor action execution where $0 < w_{xy} \leq 1.0$. The w_{xy} coefficient value indicates the percentage compliance between actor action execution and the domain, *i.e.*, $w_{gl} = 1.0$ indicates 100% compliance.

Then the time required to traverse $P(c_{xy}, g_{kh})$ is,

$$T_{xy\text{-}kh} = \sum_{i=xy}^{kh} t_i = (R_{xy} \div V_{max}) \sum_{i=xy}^{kh} (w_i)^{-1} \tag{5}$$

We (safely) assume that R_{xy} is a constant for every $c_{xy} \in \mathbf{C}_i$. V_{max} is assumed invariant during action execution. One-element pixels belonging to $\mathbf{FBD}_T(\mathbf{C}_i)$ are assigned a coefficient value of zero. Division by zero in (5) is excluded using the $\mathbf{FBD}_R(\mathbf{C}_i)$ representation for decision checks as follows.

The temporal problem solving (TPS) algorithm is a two pass algorithm. Given start and goal points c_{xy} and g_{zd}, the first pass computes the *minimal* path time for path $P(c_{xy}, h_{kh})$ where h_{kh} is a zero-element point in $\mathbf{FBD}_T(\mathbf{C}_i)$, and $h_{kh} \neq g_{zd}$. The minimal path time from h_{kh} to g_{zd} is then *estimated*. The estimated $T_{kh\text{-}zd}$ is developed by computing the *min* 4-connected path distance between points, assuming that there are no 1-element points occluding the path and that every w_i has a value of 1.0. If $min T_{xy\text{-}kh}$ + *estimated* $T_{kh\text{-}zd} \leq K_{kh\text{-}zd}$ the point h_{kh} is considered a potential temporal solution path point and is saved for second pass processing.

The second pass consists of computing $min T_{zd\text{-}kh}$ for every h_{kh} point selected during pass-one. If $min T_{xy\text{-}kh} + min T_{zd\text{-}kh} \leq K_{kh\text{-}zd}$, then the h_{kh} point must belong to at least one $P(c_{xy}, g_{kh})$ which satisfies the temporal constraint $K_{kh\text{-}zd}$. The set of all such h_{kh} becomes the TPS solution set. Likewise, a *minimal* spatial solution is developed if the w_i values are fixed as a positive, non-zero constant for all of the 0-

element points belonging to $\mathbf{FBD}_T(\mathbf{C}_i)$ and the $minK_{kh\text{-}zd}$ is developed during the problem solving process.

The Conjunctive FBD Representation

The $\mathbf{FBD}_R \circ \mathbf{FBD}_T(\mathbf{C}_i)$ possesses several inherent representational weaknesses, which arise because the pixel feature information is spatially indiscriminate. Locational information can be added, however, for essentially point sources, such as buildings. Extended line objects, such as roads and railroads, which may be represented by two or more pixels require special representation since inter-pixel trafficability may vary dramatically depending upon the features represented per pixel. In addition, pixel spatial resolution is typically less that optimal which tends to make the image of $\mathbf{FBD}_T(c_{xy})$ multi-valued!

An alternative representation methodology is to use an N-dimensional **FBD** representation where each $\mathbf{FBD}_R \circ \mathbf{FBD}_T(\mathbf{X}_{ij})$ planar image considers only a subset of features present within each $\mathbf{C}_i$, *i.e.*, $\boldsymbol{X}_{ij} \subset \mathbf{C}_i$, such that $\cup \boldsymbol{X}_{ij} = \mathbf{C}_i, j = 1, \ldots, r$. For example, $\boldsymbol{X}_{ig}$ may represent a road network, whereas $\boldsymbol{X}_{id}$ may represent bodies of water. In addition, coupled with each $\mathbf{FBD}_R \circ \mathbf{FBD}_T(\mathbf{X}_{ij})$ is a *rule-based system* which is responsible for monitoring/changing the $\boldsymbol{fbd}_h$ region binary values based upon input data normally available to the G2. For example, assume that a major road has been destroyed during a tactical engagement. This information would be given to the "road" FBD rule-based system which would create/add to the appropriate $\boldsymbol{fbd}_h$ region(s). In addition, associated with every $\mathbf{FBD}_T$ function there exists rules which monitor the associated $(r\text{-}1)$ $\mathbf{FBD}_T$ and adjust the w_i values accordingly. An N-dimensional temporal solution or path has a general form which is based upon an N-dimensional definition of 4-connectedness.

Definition: A pixel $c_{xy,k} \in \mathbf{FBD}_T(\mathbf{X}_{ik})$ is said to be *4N-connected* if and only if $c_{xy,k}$ is 4- connected to the following 0-element neighbors: $(c_{(x-1)y,k}, c_{(x+1)y,k}, c_{x(y-1),k}, c_{x(y+1),k})$ for all $k = 1, \ldots, N$.

Definition: An *N-dimensional path*, $P(c_{xy,j}, g_{zk,i})$, is defined to be the ordered list of 0-element, 4N-connected pixels which contain no loops and path-connect $c_{xy,j}$ to $g_{zk,i}$ for $i,j = 1, \ldots, N$.

The N-dimensional coplanar stack of $\mathbf{FBD}_T(\mathbf{X}_{ij})$ and their rules form the basis for a spatial representation called a *conjunctive* **FBD** (**CFBD**). The **CFBD** is developed as follows. Let $c_{xy,j}$ be the (x,y)th pixel in $\mathbf{FBD}_T(\mathbf{X}_{ij})$ Then for every pixel, $\psi_{xy} \in \mathbf{CFBD}(\boldsymbol{C}_i)$, set the value of ψ_{xy} to

$$\psi_{xy} = (c_{xy,1} \wedge c_{xy,2} \wedge, \ldots, \wedge c_{xy,r}) \tag{6}$$

the conjunctive value. We use the following symbol to show that $\mathbf{CFBD}(\boldsymbol{C}_i)$ is formed from the conjunct of the $\mathbf{FBD}_T(\mathbf{X}_{ij})$ where $j = 1, \ldots, r$,

$$\mathbf{CFBD}(\boldsymbol{C}_i) = \bigwedge_{j=1}^{r} \mathbf{FBD}_T(\mathbf{X}_{ij}) \tag{7}$$

The following theorem follows immediately.

Theorem: There exists a path $P(\psi_{xy}, \psi_{sg})$ such that every element of $P(\psi_{xy}, \psi_{sg})$ is an element of $\mathbf{CFBD}(\boldsymbol{C}_i)$ if and only if there exists an N-dimensional path, $P(c_{xy,i}, g_{sg,f})$ such that $1 \le i,f \le r$ and $\mathbf{CFBD}(\boldsymbol{C}_i) = \wedge \mathbf{FBD}_T(\mathbf{X}_{ij}), j = 1, \ldots, r$.

Proof: Follows directly from (6), (7), and the last two definitions.

We now examine an Army constraint-based logistics problem using the **CFBD** representation and the temporal problem solver.

III. A Logistics Problem Example

In this example we assume that there are three actors: (Army) *tanks*, (Army) *fuel trucks*, and a mountain *snow storm*. The domain is the northern slope of a mountain range (see Figure 3). To the north is a large lake. To the east is a large plains area. Further to the east a battle is about to begin. Two major roads are evident. One goes through the mountains, the other winds about on the northern slopes and winds about the lake. Both roads converge to the west of the eastern plains area. At the beginning of this scenario the fuel trucks are stationed in the northwestern corner of the plains, the tanks are located in the southwestern corner, to the west of the mountains, and the snow storm is about to move easterly through the mountains.

The local commander has told the tank commanders to prepare for battle along the eastern front. Army doctrine demands that the tanks assemble in a *staging area* approximately two hours prior to battle and that they be fully fueled at that time. The tanks will attempt to distribute themselves uniformly within staging area in preparation for the attack. The fuel trucks should be positioned with the tanks within the staging area to provide them with fuel. Table 1 gives the actor-V_{max} and

Actor:	Tank	Fueling Trucks	Snow Storm
V_{max} (m/m)	200	400	0-50
Start Location	(30, 2)	(1,2)	

Actors, maximum action execution velocity, and starting positions.

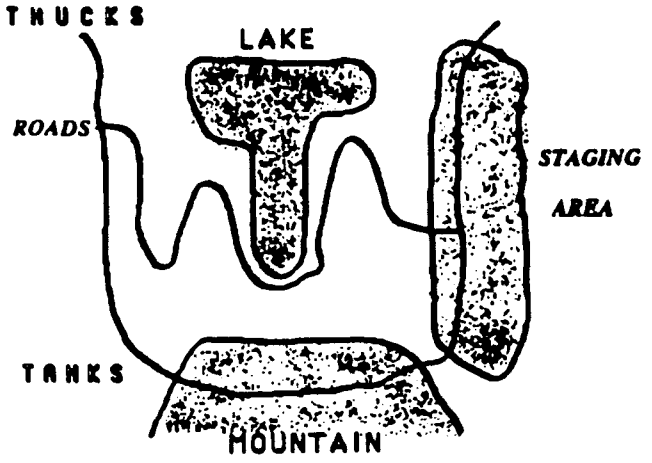
Table 1

start position data. Note that V_{max} for the storm is given as a range of values since a snow storm, though moving, may appear to be stationary to a fixed observer. In addition, once the storm has passed through an area we assume that the snow remains. The presence of snow is considered a deterrent to travel. Actor goal positions are: (4,22), (16,22), and (22, 22) which correspond respectively with the most northern, midpoint, and most southern staging area positions.

Mobility Considerations

Fueling trucks can only travel on or near the roads; most of the remaining terrain is far too rough for travel. Off-road travel is slower, depending upon terrain factors.

Tanks can move anywhere except within the mountains and the lake. Tanks can also travel on the roads however the road surface increases track wear thereby causing the tanks to move more slowly.



The domain of the logistics problem. Actors include tanks, fueling trucks, and a snow storm.

Figure 3.

A *mountain snow storm* is preparing to move southeasterly through the mountains making the mountain road impassable for both the fueling trucks and tanks.

A Logistics Problem

Assuming that both the tanks and the fuel trucks are ready to move to the staging area and that no snow is/has fallen, what is the best strategy for the tanks and trucks to use to minimize the time the tanks must wait within the staging area? How does the presence or likelihood of a mountain snow storm affect this strategy, if at all?

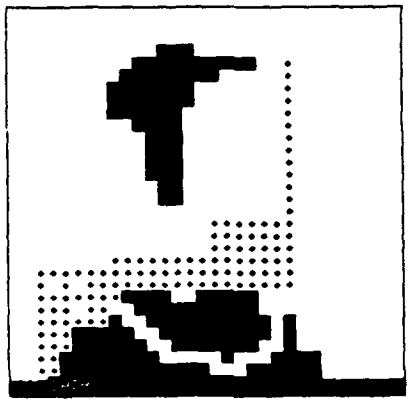
In order to solve this problem, several FBD's must be developed for this domain: a road network FBD (Figure A-1), a fuel truck FBD (Figure A-2), and a tank FBD (Figure A-3). Next, a fuel truck CFBD (Figure A-4) and a tank CFBD (Figure A-5) are developed. Finally, coefficient values are developed for the fuel truck CFBD and tank CFBD. The fuel truck coefficients are 1.0 (max weight) for road surfaces and 0.8 for off-road excursions. The tank coefficients are set to 1.0 for off-road surfaces and 0.75 for on-road surfaces. Note that both the FBD's and the coefficient values reflect a no-snow, no-traffic, "best-possible-conditions". The constraint-based model was run for every goal point with both actors (see Figures 4a,b,c, and d) with no snow present.

Table 2 lists the no-snow, minimum transit times from the tank and truck start points to the the three goal points. In this model trucks always lead the tanks by at least 2.2 minutes and no logistical problems are foreseen. Note that the minimal temporal path for the trucks to *both* (16,22) and (22, 22) is through the mountain pass. The presence of the third actor, the snow, makes the decision to use the mountain pass road less than optimal.

Start to:	(22, 22)	(16,22)	(4,22)
Tank:	16.4	19.7	26.5
Truck:	14.2(M)	15.8(M)	18.0

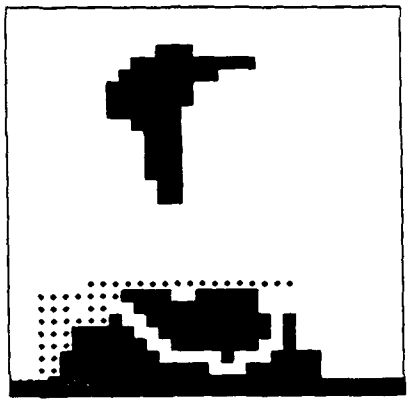
Actor minimum transit times (no snow)
Table 2.

(M) = using the southern, mountain road



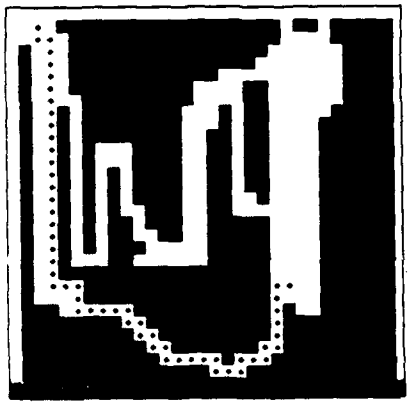
Temporal problem solver results using Tank **CFBD** representation. Points (30 2) to (4 22) yield a *min* time of 26.5 minutes.

Figure 4a



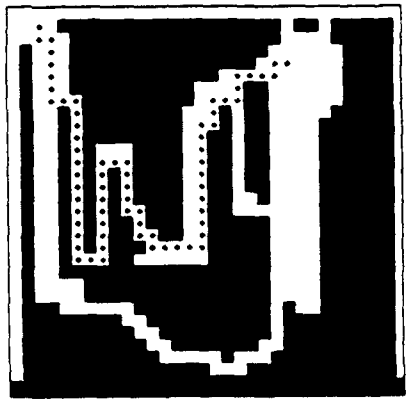
Same as Figure 4a except points are (30 2) and (22 22), the southern most point. *Min* time is 16.4 minutes.

Figure 4b



Temporal problem solver results using Fuel **CFBD** representation. Points (1 2) to (22 22) yield a *min* time of 14.3 minutes.

Figure 4c

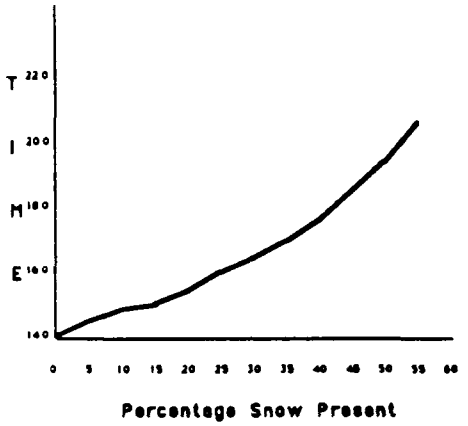


Same as Figure 4c except points are (1 2) and (4 22). *Min* time is 18.0 minutes. Note that off-road excursion is made by trucks as part of minimal time solution.

Figure 4d

Snow Present

Snow was introduced within the mountain region incrementally from no-snow present (0%) to an impassable amount of snow (100%). Mobility was simply computed to be: 1.0 - (% snow within the mountain area). Although the effect of snow upon actor mobility is linear, the presence of snow produces a *nonlinear* effect upon the the temporal computation because the spatial component (the percentage of the total road surface) affected is assumed to be unknown as is the amount of snow present within that area. Note that as each $w_i \rightarrow 0$, as in equation (5), the **FBD** characterization transforms this pixel into a region 1-element which effectively prevents any further mobility consideration.



Minimum route time for the fueling truck as a function of % mountain snow present. Note that the effect upon *min* time is nonlinear.

Figure 5

The only actor affected by the presence of snow was the fueling truck which used the most southern route (Table 2) rather than the longer valley route. Figure 5 shows the relationship between % snow present and transit time for the trucks. Note that the southern route is always shorter than the valley route until % snow exceeds 60% at which time the valley route time becomes the shortest temporal route to both (16, 22) and (22, 22) for the trucks.

Without knowing either the amount of snow present nor the percentage of the route affected by the snow, the conservative or "worst case" minimum transit times are given in Table 3.

Start to:	(22, 22)	(16,22)	(4,22)
Tank:	16.4	19.7	26.5
Truck:	20.7(V)	19.2(V)	18.0

Actor worst case minimum transit times (100% snow)

Table 3

(V) = valley route

Summary and Conclusions

It appears that the conservative logistics will service most of the tanks with fuel as they arrive in the staging area. However, the importance of this plan is not that it offers adequate logistics support but that the plan is insensitive to the mountain snow scenario; the plan is more *robust* than the other plans. The application of robust plans which are less likely to be constrained by other domain actors appears to be a powerful command strategy. We have indicated that the accuracy of a real-world planning algorithm depends upon its ability to accurately predict the interaction between action execution and the planning domain. However, the author has shown [3, 4] that for real-world domains this relationship is at best enumerable and probabilistic. At worst, the relationship appears (to the planner) to be chaotic because the planner is not aware of important domain feature data or, equivalently, because he lacks experience with similar data to make an accurate prediction. Real-world assumptions C-1 and C-2 suggest that planner experience is the most robust criteria to employ. As such, perhaps constraint-based problem solving, based upon actor experience, is best used to develop minimally constrained, maximally robust strategies as given in Table 3.

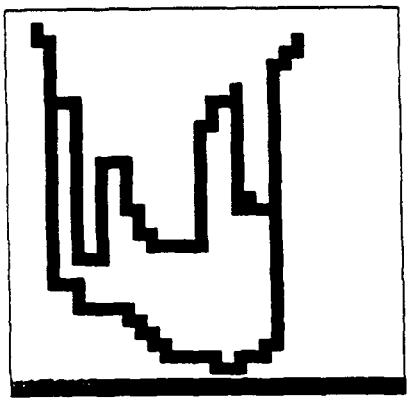
Experimental results seem to suggest that whenever two or more plans share the same goal, G, the preferred plan *maximizes* plan robustness while *minimizing* the cost of plan execution. If P1 and P2 are plans (*i.e.*, $P1 = \{(AC,A,C)_1, ..., (AC,A,C)_k\}$, $P2 = \{(AC,A,C)_1, ..., (AC,A,C)_h\}$) such that $G \subset \{C\}_{k+1}$ and $G \subset \{C\}_{h+1}$, then P1 is preferred to P2 if,

$$\sum_{P1} [robustness(AC,A,C)_i - cost[\boldsymbol{E}(AC,A,C)_i]] > \sum_{P2} [robustness(AC,A,C)_i - cost[\boldsymbol{E}(AC,A,C)_i]].$$

All of the algorithms described herein have been implemented in COMMON LISP on a Texas Instrument Explorer II AI Workstation. Appendix B contains a copy of the FBD characterization implementation, *i.e.*, equation (3).

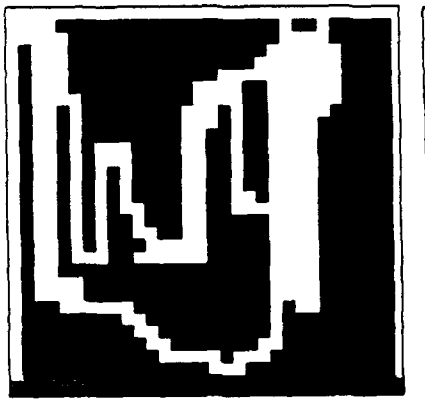
APPENDIX A

Appendix A includes **FBD** and **CFBD** figures which were generated using a *Texas Instrument Explorer II AI Workstation.*



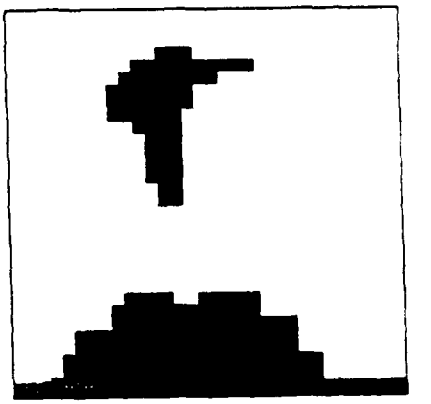
Road network **FBD**

Figure A-1



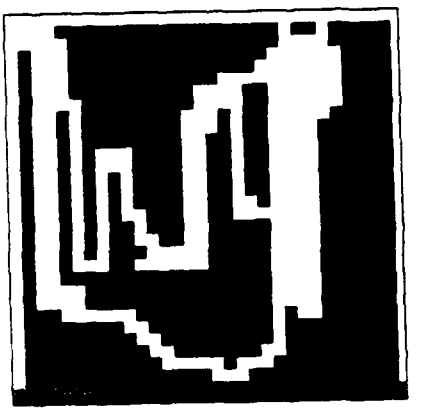
Fueling Truck **FBD**

Figure A-2



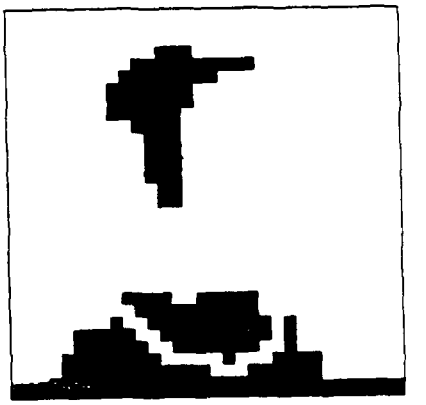
Tank **FBD**

Figure A-3



Fueling Truck **CFBD**

Figure A-4



Tank **CFBD**

Figure A-5

APPENDIX B

Appendix B contains a copy of the in-house generated **Common LISP** code used to produce a set of FBD-characterized regions, *i.e.*, equation (3). Input consists of a 1024 pixels in matrix format (32x32). The top-level function is **(MAKE-FBD R)** where $0 \leq \mathbf{R} \leq 1.0$. Function output consists of a set of *LISP atoms, i.e.,* ($\boldsymbol{fbd}_i, \boldsymbol{fbd}_{i+1}, \ldots, \boldsymbol{fbd}_k$), each with atomic properties, *i.e.,boundary list, adherent list, cut points, mc-boundary list, mc-adherent list, mc-cut points*, and the associated values as described herein.

```
;;; -*- Mode:Common-Lisp; Base:10 -*-
;
;.....................................................................................!
.
;
;   Title: FBD Program to support AISAF Situation Assessment Army Effort
;          This file contains two major programs:
;               a) ability to build a random bit array, FBD-ARRAY,
;          and  b) ability to make an FBD representation of the FBD-ARRAY.
;                  Call program with (MAKE-FBD THRSH).
;
;   Programmer:           Douglas W J Chubb
;   File:                 RACHMANINOFF:FBD-REGION;CREATE-FBD.LISP
;   Initial Development:  891018 090000
;   Last Modification:    891122 090000
;.....................................................................................!
..


(export '(xdim ydim fbd-array start-x start-y next-x next-y
          build-fbd set-display-array) 'USER)



;
; DEFVARS for program
;

(defvar xdim 32 "fbd-array width")                    ; fbd-array x-dimension
(defvar ydim 32 "fbd-array heigth")                   ; fbd-array y-dimension

(defvar fbd-array
      (make-array (list xdim ydim)
        :type 'art-1b))


(defvar thrsh 0.00)                                   ; create a 1-bit array fbd-array

(defvar start-x)                                     ; boundary start point (x,y)
(defvar start-y)
(defvar next-x)                                      ; successor point
(defvar next-y)
(defvar title 'FBD-Array-Data)
(w:make-font-purpose fonts:hl12b :flashy)

;
; Fill FBD-ARRAY with binary elements (1 or 0) depending upon value
; of random number generator.
;

(defun set-fbd-array (thrsh)
 (do ((i 0 (1+ i)))
     ((> i (1- xdim)))
 (do ((j 0 (1+ j)))
     ((> j (1- ydim)))
  (cond ((>= (random 0.99) thrsh)
           (setf (aref fbd-array i j) 1))
        (T (setf (aref fbd-array i j) 0))))) )



;
; DISPLAY-FBD creates the windows which display:
;      1. FBD-ARRAY
;      2. Boundary List
;      3. Adherent List
;      4. Interior Points
```

```
;       5. Adherent points if region multiply-connected
;       6. Boundary list if region multiply-connected.
;


(defun display-fbd (top left results)
 (let* ((mag 7)                                        ; magnification constant
        (fbd-window (make-instance 'w:window
                                   :top top       :left left
                                   :height (* mag ydim)
                                   :width (* mag xdim)
                                   :save-bits nil
                                   :borders 1
                                   :font-map '(:flashy)
                                   :label title
                                   :reverse-video-p t
                                   :expose-p t)))
       (setf fbd-bitblt (make-array (list (* mag xdim)(* mag ydim))
                          :type 'art-1b))
       (array-initialize fbd-bitblt 0)

       (do ((x 0 (1+ x)))
           ((> x (1- xdim)))
       (do ((y 0 (1+ y)))
           ((> y (1- ydim)))
       (cond ((= 1 (aref fbd-array x y))                ; magnify fbd-array results
                  (do ((i (* x mag) (1+ i)))
                      ((>= i (* (1+ x) mag)))
                  (do ((j (* y mag) (1+ j)))
                      ((>= j (* (1+ y) mag)))
                      (setf (aref fbd-bitblt i j) 1) )) )) ))

      (send fbd-window :bitblt w:alu-seta (* mag xdim) (* mag ydim) fbd-bitblt 0 0 0 0)

      (mapc (function (lambda (fbd-atom)
        (cond ((equal title 'Boundary-List)
                  (mapc (function (lambda (cpt)
                            (setf cpt (car cpt))
                            (w:make-blinker fbd-window 'w:character-blinker
                                       :font fonts:tr12b
                                       :character #\*
                                       :x-pos (* mag (second cpt))
                                       :y-pos (* mag (first cpt)) )))
                  (get fbd-atom 'cut-points)) )

              ((equal title 'M-C-Adh-Boundary-Lists)
                 (mapc #'(lambda (cp-list)
                  (mapc (function (lambda (cpt)
                            (setf cpt (car cpt))
                            (w:make-blinker fbd-window 'w:character-blinker
                                       :font fonts:tr12b
                                       :character #\*
                                       :x-pos (* mag (second cpt))
                                       :y-pos (* mag (first cpt)) )))
                  cp-list)
                ) (get fbd-atom 'mc-cut-points)) )
              (T)) )) results)

 ))

;
; MAKE-FBD is the starting routine which creates initial binary array,
; boundary list, etc.  If thrsh <= 0.0, routine being called by
; mouse routine and input data drawn, not using a random number
; generator.
;
```

```
(defun MAKE-FBD (thrsh)
 (let ((results nil))
   (setq add-regions nil)        ; add & delete variable used during
   (setq delete-regions nil)     ; TEST-FBD-ROBUSTNESS function.
   (cond ((>= thrsh 0.0)(set-fbd-array thrsh)))
   (make-frame)
   (setf title ' FBD-Array-Data)
   (display-fbd 10 300 nil)
   (do ((i 0 (1+ i)))
       ((> i (1- xdim)))
   (do ((j 0 (1+ j)))
       ((> j (1- ydim)))
   (cond ((= (aref fbd-array i j) 1)
          (setq start-x i)
          (setq start-y j)
          (setq next-x nil)
          (setq next-y nil)
          (setq results (cons (build-fbd i j (1- i) j (gensym 'fbd)) results)))) ))

   (display-results results)
    results ))

;
; DISPLAY-RESULTS sets up variables and atom properties to display
; using DISPLAY-FBD subroutine.
;

(defun display-results (results)
            (array-initialize fbd-array 0)
            (setq title 'Boundary-List)
            (set-display-array fbd-array results 'b-list)  ; display boundary points
            (display-fbd 250 150 results)

            (array-initialize fbd-array 0)
            (setq title 'Adherent-List)
            (set-display-array fbd-array results 'a-list-out) ; display adherent points
            (display-fbd 250 450 nil)

            (array-initialize fbd-array 0)
            (setq title 'Interior-Points)
            (set-display-array fbd-array results 'i-list)    ; display interior points
            (display-fbd 510 20 nil)

            (array-initialize fbd-array 0)
            (setq title 'Mul-Conn-Adherent-Lists)
            (set-display-array fbd-array results 'a-list-in) ; display adherent points
            (display-fbd 510 300 nil)

            (array-initialize fbd-array 0)
            (setq title 'M-C-Adh-Boundary-Lists)
            (set-display-array fbd-array results 'mc-list) ; display boundary points
            (display-fbd 510 580 results) )


;
; SET-DISPLAY-ARRAY sets contents of "PROP" into FBD-ARRAY for display on TV
;

(defun set-display-array (farray fbd-atom-list prop)
   (mapc (function (lambda (fbd-atm)
     (mapc (function (lambda (plst)
        (cond ((listp (first plst))
                (mapc (function (lambda (pix)
                   (setf (aref farray (first pix)(second pix)) 1))) plst) )
              (T (setf (aref farray (first plst)(second plst)) 1) )) ))
      (get fbd-atm  prop) )  )) fbd-atom-list))
```

```
;
; BUILD-FBD is the master routine which creates the FBD representation for
; each 8-connected, 1-element region in FBD-ARRAY.
;

(defun build-fbd (xp yp x2 y2 fbd-atom)
  (let* ((b-list (list (list xp yp)))
         (a-list-out nil)
         (a-list-in nil)
         (i-list nil)
         (1-element nil))
    (cond ((and next-x next-y)(setf b-list nil)))
    (mapc (function (lambda (pts)
           (cond ((and (= (aref fbd-array (first pts)(second pts)) 0)
                       (null 1-element))
                  (setf a-list-out (cons pts a-list-out)))
                 ((and (= (aref fbd-array (first pts)(second pts)) 0)
                       1-element)
                  (setf a-list-in (cons pts a-list-in)))
                 ((null 1-element)
                  (setf b-list (cons pts b-list))
                  (setf 1-element pts))                    ; save successor point
                 (T (setq i-list (cons pts i-list))))))
          (cw-scan-pattern xp yp x2 y2) )
    (cond ((null 1-element)                                ; a single 1-element point
           (setf (get fbd-atom 'b-list) b-list)
           (setf (get fbd-atom 'a-list-out) a-list-out)
           (setf (get fbd-atom 'i-list) i-list) )
          (T (setf (get fbd-atom 'b-list)
                   (append (get fbd-atom 'b-list) (reverse b-list)))
           (setf (get fbd-atom 'a-list-out)
                   (append (get fbd-atom 'a-list-out)(reverse a-list-out)))
           (setf a-list-in
                   (set-difference a-list-in (get fbd-atom 'a-list-out) :test #'equal))
           (setf (get fbd-atom 'a-list-in)
                   (union (get fbd-atom 'a-list-in) a-list-in :test #'equal))
           (setf (get fbd-atom 'i-list)
                   (set-difference (union i-list (get fbd-atom 'i-list) :test #'equal)
                                   (get fbd-atom 'b-list) :test #'equal)) ))
     (cond ((null 1-element)
                 (remove-region fbd-atom) fbd-atom)
           ((and (= xp start-x)(= yp start-y)(null next-x)(null next-y))
                 (setf next-x (first 1-element))
                 (setf next-y (second 1-element))
                 (build-fbd next-x next-y xp yp fbd-atom))
           ((and (= xp start-x)
                 (= yp start-y)
                 (= next-x (first 1-element))
                 (= next-y (second 1-element)))
                 (setf (get fbd-atom 'b-list)                    ; remove last two pts
                       (cddr (reverse (get fbd-atom 'b-list))))  ; from boundary list
                 (count-cut-points fbd-atom 'b-list 'cut-points)
                 (setf (get fbd-atom 'a-list-in)                 ; compute 0-element pts
                       (remove-duplicates
                         (set-difference (get fbd-atom 'a-list-in)
                                         (get fbd-atom 'a-list-out)
                                          :test #'equal) :test #'equal))
                 (build-interior-list fbd-atom)
                 (parse-4-space fbd-atom)
                 (rebuild-boundary-list fbd-atom)
                 (remove-region fbd-atom) fbd-atom)
           (T (build-fbd (first 1-element)(second 1-element) xp yp fbd-atom))) ))
```

```
(defun parse-4-space (fbd-atom)
  (let ((ain (remove-duplicates (get fbd-atom 'a-list-in) :test #'equal)))
  (cond (ain (setf (get fbd-atom 'a-list-in) nil)
             (parse-4-space2 fbd-atom (list (first ain)) (cdr ain))
             (order-adherent-list fbd-atom) )
        (T ) )))

;
; PARSE-4-SPACE2 parses the 4-connected ADHERENT points into a 4-connected
; subspace of 0-elements.
;

(defun parse-4-space2 (fbd-atom pt 0-elements)
 (let* ((4neigh nil)(0-list 0-elements))
  (cond ((null 0-elements)
           (setf (get fbd-atom 'a-list-in)
                 (cons  pt (get fbd-atom 'a-list-in))))
        (T (mapc (function (lambda (sps)
           (mapc (function (lambda (4pt)
              (cond ((member 4pt 0-elements :test #'equal)
                         (setf 4neigh (cons 4pt 4neigh))
                         (setf 0-list (remove 4pt 0-list :test #'equal)))
                    (T )) )) (4neighbors sps))  )) pt)
           (cond ((null 4neigh)
                    (setf (get fbd-atom 'a-list-in)
                          (cons pt (get fbd-atom 'a-list-in)))
                    (parse-4-space2 fbd-atom (list (first 0-list))(cdr 0-list)))
                 (T (parse-4-space2 fbd-atom (append pt 4neigh) 0-list))) )) ))


;
; ORDER-ADHERENT-LIST orders each 4-connected region of 0-elements
; into elements which represent the ADHERENT LIST for that region.
;

(defun order-adherent-list (fbd-atom)
 (let ((alist nil)(alist-in nil)(1-flag nil))
   (mapc (function (lambda (adherent-list)
     (setf alist nil)
     (mapc (function (lambda (adherent-point)
       (setf 1-flag nil)
       (mapc (function (lambda (8neigh)
         (cond ((= 1 (aref fbd-array (first 8neigh)(second 8neigh)))
                  (setf 1-flag T))
               (T)) )) (8neighbors adherent-point))
       (cond ((null 1-flag)(setf alist (cons adherent-point alist)) ))
        )) adherent-list)
      (setf alist-in (cons (remove-duplicates
                             (set-difference adherent-list alist :test #'equal)
                             :test #'equal)  alist-in))
       )) (get fbd-atom 'a-list-in))

   (setf (get fbd-atom 'a-list-in) nil)
   (mapc (function (lambda (adherent-list)
     (setf (get fbd-atom 'a-list-in) (cons (build-adherent-boundary adherent-list)
                                           (get fbd-atom 'a-list-in)))
     )) alist-in) ))


;
; BUILD-ADHERENT-BOUNDARY is a helper function to ORDER-ADHERENT-LIST
; function.
;

(defun build-adherent-boundary (alist)
  (let* ((aorder (copy-seq alist))(aaa nil)(boundary nil))
    (cond ((< (length alist) 3) alist)
```

```
          (T (setf aaa (stable-sort aorder #'(lambda (pt1 pt2)
                          (cond ((< (first pt1)(first pt2)) T)
                                (T NIL)) )) )
             (setf aorder (stable-sort aaa #'(lambda (pt1 pt2)
                          (cond ((and (= (first pt1)(first pt2))
                                      (< (second pt1)(second pt2))) T)
                                (T NIL)) )) )

             (setf start-x (first (car aorder)))
             (setf start-y (second (car aorder)))
             (setf next-x nil)
             (setf next-y nil)
             (setf boundary (list (list start-x start-y)))
             (build-adherent-b2 start-x start-y (1- start-x) start-y aorder boundary) ))))
)

;
; BUILD-ADHERENT-B2 is another helper function to BUILD-ADHERENT-BOUNDARY
; function.
;

(defun build-adherent-b2 (xp yp x2 y2 splist bblist)
  (let ((next nil))
     (mapc #'(lambda (sppt)
         (cond ((and (null next)(member sppt splist :test #'equal))
                     (setf next sppt)
                     (setf bblist (cons sppt bblist)) )
               (T nil)) )(cw-4-scan xp yp x2 y2) )

     (cond ((null next)(print "error"))
           ((and (= xp start-x)
                 (= yp start-y)
                 (null next-x)
                 (null next-y)) (setf next-x (first next))
                                (setf next-y (second next))
                                (build-adherent-b2 (first next)(second next)
                                                   xp yp splist bblist))
           ((and (= xp start-x)
                 (= yp start-y)
                 (= next-x (first next))
                 (= next-y (second next))) (reverse (cddr bblist)))

           (T (build-adherent-b2 (first next)(second next)
                                 xp yp splist bblist))) ))


;
; CW-4-SCAN generates a 4-connected, clock-wise scan pattern
;

(defun cw-4-scan (xp yp x2 y2)
 (let* ((4pts (4neighbors (list xp yp)))
        (cwlst (cdr (member (list x2 y2) 4pts :test #'equal))))
     (append cwlst (set-difference 4pts cwlst :test #'equal)) ))


;
; 4NEIGHBORS generates the 4-connected neighborhood.
;

(defun 4neighbors (el)
  (let* ((x (first el))(y (second el)))
    (list (list (1- x) y)(list x (1+ y))
          (list (1+ x) y)(list x (1- y))) ))

;
; REBUILD-BOUNDARY-LIST is master routine used to generate the multiply-connected
```

```
; boundary lists.  This routine uses the ordered ADHERENT LISTS in property
; "a-list-in" i.e., Adherent-List-inside-region, to generate the boundary lists.
;

(defun rebuild-boundary-list (fbd-atom)
  (cond ((null (get fbd-atom 'a-list-in)) )
        (T (rebuild-boundary-list3 fbd-atom)
           (cond ((null (get fbd-atom 'mc-list)) )
                 ((listp (first (first (get fbd-atom 'mc-list)))) )
                 (T (setf (get fbd-atom 'mc-list)
                          (list (get fbd-atom 'mc-list))) ))
           (count-cut-points fbd-atom 'mc-list 'mc-cut-points)
  )))



;
; REBUILD-BOUNDARY-LIST3 is the "work-horse' function for generating the
; multiply-connected "inner" boundary lists.
;

(defun rebuild-boundary-list3 (fbd-atom)
 (let* ((iblist nil)(ib2 nil)(fpt nil)(pattern nil)(sp-flag nil))
   (mapc #'(lambda (adlist)
    (setf iblist nil)
    (cond ((> (length adlist) 1)
             (setf fpt (first adlist))
             (mapl #'(lambda (adherent)
                (cond ((= 1 (length adherent))
                          (setf pattern (cw-scan-pattern
                                          (first  fpt)
                                          (second fpt)
                                          (first  (first adherent))
                                          (second (first adherent))) ))
                      (T (setf pattern (cw-scan-pattern
                                          (first  (second adherent))
                                          (second (second adherent))
                                          (first  (first adherent))
                                          (second (first adherent))) )) )
                (cond ((> (length adherent) 2)
                          (setf pattern (set-difference pattern
                                          (member (third adherent) pattern :test #'equal)
                                          :test #'equal)) ))

                (setf sp-flag nil)
                (setf ib2 nil)
                (mapc #'(lambda (ng)
                  (cond ((and (null sp-flag)
                              (or (member ng (get fbd-atom 'i-list) :test #'equal)
                                  (member ng (get fbd-atom 'b-list) :test #'equal)))
                         (setf ib2 (cons ng ib2)) )
                        ((and ib2
                              (null sp-flag)
                              (member ng adlist :test #'equal))
                         (setf sp-flag T))
                        (T )) ) pattern)
                (setf ib2 (reverse ib2))
                (cond ((and (> (length iblist) 1)
                            (equal (first ib2)(second (reverse iblist)))
                            (equal (second ib2)(first (reverse iblist))))
                          (setf ib2 (cddr ib2)) ))
                (setf iblist (append iblist ib2))
              )  adlist) )

           (T (mapc #'(lambda (ng)
                 (cond ((or (member ng (get fbd-atom 'i-list) :test #'equal)
                            (member ng (get fbd-atom 'b-list) :test #'equal))
```

```
                        (setf iblist (cons ng iblist)) )
                      (T )) ) (8neighbors (first adlist))) ))

     (cond (iblist (setf (get fbd-atom 'mc-list)
                         (cons (setf ib2 (remove-dups-sequences iblist))
                               (get fbd-atom 'mc-list))) )
           (T (print "ERROR..Rebuild-Boundary-List2")))
    ) (get fbd-atom 'a-list-in))  ))

;
; When the MC boundaries are first generated there are (possibly) numerous
; duplicates, both point-wise and patterns of points.  The following routines
; effectively remove these patterns.
;

(defun remove-dups-sequences (mclist)
 (let ((mc2 mclist)(duup nil))
    (cond ((and (> (length mclist) 1)
                (equal (first mclist)
                       (first (reverse mclist))))
           (remove-dups-sequences (cdr mclist)) )
          (T

    (mapl #'(lambda (mc-mc)
     (cond ((= (length mc-mc) 1)(setf duup (cons (first mc-mc) duup)) )
           ((equal (first mc-mc)(second mc-mc)) )
           (T (setf duup (cons (first mc-mc) duup))) ) ) mc2)

    (cond ((= (length mc2)(length duup))
              (remove-many-items (remove-dup-tails mc2)))
          (T (remove-dups-sequences (reverse duup)) )) )) ))



(defun remove-dup-tails (mc2)
 (let ((rmc2 (reverse mc2)))
  (cond ((and (> (length mc2) 3)
              (equal (first mc2)(second rmc2))
              (equal (second mc2)(first rmc2)))
         (remove-dup-tails (cddr mc2)) )
        ((and (> (length mc2) 5)
              (equal (first mc2)(third rmc2))
              (equal (second mc2)(second rmc2))
              (equal (third mc2)(first rmc2)) )
         (remove-dup-tails (cdddr mc2)))
        (T mc2)) ))



(defun remove-many-items (mc3)
 (let ((duf nil) (skip-flag 0))
  (cond ((< (length mc3) 4) mc3)
        (T
    (mapl #'(lambda (mc22)
      (cond ((> skip-flag 0)(setf skip-flag (1- skip-flag)))
            (T
              (cond ((< (length mc22) 4)
                        (setf duf (cons (first mc22) duf)) )
                    ((and (< (length mc22) 6)
                          (equal (first mc22)(third mc22))
                          (equal (second mc22)(fourth mc22)))
                     (setf skip-flag 1))
                    ((< (length mc22) 6)
                       (setf duf (cons (first mc22) duf)))
                    ((and (equal (first mc22)(fourth mc22))
                          (equal (second mc22)(fifth mc22))
```

```
                        (equal (third mc22)(sixth mc22)))
                    (setf skip-flag 2))
                ((and (equal (first mc22)(third mc22))
                      (equal (second mc22)(fourth mc22)))
                   (setf skip-flag 1))
                (T (setf duf (cons (first mc22) duf)))))))) mc3)

    (cond ((= (length duf)(length mc3)) mc3)
          (T (remove-many-items (reverse duf)))) )) ))

;
; 8NEIGHBORS generates the 8-connected neighborhood of points.
;

(defun 8neighbors (pixel)
 (let ((x (first pixel))(y (second pixel)))
   (list (list (1- x) y)(list (1- x)(1+ y))
         (list x (1+ y))(list (1+ x)(1+ y))
         (list (1+ x) y)(list (1+ x)(1- y))
         (list x (1- y))(list (1- x)(1- y))) ))

;
; BUILD-INTERIOR-LIST is master routine for discovering 1-element region
; interior points.  Method used is "region-growing".
;

(defun build-interior-list (fbd-atom)
 (cond ((null (get fbd-atom 'i-list)) )
       (T (interior-hunt fbd-atom (get fbd-atom 'i-list)) )) )



(defun interior-hunt (fbd-atom intlist)
  (let* ((neighborhood nil)(i-flag nil))
    (mapc (function (lambda (ipoint)
      (setf neighborhood (interior-point-check fbd-atom ipoint))
      (cond (neighborhood (setf (get fbd-atom 'i-list)
                                (union neighborhood (get fbd-atom 'i-list)
                                                    :test #'equal))
                          (setf i-flag (append i-flag neighborhood))  )
            (T))  ))  intlist)
  (cond (i-flag
            (setf i-flag (remove-duplicates i-flag :test #'equal))
            (setf (get fbd-atom 'i-list)
                  (remove-duplicates (get fbd-atom 'i-list) :test #'equal))
            (interior-hunt fbd-atom i-flag))
        (T )) ))


(defun interior-point-check (fbd-atom ip)
  (let* ((scan nil)(neighbor nil))
    (setf scan (cw-scan-pattern (first ip)(second ip)(1- (first ip))(second ip)))
    (mapc (function (lambda (bip)
      (cond ((and (= (aref fbd-array (first bip)(second bip)) 0)
                  (not (member bip (get fbd-atom 'a-list-in) :test #'equal))
                  (not (member bip (get fbd-atom 'a-list-out) :test #'equal)))
                    (setf (get fbd-atom 'a-list-in)(cons bip (get fbd-atom 'a-list-in))) !
)
            ((= (aref fbd-array (first bip)(second bip)) 0)  )
            ((or (member bip (get fbd-atom 'i-list) :test #'equal)
                 (member bip (get fbd-atom 'b-list) :test #'equal)) )
            (T (setf neighbor (cons bip neighbor)) )) ))  scan)
    neighbor ))

;
; CW-SCAN-PATTERN returns a cw scan list of 8-neighborhood points about
; point (x1,y1) starting at the first cw-position from point (x2,y2).
```

```
; The last point returned is point (x2,y2).
;

(defun cw-scan-pattern (x1 y1 x2 y2)
 (let ((cw-scan (list (list (1- x1) y1)(list (1- x1)(1+ y1))
                      (list x1 (1+ y1))(list (1+ x1)(1+ y1))
                      (list (1+ x1) y1)(list (1+ x1)(1- y1))
                      (list x1 (1- y1))(list (1- x1)(1- y1)))))
 (append (cdr (append (member (list x2 y2) cw-scan :test #'equal)
                 (set-difference cw-scan (member (list x2 y2) cw-scan :test #'equal))))
         (list (list x2 y2))) ))


;
; COUNT-CUT-POINTS discovers boundary list cut points.  Once found the points are
; annotated with the number of new regions which will be established if that
; 1-element is changed to a 0-element.
;

(defun count-cut-points (atm prop indicator)
  (let* ((c-list (get atm prop))(c-lister nil)(cut-points nil)(ncuts 0)(rest-cuts nil))
    (mapc (function (lambda (pt)
      (cond ((listp (first pt))
               (setf cut-points nil)
               (setf c-lister nil)
               (setf rest-cuts (list-cut pt 1))
               (mapc (function (lambda (pt2)
                 (setf ncuts (- (length pt)(length (remove pt2 pt :test #'equal))))
                 (cond ((and (> ncuts 1)
                             (not (member pt2 cut-points :test #'equal))
                             (list-cut (remove pt2 pt :test #'equal)
                                       (+ ncuts
                                          (length (remove pt2 rest-cuts :test #'equal))))!
 )
                        (setf cut-points (cons pt2 cut-points))
                        (setf c-lister (cons (cons pt2 ncuts) c-lister)) ))
               )) pt)

               (cond (rest-cuts
                       (mapc #'(lambda (rcut)
                         (cond ((member rcut cut-points :test #'equal) )
                               ((and (= 1 (- (length pt)
                                             (length (remove rcut pt :test #'equal))))
                                     (> (length (member rcut pt :test #'equal)) 1)
                                     (member (second (member rcut (reverse pt) :test #'eq!
ual))
                                             (8neighbors (second (member rcut pt :test #'!
equal)))
                                             :test #'equal))
                                (setf cut-points
                                  (cons
                                    (second (member rcut (reverse pt) :test #'equal))
                                    cut-points))
                                (setf c-lister (cons (cons (first cut-points) 2) c-list!
er)))
                               (T))) rest-cuts)))

               (setf (get atm indicator)(cons c-lister (get atm indicator))) )

             (T (setf ncuts (- (length c-list)(length (remove pt c-list :test #'equal))))
                (cond ((and (not (member pt cut-points :test #'equal))(> ncuts 1))
                       (setf cut-points (cons pt cut-points))
                       (setf (get atm indicator)(cons (cons pt ncuts)(get atm indicator)!
)) ))
              ) ) )) c-list)
     (setf (get atm indicator) (reverse (get atm indicator))) ))
```

```
;
; LIST-CUT checks to see if multiple point in boundary list is cut-point
; and if any other points are potential cut points.
;

(defun list-cut (mcl ncuts)
 (let ((cut-flag nil))
  (mapl #'(lambda (mc2)
    (cond ((and (= (length mc2) 1)
                (not (member (first mcl)(8neighbors (first mc2)) :test #'equal)))
           (setf cut-flag (cons (first mc2) cut-flag)) )
          ((= (length mc2) 1) )
          ((not (member (second mc2)(8neighbors (first mc2)) :test #'equal))
           (setf cut-flag (cons (first mc2) cut-flag)) )
          (T )) ) mcl)
  (cond ((>= (length cut-flag) ncuts) cut-flag)
        (T nil)) ))


;
; REMOVE-REGION removes 1-element points from FBD-ARRAY once the region
; has been completely characterized by an equivalent FBD expression.
;

(defun remove-region (atm)
 (mapc (function (lambda (pts)
    (setf (aref fbd-array (first pts)(second pts)) 0)
   )) (get atm 'a-list) )
 (mapc (function (lambda (pts)
     (setf (aref fbd-array (first pts)(second pts)) 0)
    )) (get atm 'b-list) )  )



;
; MAKE-FRAME creates a 0-element frame about the FBD-ARRAY.
;

(defun make-frame ()
  (do ((i 0 (1+ i)))
      ((> i (1- xdim)))
   (do ((j 0 (1+ j)))
       ((> j (1- ydim)))
    (cond ((or (= j 0)(= j (1- ydim))(= i 0)(= i (1- xdim)) )
           (setf (aref fbd-array i j) 0))) )))
```

Bibliography

[1] P.E. Agre, "*The Dynamic Structure of Everyday Life*", MIT AI Tech Report #AI-TR-1085, 1988.

[2] D. Chapman, "*Planning for Conjunctive Goals*", MIT AI Tech Report #AI-TR-0802, 1985.

[3] D.W.J. Chubb, "*Transitioning Mechanized Plan Recognition from Closed to Real-World Domains*", Proc.of the SPIE International Society for Optical Engineering, Sensor Fusion II: Human and Machine Strategies, pp. 538-549, 1989.

[4] D.W.J. Chubb, "*A Proof Of Chapman's Conjecture for Tactical Conjunctive Plan Recognition*", Proc. of the 1989 Tri-Service Data Fusion Symposium, Tech. Vol. I, pp. 93-102, 1989.

[5] D.W.J. Chubb, "*An Introduction and Analysis of a Straight Line Path Algorithm for Use in Binary Domains*", Proc. 1987 Workshop on Spatial Reasoning and Multi-Sensor Fusion, pp. 220-229, 1988.

[6] D.W.J. Chubb and D. Levine, "*A Proof of Robustness of the Functional Binary Decomposed Semantic Spatial Representation*", C2SW TR86-1, 1986.

[7] Jane's "*Military Vehicles and Ground Support Equipment*", 5th Edition, Ed. Christopher F. Foss and Terry J. Gander, 1984.

[8] H. Kautz and J. Allen, "*Generalized Plan Recognition*", Proc. of the Sixth National Conference on Artificial Intelligence, pp. 32-38, 1986.

[9] D. Levine, "*A Design for a Battlefield Situation Assessment System*", LNK Corporation, Silver Spring, MD., 1985.

[10] P. Maes, "*The Dynamics of Action Selection*", Proc. of the Eleventh International Joint Conference on Artificial Intelligence, Detroit, MI, Vol. 2, pp. 991-997, 1989

[11] J. A. Maier and S.C. Williams, "*An Improved Artificial Intelligence-Based Paradigm for Tactical Battlefield Situation Assessment*", The Analytic Sciences Corporation, TR-5399-3, 1989.

[12] D.V. McDermott, "*A Prototype System for Automated Tactical Situation Assessment*", Yale University U/CSD/RR#674, 1989.

[13] D.V. McDermott, "*A Temporal Logic for Reasoning About Processes and Plans*", Cognitive Science, 6(2), 1982.

[14] A.S. Rao and N.Y. Foo, "*Minimal Change and Maximal Coherence: A Basis for Belief Revision and Reasoning about Actions*", Proc. of the Eleventh Internation Joint Conference on Artificial Intelligence, Vol. 2, pp. 966-968, 1989.

[15] G.J. Sussman, "*A Computational Model of Skill Acquisition*", MIT AI Tech Report #AI-TR-297, 1973.

[16] R. Willensky, "*Planning and Understanding*", Addison-Wesley, Reading, MA., 1983.

GEOMETRIC REASONING FOR RECOGNITION OF THREE DIMENSIONAL OBJECT FEATURES * †

M. Marefat and R. L. Kashyap
School of Electrical Engineering
Purdue University, West Lafayette, IN

ABSTRACT. A method for extracting manufacturing shape features from the boundary representation of a polyhedral object is presented. In this approach, the depressions of the part are represented as cavity graphs which are in turn used as a basis for hypothesis generation-elimination. The proposed cavity graphs are an extended representation, in which the links reflect the concavity of the intersection between two faces, and the node labels reflect the relative orientation of the faces comprising the depression. The hypotheses are generated by decomposition of the cavity graphs into maximal constituents. The incorrect hypotheses are eliminated by rule-based experts which can discard a hypothesis or opportunistically improve and propose it for reexamination.

Emphasis is put on automatic analysis of depressions which are formed by interactions of primitive features, because previous methods have limited success in handling interactions. It is shown that although there is a unique subgraph for each primitive feature, every cavity graph does not correspond to a unique set of primitive features. Consequently, since the cavity graph of a depression may not be the union of the representations for the involved primitives, we introduce the concept of virtual links for the formal analysis of the depressions based on cavity graphs. Finally a suitable method for automatic determination of the virtual links is presented. This method is based on combining topologic and geometric evidences, and uses a combination of Dempster-Shafer decision theory and clustering techniques to reach its conclusions. Experimental results for a number of examples, which are not correctly analyzed by previous systems, are presented through out the paper, and implementation details are discussed.

1. INTRODUCTION. Understanding the shape of an object is essential in CAD and CAM to automate the link between design and manufacturing, and is important in computer vision for recognizing objects based on their properties. The traditional CAD description of the design of a prismatic part represents its geometry as a data structure involving faces, edges, and vertices. However to manufacture a part, we need to describe it in terms of the higher level semantic features such as slots, holes, and pockets because the machining operations create only these features and not edges, faces or vertices. Thus this higher level feature information should be extracted from the CAD description. The problem is then to find a useful description of the object in terms of the shape features given a boundary representation of the object in terms of face, edge, and vertex entities.

Several approaches have been proposed for recognition of object features from CAD data. Woo [16] used convex hull techniques to describe the object as alternating sums of volumes. Kyprianou [9] and Staley [13] applied syntactic pattern recognition methods to classify depressions. Henderson [6] and Kung [8] used logic programming and expert systems to extract shape features while Joshi [7] developed Attributed Adjacency Graphs for a part. De Floriani [5] uses connectivity properties to classify features into DP-features (protrusions or depressions) and H-features(through holes or handles). While most of the above methods use the boundary description of the object as input, Lee and Fu [10] propose algorithms for extraction and unification of some features from a CSG tree. However, previous methods have very limited success in recognition of interacting features because:

(i) Interaction between primitives produces different versions of a primitive. There is no unique representation for all different occurrences of a primitive feature.

* Supported by the U.S. Army Research Office
† A detailed discussion of the reported research appears in the October 1990 issue of IEEE Transactions on Pattern Analysis and Machine Intelligence.

(ii) During the course of interaction between primitives, faces of a primitive feature may be divided into several disconnected components thus creating new faces. Furthermore, two intersecting faces of a primitive (when the primitive is isolated) may not be adjacent in a compound feature containing the interaction of the primitive with another primitive.

In this paper we describe a novel approach for identifying and recognizing the primitive features in the depressions of a polyhedral machining part. Figure 1 illustrates the overall approach for our proposed method. It takes advantage of graph matching, expert system, and reasoning with uncertainty techniques. Briefly the contributions of this work can be summarized as follows:

(i). A new graph representation for the primitive shape features of an object is introduced. In this representation each face is represented by one face, the node labels determine the relative orientation of faces of the object in space, and the links determine the non-convexity between two faces.

(ii). A method for extracting primitive shape features of an object even when these primitives are interacting is developed. This method uses hypothesis generation-elimination approach. The hypotheses are generated by decomposing the cavity graphs of the object into maximal subgraph constituents, and the incorrect hypotheses are eliminated by rule-based experts which evaluate each hypothesis.

(iii). The concept of virtual links for the graph representation of an object is introduced. Because the graph of a depression is not necessarily the union of the representations for the involved primitives, to include the correct hypotheses in the hypothesis space, we show that unification to merge the nodes of a group of separated faces, and augmenting cavity graphs with virtual links form powerful shape analysis techniques.

(iv). We develop a method of reasoning with uncertainties to determine the virtual links to be augmented to a cavity graph. Virtual links can be determined by combining geometric and topologic evidences. The shape primitives found based on these techniques may be interpreted as the primitives most probably comprising the depression.

The remainder of this paper describes the details of our methods and of the prototyped system implemented. In the next section we describe the primitive features and methods for extracting them when they are not interacting. Section three discusses interaction between primitives. Section four describes generation of hypotheses and the reasoning strategy to select the correct subset of them. Sections five and six are devoted to presenting the concept of virtual links, the methods for determining the most probable features in a cavity, and the gathering and combining processes of topologic and geometric evidences. Section seven discusses verification. Implementation is briefly presented in section eight, and discussion and experimental results from a prototyped system interfaced to the PADL/2.0 [1] solid modeling system showing the extraction of primitive features from some parts that are not correctly analyzed by the previous systems are presented in section nine.

2. PRIMITIVE FEATURES.

Faces of a part are usually machined in groups. These groups form primitive machining features, such as pockets, slots, and steps, because chunks of manufacturing knowledge are associated with each group. A primitive shape feature (or primitive feature) of an object may informally be defined as a connected set of faces from its boundary with semantic meaning for accomplishing a desired task. Figure 2 shows the primitive features we considered. One may note that each depicted primitive represents a family of shape features since concave and convex angles may assume any value in the concave or convex range. Each member of the family is obtained by selecting a different combination of face intersection angles from the appropriate range such that the combination is physically realizable in 3D. We selected shape features that are important for machining a part, but the methods we use may be applied to shape features of interest in other areas.

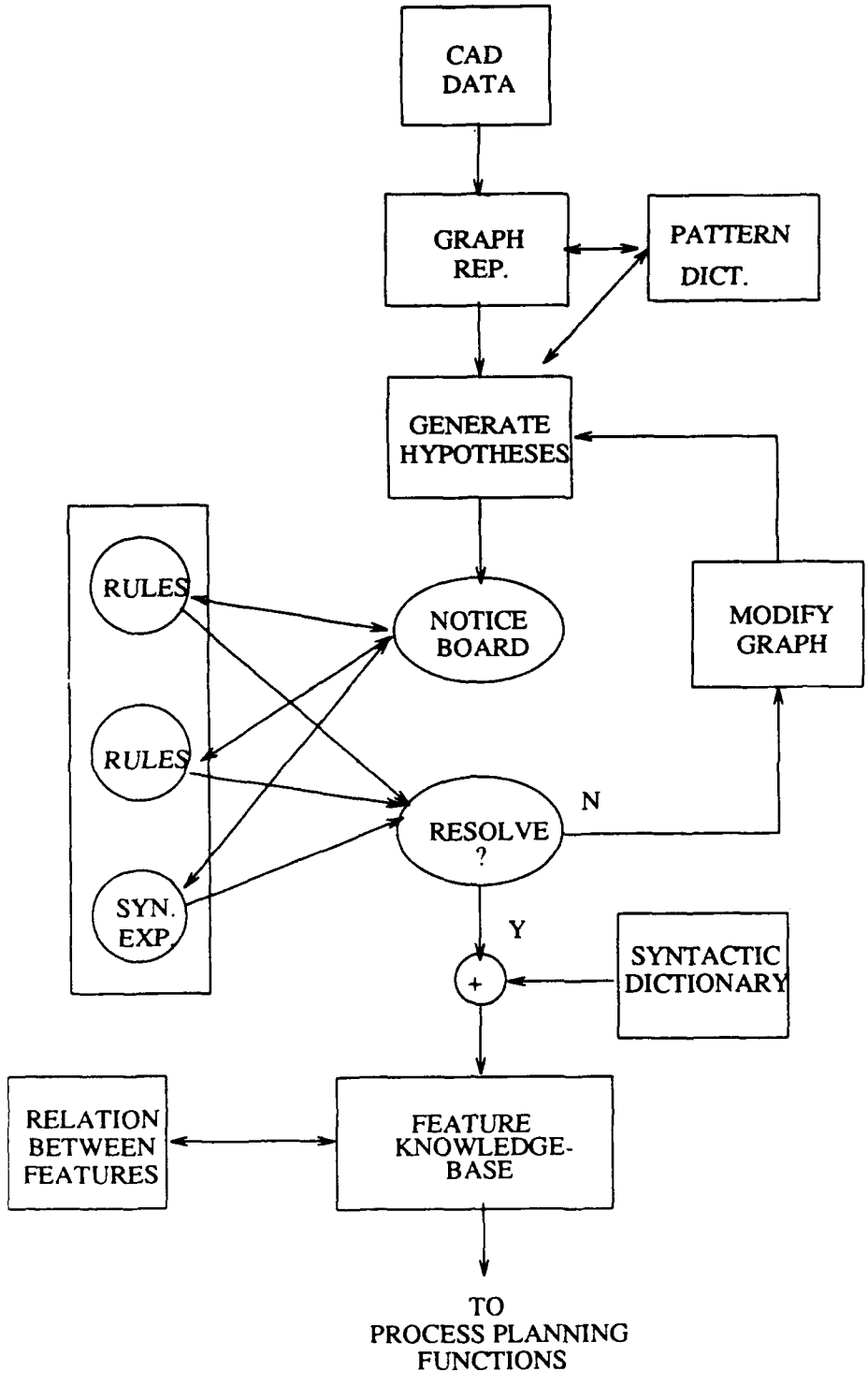


Figure 1

Schematic Diagram of the Proposed System

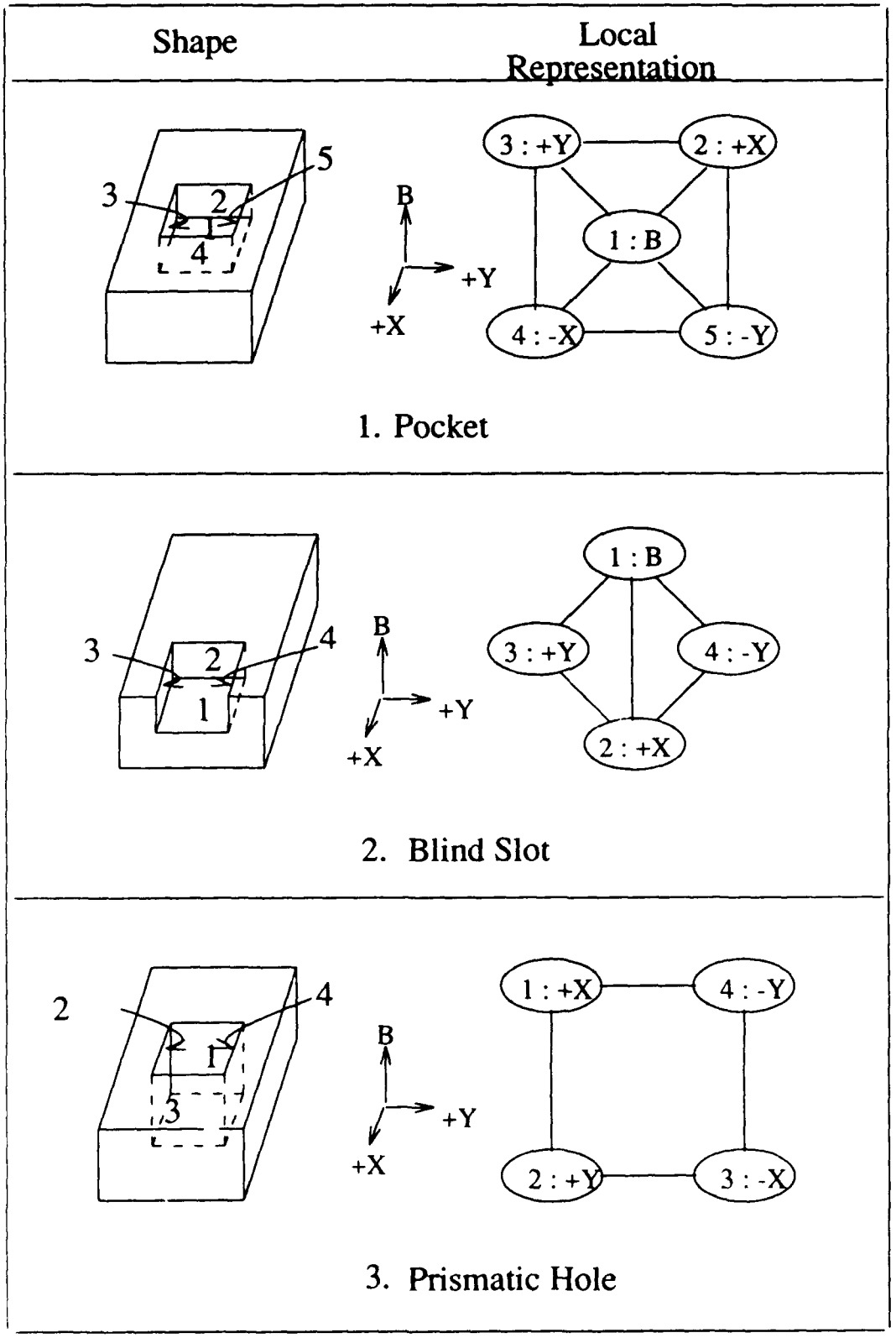


Figure 2: Primitive features and their isolated local representations. Node labels are dominant direction of normal to the face away from the material. Other consistent permutations of X, Y, and B also lead to representations for the same primitive.

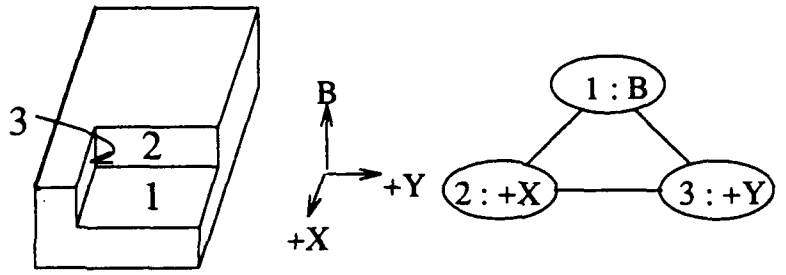
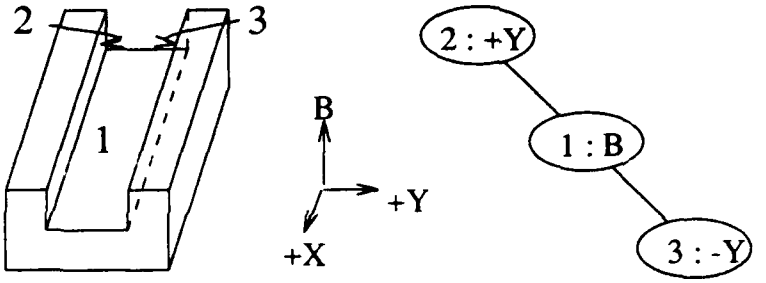
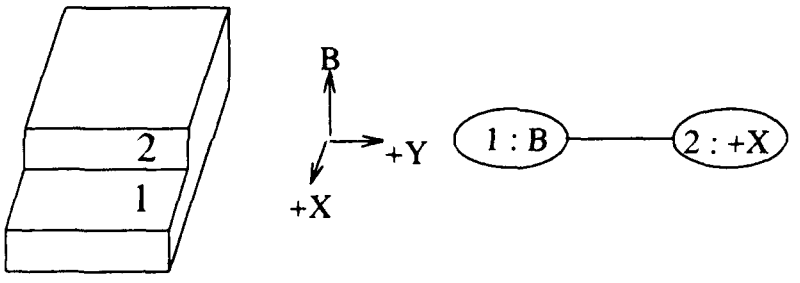
Shape	Local Representation
	
4. Blind Step	
	
5. Thru Slot	
	
6. Step	

Figure 2 (continued)

2.1 Representing Primitive Features. In order to recognize the primitives in the boundary of an object, a *unique* representation for each primitive feature is required. Our representations for the primitives are similar to the Attributed Adjacency Graphs proposed by Joshi [7] with the nodes labeled. The representations for primitives are also shown in figure 2. In our approach, each primitive is represented by a labeled graph. Each face of the primitive corresponds to a node of this graph, and two nodes are connected by a link if the corresponding faces intersect. Each node is labeled by a label from the set $\{B, -B, +X, -X, +Y, -Y\}$. The node labels show the orientation of a face in space. The node 3: +Y , for instance, indicates that the dominant component of the normal to face 3 is in the +Y direction of the *local* coordinates.

A unique subgraph pattern corresponds to each primitive, but rotating the X, Y, and B labels in a graph such that the local coordinates follow the right hand rule corresponds to a graph for an identical feature. Assuming that we keep our local coordinates fixed, replacing node labels in the template of a primitive corresponds to a rotated version of the primitive (hence the subgraph pattern is the same but node labelings are different).

2.2 Extracting Isolated Primitive Features. Using the above representation, extracting the primitive features from the boundary of an object whose features are isolated, that is the primitives are not interacting, becomes a simple task. A *global graph* of the object is first constructed. A global graph is a representation for the entire object. The global graph, G, for an object, S, has the following properties:

(i). Each face f_i of the object, S, is represented by a node in G.

(ii). For every edge, e_{ij}, of the object shared by two faces f_i and f_j, there is an edge connecting the corresponding nodes in G.

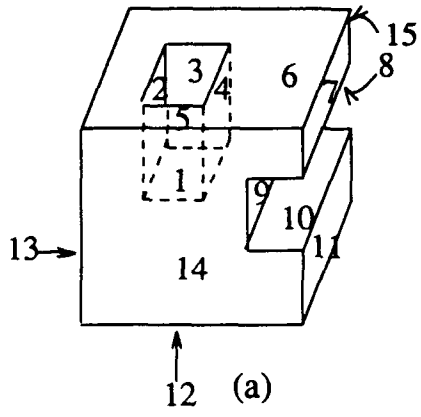
(ii). $l(G)$ is a labeling for the edges of G, which marks an edge in the graph concave if the faces sharing the corresponding edge of the object are concavely adjacent, and marks the edge convex otherwise.

To find the graphs representing the cavities in the object, the subgraph of the global graph consisting of only the concave links is obtained. Each component of this subgraph disconnected from the other components is a *cavity graph* of the object. The nodes of each cavity graph are labeled. To label the nodes, the node with the greatest number of concave edges is selected to be the base and is labeled B. A pair of X,Y directions are chosen so that together with the normal to the base they conform to the right hand rule. The other nodes are labeled according to the direction of the normal to the corresponding face away from the object. The label for the node corresponding to face f, is denoted by $l'(f)$.

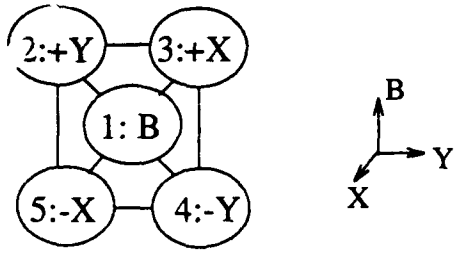
The above cavity graphs may now be matched to the representations for the primitives. Each cavity graph corresponds to one primitive's representation because the primitives are isolated. Each match generates one hypothesis stating that the depression in the boundary of the object corresponds to a primitive of the type matched. The expert rule-based systems check each hypothesis and approve the correct ones. Figure 3 shows this process of extracting the primitive features from the boundary of an object when they are isolated. While identifying a primitive the information describing its instance, such as its base face, its axis, its diameter, etc. is reconstructed. This information is represented in frames which are stored in the Feature Knowledge-Base.

3. INTERACTING FEATURES. Primitive features are usually not isolated. In fact the significance of this research is associated with recognizing interacting features and finding their constituent primitives. The number of primitive features to be recognized is finite, but the configurations of features, their intersections, and the types of interactions that may arise in a compound cavity are unlimited. It is not practical to enumerate all possible interactions of shape features.

The major hindrance in determining the constituent primitives of an object is nonuniqueness. The cavity graphs for the depression consisting of interactions may not match the representation for any of the primitives. In fact the subgraph representing a primitive in an interaction could be different from the representations shown above for isolated primitives.



(a)



(b1)



(b2)

HYPOTHESES:

(Pocket Base:1 Sides:(2 3 4 5))

(Thru_slot Base:9 Sides:(8 10))

(c)

EXTRACTED PRIMITIVES:

(verified (Pocket Base : 1 Sides : (2 3 4 5)) Certainty : 1.0)

(verified (Thru_slot Base : 9 Sides : (8 10)) Certainty : 1.0)

(d)

Figure 3: Extraction of Isolated Primitive Features

(a): a part with an isolated pocket and a thru_slot primitive.
(b1),(b2): The cavity graphs of the object.
(c): The hypotheses generated from the cavity graphs in (b1),(b2).
(d): The verified primitives extracted by the prototyped system.

Consider the part shown in figure 4. The subgraph of the global graph of the part consisting of only the concave links does not match the representation for any of the primitives. The part has a pocket (faces 13,14,15,16,20,12) and a prismatic hole (faces 16,17,19,20). The subgraph induced by nodes 13,14,15,16,20,12 is the subgraph representing the pocket in the interaction. Clearly this subgraph is not isomorphic to the representation for a pocket primitive.

The problem in the above example is a direct result of the interaction between primitive features. As a consequence of the intersection of the hole and the pocket, one of the side faces of the pocket is divided into two smaller disconnected faces, 13 and 14. A group of faces that have been produced as a result of dividing a large face because of interaction of primitive features are called *unifiable*. Faces 13 and 14 in the above example are unifiable. We can unify the faces in a unifiable group to form a larger conceptual face. The conceptual face produced by unifying faces 13 and 14 is represented by [13,14]. The following Observation expresses the minimum conditions required for a set of faces to be unifiable.

Observation 1: The set of faces $U = \{f_1, f_2, ..., f_i\}$ of an object S is *unifiable* if

(i). every face in the set, U, is embedded on the same plane, i.e. they have the same equation,

(ii). No face of the part intersects the interior of the face, $[f_1, f_2, ..., f_i]$ formed by unifying $f_1, f_2, ..., f_i$, and

(iii). normals of all elements of U point in the same direction.

The notion of unifiability is coupled with the machining of the set of unified faces. Conditions (i) and (iii) of observation 1 indicate that the unified faces have similar geometry, and condition (ii) guarantees that no other part of the object obstructs the larger face formed by unification. Therefore, all faces in a unifiable set may be machined by the same group of machining operations.

In the same manner that the node labeling helps distinguish between representations which otherwise seem identical (example in figure 9), unification helps in obtaining a unique representation for a primitive in its interaction with other primitives. Figure 4(c) shows the subgraph of the global graph 4(b) consisting of concave edges with the faces unified and the nodes labeled. It is clear that the subgraph of the cavity graph, 4(c), induced by nodes [13,14],15,16,20,12 is isomorphic to the representation for a pocket. In fact, the cavity graph is the union of the subgraphs representing a pocket and a prismatic hole (subgraph induced by 16,17,19,20). Using $l(e)$ to denote the label for edge e, and $l'(f)$ to denote the label for face f, the cavity graphs can be defined as follows:

Definition: A *Cavity Graph*, H, is a subgraph of the global graph , G, of the part such that H is connected and:

(i). unifiable faces of G are unified,

(ii). $l(e) = concave \ \forall e \in E(H)$, and

(iii). $l'(f) \in \{B, -B, +X, -X, +Y, -Y\} \ \forall f \in F(H)$.

where $E(H)$ and $F(H)$ are the edges and the faces of the cavity graph, H, respectively.

The cavity graph 4(c), as mentioned above, is composed from the union of the subgraphs for a pocket primitive and a prismatic hole primitive. Therefore, it seems logical that to obtain the primitive features of the part we must decompose the cavity graph into the subgraphs for the constituent primitives. The cavity graphs obtained according to the above definition are, in general, a proper subgraph of the union of the representations for the primitives involved in the interaction, or isomorphic to the union of the representations. Decomposing the cavity graphs into a set of maximal subgraph patterns representing the primitive features yields a set of promising hypotheses about the primitive constituents of the depressions in the part.

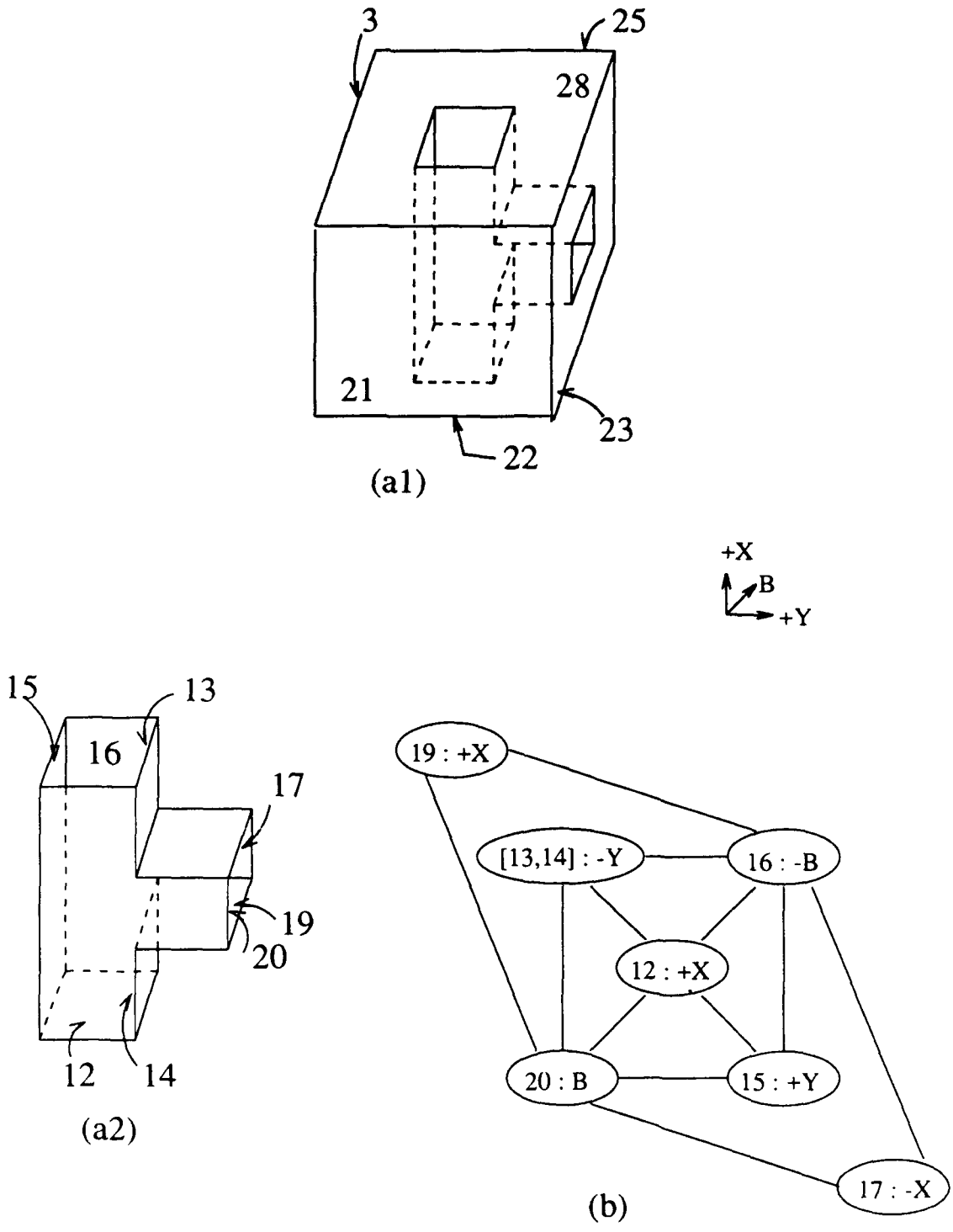


Figure 4: Cavity Graph Representation for a Depression

(a1) Shows an example part, (a2) shows the depression in (a1), which is a pocket with base-face 5, with a prismatic hole opening into it. Note that faces 13 and 14 are coplanar. (b) shows the Cavity graph of depression. In (b) (15 : +Y) means that the normal to face 15 is in direction +Y. Faces 13 and 14, which meet all criteria for unification, are unified and denoted by [13,14].

4. HOW ARE HYPOTHESES GENERATED? Generation and verification of hypotheses form the core of our method for extracting shape features. The purpose of the hypothesis generation step is to find a set of promising hypotheses from which the correct hypotheses about primitives forming the depressions can be selected by the expert consultants. An object typically consists of several depressions. Each depression is represented by one or more cavity graphs. To generate hypotheses, the cavity graphs are decomposed into maximal subgraphs corresponding to the patterns for the primitive features.

Observation 2: There is an ordering $H_1, H_2, \ldots, H_j$ of the local representations of the primitive features, such that $H_1 \supset H_2 \supset \cdots \supset H_j$.

From the local primitive representations in figure 2:

$$H_{pocket} \supset H_{blind-slot} \supset H_{prismatic-hole} \supset H_{blind-step} \supset H_{slot} \supset H_{step}$$

The ordering introduced by observation 2 is used to assure maximality of each primitive in a decomposition. We first look for subgraphs of a cavity graph isomorphic to H_i, and after all such subgraphs are found and the corresponding hypotheses are generated, we look for subgraphs isomorphic to H_{i+1}. Two subgraphs representing different hypotheses may have common vertices and even common edges, but a subgraph is not allowed to represent a hypothesis if all of its edges (and hence its vertices) are elements of subgraphs for higher order primitives. This continues until the Cavity Graph is completely decomposed into its maximal constituents.

For the purpose of efficiency, this decomposition is broken down into steps. For each cavity graph, an ordered set of faces is produced such that faces with more concave edges in their boundary are ranked first. Faces are taken one at a time from this ordered set of faces and all consistent and acceptable (with respect to primitive structure and node labels) hypotheses for primitives are generated, such that the face is the base of the primitive. A hypothesis is removed from the hypothesis set if its graph is a part of the graph of any previously generated hypothesis. This ensures maximality in the set of hypotheses. This is repeated until all faces and edges belong to some hypothesis. The process is performed for each cavity graph, thereby generating hypotheses for the depressions in the complete object.

Figure 5 shows the Cavity Graph of a depression and the set of hypotheses generated on it using the above outlined algorithm. Hypotheses suggest the existence of specific primitive features and the faces comprising each one. Each hypothesis is represented as a list. The first element of the list is a key to the type of primitive being hypothesized. The other elements are the base of the primitive and a list of its side faces in a predefined order.

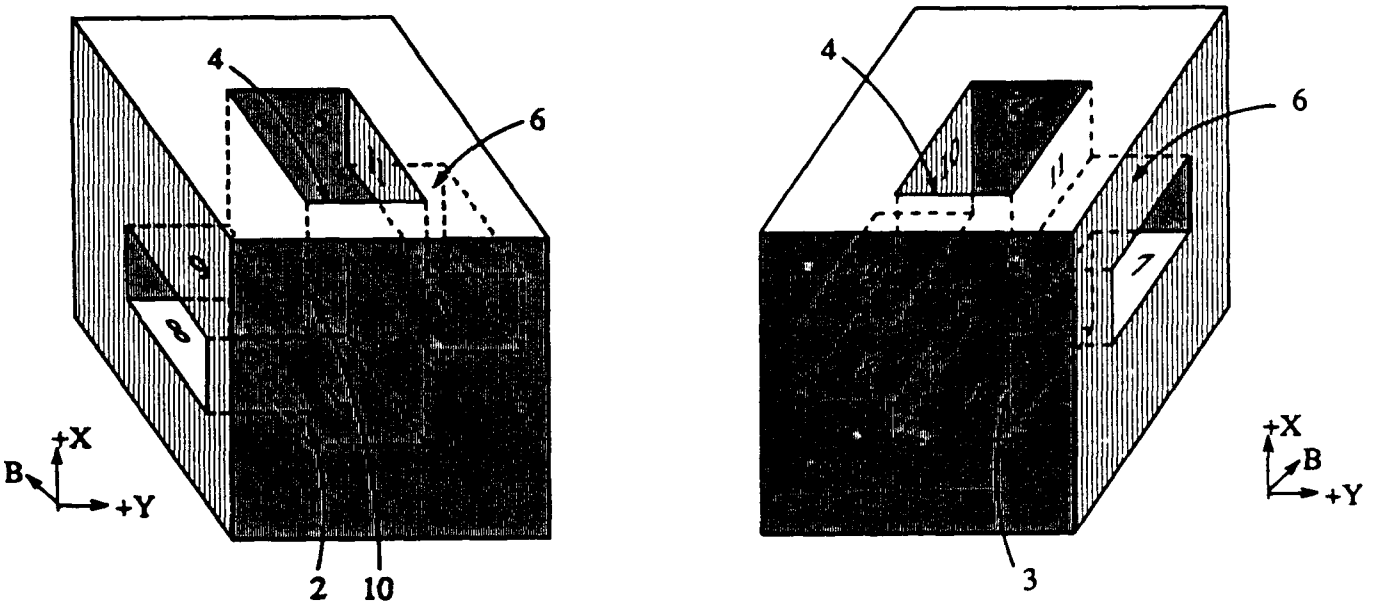
Although the algorithm stops when all the faces and edges are in some hypothesis, keeping track of what faces have been used as a base in the ordered set of faces enables us to follow our enumeration, and hence to generate new hypothesis sets from where we left off without duplicating the earlier effort.

There are certain properties which are desirable for generators. Winston [15] describes these as:

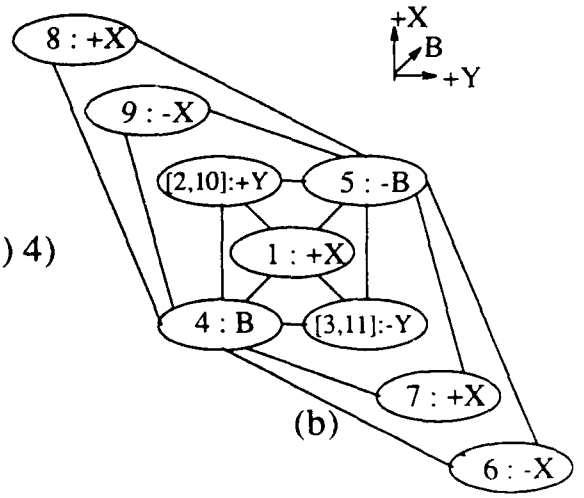
- Good generators are complete. They eventually produce all possible solutions.
- Good generators are nonredundant. They never damage efficiency by proposing the same solution twice.
- Good generators are informed. They use possibility-limiting information, restricting the solutions they propose accordingly.

The following observation shows that our hypothesis generation scheme has the above properties:

Observation 3: The hypothesis generation scheme proposed above is (i) complete, (ii) nonredundant, and (iii) informed.



(a)



(b)

1. (Pocket Base: 1 Sides: (5 (2 10) (3 11) 4)
2. (Hole Base: 4 Sides: (6 5 7))
3. (Hole Base: 4 Sides: (6 5 8))
4. (Hole Base: 4 Sides: (9 5 8))
5. (Hole Base: 4 Sides: (9 5 7))

(c)

Figure 5: Hypotheses Generated for a Part

(a): An object with a depression formed by interaction of one pocket and two polyhedral holes opening into it.

(b): Cavity graph of its depression,
Note that the coplanar face-pairs 2,10 and 3,11, are unifiable. These faces belong to the pocket primitive, and hence in the cavity graph (b), one node represents both faces of the unified pair [2,10] and one node represents the unified pair [3,11].

(c): The set of hypotheses generated for the part.
The 3rd and 5th hypotheses are incorrect.

(i). Completeness is due to the fact that at each face, all primitive templates with that face as their base are checked, and eventually all faces are used to find templates based at them.

(ii). The scheme is not redundant because it keeps a history of the hypothesized primitives, and every new one is checked in step (vi) of the algorithm against the previous ones, and if a hypothesized primitive's set of edges and faces is a subset of another primitive's set of edges and faces, the hypothesis corresponding to the subset is discarded.

(iii). The method is informed for three reasons:

(a) The faces of a depression are ranked.

(b) The order of search for primitive features is that set by observation 2.

(c) At any time it is known what faces and edges are hypothesized about.

The first set of hypotheses generated on a Cavity Graph, G, is not necessarily the only set. If the subsequent analysis shows that the generated hypotheses represent the primitives involved incorrectly, subsequent hypotheses are generated using a supergraph of the cavity graph. We describe later how this supergraph is built in the section on the combination of topologic and geometric evidences. The hypothesis generation process is recursive. The first set of the hypotheses is formed by maximal subgraphs of the supergraph. Although seldom needed, later hypothesis sets are produced by decomposing the supergraph into a new set of subgraphs. These later subgraphs constitute breaking the supergraph into smaller components which are not maximal. However, for every generated set of hypotheses, $H = \{h_1, h_2, ..., h_i\}$, we have:

$$h_1 \bigcup h_2 \bigcup \cdots \bigcup h_i = G,$$

where h_j refers to the local representation for the primitive of hypothesis j, $(1 \leq j \leq i)$,

and G = the cavity graph on which hypotheses are generated.

The hypothesized features must be checked in detail for correctness by the rule-based consultants. The generated hypotheses are made available on a notice-board to the rule-based experts.

4.1 Elimination of Invalid Hypotheses. The objective of the rule-based experts is selection from the proposed set of hypotheses, H, of the subset of correct hypotheses, V, which accurately specifies the comprising primitive features of a depression. Rule-based experts examine each generated hypothesis in detail to decide if the primitive feature suggested by the hypothesis is valid. Each expert is independent and equipped with knowledge in the form of rules and procedures. An example rule for a thru-slot looks like this in pseudo lisp:

```
(thru-slot    base: B    sides: (side1, side2)) :-
              (concave-adjacent    base    side1),
              (concave-adjacent    base    side2),
              (opposite    (principle-normal-component    side1)
                           (principle-normal-component    side2)),
              (intersect   base    side1   edge1),
              (intersect   base    side2   edge2),
              (no-concave-edge-between    edge1  edge2),
              (not   (adjacent side1  side2)),
              ((concave-adjacent side1  list1)  and  (length  list1  '1)),
              ((concave-adjacent side2  list2)  and  (length  list2  '1)).
```

The above rule says that the three faces (base, side1, and side2) form a thru-slot if the sides are concavely adjacent to the base, the sides are almost parallel and have opposing normals.

there is no face concave to the base between side1 and side2, and neither side1 nor side2 is concavely adjacent to any other face except the base of the slot.

The type field of a hypothesis is used to search for the rules applicable. The matching of the types triggers a rule. The conjuncts in the condition are evaluated when a rule is fired. If all the conjuncts succeed, the hypothesis being tested is assumed correct and is tagged to indicate so. Let us consider the example shown in figure 5 again. There is one depression in the part and one cavity graph corresponding to it. The depression in the part consists of a pocket with two holes opening into it. Five hypotheses are generated, one for a pocket and four for holes, three of which are correct. One of the conjuncts in the if part of the rule for holes states that all the faces of a hole should be convexly adjacent to the same face. This condition models the fact that at the open end for a hole there is a loop of edges produced by the intersection of the faces of the hole with the opening face of the hole. Applying this condition to the four hole hypotheses of figure 5 it becomes clear that the third and the fifth hypotheses are incorrect because faces 6 and 8 of the third hypothesis are not adjacent to the same opening face and similarly faces 9 and 7 of the fifth hypothesis are not adjacent to the same opening face. These hypotheses are eliminated.

5. VIRTUAL LINKS. In the cases that correct conclusions about the primitive features constituting a depression can be drawn from the original cavity graph of the depression, the correct hypotheses are found by the experts and a verification step checks the results. In such cases, there is no need for virtual links. However, another class of problems that arises in the analysis of primitive interactions is when the intersection of primitives causes the the adjacency relationships between the faces of a primitive to be changed. In this case, the representation of a primitive in the interaction (subgraph of the cavity graph induced by the nodes for the faces of the primitive) does not match the template for the primitive (shown in figure 2), because some links of the template are not in the induced subgraph. To clarify the problem, let us consider the example in figure 6. The object in figure 6(a) has a blind_slot *(base : 5 sides : (4, 3, 1))* and a thru_slot *(base : 5 sides : (2, 1))*. The shape primitives share the same base, but face 3 of the blind_slot is disconnected from the side face 1. Consequently there is no link between nodes 3 and 1 of the cavity graph, so the subgraph induced by vertices 5,4,3,1 does not match the template for a blind_slot. In order to generate the correct candidate hypotheses for such parts, it is necessary to augment the cavity graph with virtual links. For the cavity graph, G, of figure 6(b), the hypothesis space includes the indicated primitives only if the virtual link, (3 1), between nodes 3 and 1, is added to G. In general, we don't know how many links are required a priori. The number of virtual links *can be more than one*. The most appropriate virtual links are those whose addition makes the resulting supergraph isomorphic to the union of representations for the involved primitives.

5.1 Direct Method to Find Virtual Links The appropriate virtual links for a cavity graph may be found by approaches including enumeration, partial match, expert system, and uncertainty reasoning. Enumeration considers each possible supergraph of the cavity graph individually. Although the desirable results may be obtainable with this approach, it has practical limitations because the power set of all possible links should be considered.

We refer to the partial match approach as the direct method. In this method, the partial matches between the maximal subgraphs of the cavity graph of depression and the templates for primitive features are considered as potential hypotheses, and used for determining which virtual links to be augmented to the cavity graph. Consider the example of figure 6 again. Node 5 can be chosen as the base to generate the maximal subgraphs of the depression, since it has the highest number of incident links. Considering different primitives, we observe that if the three links (4 2), (2 1), and (1 3) are added to the cavity graph it may represent the template for a pocket, but then the labels do not form a consistent set for a pocket primitive. Next, we observe that the subgraph induced by nodes 4,5,3 and 1 partially matches the template for a blind-slot. For this partial match to be complete, a link between the nodes 3 and 1 of the cavity graph, G, is required. Proceeding in this fashion, the direct method may select this link as a strong candidate for a virtual link to be augmented to the cavity graph. Obviously, to prove the necessity for this link more reasoning and tests must be performed, however, the direct method identifies

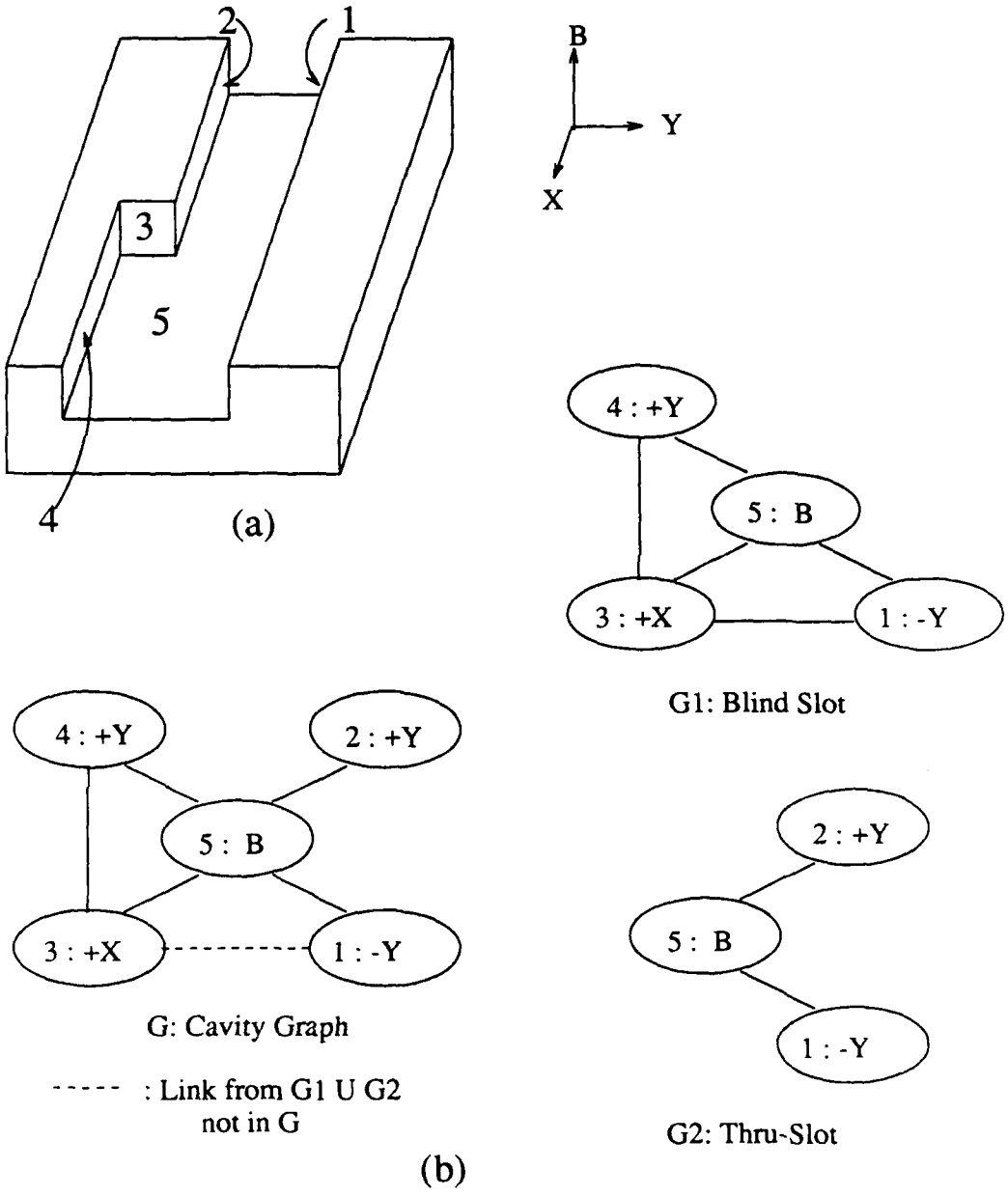


Figure 6: A Compound Feature whose Cavity Graph is not the union of the template subgraphs for the primitives involved.

The depression of object shown in (a) consists of a blind slot and a thru-slot sharing base-face 5. However, face 3 of the blind slot does not intersect face 1 as a result of the blind slot's interaction with the thru-slot. The cavity graph of the depression is G without the dotted link. Therefore a thru-slot subgraph, G1, and a blind-slot subgraph, G2, can be obtained from the Cavity Graph, G, only if the shown dotted virtual link is added to G.

strong candidates.

The direct method of considering partial matches to find virtual links is useful for reasoning with relatively small cavity graphs, but it cannot be effectively used for complicated cavity graphs, such as the cavity graphs associated with example part of figure 12. There are too many sufficiently close partial matches in these cavity graphs. A better method should be used to take into account other supporting information, and automatically generate the most useful virtual links. We show in the next section that a method based on combining geometric and topologic evidences can be used to achieve this purpose.

6. DETERMINING VIRTUAL LINKS BY COMBINING TOPOLOGIC AND GEOMETRIC EVIDENCES.

6.1 Dempster-Shafer Theory and Extraction of Primitive Features. In this section, we introduce a method to find the virtual links to be augmented to the cavity graph of a depression by combining geometric and topologic evidences, and applying a clustering technique. Figure 7 shows the basic block diagram for this approach. The approach is based on determining the subset of the virtual links in a cavity graph that should be reconstructed to obtain the supergraph embedding the most probable primitive features.

As we mentioned earlier, the number of required virtual links are not known in advance. Let us consider the example object in figure 8(a). Its depression consists of two pockets with perpendicular axes opening into each other. The base of the first pocket is face 1 (side-faces: 2,4,3,and 5), and the other pocket has face 2 as its base (side-faces: 1,5,6, and 4). The cavity graph for the depression is shown in figure 8(b). Because there is neither a link between the nodes 3 and 1, nor a link between nodes 2 and 6, neither the subgraph induced by nodes 1,2,3,4, and 5, nor the subgraph induced by nodes 1,2,4,5, and 6, matches the template for a pocket. For the example part of figure 8, two virtual links ((6 2) and (3 1)) are needed.

Dempster-Shafer Theory [12] is particularly useful for our purposes, because of its ability to narrow the set of promising hypotheses with the accumulation of evidence.

Each virtual link to be augmented to the cavity graph can be represented by a tuple $(f_i f_j)$. The set $A = \{(f_{11} f_{12}), \cdots, (f_{i1} f_{i2})\}$, then, is the hypothesis which suggests the set of i links $\{(f_{11} f_{12}), \cdots, (f_{i1} f_{i2})\}$ should be added to the cavity graph of a depression.

The hypothesis that contains the set of all possible virtual links in a cavity graph is the set that is referred to as the frame of discernment, denoted Θ. Let G be the cavity graph of a depression. The set Θ is obtained by determining the set of all links required to make G a complete graph. (A complete graph is a graph that is simple* and each pair of distinct nodes in it is joined by an edge.) Hence, the set of links in the frame of discernment, Θ, is $E^c(G)$, the complement of $E(G)$, the set of edges of G with respect to the set of links connecting all distinct pairs of nodes of G. The set Θ for the example part of figure 8 would be:

$$\Theta = \{(f_6 f_3), (f_6 f_1), (f_6 f_2), (f_3 f_1), (f_3 f_2), (f_4 f_5)\}.$$

An underlying assumption of the Dempster-Shafer Theory is that the hypotheses under consideration are mutually exclusive and exhaustive. These assumptions are satisfied by the set of hypotheses constituting a frame of discernment for a cavity graph.

Different pieces of information, such as whether the planes containing two faces are perpendicular or not, form evidences helping support or helping exclude a one element subset of Θ. The impact of each evidence is represented by a basic probability assignment (bpa). A bpa assigns a number, m, in the range [0 1] to each subset of Θ. Shafer describes a basic probability assignment as:

> The quantity $m(A)$ is called A's basic probability number, and it is understood to be the measure of the belief that is committed exactly to A [12].

---

* : A graph is simple if it has no loops and no two of its links join the same pair of nodes.

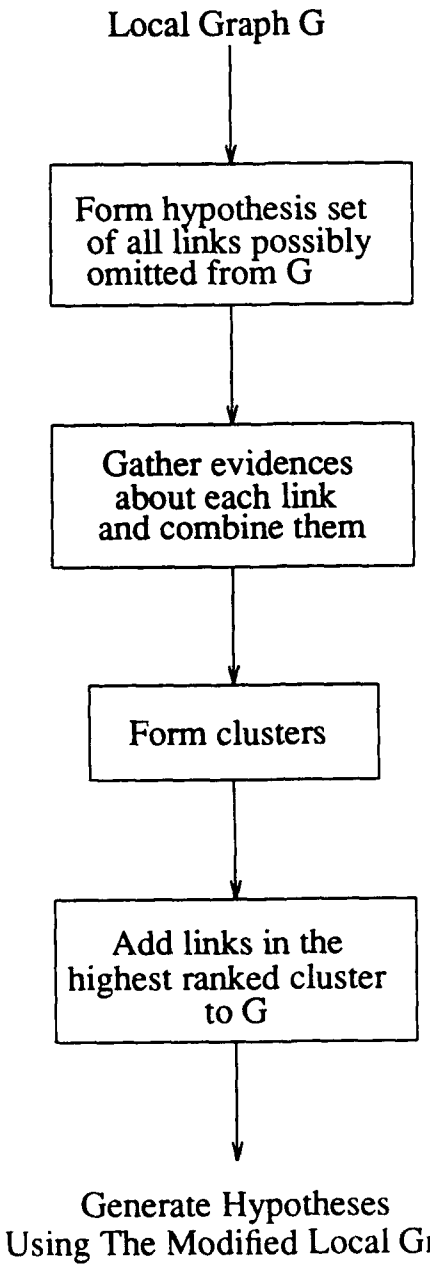


Figure 7:
Basic Block Diagram For Finding The Most Probable Features

This proportion of belief does not imply belief in the subsets of A. The quantity $Bel(A)$, for some bpa, m, is used to represent the total belief in A, which indicates certainty in A and its subsets. Shafer further explains:

> ... To obtain the measure of the total belief committed to A, one must add to $m(A)$ the quantities $m(B)$ for all proper subsets B of A:
>
> $$Bel(A) = \sum_{B \subset A} m(B)$$

Evidences and bpas associated with each evidence should be selected such that they confirm or disconfirm what is implied physically by adding the virtual link to the Cavity Graph. Some relevant information might be: Are the normals to faces f_i and f_j almost perpendicular? If the addition of $(f_i f_j)$ completes the representation for the maximal primitive, P, are there indications that P is part of the depression? ... Each of these pieces of information forms the basis for an evidence helping support or helping exclude a one element subset of Θ. The bpa, m_i, assigned by evidence i must be chosen such that:

$$\sum_j m_i(A_j) = 1 \quad ; \quad A_j \subset \Theta$$

and $m(\varnothing) = 0.$

$m(\Theta)$ is the quantity of belief that remains unassigned after various amounts of belief are assigned to all proper subsets of Θ.

After relevant evidences are gathered and the corresponding bpa assignments are made, the supports for hypotheses are combined. Given two observations corresponding to hypothesis sets X and Y, with bpa assignments $m_1(X)$ and $m_2(Y)$, we use Dempster's combination rule to compute a new function that represents the impact of the combined evidence. This function assigns $m_1(X)m_2(Y)$ to the intersection of X and Y. Since there are typically several subsets of Θ whose intersection is the same as that of X and Y, the bpa assignment of the combined function, denoted $m_1 \oplus m_2$, for every hypothesis set A, is computed from:

$$m_1 \oplus m_2(A) = \frac{1}{1-\kappa} \sum_{\substack{i,j \\ X_i \cap Y_j = A}} m_1(X_i) m_2(Y_j) \quad ; \quad A \neq \varnothing \tag{2}$$

$$m_1 \oplus m_2(A) = 0 \quad ; \quad A = \varnothing$$

where:

$$\kappa = \sum_{\substack{i,j \\ X_i \cap Y_j = \varnothing}} m_1(X_i) m_2(Y_j)$$

The result of the combination is independent of the order in which the evidences are gathered and combined, an essential property fulfilled because of the commutativity of multiplication. Also, the sum of all bpas assigned by $m_1 \oplus m_2$ adds up to 1, thus satisfying the definition of a bpa.

To combine all supporting evidences, equation (2) is applied repeatedly. Assuming there are j pieces of evidence, m_1, m_2, ..., m_j, we use:

$$m = (...((m_1 \oplus m_2) \oplus m_3) \oplus ...) \oplus m_j)$$

to come up with m, the combined bpa for each singleton. If A is a singleton, $Bel(A)$, the total belief committed to A, is equal to $m(A)$ since the contribution from the bpa of its only proper subset, $\varnothing$, is zero. Therefore, to determine the virtual links to be augmented, the combined bpa m can be used to rank the i singleton subsets of Θ, and cluster the singletons into disjoint groups.

Each singleton in a cluster has a combined bpa value which is similar to the other singletons in its cluster, and the bpas of the singletons in one cluster are notably different from the bpas of singletons in any other cluster. The hypotheses associated with the cluster containing

the highest ranked singleton are the hypotheses that correspond to the most probable virtual links. These links are augmented to the cavity graph of the depression. The resulting supergraph is then analyzed using the hypothesis generate-verify method described in previous sections to find its constituting primitive features. The next example clarifies the method:

Example: Consider the part of figure 8(a). The hypotheses generated using the cavity graph shown in 8(b) do not model the depression; we already determined the frame of discernment to be:

$$\Theta = \{(f_6f_3),\ (f_6f_1),\ (f_6f_2),\ (f_3f_1),\ (f_3f_2),\ (f_4f_5)\}.$$

Evidences: For clarity, let us consider three simple bpa assignments for links in Θ: If two faces, f_i and f_j, have orthogonal principle normal components, assign: $m(\{(f_if_j)\}) = 0.7$, and $m(\Theta) = 0.3$; if two faces, f_i and f_j,are convexly adjacent, assign: $m(\{(f_if_j)\}^c) = 0.7$, and $m(\Theta) = 0.3$; and if two faces, f_i and f_j, are almost parallel with normals in the opposite direction, $m(\{(f_if_j)\}^c) = 0.8$, and $m(\Theta) = 0.2$. There are seven pieces of information applicable to the links in Θ with the following bpa assignments:

1. $m_1(\{(f_6f_3)\}) = 0.7$; $m_1(\Theta) = 0.3$
2. $m_2(\{(f_6f_3)\}^c) = 0.7$; $m_2(\Theta) = 0.3$
3. $m_3(\{(f_6f_1)\}^c) = 0.8$; $m_3(\Theta) = 0.2$
4. $m_4(\{(f_6f_2)\}) = 0.7$; $m_4(\Theta) = 0.3$
5. $m_5(\{(f_3f_1)\}) = 0.7$; $m_5(\Theta) = 0.3$
6. $m_6(\{(f_3f_2)\}^c) = 0.8$; $m_6(\Theta) = 0.2$
7. $m_7(\{(f_4f_5)\}^c) = 0.8$; $m_7(\Theta) = 0.2$

After combining them, $m(A) = m_1 \oplus m_2 \oplus \cdots \oplus m_7(A)$ and we get the following combined bpas for the singleton hypotheses:

$m(\{(f_3f_1)\}) = 0.366$ $m(\{(f_6f_2)\}) = 0.366$
$m(\{(f_6f_3)\}) = 0.110$ $m(\{(f_6f_1)\}) = 0.0$
$m(\{(f_3f_2)\}) = 0.0$ $m(\{(f_4f_5)\}) = 0.0$

Therefore we obtain:

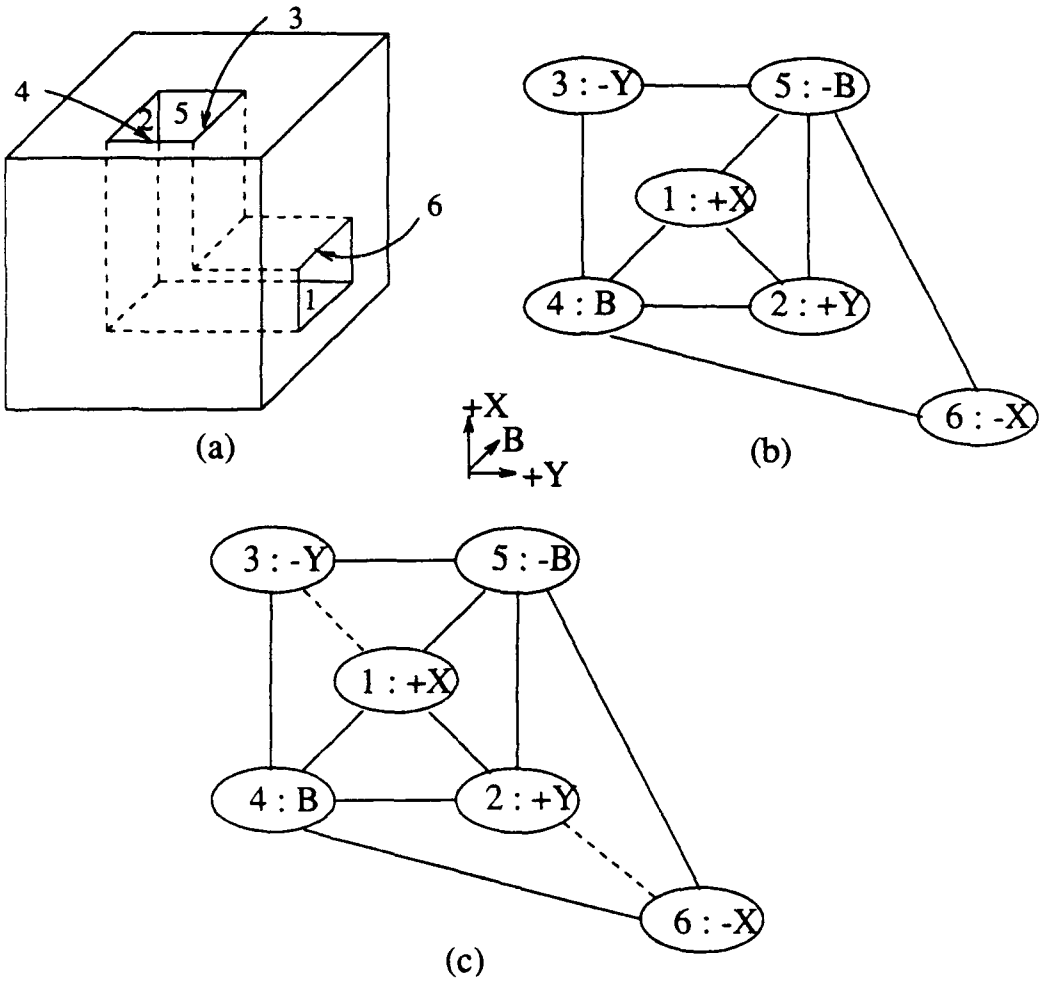
cluster1 $= \{(f_1f_3),\ (f_2f_6)\}$
cluster2 $= \{(f_3f_6)\}$
cluster3 $= \{(f_3f\ 2),\ (f_4f_5),\ (f_6f_1)\}$.

The links, from the highest ranked cluster are restored. The resulting supergraph is shown in Figure 8(c). Based on the graph in 8(c) the following primitive constituents of the depression can be identified:

((pocket Base:1 sides:(2 4 3 5)) certainty:0.8)
((pocket Base:2 sides:(1 5 6 4)) certainty:0.8)

It is important to note that in assigning the above bpas to the corresponding subsets of Θ, the actual numbers, e.g. 0.7, do not need to be exact for our purposes because we *are not drawing conclusions from the value* of the combined bpas. Conclusions as to what links to be augmented to the Cavity Graph are based on the final *ranking* of the links of Θ obtained from combining evidences.

7. VERIFICATION. Whether the cavity graph or its supergraph is the basis for the generated hypotheses, the subset of hypotheses selected as the correct primitives modeling the depression needs to be verified. Selection of a hypothesis by an expert is a temporary acceptance for that specific hypothesis. In recognizing interacting features, it is important to analyze the cavity as a whole. To achieve integrity, a selected set of hypotheses, V, must model the whole depression. This requires all faces and edges of the underlying cavity graph to be associated with at least one of the primitive features in V. If there are edges or nodes of Cavity Graph not associated with any selected hypothesis (dangling edges or nodes), then all hypotheses in V are disbelieved. This strategy avoids misclassification of a primitive by assigning some of its topologic entities to a neighboring primitive feature.



---- Reconstructed Link

Recognized Primitive Features:

(verified (pocket Base : 1 Sides : (2 4 3 5)) Certainty : 0.8)
(verified (pocket Base : 2 Sides : (1 5 6 4)) Certainty : 0.8)

(d)

Figure 8: Adding virtual links to the cavity graph.

(a): A part with two interacting pockets.

(b): Cavity graph of the depression
The subgraphs of this cavity graph induced by the faces for each pocket do not represent a pocket template because the links (3 1) and (2 6) do not exist in this cavity graph. Specifically the maximal decomposition of this cavity graph contains representations for two holes and one blind slot, any combination of which is incorrect.

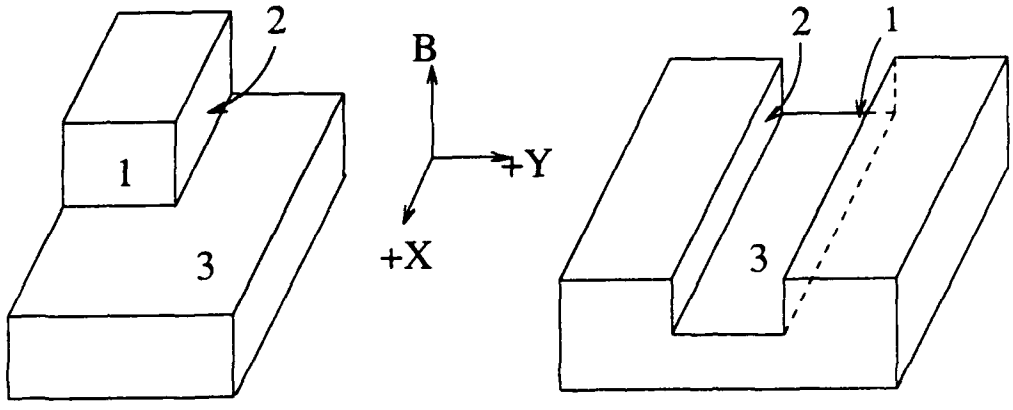
(c): The Cavity graph with virtual links added.
This augmented cavity graph is readily decomposed into two maximal constituents each representing one desired pocket.

(d): The most probable features extracted from (c).

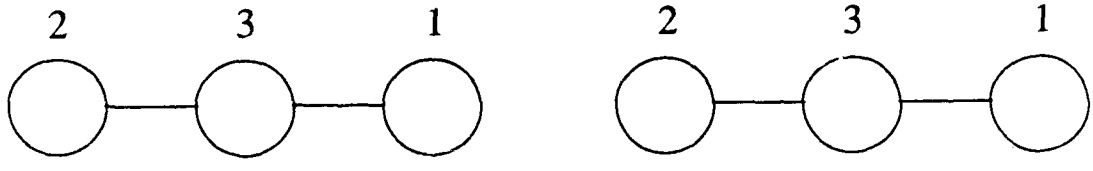
8. RESULTS AND DISCUSSION. Cavity graphs, one of the representation tools we have used in this research, are an extension of the attributed adjacency graphs proposed by Joshi [7]. Our cavity graphs not only explicitly represent the convexity-concavity information between two intersecting faces, but they also explicitly represent the spatial orientation of faces relative to one another in 3D space. These relative spatial orientations are available in the labels associated with the cavity graph nodes to directly participate in the matching process. Node labels are important for the correct classification of primitive features. Their utility can be seen with the use of a synthetic simple example. Consider the two simple parts shown in figure 9. Their cavity graphs without the node labels and with the node labels are shown in figures 9(b1-b2) and 9(c1-c2) respectively. There is no difference in the graphs for the features without the node labels, but object (a) contains two steps and object (b) contains a thru-slot. It is clear that using the graphs without the face-labelings as the basis for recognition would result in classifying both features in the same class. However, a quick and rough labeling of the nodes reveals that the two faces concavely adjacent to the base face are almost perpendicular in object (a) and almost parallel in object (b).

The advantage of this extended representation is not contained to representing certain depressions uniquely. It is directly accountable for reducing complexity greatly in generation of hypotheses. This improvement in complexity is achieved in two ways. First, only the template matches which are supported by an acceptable combination of node labels produce a hypothesis, and secondly, the node unification of separate faces into one node reduces the number of nodes and therefore complexity. The next example part illustrates how these two considerations work to reduce complexity. In figure 10, five slots interact. The coplanar faces 10, 11, 12, and 13 are produced because of the crossing of the three parallel slots on the left into the thru-slot in the middle. The thru-slot in the middle consists of seven machining faces: Face 16 is the base; faces 10, 11, 12, and 13 form one side of this slot; and faces 14 and 15 form the other side. In addition to illustrating how all faces of a primitive, even when they are arbitrarily split in an interaction, are extracted and incorporated in its description, this example serves to show how the node labeling mechanism prevents the generation of many incorrect hypotheses. The only cavity graph for the object is shown in figure 10(b). A blind decomposition of the cavity graph into primitive templates without regard to node labels produces as many as $\binom{10}{2} = 45$ slot hypotheses, since there are 45 possible pairings of faces within the cavity graph. However, using the face labeling mechanism, only the faces with opposite principle normal components are used to form hypotheses. Hence, only 17 hypotheses are generated. This is a much smaller subset which may carefully be examined by the experts. The five slots correctly constituting the interaction are found and verified from among these 17 hypotheses. The resulting primitives are shown in 10(c).

The next example shows the importance of unification for finding machining faces which have no concave edges. Although these faces are not part of the subgraph of the part induced by just the concave links, since all of their edges are formed by convex face-intersections, they must be included in the description of the primitives they are a part of, so that in planning or recognition tasks correct results are obtained. Figure 11(a1) shows an example part with four thru-slots. Two of the slots cross each other while the remaining two open into one of the former slots and are parallel with each other. (a1) and (a2) show magnified views of these two latter slots. The intersection of the slots causes three of the side faces to break into disconnected components forming coplanar face pairs 2,3 and 4,5 and 9,13. Note that faces 5 and 13 are adjacent to all their neighboring faces via convex edges. This example shows is that although in a cavity graph a link corresponds to a pair of faces being concavely adjacent, machining faces with no concave edges are properly identified. Faces 5 and 13 are two such faces. Face 5, for instance, has no concave edges, but it is one of the side faces of the slot having base face 14 and side faces: 5,4,13,9. There are three cavity graphs associated with this part which are shown in figure 11(b). Face 5, as a result of unification, shares the same node with face 4 in the left cavity graph. Therefore, it is correctly included as a machined surface of the slot in the surfaces of extracted primitives. The parts giving rise to these types of interactions can not be properly handled by previous systems such as [7].

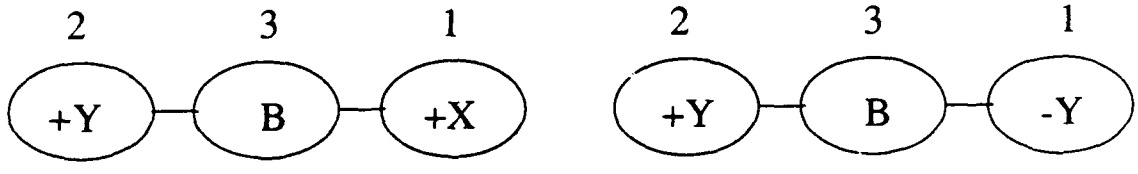


(a1) (a2)



(b1) (b2)

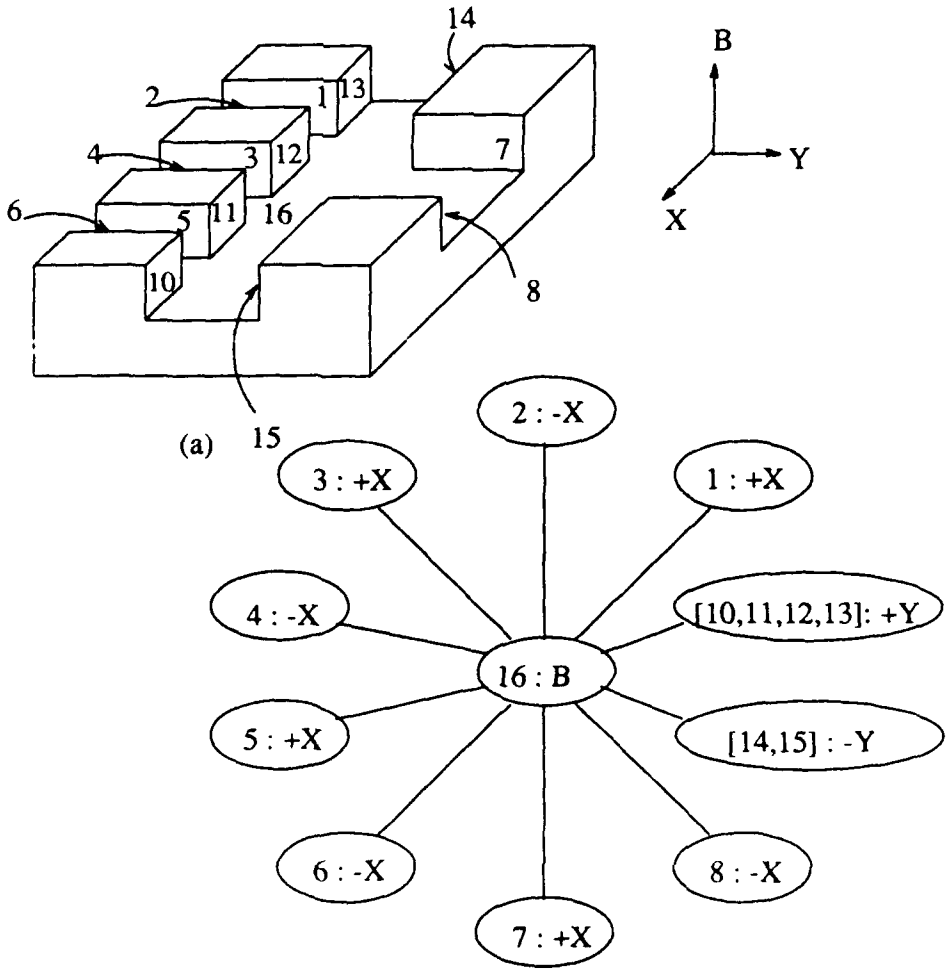
Without
Face Labeling



(c1) (c2)

With
Face Labeling

Figure 9
The Need for Face Labeling:
Shown in (a1) and (b1) are two simple objects; There is no difference in their Cavity graphs without face labels, (a2) and (b2); Cavity graphs including the face labels (c1) and (c2) capture the difference.



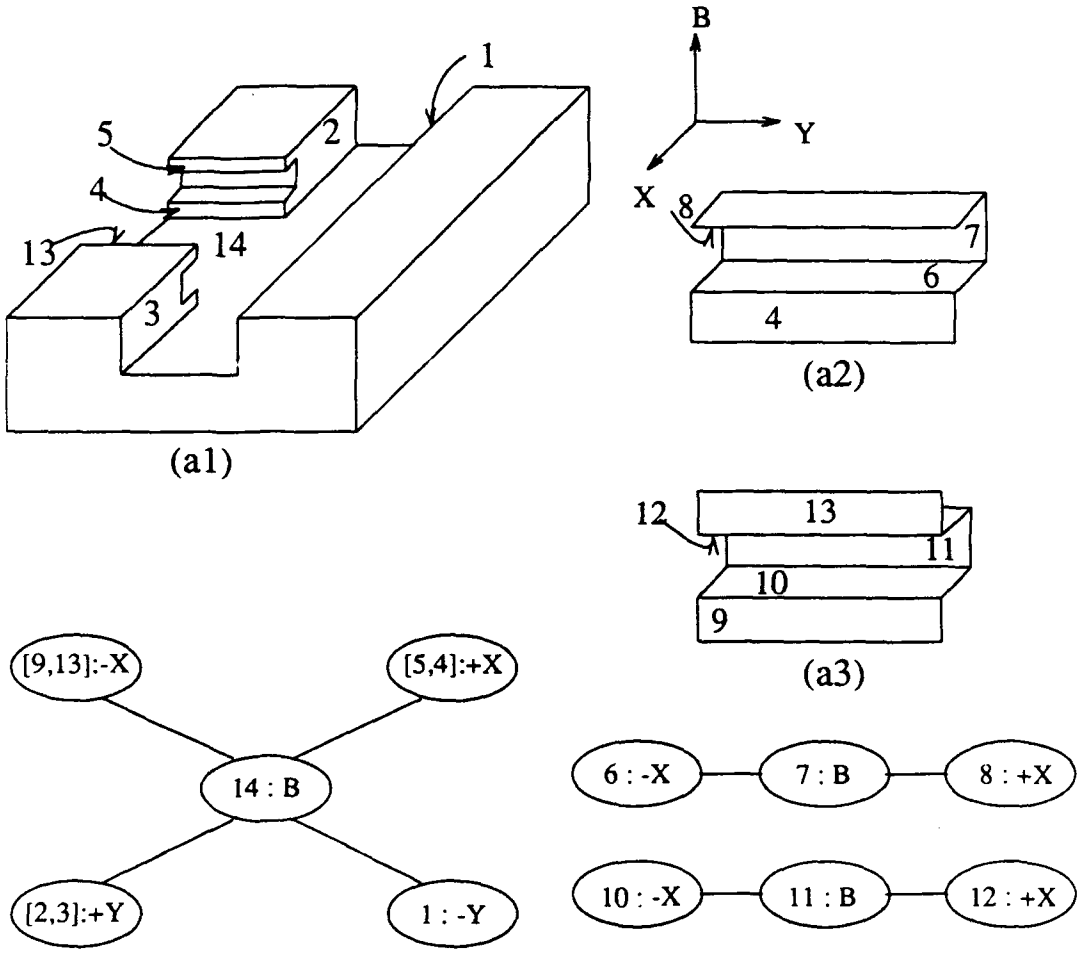
Verified Primitive Features:

(verified (thru_slot Base : 16 Sides : ((14 15) (10 11 12 13))) Certainty : 1.0)
(verified (thru_slot Base : 16 Sides : (1 2) Certainty : 1.0)
(verified (thru_slot Base : 16 Sides : (3 4) Certainty : 1.0)
(verified (thru_slot Base : 16 Sides : (5 6) Certainty : 1.0)
(verified (thru_slot Base : 16 Sides : (7 8) Certainty : 1.0)

(c)

Figure 10: Limiting the Number of Generated Hypotheses

(a) shows an example object with 5 interacting slots. The group of faces [10,11,12,13] are unifiable. The pair of coplanar faces 14 and 15 are produced by splitting a larger conceptual face, and are also unifiable. (b) shows the cavity graph generated by the system for the object. Blind graph matching would produce as many as 45 possible slot subgraphs from this cavity graph, but because only appropriate combination of node labels are considered, our system generated only 17 hypotheses. (c) shows the five hypotheses verified to be correct by the system.



(verified (thru_slot Base : 14 Sides : ((5 4) (9 13))) Certainty : 1.0)
(verified (thru_slot Base : 14 Sides : (1 (2 3))) Certainty : 1.0)
(verified (thru_slot Base : 7 Sides : (8 6)) Certainty : 1.0)
(verified (thru_slot Base : 11 Sides : (10 12)) Certainty : 1.0)

(c)

Figure 11: Importance of Unification in Finding Machining Faces with no Concave Edges

The part shown in (a1) has four thru-slots. Note that the coplanar face pairs [2,3], [5,4], and [9,13] are unifiable since they have been split as a result of interaction of slots. (a2) and (a3) show a magnified view of the two slots which are parallel with each other.
Also note that faces 5 and 13 are adjacent to all their neighboring faces via convex edges. Although methods such as those in [7] do not classify these faces as part of shape primitives, the methods proposed in this paper use unification to capture these faces in the proper shape primitive.
(b) shows the cavity graphs for the part in (a1), and (c) shows the features extracted by the prototyped system.

In this section we discussed some sample results which will shine light on the advantages obtained using the methods proposed in this work in comparison with the previous existing systems. The primitives in these test parts are not successfully extracted by previous systems.

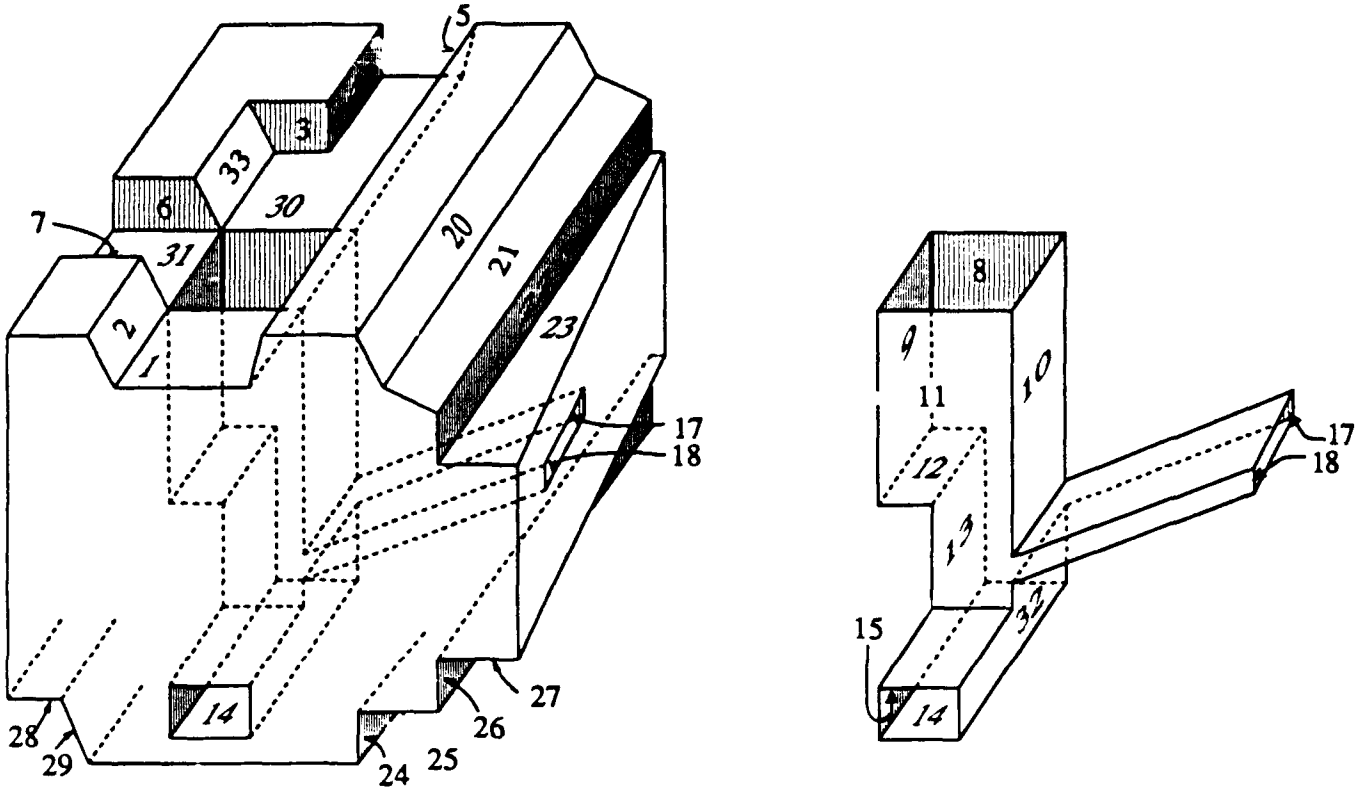
9.1 Extension to Non-Perpendicular Face Intersections. In this section we give a straightforward extension of the methods to accommodate identification and extraction of the primitives in parts in which the intersecting faces are not necessarily at right angles to each other. As we mentioned earlier, the primitive representations of figure 2 each represent a family of primitives which are obtained by varying the face-face intersection angles between all the allowed values in the respective convex or concave range that are physically realizable. In order to find the cavity graphs for a part with inclined faces, the nodes must be labeled appropriately. As before, the node corresponding to the face with the greatest number of concave edges is selected to be the base and is labeled B. A set of local coordinates, $(\vec{x_l}, \vec{y_l}, N_b)$, for the depression is constructed from the unit normal vector to the base, N_b, together with two other unit vectors, $\vec{x_l}$ and $\vec{y_l}$, such that they are mutually orthogonal to each other. For any other non-base node, the unit normal, N_f, of the corresponding face(s), f, of depression, can be decomposed in to:

$$\vec{N_f} = n_1 \cdot \vec{N_b} + n_{x_l} \cdot \vec{x_l} + n_{y_l} \cdot \vec{y_l}$$

The dominant component of $\vec{N_f}$ determines the label for the face, f.

After the construction of the cavity graphs, geometric reasoning by hypothesis generation-elimination and addition of virtual links can be carried on as highlighted in previous sections. Our next example, shown in figure 12, demonstrates the extraction of shape primitives from a part with non-perpendicular intersecting faces. It is a little more complicated than the previous examples, but it illustrates the potential of the proposed methods for handling fairly complex interactions that arise in machined parts. The part in figure 12(a1) has twelve features: five steps, one blind slot, two thru-slots, three pockets, and one prismatic-hole. To observe some of the difficulties one may note the following: The coplanar faces 1, 30, and 31 are shared by two thru-slots and one blind slot. The blind-slot and one of the thru-slots (the one formed by side faces 4 and 5) have the same axis so that the end-face 3 of the blind-slot is disconnected from its side-face 5. The other thru-slot (with side-faces 6 and 7) crosses the blind-slot splitting its side into coplanar faces 33 and 2. Both side-faces of the blind-slot intersect its base non-orthogonally. The cavity opened in the middle of the blind-slot is shown separately for clarity in figure 12(a2). There are two pockets, one with base-face 12, and the other with the base-face 14, which have the same axis and share the side-faces 8, [10,32], and 11. These two pockets open into each other. A third pocket, which has an axis perpendicular to the first two pockets, opens into the bottom of the second one (this pocket has face 8 for its base). Note that the base-face of none of the pockets intersects all of its sides. In the pocket with base 12, there is no intersection between base and side-face [10,32], in the pocket with base 14, there is no intersection between the base and side-face 11, and in the pocket with base 8, there is no intersection between the base and side-face 15. Therefore, the methods based on topology alone [5,7], or generic rules [6] cannot capture these classes of interactions. Finally, it is worth noting that the prismatic-hole in the right with faces 18, 17, 8, and 11 opens at an inclined angle into the pockets. Generally this type of interaction is difficult to handle for subtractive methods.

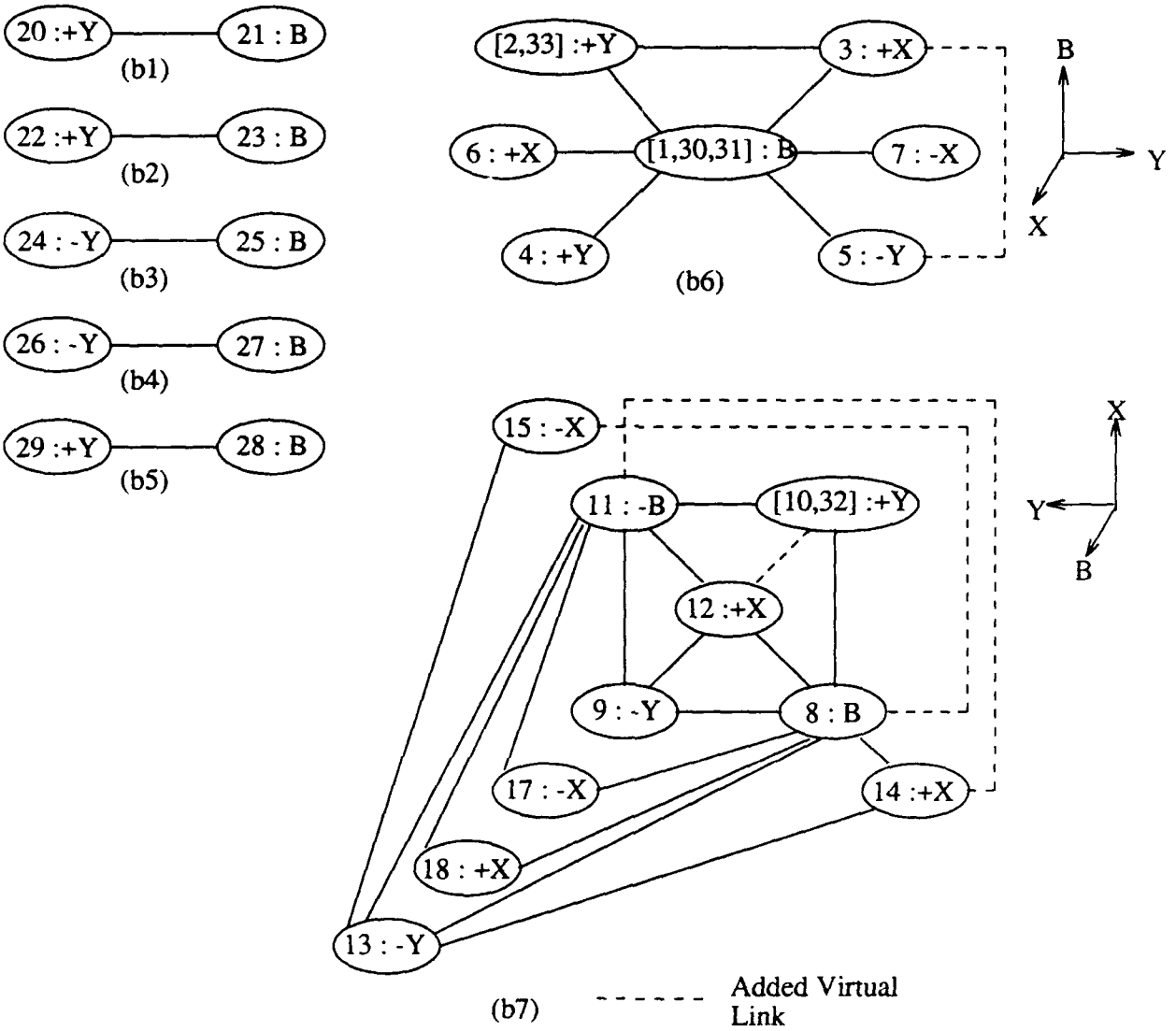
The depressions in this part are represented by seven cavity graphs. Figures (b1)-(b7) show the seven cavity graphs finally constructed by our system. Note that each cavity graph has its own local coordinates and, for example, the local coordinates for cavity graph (b6) are different than those of cavity graph (b7) as shown in the figure. We also observe that the system automatically constructed the four links (3 5), (15 8), (12 [10,32]), and (14 11), which are shown with dotted line segments in figure 12(b). The first added link completes the representation for the blind-slot primitive, (base: 1,30,31 sides:2,33,3,5), and the other three links are used to extract the three pockets, one with base-face 12, one with base-face 14, and the other with face 8 as its base. Without these links, these primitives would not be present in the hypothesis space of the problem and could not be considered. The result of primitives extracted and classified by the system is shown in figure 12(c).



(a)

Figure 12: Recognizing Primitives with Inclined Faces

(a) shows a part with inclined faces.
(b1)-(b7) show the seven cavity graphs with added virtual links for its depressions.
(c) shows the primitive features recognized by the system.



Extracted Features:

(verified (step Base : 21 Sides : (20)) Certainty : 1.0)
(verified (step Base : 23 Sides : (22)) Certainty : 1.0)
(verified (step Base : 25 Sides : (24)) Certainty : 1.0)
(verified (blind_slot Base : (1 30 31) Sides : ([2,33] 3 5)) Certainty : 0.8)
(verified (thru_slot Base : (1 30 31) Sides : (4 5)) Certainty : 1.0)
(verified (thru_slot Base : (1 30 31) Sides : (6 7)) Certainty : 1.0)
(verified (pocket Base : 12 Sides : (8 9 [10,32] 11)) Certainty : 0.8)
(verified (pocket Base : 8 Sides : (14 [10,32] 13 15)) Certainty : 0.8)
(verified (prismatic_hole Base : 18 Sides : (8 7 11)) Certainty : 1.0)
(verified (step Base : 27 Sides : (26)) Certainty : 1.0)
(verified (step Base : 28 Sides : (29)) Certainty : 1.0)
(verified (pocket Base : 14 Sides : (13 11 [10,32] 8)) Certainty : 0.8)

(c)

10. CONCLUDING REMARKS. We have described methods and algorithms to extract a wide range of interacting features and find their constituent primitive machining features. We have discussed both fundamental concepts and practical implementations. The recognition method is based on Cavity Graphs, which provide a topologic and geometric description of the depressions in the object boundary. Concepts have also been developed to combine topologic and geometric evidences to extract the most probable features of a depression. This method enables us to identify the comprising primitive features when the topologic relationships between the faces of a primitive in interaction do not include all the relationships found in an isolated primitive. The proposed representations carry more geometric information than other methods, thus allowing a more accurate machine understanding of the shape of the part.

Although the proposed method successfully analyzes the feature content of many parts, much work still needs to be done to automate the interpretation of engineering designs. A fundamental project would be to keep a record of the design session, and use this incremental information together with the CAD representation of the designed part rather than using only the CAD representation for automatic extraction of the machining features in the part.

References

[1]. Brown, C. M., "PADL-2: A technical summary," *Computing Surveys*, Vol. 12, No. 2, pp. 69-84, March 1982.

[2]. Buchanan, Bruce G., "Rule-Based Expert Systems", *Addison-Wesley*, pp. 272-292, 1985.

[3]. Chang, T. C., and Wysk, R. A., "An Introduction to Automated Process Planning Systems," *Prentice-Hall*, 1985.

[4]. Choi, B. K. "CAD/CAM compatible, Tool-Oriented Process Planning For Machining Centers," *Ph.D Thesis, Purdue University*, West Lafayette, Indiana, December 1982.

[5]. De Floriani, Leila, "A Graph Based Approach to Object Feature Recognition," *Proceedings of the Third Annual Symposium on Computational Geometry*, Waterloo, Canada, 1987.

[6]. Henderson, M. R., "Extraction of Feature Information from Three Dimensional CAD Data," *Ph.D Thesis, Purdue Univ.*, West Lafayette, Indiana, 1984.

[7]. Joshi, Sanjay B., "CAD Interface for Automated Process Planning," *Ph.D Thesis, Purdue University*, West Lafayette, Indiana, August 1987.

[8]. Kung, H., "An Investigation into Development of Process Plans from Solid Geometric Modeling Representation," *Ph.D. Thesis*, Oklahoma State Univ., 1984.

[9]. Kyprianou, L. K., "Shape Classification in Computer-Aided Design," *Ph.D Thesis, University of Cambridge*, Cambridge, England, July 1980.

[10]. Lee, Y. C., and Fu, K. S., "Machine Understanding of CSG: Extraction and Unification of Manufacturing Features," *IEEE Computer Graphics and Applications*, pp. 20-32, 1987.

[11]. Requicha, A. A. G., "Representations for Rigid Solids: Theory, Methods, and Systems," *IEEE Computer Graphics and Applications*, Vol. 12, No. 4, pp. 45-60, December 1980.

[12]. Shafer, Glenn, "A mathematical theory of evidence," *Princeton University Press*, 1976.

[13]. Staley, S. M., "Using Syntactic Pattern Recognition to Extract Feature Information From a Solid Geometric Data Base," *Computers in Mechanical Engineering*, Vol. 2, No. 2, pp. 61-66, Sept. 1983.

[14]. Weiler, K., "Topological Structures for Geometric Modeling," *Ph.D Thesis, Rensselaer Polytechnic Institute*, New York, August 1986.

[15]. Winston, Patrick Henry, "Artificial Intelligence," *Addison-Wesley*, pp. 159-166, 1984.

[16]. Woo, T. C., "Feature Extraction by Volume Decomposition," *Proceedings conference on CAD/CAM Technology in Mechanical Engineering*, MIT, Massachusetts, pp. 76-94, March 1982.

A New Class of Schemes for the Time-Dependent Stokes Equations*

John C. Strikwerda
Computer Sciences Department
and
Center for the Mathematical Sciences
University of Wisconsin–Madison
Madison, WI 49506

ABSTRACT.

A new class of finite difference scheme, using the multigrid iterative method, is presented for solving the time-dependent, incompressible Navier-Stokes equations. The schemes are quite flexible, maintaining second-order accuracy with overlapping grids in domain decomposition and with irregular grid systems. A variant of this method may be used to solve the steady-state equations.

The multigrid iteration procedure, although not essential to the method, provides a dramatic speed-up over the use of iterative methods such as S.O.R. As is typical of multigrid methods, the work per grid point is essentially independent of the finest grid spacing.

Results are presented of the application of one of the schemes to the computation of the two-dimensional flow past a rectangle in a channel. The computation makes use of overlapping grid domains. Numerical tests are presented demonstrating the second-order accuracy of the method.

1. INTRODUCTION.

In this paper, a class of finite differences schemes for the time-dependent incompressible Stokes and Navier-Stokes equations are presented. The significant features of these schemes are that they are second-order accurate, they use standard grids, that is, not staggered grids, and the schemes retain their second-order accuracy when used with non-uniform and non-orthogonal grids. In addition, the schemes are unconditionally stable. Because the schemes are inherently implicit, iterative methods are used to solve for the solution at the next time level. The method presented here is a multigrid method, but other methods may also be used.

For ease of exposition the Stokes equations are used to introduce the basic schemes. Later, in section 4 it is shown how to treat the additional terms of the Navier-Stokes equations.

* Supported by the U.S. Army Research Office

2. THE FINITE DIFFERENCE SCHEMES.

The time-dependent Stokes equations are:

$$\vec{u}_t - \nabla^2 \vec{u} + \vec{\nabla} p = \vec{f} \tag{2.1a}$$

$$\vec{\nabla} \cdot \vec{u} = g \tag{2.1b}$$

The vector function $\vec{u}$ is the velocity, and the scalar function p is the pressure. The functions $\vec{f}$ and g are considered to be given data. Notice that the pressure appears only in (2.1a) and only in terms of its spatial derivatives. In most problems the function g is identically zero, but we include the general case because it fits in naturally with the multigrid iteration method.

The equations (2.1) hold in some domain Ω, and to specify a unique solution, boundary conditions must be given. The simplest conditions are to specify the velocity $\vec{u}$ on the boundary, i.e.,

$$\vec{u} = \vec{b} \quad \text{on} \quad \partial\Omega \tag{2.2}$$

This is called the Dirichlet boundary condition.

The data g in (2.1b) and $\vec{b}$ in (2.2) must satisfy a constraint in order for a solution to exist. Using integration by parts we have

$$\int_\Omega g = \int_\Omega \vec{\nabla} \cdot \vec{u} = \int_{\partial\Omega} \vec{u} \cdot \vec{n} = \int_{\partial\Omega} \vec{b} \cdot \vec{n} \tag{2.3}$$

where $\vec{n}$ is the outer unit normal to $\partial\Omega$. For boundary conditions other than (2.2) there may or may not be a constraint.

Initial data is specified for the velocity and the pressure

$$\begin{aligned} \vec{u}(0, x, y) &= \vec{u}_0(x, y) \\ p(0, x, y) &= p_0(x, y) \end{aligned} \tag{2.4}$$

The initial data should be consistent with the equations (2.1), i.e.,

$$\vec{\nabla} \cdot \vec{u}_0 = g(0, x, y).$$

and

$$g_t - \nabla^2 g + \vec{\nabla} p_0 = \vec{\nabla} \cdot \vec{f}$$

For an introduction to the theory of the Navier-Stokes equations we refer to [6].

The first finite difference scheme is based on the Crank-Nicolson scheme for the heat equation. We let $t_n = n\Delta t$, and for a cartesian grid, let $x_\ell = \ell \Delta x$ and $y_m = m\Delta y$. We first discretize only in time

$$\frac{\vec{u}^{n+1} - \vec{u}^n}{\Delta t} - \frac{1}{2}(\nabla^2 \vec{u}^{n+1} + \nabla^2 \vec{u}^n) + \frac{1}{2}(\vec{\nabla} p^{n+1} + \vec{\nabla} p^n) = \vec{f}^{n+\frac{1}{2}} + O(\Delta t^2) \tag{2.5}$$

$$\vec{\nabla} \cdot \vec{u}^{n+1} = g^{n+1} \tag{2.5}$$

The Laplacian is discretized using the standard five-point Laplacian

$$\nabla^2 \vec{u} = \frac{\vec{u}_{\ell+1,m} + \vec{u}_{l-1,m} - 2\vec{u}_{l,m}}{\Delta x^2} + \frac{\vec{u}_{\ell,m+1} + \vec{u}_{\ell,m-1} - 2\vec{u}_{\ell,m}}{\Delta y^2} + O(\Delta x^2 + \Delta y^2) \qquad (2.6)$$

The discretization of the gradient of the pressure in (2.1a) and the divergence of the velocity in (2.1b) must be done so as to insure smoothness or regularity of the solution. Since the theory for regularity of solutions to finite difference schemes for the time-dependent Stokes equations is not developed, we use the theory for the steady Stokes equations as a guide. Considering (2.1) on all of R^n, we may transform equation (2.1) with the Fourier transform in space and the Laplace transform in time, obtaining for the case of two x dimensions

$$\begin{pmatrix} s+|\omega|^2 & 0 & i\omega_1 \\ 0 & s+|\omega|^2 & i\omega_2 \\ i\omega_1 & i\omega_2 & 0 \end{pmatrix} \begin{pmatrix} \hat{u} \\ \hat{v} \\ \hat{p} \end{pmatrix} = \begin{pmatrix} \hat{f}^1 \\ \hat{f}^2 \\ \hat{g} \end{pmatrix} \qquad (2.7)$$

where $|\omega|^2 = \omega_1^2 + \omega_2^2$. The dual variables are s, for the Laplace transform, and $\vec{\omega} = (\omega_1, \omega_2)$ for the Fourier transform.

The regularity, or smoothness properties, of (2.1) depend on the behavior of the inverse of the matrix, which we call S, on the left of (2.7). The matrix S is the symbol of the Stokes equation (2.1). In particular, the determinant of S is

$$\det S = -(s + |\omega|^2)|\omega|^2$$

and is bounded well away from zero as $|\omega|$ and $Re\ s$ become large.

We now consider the symbol of finite difference schemes for the Stokes equations, see [5]. We begin by considering central difference operations for both the gradient and divergence operators in (2.5). The symbol is

$$S_1 = \begin{pmatrix} \sigma & 0 & \dfrac{i \sin \xi_1 \Delta x}{\Delta x} a \\ 0 & \sigma & \dfrac{i \sin \xi_2 \Delta y}{\Delta y} a \\ \dfrac{i \sin \xi_1 \Delta x}{\Delta x} & \dfrac{i \sin \xi_2 \Delta y}{\Delta y} & 0 \end{pmatrix}$$

where

$$a = \frac{e^{s\Delta t} + 1}{2} \quad \text{and} \quad \sigma = \frac{e^{s\Delta t} - 1}{\Delta t} + 4a \left(\frac{\sin^2 \xi_1 \Delta x/2}{\Delta x^2} + \frac{\sin^2 \xi_2 \Delta y/2}{\Delta y^2} \right)$$

The determinant of S_1 is

$$\det S_1 = -\sigma \cdot a \cdot \left(\frac{\sin^2 \xi_1 \Delta x}{\Delta x^2} + \frac{\sin^2 \xi_2 \Delta y}{\Delta y^2} \right)$$

The range of ξ_1 and ξ_2 are given by $|\xi_1|\Delta x \le \pi$ and $|\xi_2|\Delta y \le \pi$. Note that $\det S_1$ vanishes for $|\xi_1|\Delta x$ and $|\xi_2|\Delta y$ equal to π. The vanishing of the determinant of the symbol

for certain frequencies implies that these frequencies can not be determined by the data. This statement applies as well for domains other than R^n.

In place of central differences we consider the regularized differences inroduced in [2]. These are:

$$\frac{\partial p}{\partial x} \simeq \delta_{x,r}p = \frac{p_{\ell+1,m} - p_{\ell-1,m}}{2\Delta x} - \frac{p_{\ell+2,m} - 3p_{\ell+1,m} + 3p_{\ell,m} - p_{\ell-1,m}}{6\Delta x} \tag{2.8a}$$

$$\frac{\partial p}{\partial y} \simeq \delta_{y,r}p = \frac{p_{\ell,m+1} - p_{\ell,m-1}}{2\Delta y} - \frac{p_{\ell,m+2} - 3p_{\ell,m+1} + 3p_{\ell,m} - p_{\ell,m-1}}{6\Delta y} \tag{2.8b}$$

$$\frac{\partial u}{\partial x} \simeq \delta_{x,r}u = \frac{u_{\ell+1,m} - u_{\ell-1,m}}{2\Delta x} - \frac{u_{\ell+1,m} - 3u_{\ell,m} + 3u_{\ell-1,m} - u_{\ell-2,m}}{6\Delta x} \tag{2.8c}$$

$$\frac{\partial v}{\partial y} \simeq \delta_{y,r}v = \frac{v_{\ell,m+1} - v_{\ell,m-1}}{2\Delta y} - \frac{v_{\ell,m+1} - 3v_{\ell,m} + 3v_{\ell,m-1} - v_{\ell,m-2}}{6\Delta y}. \tag{2.8d}$$

Notice that the additional regularizing terms are third-order divided differences which are shifted forward for the pressure and shifted backward for the velocity derivatives. At grid points near the boundaries, where the approximations (2.8) can not be applied, the third-order difference is shifted the other way.

For the scheme using the regularized differences (2.8) the symbol is

$$S_2 = \begin{pmatrix} \sigma & 0 & i\zeta_1 a \\ 0 & \sigma & i\zeta_2 a \\ i\bar{\zeta}_1 & i\bar{\zeta}_2 & 0 \end{pmatrix}$$

with

$$\zeta_1 = \frac{\sin \xi_1 \Delta x}{\Delta x} + \frac{4}{3} e^{i\xi_1 \Delta x/2} \frac{\sin^3 \xi_1 \Delta x/2}{\Delta x}$$

$$\zeta_2 = \frac{\sin \xi_2 \Delta y}{\Delta y} + \frac{4}{3} e^{i\xi_2 \Delta y/2} \frac{\sin^3 \xi_2 \Delta y/2}{\Delta y}.$$

The determinant of S_2 is

$$\det S_2 = -\sigma a(|\zeta_1|^2 + |\zeta_2|^2).$$

Since ζ_1 and ζ_2 do not vanish for $0 < |\xi_1|\Delta x \leq \pi$ and $0 < |\xi_2|\Delta y \leq \pi$, we see that the determinant of S_2 does not vanish except for $\xi_1 = \xi_2 = 0$, the same as for the symbol of the Stokes equations (2.1). Thus, it seems likely, based on the theory for finite difference approximations of the steady Stokes equation, that the solutions of the scheme using the regularized differences (2.8) will be smooth.

The Crank-Nicholson scheme for the heat equation is not dissipative unless the ratio $\Delta t/\Delta x^2$ and $\Delta t/\Delta y^2$ are bounded, e.g., [5]. The scheme based on (2.5) is also not dissipative and this also effects the smoothness of the solutions of the finite difference scheme. Note that dissipativity is related to the behavior of the symbol as $Re\, s \to 0$ with

$|Im\ s|\Delta t \leq \pi$. Thus the determinant of S_1 vanishes for $s\Delta t = \pi$, and this is related to the non-dissipativity of the scheme.

The next two schemes for the Stokes equations are both dissipative schemes. The first of these uses the second-order backward difference time, and is

$$\begin{aligned} \frac{3\vec{u}^{n+1}_{\ell,m} - 4\vec{u}^{n}_{\ell,m} + \vec{u}^{n-1}_{\ell,m}}{2\Delta t} - \nabla_h^2 \vec{u}^{n+1}_{\ell,m} + \nabla_h p^{n+1} &= \vec{f}^{n+1} \\ \nabla_h \cdot \vec{u}^{n+1} &= g^{n+1} \end{aligned} \tag{2.9}$$

Scheme (2.9) is second-order accurate in both time and space. Using the regularized differences (2.8), the symbol of the scheme is non-singular for $\vec{\xi}$ and s nonzero, and in the ranges $|\xi_1|\Delta x \leq \pi, |\xi_2|\Delta y \leq \pi$, and $|Im\ s|\Delta t \leq \pi$.

The third scheme to be considered here may be regarded either as a second-order accurate discretization about the time level $(n + 2/3)\Delta t$ or as a weighted average of the first two schemes. Taking 2/3 of scheme 1 and 1/3 of scheme 2 we arrive at scheme 3

$$\begin{aligned} \frac{7\vec{u}^{n+1}_{\ell,m} - 8\vec{u}^{n}_{\ell,m} + \vec{u}^{n-1}_{l,m}}{6\Delta t} - \frac{2}{3}\nabla_h^2 \vec{u}^{n+1} - \frac{1}{3}\nabla_h^2 \vec{u}^{n} + \frac{2}{3}\vec{\nabla}_h p^{n+1} + \frac{1}{3}\vec{\nabla}_h p^{n} &= \vec{f}^{n+2/3} \\ \vec{\nabla}_h \cdot \vec{u}^{n+1} &= g^{n+1}. \end{aligned} \tag{2.10}$$

Symbols for these two schemes do not vanish for $Re\ s > 0$ and $|\xi| > 0$. Schemes 2 and 3 are both dissipative, that is, the values of s for which the symbol vanishes satisfy

$$|e^s| \leq 1 - c\Delta t \left(\frac{\sin^2 \xi_1 \Delta x/2}{\Delta x^2} + \frac{\sin^2 \xi_2 \Delta y/2}{\Delta y^2} \right)$$

for some positive constant c.

3. THE MULTIGRID ALGORITM.

Each of the three schemes discussed in the previous section are implicit and require some means to solve the linear system of equations for the velocity and pressure at the next time level. Each of these systems can be written in the form

$$\vec{u}^{n+1}_{\ell,m} - \alpha\Delta t \nabla_h^2 \vec{u}^{n+1}_{\ell,m} + \alpha\Delta t \vec{\nabla}_h p^{n+1}_{\ell,m} = \vec{F} \tag{3.1a}$$

$$\vec{\nabla} \cdot \vec{u}^{n+1}_{\ell,m} = g \tag{3.1b}$$

We present here an iterative method using multigrid techniques to solve systems of the form (3.1). The general multigrid method, see [1], involves using a sequence of grids with a smoothing operator on each grid and methods for mapping function between neighboring grids. The smoothing operator should be designed to reduce the amplitude i of those modes of the error that are of highest frequency on that grid.

For the results reported in this paper the grids had $\Delta x = L_1/2^n$ and $\Delta y = L_2/2^n$ for some integer n on a rectangle with sides of length L_1 and L_2 in the x and y directions.

respectively. The sequence of coarser grids were defined by $\Delta x = L_1/2^j$ and $\Delta y = L_2/2^j$ for $j = n-1, n-2, \ldots, 1$.

On each grid the basic smoother for the velocity was a Gauss-Seidel iteration on (3.1a) using the checkerboard ordering. At each point the smoothing step is given by

$$\vec{u}_{\ell,m} \leftarrow \vec{u}_{\ell,m} - \beta(\vec{u}_{\ell,m} - \alpha\Delta t\nabla_h^2\vec{u}_{\ell,m} + \alpha\Delta t\vec{\nabla}_h p_{\ell,m} - \vec{F}_{\ell,m}) \tag{3.2}$$

where $\beta = (1 + 2\alpha\Delta t(1/\Delta x^2 + 1/\Delta y^2))^{-1}$. Note that β^{-1} is the coefficient of $\vec{u}_{\ell,m}$ on the right-hand side of (3.1a).

After the velocity was smoothed once with the Gauss-Seidel iteration, the pressure was smoothed by the operation

$$p_{\ell,m} \leftarrow p_{\ell,m} - \gamma(\vec{\nabla}_h \cdot \vec{u}_{\ell,m} - g_{\ell,m}) \tag{3.3}$$

The parameter γ in (3.3) was determined in several ways, this is discussed in more detail later.

The overall smoother for the solution consisted of two steps, each of which consisted of a velocity smoothing and a pressure smoothing.

The prolongation from a coarse grid to a finer grid was bi-quadratic interpolation and the restriction from a fine grid to a coarse grid was the adjoint of the bi-quadratic interpolation operator.

On the coarsest grid, with $\Delta x = L_1/2$ and $\Delta y = L_2/2$, there is one equation for each of the velocity components and the pressure gradients were set to zero.

In analyzing the behaviour of iterative methods for the system (3.1) it must be kept in mind that the velocity and pressure are coupled in an elliptic system. Thus, modes of the pressure error affect the velocity error through equation (3.2) and this in turn affects the pressure via equation (3.3). The overall effect must be to reduce the error in both the velocity and the pressure.

It was found that the velocity errors were reduced quite rapidly by the multigrid iterations, taking only a couple of V-cycles to reduce the error below the desired tolerance. However, the pressure errors were reduced more slowly. Why this is so is the result of continuing investigation.

The parameter γ in (3.3) was chosen to depend on the grid spacing. The formula is derived as follows. A change of ε in the pressure at one grid point contributes a change to the velocity at neighboring grid points through (3.2) proportional to $\alpha\beta\Delta t\varepsilon/\Delta x$ or $\alpha\beta\Delta t\varepsilon/\Delta y$. The proportionality depends on the relative locations of the points, but is independent of the grid spacing. The changes in the velocity result in changes in the pressure through (3.3) proportional to $\alpha\beta\gamma\Delta t\varepsilon(1/\Delta x^2 + 1/\Delta y^2)$. This indicates that γ should be given by

$$\gamma = c_0\alpha^{-1}\left(1 + 2\alpha\Delta t(1/\Delta x^2 + 1/\Delta y^2)\right)\left(\Delta/\Delta x^2 + \Delta/\Delta y^2\right)^{-1} \tag{3.4}$$

for some value of c_0.

4. NAVIER-STOKES EQUATIONS.

The incompressible Navier-Stokes equations are

$$\vec{u}_t - \nabla^2 \vec{u} + \nabla \vec{u}\vec{u}^T + \vec{\nabla} p = \vec{f} \tag{5.1a}$$

$$\vec{\nabla} \cdot \vec{u} = g \tag{5.1b}$$

(The superscript T on $\vec{u}$ denotes transpose.)

The modifications to the schemes so as to include the nonlinear convection terms can be done so as to maintain the linearity of the equations that must be solve at each time step. To show how this is done, consider the first scheme (2.5) with the addition of the convection terms. We have

$$\frac{\vec{u}^{n+1} - \vec{u}^n}{\Delta t} - \frac{1}{2}(\nabla^2 \vec{u}^{n+1} + \nabla^2 \vec{u}^n) + \nabla \vec{u}^{n+1/2} \vec{u}^{n+1/2,T} + \frac{1}{2}(\vec{\nabla} p^{n+1} + \vec{\nabla} p^n) = \vec{f}^{n+\frac{1}{2}} + O(\Delta t^2).$$

By differencing the convection term as

$$\nabla \vec{u}^{n+\frac{1}{2}} \vec{u}^{n+\frac{1}{2},T} = \frac{1}{2} \nabla \vec{u}^{n+1} \vec{u}^{n,T} + \frac{1}{2} \nabla \vec{u}^n \vec{u}^{n+1,T} + O(\Delta t^2)$$

the second-order accuracy is maintained and the system for $\vec{u}^{n+1}$ is linear.

Similarly, scheme 2, i.e., (2.9), is modified by the approximation

$$\nabla \vec{u}^{n+1} \vec{u}^{n+1,T} = \nabla \vec{u}^{n+1} \vec{u}^{n,T} + \nabla \vec{u}^n \vec{u}^{n+1,T} - \nabla \vec{u}^n \vec{u}^{n,T} + O(\Delta t^2)$$

to preserve the second-order accuracy and linearity.

Scheme 3, i.e., (2.10), is modified by the approximation

$$\nabla \vec{u}^{n+\frac{2}{3}} \vec{u}^{n+\frac{2}{3},T} = \frac{2}{3} \nabla \vec{u}^{n+1} \vec{u}^{n,T} + \frac{2}{3} \nabla \vec{u}^n \vec{u}^{n+1,T} - \frac{1}{3} \nabla \vec{u}^n \vec{u}^{n,T} + O(\Delta t^2).$$

5. TEST RESULT.

We give the results of a test case, for which the domain is the square given by $0 \le x \le 1$ and $0 \le y \le 1$. The exact solution is given by

$$\begin{aligned} u &= \quad te^{tx} \cos(ty) \\ v &= -te^{tx} \sin(ty) \\ p &= -e^{tx}(x \cos(ty) - y \sin(ty)) + 5. \end{aligned} \tag{5.1}$$

where $\vec{u} = (u, v)$. At each time step the finite difference equations were solved to an accuracy of $5 \cdot 10^{-6}$. Dirichlet boundary conditions were used, e.g., (2.2) where $\vec{b}$ was the exact data from (4.1).

As shown in Table 1 the solutions were computed to second-order accuracy. For the results shown in Table 1 the constant c_0, see (3.4) was 0.2.

The method gives a substantial increase in speed over methods based on successive-over-relaxation, see e.g., [3]. Further improvement should give more efficiency.

The finite difference schemes introduced here can be used with domain decomposition, see [4]. Results of a computation of two-dimensional flow past a rectangular shaped obstruction are displayed in Figure 1. The domain is decomposed into four sub-domains. The velocities are interpolated from one subdomain to the boundaries of the other subdomains.

Table 1			
	Errors		
M	u	v	p
8	1.3(−5)	1.5(−5)	2.1(−3)
16	9.1(−7)	1.3(−6)	5.8(−4)
32	2.6(−7)	2.3(−7)	1.5(−4)
64	7.2(−8)	5.3(−8)	3.8(−5)

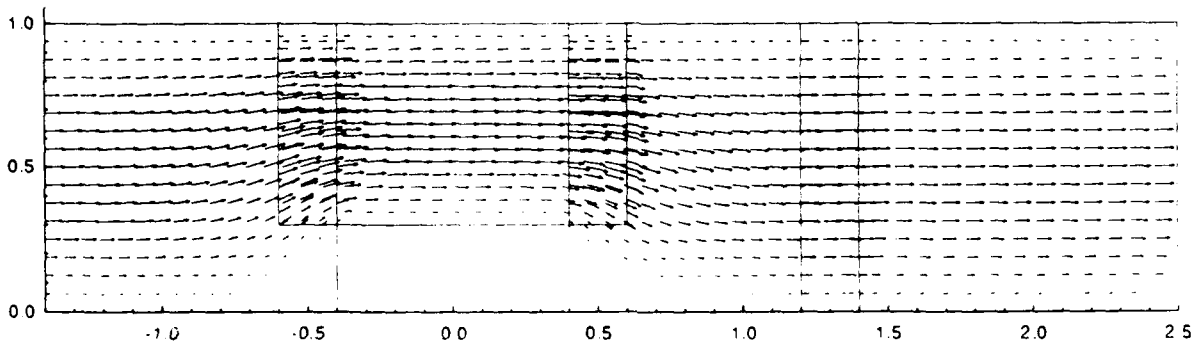


Figure 1

REFERENCES

[1] A. Brandt and N. Dinar, *Multigrid solutions to Elliptic Flow Problems, Numerical Methods for Partial Differential Equations*, S. V. Parter, ed., Academic Press, Inc., New York, 1979.

[2] J. C. Strikwerda, *Finite difference methods for the Stokes and Navier-Stokes equations*, SIAM J. Sci. Stat. Comput., 5 (1984), pp. 56-68.

[3] J. C. Strikwerda, *An iterative method for solving finite difference approximations to the Stokes equations*, SIAM J. Numer. Anal., 21 (1984), pp. 447-458.

[4] J. C. Strikwerda and C. D. Scarbnick, *A Domain Decomposition Method for Incompressible Viscous Flow*, SIAM J. Sci. Stat. Comput., to appear (1990).

[5] J.C. Strikwerda, *Finite Difference Schemes and Partial Differential Equations*, Wadsworth & Brooks/Cole, Pacific Grove, CA, 1989.

[6] R. Temam, *Navier-Stokes Equations, Theory and Numerical Analysis*, North-Holland, Amsterdam, 1984.

Diagonal Implicit Multigrid Solution of Compressible Turbulent Flows

R. R. Varma and D. A. Caughey

Sibley School of Mechanical and Aerospace Engineering
Cornell University
Ithaca, New York 14853

1 Abstract

A multigrid Alternating Direction Implicit scheme has been developed to solve the compressible Navier-Stokes equations for two-dimensional problems. The scheme is an extension of that developed by Caughey [1] to solve the Euler equations of inviscid compressible flow.

Spatial discretization of the governing equations is done using a finite volume approximation to provide flexibility in dealing with complicated geometries. In order to prevent decoupling of the solution at odd- and even-numbered points of the grid, and to prevent oscillations of the solution near shock waves, artificial dissipation is added in the form of an adaptive blend of second and fourth differences of the solution. The time-linearized implicit operator is approximated as the product of two one-dimensional factors. In order to improve computational efficiency, each of the implicit factors is diagonalized using a local similarity transformation. This diagonalization is possible only when the contributions of the viscous terms to the implicit operator are approximated or eliminated altogether. But it is fairly common to treat the viscous terms explicitly when using even non-diagonalized ADI schemes. The resulting scheme requires the solution of four scalar pentadiagonal systems along each line in each of the two mesh directions for each time step. The implicit scheme is used within the framework of the multigrid method to further accelerate convergence to a steady state.

The motivation for the development of the method is to improve upon the convergence rates of explicit multigrid methods on the highly stretched grids required for high Reynolds number flows. The turbulence model used here is based on the algebraic model developed by Baldwin and Lomax [2]. Results are presented for flows past airfoils. Flow field results are presented to confirm the accuracy of the method, and convergence rates are

compared with other methods to demonstrate the efficiency of the implicit ADI multigrid method.

2 Analysis

2.1 The Equations

The Reynolds Averaged Navier-Stokes equations in two dimensions can be written as

$$\frac{\partial \vec{w}}{\partial t} + \frac{\partial \vec{f}}{\partial x} + \frac{\partial \vec{g}}{\partial y} = \frac{\partial \vec{f}_v}{\partial x} + \frac{\partial \vec{g}_v}{\partial y}, \tag{1}$$

where

$$\vec{w} = \{\rho,\ \rho u,\ \rho v,\ e\}^T \tag{2}$$

is the vector of conserved variables,

$$\vec{f} = \{\rho u,\ \rho u^2+p,\ \rho uv,\ (e+p)u\}^T, \tag{3}$$

$$\vec{g} = \{\rho v,\ \rho uv,\ \rho v^2+p,\ (e+p)v\}^T, \tag{4}$$

are the inviscid flux vectors in the x and y directions respectively, and

$$\vec{f}_v = \{0,\ \sigma_{xx},\ \sigma_{xy},\ u\sigma_{xx}+v\sigma_{xy}-q_x\}^T, \tag{5}$$

$$\vec{g}_v = \{0,\ \sigma_{xy},\ \sigma_{yy},\ u\sigma_{xy}+v\sigma_{yy}-q_y\}^T, \tag{6}$$

are the viscous flux vectors in the x and y directions respectively. The variables ρ and p are the fluid density and pressure, u and v are the velocity components in the x and y directions, and e is the total energy per unit volume. The equation of state for a calorically perfect gas is used to relate the pressure to the total energy

$$p = (\gamma - 1)\left\{e - \rho\frac{u^2 + v^2}{2}\right\}, \tag{7}$$

where γ is the ratio of specific heats. For air, $\gamma = 1.4$. The viscous stresses and heat fluxes, with the assumption of Stokes' hypothesis, are given by

$$\sigma_{xx} = 2\mu\frac{\partial u}{\partial x} - \tfrac{2}{3}\mu\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right), \tag{8}$$

$$\sigma_{xy} = \mu\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right), \tag{9}$$

$$\sigma_{yy} = 2\mu\frac{\partial v}{\partial y} - \tfrac{2}{3}\mu(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}), \quad (10)$$

$$q_x = -k\frac{\partial T}{\partial x}, \quad (11)$$

$$q_y = -k\frac{\partial T}{\partial y}. \quad (12)$$

Here T is the temperature, μ is the viscosity and k is the thermal conductivity of the fluid.

2.2 Turbulence Model

The effects of turbulence are modeled using the eddy diffusivity concept for the Reynolds stresses and eddy thermal conductivity for the turbulent heat fluxes. The total diffusivities are given by

$$\mu = \mu_{mol} + \mu_t, \quad (13)$$

$$k = k_{mol} + k_t, \quad (14)$$

where μ_{mol} and k_{mol} are the molecular quantities, and μ_t and k_t are the turbulent quantities. We obtain closure by modeling μ_t analytically using a zero equation model and calculating k_t from Pr_t, the turbulent Prandtl number, which is chosen to be equal to 0.9 .

The turbulence model is based on the algebraic model of Baldwin and Lomax [2]. This is a two-layer zero-equation eddy-viscosity model. The eddy viscosity μ_t is given by

$$\mu_t = \begin{cases} (\mu_t)_{inner} & y \le y_{crossover} \\ (\mu_t)_{outer} & y > y_{crossover} \end{cases} \quad (15)$$

where y is the distance normal to the wall and $y_{crossover}$ is the smallest value of y at which the value calculated from the inner formula exceeds the value from the outer formula. Both the inner and outer formulas have the general form

$$\mu_t = \alpha\, K\, \ell\, \mathrm{u}\, \rho, \quad (16)$$

where ℓ is the length scale and u is the velocity scale prescribed by the model.

In the *inner region* the length scale is the Prandtl mixing length modified by the van Driest damping factor. The velocity scale is calculated as the

product of the modified mixing length ℓ and the magnitude of the local vorticity $|\omega|$, according to the Prandtl-van Driest formulation.

$$\ell = \kappa y \left(1 - e^{-y^+/A^+}\right) \text{ where } y^+ = \frac{y u_\tau}{\nu} = \frac{y}{\nu}\sqrt{\frac{\tau_w}{\rho}}, \tag{17}$$

$$\mathrm{u} = \ell\,|\omega|\,, \tag{18}$$

$$\alpha = 1.0\,;\; K = 1.0, \tag{19}$$

where $\kappa = 0.4$ is the von Karman constant and $A^+ = 26.0$ is an effective sub-layer thickness.

To determine the scales in the *outer region* Baldwin and Lomax defined a function

$$F(y) = y\,|\omega|\,\left(1 - e^{-y^+/A^+}\right). \tag{20}$$

This function $F(y)$ is used to compute the length and velocity scales in the outer region according to

$$\ell = 1.6\,y_{Fmax}, \tag{21}$$

$$\mathrm{u} = \min\left(F_{max}, \frac{(U_{Fmax} - U_{min})^2}{F_{max}}\right), \tag{22}$$

where F_{max} is the maximum value of $F(y)$ that occurs in a profile, y_{Fmax} is the y-location of that maximum, U_{Fmax} is the total velocity at that location and U_{min} is the minimum velocity in the profile. The Clauser constant α and the Klebanoff intermittency factor K are given by

$$\alpha = 0.0168, \tag{23}$$

$$K = \left[1 + 5.5\left(\frac{0.3\,y}{y_{Fmax}}\right)^6\right]^{-1}. \tag{24}$$

The model is modified slightly when applied to the wake. In the wake the van Driest damping factor is set equal to 1, and the y-distance is measured from the first coordinate line in the wrap-around direction of the C-grid, *i.e.* from the $\eta = 0$ line.

2.3 Finite Volume Formulation

To facilitate the handling of complex geometries a finite volume formulation is used. First the equations are transformed from the physical plane to the

computational plane through a non-singular transformation. In the new plane the system of equations can be written as

$$\frac{\partial \vec{W}}{\partial t} + \frac{\partial \vec{F}}{\partial \xi} + \frac{\partial \vec{G}}{\partial \eta} = \frac{\partial \vec{F}_v}{\partial \xi} + \frac{\partial \vec{G}_v}{\partial \eta}, \tag{25}$$

where

$$W = hw$$

is the vector of transformed dependent variables and $h = x_\xi y_\eta - y_\xi x_\eta$ is the determinant of the Jacobian of the transformation. The transformed inviscid and viscous fluxes are given by

$$F = f y_\eta - g x_\eta, \tag{26}$$

$$G = -f y_\xi + g x_\xi, \tag{27}$$

$$F_v = f_v y_\eta - g_v x_\eta, \tag{28}$$

$$G_v = -f_v y_\xi + g_v x_\xi. \tag{29}$$

The dependent variables are taken to be the cell average quantities. Spatial derivatives are approximated by evaluating the fluxes across the faces of each cell. This requires the values of the velocities on all cell faces, which are taken to be the average of the cell average values of the two cells sharing the face.

2.4 Thin Layer Approximation

The thin layer approximation neglects all diffusion processes parallel to the body surface. In this respect it is similar to the classical boundary layer approximation, but it is different in that no assumptions are made regarding the pressure, and the momentum equation normal to the body surface is retained. In the present formulation the ξ-direction is the coordinate direction approximately parallel to the surface and the η-direction is approximately normal to the surface. Therefore all ξ-derivatives are neglected while all η-derivatives are retained in all the viscous terms and in the evaluation of the cartesian derivatives in the viscous terms. In particular, the viscous flux in the η-direction $\vec{G}_v$ which involves the calculation of cartesian derivatives (Eqs. 8 - 12) is modified to a simpler form $\vec{G}'_v$ containing only η-derivatives. The system of equations reduces to

$$\frac{\partial \vec{W}}{\partial t} + \frac{\partial \vec{F}}{\partial \xi} + \frac{\partial \vec{G}}{\partial \eta} = \frac{\partial \vec{G}'_v}{\partial \eta} \tag{30}$$

In most Navier-Stokes solutions for high Reynolds number turbulent flows the diffusion terms involving derivatives parallel to the body surface have not been resolved even when the appropriate terms have been retained in the equations. This is due to the coarseness of the mesh in that direction. In other words, if the mesh is not fine enough in the direction parallel to the body surface to resolve these diffusion terms, it is a wasted effort to try to calculate them.

2.5 Artificial Dissipation

The finite volume scheme for the Euler equations does not contain any dissipative terms. In order to prevent odd-even point decoupling and oscillations near shock waves or stagnation points artificial dissipation terms must be added when solving the Euler equations. The Navier-Stokes equations on the other hand possess dissipative properties due to the presence of the viscous terms. However the physical dissipation provided by these terms in regions away from the shear layer may not be sufficient to guarantee stability. So in order to maintain the stability and robustness of the numerical procedure it was necessary to add artificial dissipation. The terms were constructed as an laptive blend of second and fourth differences with the directional scalin of the terms suggested by Caughey [1].

The modified set of equations is

$$\frac{\partial \vec{W}}{\partial t} + \frac{\partial \vec{F}}{\partial \xi} + \frac{\partial \vec{G}}{\partial \eta} = \frac{\partial \vec{G}_v}{\partial \eta} +$$

$$\frac{\partial}{\partial \xi}\left\{(\epsilon^{(2)}\frac{\partial \vec{w}}{\partial \xi} - \epsilon^{(4)}\frac{\partial^3 \vec{w}}{\partial \xi^3})\right\} + \frac{\partial}{\partial \eta}\left\{(\epsilon^{(2)}\frac{\partial \vec{w}}{\partial \eta} - \epsilon^{(4)}\frac{\partial^3 \vec{w}}{\partial \eta^3})\right\} \qquad (31)$$

where $\epsilon^{(2)}$ and $\epsilon^{(4)}$ are as defined in [1].

2.6 Iterative Scheme

To construct an iterative scheme to solve the difference equations, the spatial derivatives are first approximated implicitly and the changes in the flux vectors are linearized in time. The implicit operator thus obtained is approximated as the product of two one-dimensional factors. The development thus far follows that of Briley and McDonald [3] and Beam and Warming [4]. Since the artificial dissipation terms must be treated implicitly for rapid

convergence, this scheme would lead to the requirement to solve block pentadiagonal systems for each of the two factors. An alternative suggested by Pulliam and Chaussee [5] is to diagonalize the implicit factors using a local similarity transformation. This yields the decoupled set of equations

$$\{I + \theta\Delta t\,[\Lambda_{A^n_{i,j}}\delta_\xi - \epsilon^{(2)}\delta^2_\xi(1/h) + \epsilon^{(4)}\delta^4_\xi(1/h)]\}Q^{-1}_{A^n_{i,j}}$$

$$\times\, Q_{B^n_{i,j}}\{I + \theta\Delta t[\Lambda_{B^n_{i,j}}\delta_\eta - \epsilon^{(2)}\delta^2_\eta(1/h) + \epsilon^{(4)}\delta^4_\eta(1/h)]\}\Delta\vec{V}^n_{i,j}$$

$$= -\Delta t\, Q^{-1}_{A^n_{i,j}}\{\delta_\xi\vec{F}_{i,j} + \delta_\eta\vec{G}_{i,j} - \delta_\eta(\vec{G}'_v)_{i,j} - \epsilon^{(2)}_{i,j}(\delta^2_\xi + \delta^2_\eta)\vec{w} + \epsilon^{(4)}_{i,j}(\delta^4_\xi + \delta^4_\eta)\vec{w}\}^n. \tag{32}$$

Here A and B are the Jacobians of the transformed flux vectors

$$A = \left\{\frac{\partial F}{\partial W}\right\}\ ;\ B = \left\{\frac{\partial G}{\partial W}\right\} \tag{33}$$

and Λ_A and Λ_B are diagonal matrices whose diagonal elements are the eigenvalues of A and B. The modal matrices Q_A and Q_B diagonalize A and B according to

$$Q_A^{-1}AQ_A = \Lambda_A\ ;\ Q_B^{-1}BQ_B = \Lambda_B. \tag{34}$$

The correction ΔW to the solution in each computational cell is given by

$$\Delta W^n_{i,j} = Q_B\Delta V^n_{i,j}. \tag{35}$$

The elements of the Jacobian matrices, their modal matrices and the diagonal matrices have been given by Pulliam and Chaussee [5]. The system of equations (Eq. 32) are solved at each time step by solving four scalar pentadiagonal systems along each line in each of the two directions.

2.7 Boundary Conditions

2.7.1 Explicit Boundary Conditions

The solutions computed to date have been for subsonic freestream Mach numbers. For subsonic inflow the boundary conditions are based on the Riemann invariants for the one-dimensional problem normal to the boundary. The first Riemann invariant

$$R_1 = \frac{x_\xi v - y_\xi u}{\sqrt{x^2_\xi + y^2_\xi}} + \frac{2c}{\gamma - 1} \tag{36}$$

is extrapolated from the interior of the domain, while the second Riemann invariant

$$R_2 = \frac{x_\xi v - y_\xi u}{\sqrt{x_\xi^2 + y_\xi^2}} - \frac{2c}{\gamma - 1} \tag{37}$$

is specified, as are the entropy and the tangential velocity component. At a subsonic outflow boundary ρu, ρv and the entropy are extrapolated from the interior, while pressure is specified to be the freestream value.

On the body surface the no-slip boundary condition is applied, i.e., the velocity components u and v are set equal to zero. Also, the surface is assumed to be adiabatic, i.e. $\partial T/\partial n = 0$.

2.7.2 Implicit Boundary Conditions

In the far-field the implicit boundary conditions are treated in a manner consistent with characteristic theory; on the body surface homogeneous Dirichlet conditions are applied.

2.8 Multigrid Algorithm

The scheme is implemented within the framework of the multigrid algorithm following Jameson [6] and Caughey [1]. The algorithm consists of the following steps:

1. Form a coarse grid by eliminating every second line of the fine grid in each coordinate direction.
2. Restrict the flow variables to the coarse mesh using area-weighted averages of the values on the fine mesh.
3. Drive the corrections on the coarse mesh with the residual computed on the fine mesh.
4. Continue until the coarsest mesh.
5. Prolong corrections back to the finer meshes using bilinear interpolation.
6. Add corrections on the finest mesh to the solution.
7. Repeat cycle.

Both the body-surface and the far-field boundary conditions are updated on coarser meshes. A fixed V-cycle is used in which the solution is advanced one time step on each mesh as the grid is coarsened and refined. A fixed coefficient second difference form of the dissipation is used on all but the finest mesh.

3 Results

The scheme described above has been applied to compute transonic flows past the NACA 0012 airfoil. The following cases are presented here:

1. $M_\infty = 0.7, \alpha = 1.49, Re_c = 9 \times 10^6$
2. $M_\infty = 0.799, \alpha = 2.26, Re_c = 9 \times 10^6$

These are cases A1 and A3 of the Viscous Transonic Airfoil Workshop of 1987 [7]. All results were calculated on C-grids containing 192×48 cells in the wrap-around and body-normal directions respectively. Of the 192 points in the wrap-around direction 120 were on the airfoil. The distance from the airfoil to the first coordinate line was 5×10^{-5} of a chord which corresponds to a y^+ less than 4 for the given Reynolds number. The farfield boundaries were about 7 chord lengths from the airfoil. The cells are highly clustered in the η-direction near the surface of the airfoil and have large aspect ratios. The largest aspect ratio is of the order 10^3.

The calculations were performed on an IBM 3090/600J. A typical calculation took about $100\mu s$ per work-unit per point as compared to $93\mu s$ for an Euler calculation using the same method. A work unit is the amount of computation required for advancing one time step on the finest mesh.

Airfoil surface pressure distributions for the two cases are presented to verify the accuracy of the scheme. The results of the two cases are compared with the experiments of Harris [8] and the computational results from the VTA Workshop [7]. For case(1) the flow is attached and just slightly supersonic near the leading edge upper surface. The measured experimental angle of attack for this case was 1.86°. This was corrected to 1.49° by Harris using a linear method for accounting for wind tunnel wall effects. Figure 1 shows that the computed surface pressures are in excellent agreement with experimental data. Table 1 gives a comparison of the force coefficients. The lift coefficient obtained using the present method is about 2% less than the experimental value and is within the range of values obtained computationally.

Drag coefficients are difficult to calculate accurately because the pressure integration for drag is very sensitive. Even so the value obtained 0.0084 is only about 6% different from the experimental value, and is within the range of the VTA Workshop values. Of the total computed drag about 17% is due to skin friction and the rest due to pressure drag.

Convergence results are presented in Figures 2 and 3. Grid sequencing is used, i.e. multigrid solutions are first obtained on coarser grids and then interpolated for use as initial conditions on fine grids. The error is defined as the residual of the continuity equation averaged over all the grid cells. The logarithm of this error is plotted against the number of work units in Figure 2 for a single grid and for 4 levels of multigrid. We see that with 4 levels of multigrid the error has been reduced 8 orders of magnitude in 500 work units, whereas for the single grid it has been reduced only 3 orders of magnitude. The asymptotic rate of convergence is clearly much improved with multigrid. The CFL number for both cases was 24, and local time-stepping was used. Figure 3 shows that the three measures of global convergence - the lift coefficient C_L, the drag coefficient C_D and the number of cells N_{sup} in which the local velocity is supersonic, have converged to within plottable accuracy of the final values within 50 work units when using 4 levels of multigrid.

Figure 4 compares the convergence history of the implicit multigrid scheme presented in this paper with the explicit multigrid Runge-Kutta scheme of Martinelli and Jameson [9]. The overall convergence rate and the asymptotic rate are improved with the present implicit scheme.

The flow conditions for case(2) are $M_\infty = 0.799, \alpha = 2.26°$ and $Re_c = 9 \times 10^6$. The flow field contains a shock on the airfoil upper surface at an x/c of about 0.5. The shock is strong enough to induce a significant boundary layer separation. The experimental data obtained by Harris [8] are compared with the computational results in Figure 5. The computational angle of attack (2.26°) is obtained from the measured angle of attack (2.86°) using a linear wind tunnel wall correction procedure [8]. Our results are generally in close agreement with other computational results [10] that use the same turbulence model but the shock strength and the shock position are incorrectly predicted. The computed shock is both stronger and farther downstream than that measured experimentally. The convergence history is shown in Figure 6. The rate is comparable to that obtained for the simpler case(1).

4 Conclusions

The multigrid diagonalized Alternating Direction Implicit scheme developed by Caughey has been extended to solve the thin-layer Navier-Stokes equation for compressible flow. The Baldwin-Lomax algebraic turbulence model was used. Results for transonic flows past airfoils were presented. They show that for attached flow the computed flowfield data are in good agreement with the experimental data, but for flows with strong shocks and shock-induced separation the agreement is poor. This can be attributed to the equilibrium nature of the turbulence model used. The convergence rates obtained using the implicit method described above are better than those obtained using the explicit Runge-Kutta method.

5 Acknowledgments

This research has been supported in part by the Independent Research and Development Program of the McDonnell Douglas Corporation and by the U. S. Army Research Office through the Mathematical Sciences Institute of Cornell University. The calculations reported here were performed at the Cornell National Supercomputer Facility, a resource of the Cornell Theory Center, which receives major funding from the National Science Foundation and the IBM Corporation, with additional support from New York State, and the Corporate Research Institute.

References

[1] Caughey, D. A., *Diagonal Implicit Multigrid Algorithm for the Euler Equations,* **AIAA Journal**, Vol. 26, No.7, July 1988, pp 841-851.

[2] Baldwin, B. S. and H. Lomax, *Thin Layer Approximation and Algebraic Model for Separated Turbulent Flows.* **AIAA Paper 78-257**, 16th Aerospace Sciences Meeting, Huntsville, Alabama, January 1978.

[3] Briley, W. R. and H. McDonald, *Solution of the Three-Dimensional Compressible Navier-Stokes Equations by an Implicit Technique.* **Proceedings of the Fourth International Conference on Numerical Methods in Fluid Dynamics, Lecture Notes in Physics**, Vol. 35, Springer-Verlag, New York, 1974, pp 205-110.

[4] Beam, R. M. and R. F. Warming, *An Implicit Finite-Difference Algorithm for Hyperbolic Systems in Conservation Law Form*, **Journal of Computational Physics**, Vol. 22, No.1, Sept. 1976, pp 87-110.

[5] Pulliam, T. H. and D. S. Chaussee, *A Diagonal Form of an Implicit Approximate-Factorization Algorithm*, **Journal of Computational Physics**, Vol. 39,1981, pp 347-363.

[6] Jameson, A., *Solution of the Euler Equations by a Multigrid Method*, **MAE Report 1613** Mechanical and Aerospace Engineering, Princeton University, Princeton, N.J., June 1983.

[7] Holst, T. L., *Viscous Transonic Airfoil Workshop Compendium of Results*, **Journal of Aircraft** Vol. 25, No. 12, December 1988, pp 1073-1087.

[8] Harris, C. D., *Two-Dimensional Aerodynamic Characteristics of the NACA 0012 Airfoil in the Langley 8-Foot Transonic Pressure Tunnel*, **NASA TM 81927, 1987**.

[9] Martinelli, L. and A. Jameson, *Validation of a Multigrid Method for the Reynolds Averaged Equations*, **AIAA-88-0414** AIAA 26th Aerospace Sciences Meeting, January 11-14, 1988, Reno, Nevada.

[10] King, L.S., *A Comparison of Turbulence Closure Models for Transonic Flows about Airfoils*, **AIAA-87-0418** AIAA 25th Aerospace Sciences Meeting, January 12-15, 1987, Reno, Nevada.

	Experimental Results of Harris [1981]	FLO53MDI (this work)	Computational Results VTA Workshop [1987]
Lift Coeff. (Cl)	0.2410	0.2379	0.2350 - 0.2620
Drag Coeff. (Cd)	0.0079	0.0084	0.0074 - 0.0100

Table 1: Comparison of Force Coefficients for case(1)

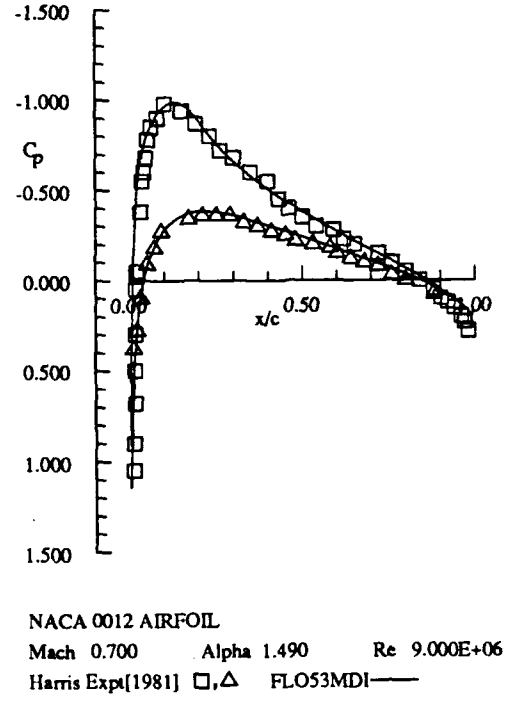


Figure 1: Comparison of computed (FLO53MDI) surface pressure coefficients with values from the Harris experiment

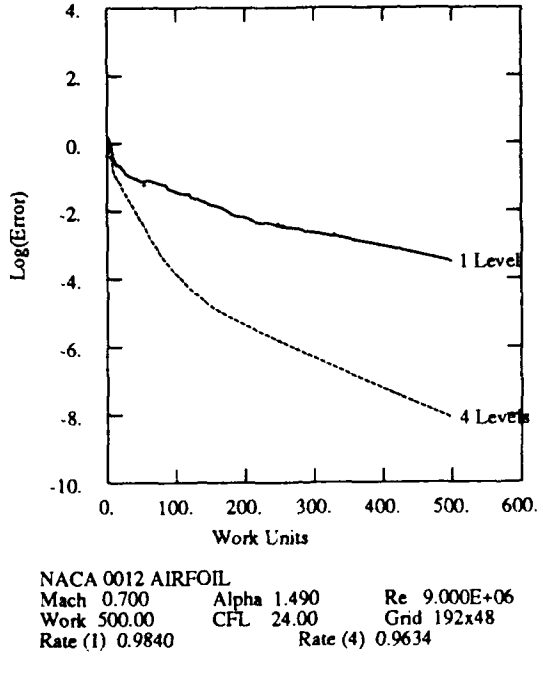


Figure 2: Convergence rates using a single grid and using 4 levels of multigrid

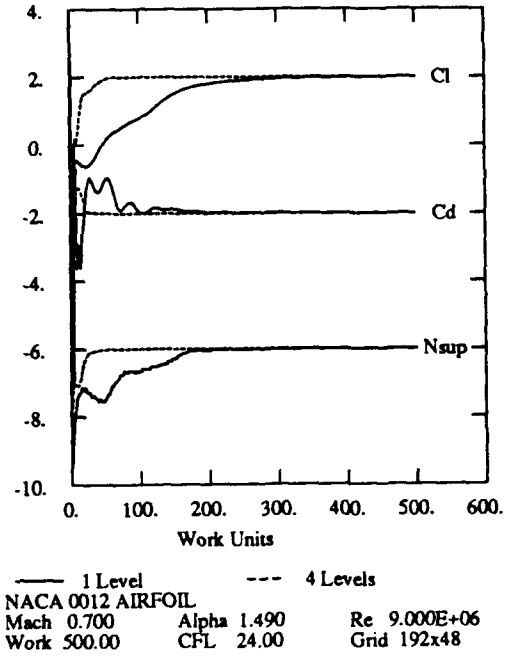


Figure 3: Convergence of global measures

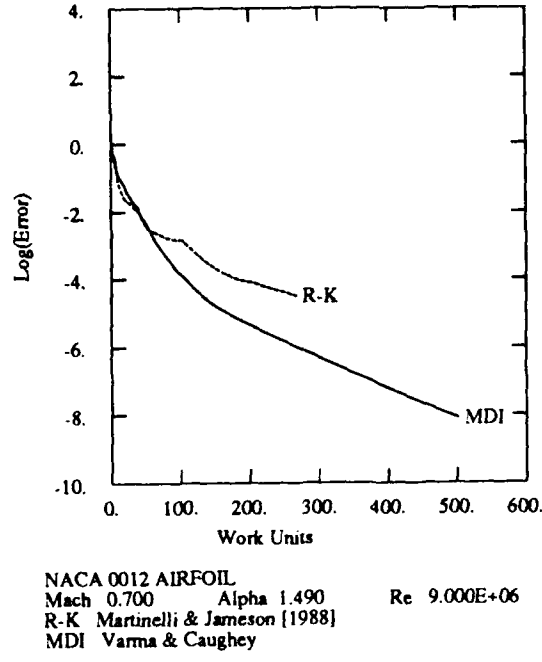


Figure 4: Comparison of the convergence rate of the present method with the explicit Runge-Kutta method of Martinelli and Jameson[1988]

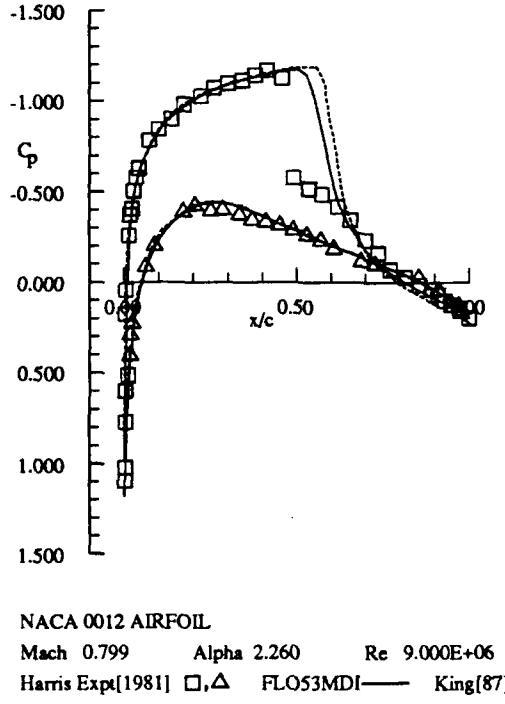


Figure 5: Comparison of pressure coefficients for case(2): $M_\infty = 0.799$

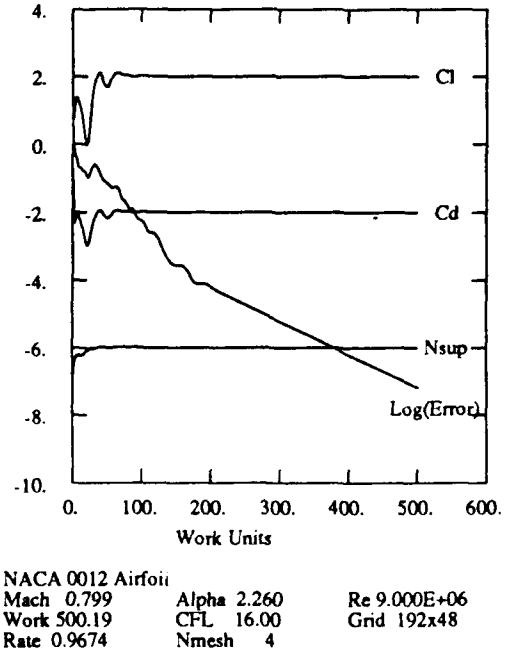


Figure 6: Convergence history for case(2): $M_\infty = 0.799$

Multigrid Diagonal Implicit Algorithm for Compressible Laminar Flows

Thomas L. Tysinger & David A. Caughey
Sibley School of Mechanical and Aerospace Engineering
Cornell University
Ithaca, New York

Abstract

An Alternating Direction Implicit diagonal multigrid algorithm is presented for the solution of the Navier-Stokes equations of viscous, compressible flow. Attention is focused on the inclusion of viscous contributions to the implicit factors in a way that will enhance the stability, yet not disturb the efficiency, of the diagonal algorithm. Flows past two-dimensional airfoils are computed to demonstrate the stability and efficiency of the scheme.

I. Introduction

In the numerical simulation of viscous flows at high Reynolds numbers, it is necessary to resolve the thin shear regions which develop near solid boundaries. These thin shear regions require the use of grids with cells of very high aspect ratio, which are known to hinder convergence for steady problems in explicit schemes. To overcome such difficulties, Caughey has developed an diagonal implicit algorithm for the solution of the Euler equations of inviscid, compressible flow [1]. Rapid convergence is achieved with the use of the implicit scheme within the multigrid framework.

Here, the method is extended to solve the Navier-Stokes equations, and specifically the thin-layer approximation to those equations. Aspects of the algorithm including artificial dissipation, boundary conditions, multigrid, and other topics not directly related to the implementation of the Navier-Stokes equations are not addressed here in great length; details of those issues can be found in [1]. Instead, emphasis is directed at methods of adding viscous contributions to the algorithm in a way which does not disturb the overall stability and efficiency of the implicit scheme. Since no attempt in the present analysis is made to incorporate a turbulence model, discussion will be limited to laminar flows.

II. Governing Equations

Navier-Stokes Equations

The equations which govern compressible viscous flows are the Navier-Stokes equations. In Cartesian coordinates, the Navier-Stokes equations in two-dimensions can be written

$$\frac{\partial \underline{w}}{\partial t} + \frac{\partial \underline{f}_C}{\partial x} + \frac{\partial \underline{g}_C}{\partial y} = \frac{\partial \underline{f}_V}{\partial x} + \frac{\partial \underline{g}_V}{\partial y}, \tag{1}$$

where $\underline{w} = \{\rho, \rho u, \rho v, e\}^T$ is the vector of conserved dependent variables. Here, ρ denotes the density, u and v the cartesian velocities, and ϵ the total energy per unit volume. The convective flux vectors in the x- and y- directions, respectively $\underline{f}_C$ and $\underline{g}_C$, and the viscous flux vectors $\underline{f}_V$ and $\underline{g}_V$ are given by

$$\begin{aligned}
\underline{f}_C &= \left\{\rho u, \rho u^2 + p, \rho uv, (e+p)u\right\}^T, \\
\underline{g}_C &= \left\{\rho v, \rho uv, \rho v^2 + p, (e+p)v\right\}^T, \\
\underline{f}_V &= \{0, \tau_{xx}, \tau_{xy}, (\bar{\bar{\tau}} \cdot \vec{u})_x - q_x\}^T. \\
\underline{g}_V &= \{0, \tau_{yx}, \tau_{yy}, (\bar{\bar{\tau}} \cdot \vec{u})_y - q_y\}^T.
\end{aligned}$$

The viscous shear stresses and the heat fluxes are of the form

$$\begin{aligned}
\tau_{xx} &= 2\mu u_x + \lambda(u_x + v_y), \\
\tau_{yy} &= 2\mu v_y + \lambda(u_x + v_y), \\
\tau_{xy} &= \mu(u_y + v_x), \\
q_x &= -kT_x, \\
q_y &= -kT_y,
\end{aligned}$$

where k is the coefficient of thermal conductivity and T is the temperature. The second coefficient of viscosity λ is related to the molecular viscosity μ by Stokes' hypothesis,

$$\lambda = -\frac{2}{3}\mu. \tag{2}$$

An equation of state is needed to relate the pressure and total energy:

$$p = (\gamma - 1)\left[e - \frac{1}{2}\rho(u^2 + v^2)\right]. \tag{3}$$

To allow treatment of arbitrary geometries, the equations are transformed into curvilinear coordinates and written

$$\frac{\partial \underline{W}}{\partial t} + \frac{\partial \underline{F_C}}{\partial \xi} + \frac{\partial \underline{G_C}}{\partial \eta} = \frac{\partial \underline{F_V}}{\partial \xi} + \frac{\partial \underline{G_V}}{\partial \eta}, \tag{4}$$

where $\underline{W} = h\underline{w}$ is the transformed dependent variable and

$$\underline{F_C}(\underline{W}) = \begin{bmatrix} \rho hU \\ \rho huU + y_\eta p \\ \rho hUv - x_\eta p \\ (e+p)hU \end{bmatrix}, \quad \underline{G_C}(\underline{W}) = \begin{bmatrix} \rho hV \\ \rho hVu - y_\xi p \\ \rho hVv + x_\xi p \\ (e+p)hV \end{bmatrix},$$

$$\underline{F_V}(\underline{W}, \underline{W_\xi}, \underline{W_\eta}) = \begin{bmatrix} 0 \\ y_\eta \tau_{xx} - x_\eta \tau_{xy} \\ y_\eta \tau_{xy} - x_\eta \tau_{yy} \\ y_\eta(u\tau_{xx} + v\tau_{xy} - q_x) - x_\eta(u\tau_{xy} + v\tau_{yy} - q_y) \end{bmatrix},$$

$$\underline{G_V}(\underline{W}, \underline{W_\xi}, \underline{W_\eta}) = \begin{bmatrix} 0 \\ -y_\xi \tau_{xx} + x_\xi \tau_{xy} \\ -y_\xi \tau_{xy} + x_\xi \tau_{yy} \\ -y_\xi(u\tau_{xx} + v\tau_{xy} - q_x) + x_\xi(u\tau_{xy} + v\tau_{yy} - q_y) \end{bmatrix},$$

are the transformed flux vectors. The contravariant velocities U and V are related to the cartesian velocities by

$$\begin{pmatrix} U \\ V \end{pmatrix} = \frac{1}{h}\begin{pmatrix} y_\eta u - x_\eta v \\ -y_\xi u + x_\xi v \end{pmatrix},$$

where $h = x_\xi y_\eta - x_\eta y_\xi$ is the determinant of the Jacobian of the transformation.

Thin-Layer Approximation

Under certain conditions it is possible to neglect viscous diffusion in the mainstream direction of the flow without adversely affecting the quality of the solution. The validity of such a simplification requires that the flow have a predominant direction, and is without massive separation. High Reynolds number flows over wings are one such example. Implementation of such a model necessitates that body surfaces be mapped onto coordinate surfaces, and there be sufficient clustering normal to the

shear surface to allow the boundary layer to be resolved. It can be argued that even if the full Navier-Stokes approximation is used, viscous diffusion in the streamwise direction cannot be resolved unless the grid is sufficiently fine in that direction [2], and for many practical flows, current computational limitations prevent the use of grids with sufficient resolution in both the normal and streamwise directions.

The transformed viscous flux vectors can be decoupled into components which depend only on the vector of dependent variables and its derivative in either the ξ- or η- direction:

$$\begin{aligned} \underline{F}_V &= \underline{F}_V(\underline{W}, \underline{W}_\xi, \underline{W}_\eta) = \tilde{\underline{F}}_V(\underline{W}, \underline{W}_\xi) + \hat{\underline{F}}_V(\underline{W}, \underline{W}_\eta), \\ \underline{G}_V &= \underline{G}_V(\underline{W}, \underline{W}_\xi, \underline{W}_\eta) = \tilde{\underline{G}}_V(\underline{W}, \underline{W}_\xi) + \hat{\underline{G}}_V(\underline{W}, \underline{W}_\eta). \end{aligned}$$

The thin-layer approximation entails retaining only the surface normal- or η- derivatives from the viscous terms in the Navier-Stokes equations (Eq. 4); that is only the term $\hat{\underline{G}}_V(\underline{W}, \underline{W}_\eta)$ is kept when the body surface is a line of η = constant. The thin-layer equations are then written

$$\frac{\partial \underline{W}}{\partial t} + \frac{\partial \underline{F}_C}{\partial \xi} + \frac{\partial \underline{G}_C}{\partial \eta} = \frac{\partial \hat{\underline{G}}_V}{\partial \eta}, \tag{5}$$

where

$$\hat{\underline{G}}_V(\underline{W}, \underline{W}_\eta) = \begin{bmatrix} 0 \\ -y_\xi \hat{\tau}_{xx} + x_\xi \hat{\tau}_{xy} \\ -y_\xi \hat{\tau}_{xy} + x_\xi \hat{\tau}_{yy} \\ -y_\xi (u\hat{\tau}_{xx} + v\hat{\tau}_{xy} - \hat{q}_x) + x_\xi (u\hat{\tau}_{xy} + v\hat{\tau}_{yy} - \hat{q}_y) \end{bmatrix},$$

and

$$\begin{aligned} \hat{\tau}_{xx} &= -\left(\frac{2\mu + \lambda}{h}\right) y_\xi u_\eta + \frac{\lambda}{h} x_\xi v_\eta, \\ \hat{\tau}_{yy} &= -\frac{\lambda}{h} y_\xi u_\eta + \left(\frac{2\mu + \lambda}{h}\right) x_\xi v_\eta, \\ \hat{\tau}_{xy} &= \frac{\mu}{h} (x_\xi u_\eta - y_\xi v_\eta), \\ \hat{q}_x &= \frac{k}{h} y_\xi T_\eta, \\ \hat{q}_y &= -\frac{k}{h} x_\xi T_\eta. \end{aligned}$$

III. Numerical Method

As with the algorithm for the Euler equations, spatial derivatives are approximated using a finite-volume formulation equivalent to a centered-difference approximation [1]. The approximation is second-order accurate when the mesh is smooth. Artificial dissipation consisting of an adaptive blend of second and fourth differences of the solution is added to insure convergence to steady state and to enable accurate shock capturing for transonic flows. Local time stepping is used to increase the convergence rate for steady problems. To further accelerate convergence, a recursive multigrid algorithm similar to that described by Smith and Caughey [3] is implemented.

The difference equations have the form

$$\frac{d}{dt}\underline{W}_{i,j} + C\underline{w}_{i,j} - V\underline{w}_{i,j} - D\underline{w}_{i,j} = 0, \tag{6}$$

where $C\underline{w}_{i,j}$ and $V\underline{w}_{i,j}$ represent contributions due to convection and viscous diffusion respectively, and $D\underline{w}_{i,j}$ represents the artificial dissipation defined in [1]. To simplify the expressions, contributions from the artificial dissipation will no longer be shown; it should be noted, however, that these terms play an important role in the overall algorithm, and a detailed description of the terms can be found in [1,4].

The first step in developing an ADI scheme is to approximate the spatial derivatives as weighted averages at new and old time levels. Such an approximation to the thin layer equation (Eq. 5) can be written

$$\begin{aligned} \Delta\underline{W}_{ij}^{n} \;+\; & \theta\Delta t\left\{\delta_\xi\left(\underline{F}_{C\,ij}^{n+1} - \underline{F}_{C\,ij}^{n}\right) + \delta_\eta\left(\underline{G}_{C\,ij}^{n+1} - \underline{G}_{C\,ij}^{n}\right)\right. \\ & \left. -\delta_\eta\left(\underline{\hat{G}}_{V\,ij}^{n+1} - \underline{\hat{G}}_{V\,ij}^{n}\right)\right\} \\ = \;& -\Delta t\left\{\delta_\xi\underline{F}_{C\,ij}^{n} + \delta_\eta\underline{G}_{C\,ij}^{n} - \delta_\eta\underline{\hat{G}}_{V\,ij}^{n}\right\}, \end{aligned} \tag{7}$$

where $\Delta\underline{W}_{ij}^{n} = \underline{W}_{ij}^{n+1} - \underline{W}_{ij}^{n}$ is the correction added to the solution, and θ represents the implicitness of the scheme with $0 \le \theta \le 1$.

The changes in the convective flux vectors can be linearized with a local Taylor series expansion in time to give

$$\underline{F}_{C\,ij}^{n+1} - \underline{F}_{C\,ij}^{n} = \mathbf{A}_{ij}^{n}\Delta\underline{W}_{ij}^{n} + O\left(\Delta t^2\right) \tag{8}$$

and

$$\underline{G}_{C\,ij}^{n+1} - \underline{G}_{C\,ij}^{n} = \mathbf{B}_{ij}^{n}\Delta\underline{W}_{ij}^{n} + O\left(\Delta t^2\right), \tag{9}$$

where $\mathbf{A} = \{\partial \underline{F}_C / \partial \underline{W}\}$ and $\mathbf{B} = \{\partial \underline{G}_C / \partial \underline{W}\}$ are the Jacobians of the transformed convective flux vectors with respect to the solution. Since the transformed viscous flux vector $\underline{\hat{G}}_V$ is a function of both $\underline{W}$ and $\underline{W}_\eta$, the appropriate linearization is

$$\underline{\hat{G}}_{V\,ij}^{\,n+1} - \underline{\hat{G}}_{V\,ij}^{\,n} = \hat{\mathbf{M}}_{ij}^n \Delta \underline{W}_{ij}^n + \hat{\mathbf{N}}_{ij}^n \Delta \underline{W}_{\eta ij}^n + O\left(\Delta t^2\right) \tag{10}$$

$$= (\hat{\mathbf{M}}_{ij} - \hat{\mathbf{N}}_{\eta ij})^n \Delta \underline{W}_{ij}^n + \frac{\partial}{\partial \eta}\left(\hat{\mathbf{N}}_{ij}^n \Delta \underline{W}_{ij}^n\right) + O\left(\Delta t^2\right), \tag{11}$$

where $\hat{\mathbf{M}} = \{\partial \underline{\hat{G}}_V / \partial \underline{W}\}$ is the Jacobian of the transformed viscous flux vector with respect to the solution and $\hat{\mathbf{N}} = \{\partial \underline{\hat{G}}_V / \partial \underline{W}_\eta\}$ is the Jacobian with respect to the derivative of the solution. Recognizing that $\hat{\mathbf{M}} - \hat{\mathbf{N}}_\eta \equiv \mathbf{0}$ if the the transport coefficients are approximated to be locally constant [5], the linearization reduces to

$$\underline{\hat{G}}_{V\,ij}^{\,n+1} - \underline{\hat{G}}_{V\,ij}^{\,n} = \frac{\partial}{\partial \eta}\left(\hat{\mathbf{N}}_{ij}^n \Delta \underline{W}_{ij}^n\right) + O\left(\Delta t^2\right). \tag{12}$$

The viscous flux Jacobian is

$$\hat{\mathbf{N}} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ n_{21} & n_{22} & n_{23} & 0 \\ n_{31} & n_{32} & n_{33} & 0 \\ n_{41} & n_{42} & n_{43} & n_{44} \end{pmatrix} h^{-1} \tag{13}$$

where

$$\begin{aligned}
n_{21} &= -\alpha_1 \left(\frac{u}{\rho}\right) - \alpha_2 \left(\frac{v}{\rho}\right), \\
n_{22} &= \alpha_1 \left(\frac{1}{\rho}\right), \\
n_{23} &= \alpha_2 \left(\frac{1}{\rho}\right), \\
n_{31} &= -\alpha_2 \left(\frac{u}{\rho}\right) - \alpha_3 \left(\frac{v}{\rho}\right), \\
n_{32} &= n_{23}, \\
n_{33} &= \alpha_3 \left(\frac{1}{\rho}\right), \\
n_{41} &= -\alpha_1 \left(\frac{u^2}{\rho}\right) - 2\alpha_2 \left(\frac{uv}{\rho}\right) - \alpha_3 \left(\frac{v^2}{\rho}\right) + \alpha_4 \left(-\frac{e}{\rho^2} + \frac{u^2+v^2}{\rho}\right), \\
n_{42} &= -\alpha_4 \left(\frac{u}{\rho}\right) - n_{21}, \\
n_{43} &= -\alpha_4 \left(\frac{v}{\rho}\right) - n_{31},
\end{aligned}$$

$$n_{44} = \alpha_4\left(\frac{1}{\rho}\right),$$

with

$$\alpha_1 = \frac{2\mu+\lambda}{h}{y_\xi}^2 + \frac{\mu}{h}{x_\xi}^2, \quad \alpha_2 = -\frac{\lambda+\mu}{h}x_\xi y_\xi,$$
$$\alpha_3 = \frac{\mu}{h}{y_\xi}^2 + \frac{2\mu+\lambda}{h}{x_\xi}^2, \quad \alpha_4 = \frac{\gamma\mu}{h}{P_r}^{-1}({x_\xi}^2 + {y_\xi}^2).$$

The matrix $\hat{\mathbf{N}}$ is not to be confused with the viscous flux Jacobian described by Steger [6] in which the elements of the matrix are differential operators. Introducing the approximations of Eqs. 8, 9, and 12 into Eq. 7 results in what is commonly called the "delta" form of the algorithm:

$$\begin{aligned} &\{\mathbf{I} + \theta\Delta t(\mathbf{A}_{ij}\delta_\xi + \mathbf{B}_{ij}\delta_\eta - \hat{\mathbf{N}}_{ij}\delta_{\eta\eta})\}^n \Delta\underline{W}^n_{ij} \\ = \; &-\Delta t\{\delta_\xi \underline{F_C}^n_{ij} + \delta_\eta \underline{G_C}^n_{ij} - \delta_\eta \underline{\hat{G}_V}^n_{ij}\}. \end{aligned} \tag{14}$$

Approximating the left hand side of Eq. 14 as the product two one-dimensional factors results in a block ADI scheme and is written

$$\begin{aligned} &\{\mathbf{I} + \theta\Delta t\mathbf{A}_{ij}\delta_\xi\} \times \{\mathbf{I} + \theta\Delta t(\mathbf{B}_{ij}\delta_\eta - \hat{\mathbf{N}}_{ij}\delta_{\eta\eta})\}^n \Delta\underline{W}^n_{ij} \\ = \; &-\Delta t\{\delta_\xi \underline{F_C}^n_{ij} + \delta_\eta \underline{G_C}^n_{ij} - \delta_\eta \underline{\hat{G}_V}^n_{ij}\}. \end{aligned} \tag{15}$$

With the addition of fourth order artificial dissipation, block pentadiagonal systems must be solved in each factor of Eq. 15. For the Euler equations, the convective flux Jacobians can be diagonalized with local similarity transformations as $\mathbf{A} = \mathbf{Q}_A\mathbf{\Lambda}_A\mathbf{Q}_A^{-1}$ and $\mathbf{B} = \mathbf{Q}_B\mathbf{\Lambda}_B\mathbf{Q}_B^{-1}$, where $\mathbf{\Lambda}_A$ and $\mathbf{\Lambda}_B$ are diagonal matrices whose diagonal elements are the eigenvalues of their respective Jacobians, and $\mathbf{Q}_A$ and $\mathbf{Q}_B$ are the modal matrices whose elements are given in [7]. This allows the block equations to be decoupled into equations which can be solved as scalar pentadiagonal systems, greatly reducing the amount of computational labor needed for a solution.

For the Navier-Stokes equations however, it is not possible to both include the viscous terms in the implicit factor and to diagonalize the system, since the convective and viscous Jacobians are not simultaneously diagonalizable. If viscous contributions are neglected completely from implicit consideration, a diagonalized system can be written

$$\begin{aligned} &\{\mathbf{I} + \theta\Delta t\mathbf{\Lambda}_{Aij}\delta_\xi\}\mathbf{Q}^{-1}_{Aij} \\ \times \; &\mathbf{Q}_{Bij}\{\mathbf{I} + \theta\Delta t\mathbf{\Lambda}_{Bij}\delta_\eta\}\mathbf{Q}^{-1}_{Bij}\Delta\underline{W}^n_{ij} \\ = \; &-\Delta t\mathbf{Q}^{-1}_{Aij}\{\delta_\xi \underline{F_C}^n_{ij} + \delta_\eta \underline{G_C}^n_{ij} - \delta_\eta \underline{\hat{G}_V}^n_{ij}\}. \end{aligned} \tag{16}$$

Neglecting the viscous terms completely from the implicit factors would jeopardize the stability of the scheme. It is desirable to maintain the efficiency of the diagonalized scheme without degrading its stability properties, so alternate approaches must be explored.

Method I

The first method consists of using the largest eigenvalue of the viscous Jacobian to add contributions to the existing implicit factors. This is similar to what was suggested by Pulliam [8]. The eigenvalues of $\hat{\mathbf{N}}$ are

$$\begin{aligned}
\lambda_1 &= (2\mu+\lambda)\left(\frac{x_\xi{}^2+y_\xi{}^2}{h}\right)\left(\frac{1}{h\rho}\right),\\
\lambda_2 &= \gamma\mu P_r{}^{-1}\left(\frac{x_\xi{}^2+y_\xi{}^2}{h}\right)\left(\frac{1}{h\rho}\right),\\
\lambda_3 &= \mu\left(\frac{x_\xi{}^2+y_\xi{}^2}{h}\right)\left(\frac{1}{h\rho}\right),\\
\lambda_4 &= 0,
\end{aligned} \tag{17}$$

where P_r is the Prandtl number. A scheme is constructed by adding the diagonal approximation $\hat{\mathbf{\Lambda}}_{\hat{N}} \approx \mathbf{Q}_B^{-1}\hat{\mathbf{N}}\mathbf{Q}_B$ to the appropriate implicit factor:

$$\begin{aligned}
&\{\mathbf{I}+\theta\Delta t\mathbf{\Lambda}_{Aij}\delta_\xi\}\mathbf{Q}_{Aij}^{-1}\\
\times\ &\mathbf{Q}_{Bij}\{\mathbf{I}+\theta\Delta t(\mathbf{\Lambda}_{Bij}\delta_\eta-\hat{\mathbf{\Lambda}}_{\hat{N}ij}\delta_{\eta\eta})\}\mathbf{Q}_{Bij}^{-1}\Delta\underline{W}_{ij}^n\\
=\ &-\Delta t\mathbf{Q}_{Aij}^{-1}\{\delta_\xi\underline{F_C}_{ij}^n+\delta_\eta\underline{G_C}_{ij}^n-\delta_\eta\underline{\hat{G}_V}_{ij}^n\}.
\end{aligned} \tag{18}$$

The diagonal approximation $\hat{\mathbf{\Lambda}}_{\hat{N}}$ is for example

$$\hat{\mathbf{\Lambda}}_{\hat{N}} = \lambda_2\mathbf{I} = \gamma\mu P_r{}^{-1}\left(\frac{x_\xi{}^2+y_\xi{}^2}{h}\right)\left(\frac{1}{h\rho}\right)\mathbf{I}. \tag{19}$$

The number of additional operations needed to implement this scheme is negligible since it involves only the calculation of an eigenvalue whose analytical form is known.

Method II

Another option is to use an additional implicit operator which contains the exclusive contributions from the viscous terms. Since the eigenvalues of the viscous Jacobian are distinct (Eqs. 17), a modal matrix $\mathbf{Q}_{\hat{N}}$ exists which diagonalizes $\hat{\mathbf{N}}$ through a similarity transformation. This results in the scheme

$$\begin{aligned} & \{\mathbf{I}+\theta\Delta t\mathbf{\Lambda}_{Aij}\delta_\xi\}\mathbf{Q}_{Aij}^{-1} \\ \times\ & \mathbf{Q}_{Bij}\{\mathbf{I}+\theta\Delta t\mathbf{\Lambda}_{Bij}\delta_\eta\}\mathbf{Q}_{Bij}^{-1} \\ \times\ & \mathbf{Q}_{\hat{N}ij}\{\mathbf{I}-\theta\Delta t\mathbf{\Lambda}_{\hat{N}ij}\delta_{\eta\eta}\}\mathbf{Q}_{\hat{N}ij}^{-1}\Delta\underline{W}_{ij}^{n} \\ =\ & -\Delta t\mathbf{Q}_{Aij}^{-1}\{\delta_\xi\underline{F_C}_{ij}^{n}+\delta_\eta\underline{G_C}_{ij}^{n}-\delta_\eta\underline{\hat{G}_V}_{ij}^{n}\}. \end{aligned} \tag{20}$$

The modal matrix and its inverse are written

$$\mathbf{Q}_{\hat{N}}=\begin{bmatrix} 0 & 0 & 0 & 1 \\ \tilde{\kappa}_x & 0 & \tilde{\kappa}_y & u \\ \tilde{\kappa}_y & 0 & -\tilde{\kappa}_x & v \\ \tilde{\theta} & 1 & -\tilde{\mu} & \frac{e}{\rho} \end{bmatrix} \tag{21}$$

$$\mathbf{Q}_{\hat{N}}^{-1}=\begin{bmatrix} -\frac{\tilde{\theta}}{\tilde{\kappa}_x^2+\tilde{\kappa}_y^2} & \frac{\tilde{\kappa}_x}{\tilde{\kappa}_x^2+\tilde{\kappa}_y^2} & \frac{\tilde{\kappa}_y}{\tilde{\kappa}_x^2+\tilde{\kappa}_y^2} & 0 \\ -\frac{e}{\rho}+u^2+v^2 & -u & -v & 1 \\ \frac{\tilde{\mu}}{\tilde{\kappa}_x^2+\tilde{\kappa}_y^2} & \frac{\tilde{\kappa}_y}{\tilde{\kappa}_x^2+\tilde{\kappa}_y^2} & -\frac{\tilde{\kappa}_x}{\tilde{\kappa}_x^2+\tilde{\kappa}_y^2} & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \tag{22}$$

where $\kappa = \xi$ or η, and

$$\tilde{\kappa}_x=\frac{\kappa_x}{\sqrt{\kappa_x^2+\kappa_y^2}},\quad \tilde{\kappa}_y=\frac{\kappa_y}{\sqrt{\kappa_x^2+\kappa_y^2}}$$

$$\tilde{\theta}=\tilde{\kappa}_x u+\tilde{\kappa}_y v,\quad \tilde{\mu}=\tilde{\kappa}_x v-\tilde{\kappa}_y u.$$

Although the above methods are developed for the thin-layer approximation. the schemes can be readily modified to accommodate the full Navier-Stokes approximation. Analogous terms must be added to account for the viscous contributions in the ξ-direction. Also, the stability properties of these methods still remain an issue and must be further explored.

IV. Stability Analysis

The stability properties of these schemes are examined using von Neumann (Fourier) analysis on a scalar model equation. The model equation for the thin-layer approximation contains fourth order artificial dissipation and is written

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} + c\frac{\partial u}{\partial y} + c\epsilon\left(\Delta x^3 \frac{\partial^4 u}{\partial x^4} + \Delta y^3 \frac{\partial^4 u}{\partial y^4}\right) = \nu \frac{\partial^2 u}{\partial y^2}. \tag{23}$$

Substitution of the Fourier term $u_{ij}^n = G^n e^{\imath\beta_x x} e^{\imath\beta_y y}$, into the model and writing it as the product of one-dimensional operators leads to

$$\begin{aligned}
(G-1) \quad & \{1 + \imath\theta_I \lambda_x \sin\xi + 16\theta_D \lambda_x \epsilon \sin^4\frac{\xi}{2}\} \\
\times \quad & \{1 + \imath\theta_I \lambda_x A_r^{-1} \sin\eta + 16\theta_D \lambda_x A_r^{-1} \epsilon \sin^4\frac{\eta}{2} + 4\theta_V \lambda_x Re_x^{-1} A_r^{-2} \sin^2\frac{\eta}{2}\} \\
= \quad & -\lambda_x\{\imath(\sin\xi + A_r^{-1}\sin\eta) + 16\epsilon(\sin^4\frac{\xi}{2} + A_r^{-1}\sin^4\frac{\eta}{2}) \\
& + 4Re_x^{-1} A_r^{-2} \sin^2\frac{\eta}{2}\}.
\end{aligned} \tag{24}$$

From Eq. 24, the magnitude of G can be calculated,

$$|G| = f(\xi, \eta; \lambda_x, Re_x, A_r, \epsilon, \theta_I, \theta_D, \theta_V),$$

where ξ and η represent the mesh wave numbers. In addition to the Courant number $\lambda_x = c\Delta t/\Delta x$ and artificial dissipation ϵ, the numerical stability of the implicit viscous equations is governed primarily by the aspect ratios (A_r) of the mesh cells and the mesh Reynolds numbers (Re_x). The expression in Eq. 24 corresponds to what is done in Method I. Similar expressions representative of Method II can also be derived.

Using such a model, it is found that when viscous terms are added directly to the convective operators, analogous to what is done with Method I, unconditional stability is achieved. If the viscous terms are evaluated explicitly without implicit contributions, a conditionally stable scheme results. This can be seen in Figure 1 in which the dark areas represent regions in parameter space where von Neumann analysis predicts an amplification factor greater than unity. This figure represents the properties of the scheme applied to a model problem using values of the dissipation parameters characteristic of those used in the computations, and a Courant number of 16. The possibility, however, of obtaining a converged solution without including viscous contributions in the implicit factors cannot be ruled out. If additional viscous operators are added to the scheme (as is done in Method II), the solution will remain conditionally stable, although the region of stability is increased slightly

as shown in Figure 2. The stability analysis indicates that the most promising algorithm would be one similar to Method I. Method II should also be considered, however, in so far as it represents less of an approximation than Method I.

V. Results

Both methods are implemented in a computer code to calculate transonic flows in two-dimensions. A number of test cases have been computed for flows past a two-dimensional NACA 0012 symmetric airfoil. The case presented here is for subcritical laminar flow ($Re = 5000$, $M_\infty = 0.5$) past a two-dimensional NACA 0012 symmetric airfoil at zero degrees angle of attack. The calculation is performed using the thin-layer approximation on a 192×48 cell "C"-grid generated using the GRAPE code elliptic mesh generator [9]. The outer boundary of the mesh is located about 8 chords from the body. Care is taken to insure sufficient clustering in the region close to the body surface where viscous effects are significant. Approximately 10 mesh points are included within the boundary layer at the airfoil trailing edge, and the first point normal to the body surface is located at about .001 chords.

The surface pressure distribution, presented in Figure 3, agrees well with that presented by Martinelli, Jameson, and Grasso [10]. The flow separates at approximately 85% of the chord, as can be seen from the contour plot of the streamwise component of mass-flux density in Figure 4; this value is close to the values reported by both Swanson and Turkel [11] and Jayaram and Jameson [12] for this case.

The iterative process is begun by initializing the solution to free stream values. A plot of the convergence history is shown in Figure 5. Using five levels of multigrid and local time stepping, the solution converged to a steady state in approximately 35 work units which corresponds to 20 multigrid cycles. One work unit is defined as the amount of work required to advance the solution one time step on the finest mesh level; for the strategy used here, each multigrid cycle requires approximately 1 2/3 work units. Overall, the average residual is reduced by 6 orders of magnitude in about 300 work units or 180 multigrid cycles. This represents a significant improvement over rates reported by other researchers [10.11].

This solution is computed at a Courant number of 16 using Method I. At this Courant number, both methods I and II produce converged solutions as illustrated in Figures 5 and 6. However, a converged solution is not attainable if viscous terms are neglected from the implicit factors as is evident in Figure 7. This demonstrates the importance of maintaining an implicit viscous contribution to the numerical scheme. Converged solutions from a completely explicit viscous scheme have also been obtained, but at the expense of a lower Courant number, hence, a lower rate of convergence.

The importance of including viscous contributions in the implicit operator has been

demonstrated. Although several options are available for implicit inclusion, the addition of approximate terms to the existing operators seems the most effective. The laminar solutions obtained are in good agreement with results reported by other researchers [12,10,11], while significant improvements in rates of convergences are achieved. Work is underway to extend these methods to three dimensions, and incorporate a turbulence model so that engineering flows can be studied.

Acknowledgments

This research has been supported in part by the Independent Research and Development Program of the McDonnell Douglas Corporation and by the U. S. Army Research Office through the Mathematical Sciences Institute of Cornell University. The calculations reported here were performed at the Cornell National Supercomputer Facility, a resource of the Cornell Theory Center, which receives major funding from the National Science Foundation and the IBM Corporation, with additional support from New York State, and the Corporate Research Institute.

References

[1] Caughey, David A. Diagonal Implicit Multigrid Algorithm for the Euler Equations. *AIAA Journal*, **26**(7):841–851, July 1988.

[2] Degani, David and Steger, Joseph L. Comparison Between Navier-Stokes and Thin-Layer Computations for Separated Supersonic Flow. *AIAA Journal*, **21**(11):1604–1605, November 1983.

[3] Smith, W. A. and Caughey, D. A. Multigrid Solution of Inviscid Transonic Flow Through Rotating Blade Passages. AIAA Paper 87-0608, AIAA 25th Aerospace Sciences Meeting, Reno, Nevada, January 12-15, 1987.

[4] Caughey, D. A. and Turkel, E. Effects of Numerical Dissipation on Finite-Volume Solutions of Compressible Flow Problems. AIAA Paper 88-0621, AIAA 26th Aerospace Sciences Meeting, Reno, Nevada, January 11-14, 1988.

[5] Beam, R. M. and Warming, R. F. An Implicit Factored Scheme for the Compressible Navier-Stokes Equations. *AIAA Journal*, **16**(4):393–402, April 1978.

[6] Steger, Joseph L. Implicit Finite-Difference Simulation of Flow about Arbitrary Two-Dimensional Geometries. *AIAA Journal*, **16**(7):679–686, July 1978.

[7] Chaussee, D. S. and Pulliam, T. H. Two-Dimensional Inlet Simulation Using a Diagonal Implicit Algorithm. *AIAA Journal*, **19**(2):153–159, February 1981.

[8] Pulliam, T. H. Efficient Solution Methods for the Navier-Stokes Equations. Lecture Notes for The Von Karman Institute for Fluid Dynamics Lecture Series, Numerical Techniques for Viscous Flow Computation in Turbomachinery Bladings, Brussels, Belgium, January 20-24, 1986.

[9] Sorenson, Reese L. A Computer Program to Generate Grids about Two- Dimensional Airfoils and Other Shapes by the Use of Poisson's Equation. NASA TM-81198, (1980).

[10] Martinelli, L., Jameson, A., and Grasso, F. A Multigrid Method for the navier-Stokes Equations. AIAA Paper 86-0208, AIAA 24th Aerospace Sciences Meeting, Reno, Nevada, January 6-9, 1986.

[11] Swanson, R. C. and Turkel, E. A Multistage Time-Stepping Scheme for the Navier-Stokes Equations. AIAA Paper 85-0035, AIAA 23rd Aerospace Sciences Meeting, Reno, Nevada, January 14-17, 1985.

[12] Jayaram, Mohan and Jameson, Anthony. Multigrid Solution of the Navier-Stokes Equations for Flow Over Wings. AIAA Paper 88-0705, AIAA 26th Aerospace Sciences Meeting, Reno, Nevada, January 11-14, 1988.

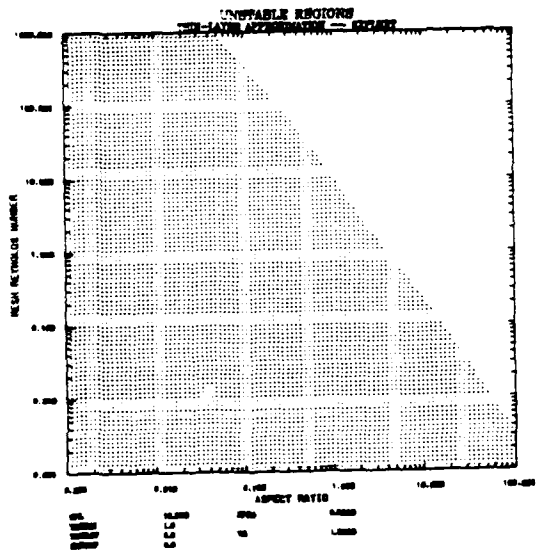


Figure 1: Explicit Viscous Treatment Only

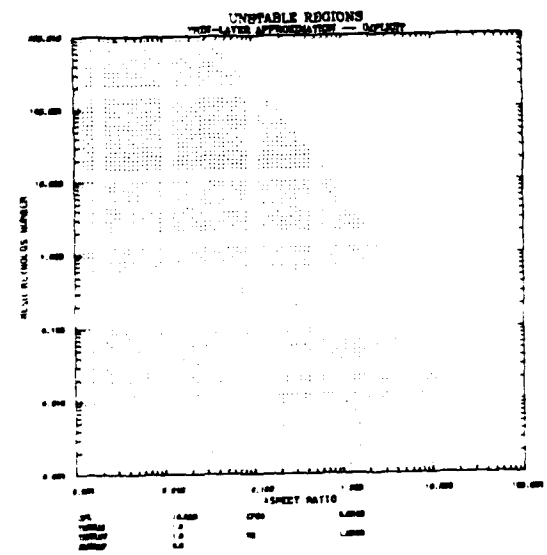


Figure 2: Additional Viscous Operator

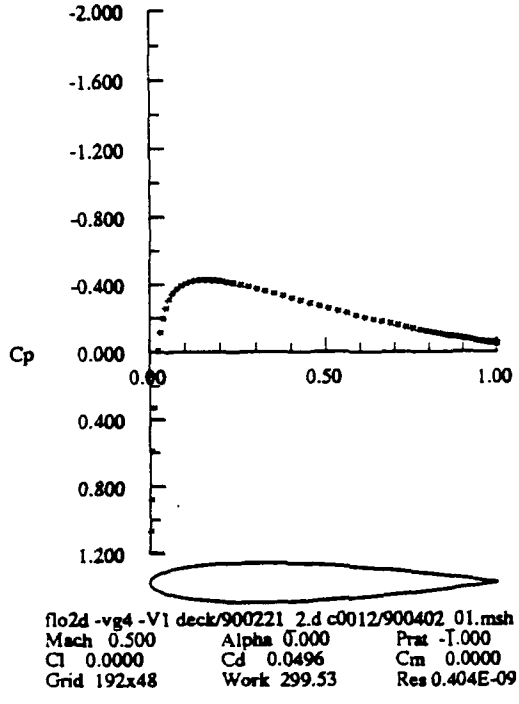


Figure 3: Surface pressure distribution: $Re = 5000.0$, $M_\infty = 0.50$, $\alpha = 0.0$

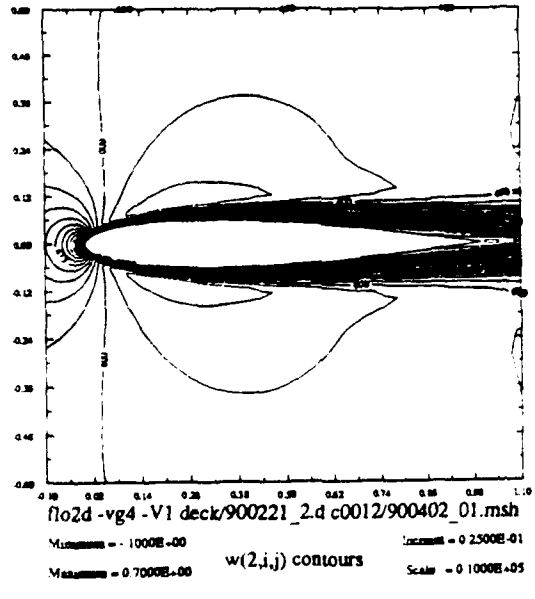
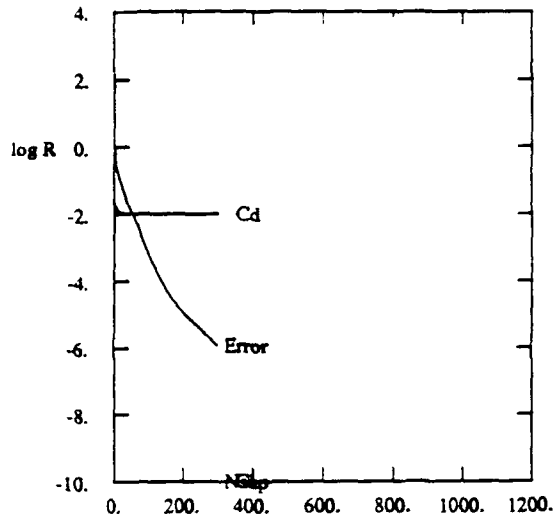


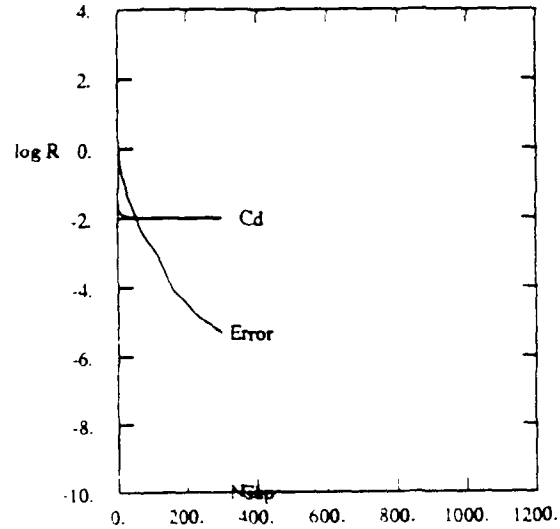
Figure 4: Streamwise mass-flux density: $Re = 5000.0$, $M_\infty = 0.50$, $\alpha = 0.0$



flo2d -vg4 -V1 deck/900221_2.d c0012/900402_01.msh

Mach 0.500	Alpha 0.000	Prat -1.000
Res1 0.350E-03	Re 5000.	CFL 16.00
Res2 0.404E-09	Fbc -1.	Grid 192x48
Work 299.53	Rate 0.9554	Nmesh 5

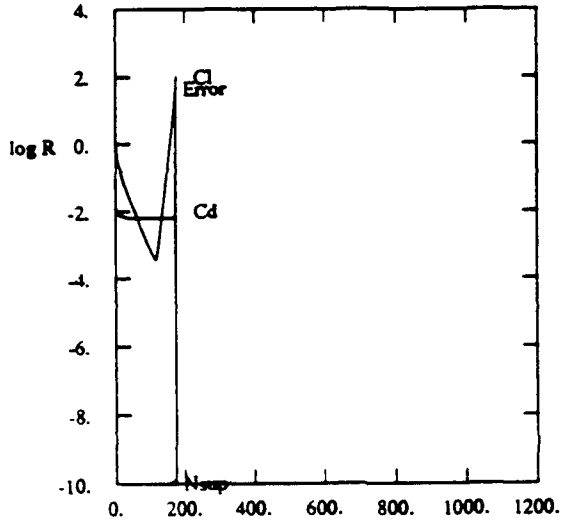
Figure 5: Residual convergence history – Method I:
$Re = 5000.0,\ M_\infty = 0.50,\ \alpha = 0.0$



flo2d -vg4 -V2 deck/900221_2.d c0012/900402_01.msh

Mach 0.500	Alpha 0.000	Prat -1.000
Res1 0.371E-03	Re 5000.	CFL 16.00
Res2 0.187E-08	Fbc -1.	Grid 192x48
Work 299.53	Rate 0.9601	Nmesh 5

Figure 6: Residual convergence history – Method II:
$Re = 5000.0,\ M_\infty = 0.50,\ \alpha = 0.0$



flo2d -vg4 -V0 expt.d c0012/900402_01.msh

Mach 0.500	Alpha 0.000	Pret -1.000
Res1 0.369E-03	Re 5000.	CFL 16.00
Res2 0.167E-01	Fbc -1.	Grid 192x48
Work 174.73	Rate 1.0220	Nmesh 5

Figure 7: Residual convergence history – Explicit Viscous:
$Re = 5000.0$, $M_\infty = 0.50$, $\alpha = 0.0$

Extremum Control:

The Effects of Artificial Viscosity

Culbert B. Laney, *Center for Applied Mathematics*
David A. Caughey, *Sibley School of Mechanical and Aerospace Engineering*
Cornell University, Ithaca, NY 14850

This paper concerns numerical approximation to discontinuous solutions of conservation laws. First order spatially accurate methods routinely capture discontinuities smoothly. The best capture grid-aligned steady shocks with only one transition point. Unfortunately, formally higher-order accurate methods include terms which may become large near discontinuities, resulting in spurious oscillations and overshoots. The goal is to design higher-order methods with the shock capturing abilities of first order methods. The 1980s saw the introduction of several successful approaches [1,2,3,4]. We propose to evaluate such numerical methods in terms of their effect on the growth and creation of extrema. The model problem is a one-dimensional scalar semidiscrete approximation. The results for semidiscrete approximations apply immediately to steady state solutions, since in this case time discretization affects only convergence rate. We briefly consider extensions to multidimensional problems.

1 Basic Theory

Consider the following scalar one-dimensional initial value problem on an unbounded domain:

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0 \tag{1}$$

$$u(x,0) = \phi(x), \quad \frac{\partial f(u)}{\partial u} = a(u).$$

The solution u is constant along straight-line characteristics given by $\partial u/\partial t = a(u)$. If two characteristics intersect then no continuous solutions exist—'weak' solutions containing jump discontinuities must be allowed. Weak solutions to equation (1) have the following interesting properties [2,5]:

E1 Maxima do not increase in time. Minima do not decrease.

E2 No new extrema are created.

Consider the semidiscrete finite-difference approximation to equation (1):

$$\frac{du_k(t)}{dt} = \frac{1}{\Delta x_k} H_k(t) \tag{2}$$

$$H_k(t) \equiv H[u_{k+K}(t), ..., u_{k-K}(t)] \tag{3}$$

where the difference stencil is $2K + 1$ points wide, $K \geq 1$; the grid spacing is $\Delta x_k = x_{k+1/2} - x_{k-1/2}$; the cell boundaries occur at $x_{k+1/2}$; and $u_k(t)$ approximates $u(x_k, t)$. When will $u_k(t)$ inherit properties E1 and E2 of the exact solution?

Theorem 1: The solution to equation (2) has property E1 if

$$H_k(t) \leq 0 \text{ for } u_k(t) \text{ maximum}$$

$$H_k(t) \geq 0 \text{ for } u_k(t) \text{ minimum}$$

Theorem 2: Property E1 implies property E2.

The conditions of Theorem 1 are necessary and sufficient if du_k/dt exists. Otherwise, the conditions hold in the limit from the right and/or left. To prove Theorem 2, note that continuous extrema must start out infinitely small, at which point E1 acts to prevent further growth. Otherwise, replace discontinuous extrema by continuous extrema with slope approaching infinity.

Enforcing E1 leads to 'clipping' error at moving continuous extrema. For example, suppose the exact solution contains a maximum M moving to the left. In general, the maximum falls somewhere between two grid points j and $j + 1$. At some time, j should become a maximum on the grid, with a value less than M. E1 prevents further increase at j, and the top of the maximum is 'clipped' off as time progresses. This process limits accuracy at unsteady extrema to roughly second order, but steeply peaked extrema are obviously more affected than flattish extrema. As a matter of fact, many methods uniformly reduce extrema, which affects steady as well as unsteady extrema e. g. [2]. Because of these problems, one might reasonably wish to enforce only condition E2. While this is possible for fully-discrete approximations, it is as yet unclear how this could be accomplished in semidiscrete cases.

Consider the following three forms for $H_k(t)$:

Conservation Form

$$H_k = -(h_{k+1/2} - h_{k-1/2})$$

where $h_{k+1/2} = h(u_{k-K+1}, ..., u_{k+K})$ is a Lipschitz continuous function consistent with $f(u)$ in the sense that $h(u, u, ..., u) = f(u)$.

Viscosity Form

$$H_k = -\frac{1}{2}[f(u_{k+1}) - f(u_{k-1}) - q_{k+1/2}\Delta_{k+1/2}u + q_{k-1/2}\Delta_{k-1/2}u]$$

where $q_{k+1/2} = q(u_{k-K+1}, ..., u_{k+K})$ is Lipschitz continuous and $\Delta_{k+1/2}u \equiv u_{k+1} - u_k$. Viscosity form is related to conservation form by:

$$h_{k+1/2} = \frac{1}{2}[f(u_{k+1}) + f(u_k) - q_{k+1/2}\Delta_{k+1/2}u]$$

Incremental Form

$$H_k = C^+_{k+1/2}\Delta_{k+1/2}u - C^-_{k-1/2}\Delta_{k-1/2}u$$

where $C^{\pm}_{k+1/2} = C^{\pm}(u_{k-K+1}, ..., u_{k+K})$ are Lipschitz continuous function such that the following conservation condition holds:

$$C^-_{k+1/2} - C^+_{k+1/2} = a_{k+1/2} \tag{4}$$

$$a_{k+1/2} \equiv \begin{cases} \frac{f(u_{k+1})-f(u_k)}{u_{k+1}-u_k} & \text{if } u_{k+1} \neq u_k \\ a(u_k) & \text{otherwise} \end{cases}$$

Incremental form relates to conservation form via:

$$h_{k+1/2} = -C^+_{k+1/2}\Delta_{k+1/2}u + f(u_k) = -C^-_{k+1/2}\Delta_{k+1/2}u + f(u_{k+1})$$

Incremental form relates to artificial viscosity form via:

$$q_{k+1/2} = C^+_{k+1/2} + C^-_{k+1/2} \tag{5}$$

$$C^+_{k+1/2} = \frac{1}{2}(q_{k+1/2} - a_{k+1/2}) \tag{6}$$

$$C^-_{k+1/2} = \frac{1}{2}(q_{k+1/2} + a_{k+1/2}) \tag{7}$$

Examples of the incremental and viscosity forms include:
Central Differences $H_k = -(f(u_{k+1}) - f(u_{k-1}))/2$

$$C^+_{k+1/2} = -\frac{1}{2}a_{k+1/2}, \quad C^-_{k+1/2} = \frac{1}{2}a_{k+1/2}, \quad q_{k+1/2} = 0$$

Roe's Method (First order upwind) [6]
(The scalar version is also known as the Cole-Murman method [7].)

$$C^+_{k+1/2} = \max(0, -a_{k+1/2}), \quad C^-_{k+1/2} = \max(0, a_{k+1/2}), \quad q_{k+1/2} = |a_{k+1/2}|$$

Consider the following simple corollary to Theorem 1:
Corollary 3: The incremental form has property E1 if

$$C^+_{k+1/2}\Delta_{k+1/2}u \leq C^-_{k-1/2}\Delta_{k-1/2}u \text{ for } u_k \text{ max} \quad (\Delta_{k+1/2}u \leq 0 \;\&\; \Delta_{k-1/2}u \geq 0)$$

$$C^+_{k+1/2}\Delta_{k+1/2}u \geq C^-_{k-1/2}\Delta_{k-1/2}u \text{ for } u_k \text{ min} \quad (\Delta_{k+1/2}u \geq 0 \;\&\; \Delta_{k-1/2}u \leq 0)$$

This implies that increasing $C^{\pm}$ tends to enforce E1; in particular, E1 holds if $C^{\pm} \geq 0$. Harten [2] introduced this popular 'positivity' condition; when true, incremental form becomes a scalar version of flux vector splitting [8]. Clearly, Roe's method is uniformly positive while the central difference method is not. We examine the positivity condition further in [9].

2 Second Order Artificial Viscosity

We may enforce Corollary 3 by adding second order artificial viscosity, thereby increasing C^+ and C^- at extrema:

$$H_k = (C^+_{k+1/2} + \frac{1}{2}\epsilon_{k+1/2})\Delta_{k+1/2}u - (C^-_{k-1/2} + \frac{1}{2}\epsilon_{k-1/2})\Delta_{k-1/2}u$$

where $\epsilon_{k+1/2} \geq 0$ is the coefficient of second order artificial viscosity and $C^\pm$ belongs to some higher-order method. Suppose the higher-order method violates Corollary 3 at point k. Choose $\epsilon_{k\pm1/2} \geq 0$ so that:

$$(C^+_{k+1/2} + \frac{1}{2}\epsilon_{k+1/2})\Delta_{k+1/2}u = (C^-_{k-1/2} + \frac{1}{2}\epsilon_{k-1/2})\Delta_{k-1/2}u. \qquad (8)$$

Corollary 3 now holds with the least possible deviation from the higher-order method. If $k \pm 1$ is also an extremum, set $\epsilon_{k\pm1/2}$ large to damp the $2\Delta x$ component. If neither $k+1$ nor $k-1$ is an extremum, equation (8) has one degree of freedom. Consider the following:

- If $C^+_{k+1/2} < 0$ and $C^-_{k-1/2} \geq 0$ then:

$$\epsilon_{k+1/2} = 2\left(-C^+_{k+1/2} + C^-_{k-1/2}\frac{\Delta_{k-1/2}u}{\Delta_{k+1/2}u}\right), \quad \epsilon_{k-1/2} = 0$$

- If $C^+_{k+1/2} \geq 0$ and $C^-_{k-1/2} < 0$ then:

$$\epsilon_{k-1/2} = 2\left(-C^-_{k-1/2} + C^+_{k+1/2}\frac{\Delta_{k+1/2}u}{\Delta_{k-1/2}u}\right), \quad \epsilon_{k+1/2} = 0$$

Various strategies may be employed when both coefficients are negative, which occurs only near sonic points (where $a(u) = 0$). We do not wish to address the, as yet, undeveloped art of sonic point capturing here.

Increasing the negative incremental coefficient corresponds to reducing the downwind contribution. In the most extreme case, such as near shocks the method becomes fully upwind and first order. For example, if the basis higher-order method is central differences:

$$a_{k+1/2} \geq 0, a_{k-1/2} \geq 0: \quad \epsilon_{k+1/2} = a_{k+1/2} + a_{k-1/2}\frac{\Delta_{k-1/2}u}{\Delta_{k+1/2}u} \leq a_{k+1/2}$$

$$a_{k+1/2} \leq 0, a_{k-1/2} \leq 0: \quad \epsilon_{k-1/2} = -a_{k-1/2} - a_{k+1/2}\frac{\Delta_{k+1/2}u}{\Delta_{k-1/2}u} \leq a_{k+1/2}$$

(If these give $\epsilon \leq 0$, Corollary 3 is not violated and no viscosity is required.) If $a_{k\pm1/2}$ are non-negative, $\epsilon_{k+1/2} = a_{k+1/2} - a_{k-1/2} = O(\Delta x)$ when $\Delta_{k+1/2}u = -\Delta_{k-1/2}u$. Thus, for a symmetric or nearly symmetric extremum moving to the right, $\epsilon_{k+1/2}\Delta_{k+1/2}u = O(\Delta x^2)$ and second order accuracy is retained.

Artificial viscosity increases as $|\Delta_{k+1/2}u|$ grows larger relative to $|\Delta_{k-1/2}u|$, but does not exceed $a_{k+1/2}$, the viscosity of Roe's method. (A similar analysis holds if $a_{k+1/2}$ and $a_{k-1/2}$ are non-positive. Similar conclusions also hold for other choices of $\epsilon_{k\pm1/2}$ in equation (8).)

The above represents a lower bound on artificial viscosity. For a steady state method to converge stably, one also needs to add viscosity in monotone regions, and perhaps increase viscosity at extrema. We need to know $C^{\pm}$ and the time-stepping method to say anything more specific.

3 Fourth Order Artificial Viscosity

Incremental form plus fourth order artificial viscosity gives:

$$\begin{aligned} H_k &= -\epsilon_{k+1/2}\Delta_{k+3/2}u + (C^+_{k+1/2} + 2\epsilon_{k+1/2} + \epsilon_{k-1/2})\Delta_{k+1/2}u \\ &\quad +\epsilon_{k-1/2}\Delta_{k-3/2}u - (C^-_{k-1/2} + 2\epsilon_{k-1/2} + \epsilon_{k+1/2})\Delta_{k-1/2}u \end{aligned}$$

where $\epsilon_{k+1/2} \geq 0$ is the coefficient of fourth order artificial viscosity. Fourth order artificial viscosity increases the coefficients of $\Delta_{k\pm1/2}u$, which tends to enforce E1. However if, for example, $\Delta_{k+3/2}u$ is very large and negative, the first term will overwhelm the others, causing u_k to overshoot, violating E1 or E2. Since u_k is too large, u_{k-1} will tend to be too small; if u_{k-1} is too small, u_{k-2} will tend to be too large; etc. In this way, the overshoot at k causes oscillations to the left. This validates the common wisdom that fourth order artificial viscosity should not be used near shocks. On the other hand, fourth order viscosity works quite well in smooth regions, where it will tend to enforce E1/E2 with less accuracy penalty than second order viscosity. It also strongly damps $2\Delta x$ waves: if u_k is a max in a $2\Delta x$ wave, all terms in H_k are negative resulting in a decrease in u_k (and similarly minima are increased). By the Nyquist sampling theorem, $2\Delta x$ waves should be eliminated, since a grid cannot accurately represent wavelengths shorter than $4\Delta x$ [9].

It seems sensible to use a blend of second and forth order artificial viscosity. Consider the 'self-adjusting hybrid method' [10] in conservation form:

$$h_{k+1/2} - \epsilon_{k+1/2}[\theta_{k+1/2}(u_{k+1} - u_k) - (1 - \theta_{k+1/2})(u_{k+2} - 3u_{k+1} + 3u_k - u_{k-1})]$$

where $\theta_{k+1/2}$ varies between 0 and 1: $\theta_{k+1/2}$ should be 1 near shocks and near 0 elsewhere and will formally depend on the choice of $h_{k+1/2}$, the conservative flux of a higher-order method. In practice, the modified blending

$$h_{k+1/2} - \epsilon_{j+1/2}[\theta_{k+1/2}(u_{k+1} - u_k) - \max(0, \delta - \theta_{k+1/2})(u_{k+2} - 3u_{k+1} + 3u_k - u_{k-1})]$$

allows greater flexibility in choosing $\theta_{k+1/2}$. The parameter δ is chosen so that $\theta_{k+1/2}$ is greater than δ near shocks. Note that this leads to an adaptive width stencil of fixed center: near shocks the stencil width shrinks from five points to

three. (Contrast this with the ENO stencil, which has fixed width and adaptive centering [4].) Jameson's method, a self-adjusting hybrid, has proven highly successful for solving the steady Euler equations of a perfect gas [1]. In the closest scalar equivalent, $h_{k+1/2}$ is central differences; $\epsilon_{k+1/2} = \kappa|a_{k+1/2}|$; and θ is a normalized second difference of u_k. (In the original vector version, θ is a normalized second difference of pressure; also, replace $|a_{k+1/2}|$ by $\rho(A_{k+1/2})$, the spectral radius of some average of the Jacobian matrices $A(u_{k+1})$ and $A(u_k)$. Recall that the spectral radius of a matrix with eigenvalues $\lambda^{(j)}, j = 1, ..., N$, is defined as $\max(|\lambda^{(j)}|)$. The algorithm is applied to each component of the vector u_k.) If $\kappa = 1/2$, $\delta = 1$, and $\theta_{k+1/2} = 1$, then the scalar method becomes Roe's method, which is optimal near steady shocks. However, usually one would choose κ somewhat less than $1/2$, trading accuracy at shocks for accuracy in smooth regions. In general, δ and κ are chosen by trial-and-error to yield the best accuracy compromise in each particular situation. Despite its success for steady Euler equations, it may not be easy or even possible to discover a satisfactory θ for use with other equations or for unsteady problems. Also, while intuitively appealing, it is unclear how to choose θ to guarantee rigorously E1/E2, or any other sufficient oscillation control condition.

4 Multi-dimensional Equations

Our theory easily extends to multi-dimensional scalar equations. Consider the two-dimensional equation:

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} + \frac{\partial g(u)}{\partial y} = 0 \tag{9}$$

When will the semi-discrete, finite-difference approximation $u_k(t)$ given by:

$$\frac{du_k(t)}{dt} = \frac{1}{\Delta x_k} H_k(t) + \frac{1}{\Delta y_k} G_k(t) \tag{10}$$

have properities E1 and E2? Theorem 2 is unchanged while Theorem 1 becomes

Theorem 4: Equation (10) has property E1 if

$$\frac{1}{\Delta x_k} H_k(t) + \frac{1}{\Delta y_k} G_k(t) \leq 0 \text{ for } u_k(t) \text{ maximum}$$

$$\frac{1}{\Delta x_k} H_k(t) + \frac{1}{\Delta y_k} G_k(t) \geq 0 \text{ for } u_k(t) \text{ minimum}$$

Corollary 5: Equation (10) has property E1 if

$$H_k(t) \leq 0 \text{ and } G_k(t) \leq 0 \text{ for } u_k(t) \text{ max}$$

$$H_k(t) \geq 0 \text{ and } G_k(t) \geq 0 \text{ for } u_k(t) \text{ min}$$

This corollary justifies the common approach of adding artificial viscosity on a dimension-by-dimension basis, at least in the scalar case. For one-dimensional hyperbolic systems of conservation laws, the characteristic variables will have properties E1 and E2; unfortunately, in more than one dimension, E1/E2 may not hold for all characteristic variables in all regions of flow [9]. For nonlinear vector problems, the characteristic variables can only be found only approximately; thus, it is not yet clear how one might rigorously guarantee E1/E2 when appropriate.

5 Conclusions

Artificial viscosity, TVD, UNO, and ENO, among other recent methods, can successfully combat the spurious oscillations commonly found in higher-order approximations to weak solutions of conservation laws. Here we have presented a new framework for understanding the action of artificial viscosity and, in a subsequent paper, we use the same approach to elucidate TVD, UNO, and ENO conditions [9].

We end with a brief discussion of the potential practical implications of this work. In the common 'method-of-lines' approach, the fully discrete method derives from a semidiscrete method. Popular time-discretizations include forward Euler, Lax-Wendroff (more generally, Cauchy-Kowalewski), and Runge-Kutta. The time-discretization may or, more likely, may not fully preserve extremum control properties. For steady problems, extremum control at intermediate times matters only to the extent that it affects the rate of convergence. For unsteady solutions, one can sometimes tolerate the effects of the time-discretization with good results, particularly if the semidiscrete method is overdamped e. g., [11]. However, rigorous enforcement of E1/E2 in unsteady problems requires consideration of the time-discretization along with the spatial discretization. We currently are engaged in an analysis of the forward Euler and Lax-Wendroff time-discretizations. We hope that success in the simplest cases will pave the way to understanding more efficient and accurate time-stepping algorithms, including Runge-Kutta.

References

[1] A. Jameson, W. Schmidt, and E. Turkel. Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes. *AIAA 14th Fluid and Plasma Dynamics Conference, Palo Alto, CA*, June 22-23, 1981.

[2] A. Harten. High Resolution Schemes for Hyperbolic Conservation Laws. *J. Comp. Phys.*, 49:357, 1983.

[3] A. Harten and S. Osher. Uniformly High-Order Accurate Nonoscillatory Schemes, I. *SIAM J. Numer. Anal.*, 24:279, 1987.

[4] A. Harten, B. Engquist, S. Osher, and S. R. Chakravarthy. Uniformly High Order Accurate Essentially Non-oscillatory Schemes. III. *J. Comp. Phys.*, 71:231, 1987.

[5] J. P. Boris and D. L. Book. Flux Corrected Transport. I. SHASTA, A Fluid Transport Algorithm that Works. *J. Comp. Physics*, 11:38, 1973.

[6] P. L. Roe. Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes. *J. Comp. Phys.*, 43:357, 1981.

[7] E. M. Murman. Analysis of Embedded Shock Waves Calculated by Relaxation Methods. *AIAA Jour.*, 12:626, 1974.

[8] J. L. Steger and R. F. Warming. Flux Vector Splitting of the Inviscid Fluid Dynamics Equation with Application to Finite-Difference Methods. *J. Comp. Physics*, 40:263, 1979.

[9] C. B. Laney and D. A. Caughey. Extremum Control in Numerical Approximations to Conservation Laws. *AIAA Paper 91-0632, AIAA 29th Aerospace Sciences Meeting, Reno, NV*, Janurary 1991.

[10] A. Harten and G. Zwas. Self-Adjusting Hybrid Schemes for Shock Computations. *J. Comp. Physics*, 9:568, 1972.

[11] S. Osher and S. Chakravarthy. High Resolution Schemes and the Entropy Condition. *SIAM J. Numer. Anal.*, 5:955, 1984.

CRITICAL TIME-STEP OF VARIOUS NUMERICAL SCHEMES FOR TRASIENT HEAT CONDUCTION

Rao Yalamanchili and S. Yalamanchili*
Light Armament Division
Close Combat Armament Center
U.S. Army Armament Research, Development and Engineering Center
Picatinny Arsenal, NJ 07806-5000

ABSTRACT. An approach to solve a pressing practical problem, aerodynamic heating of hypervelocity projectiles, is discussed. Since huge amounts of supercomputer time is needed for simulation, a thorough and detailed investigation is initiated on numerical methods for transient three dimensional heat transfer around hypervelocity projectiles and for establishing critical time steps. A methodology is established for comparing various numerical methods, in particular, finite-element, finite difference and method of weighted-residuals (MWR). Starting with the variational principle, an equivalent finite - element equation with 27 nodal temperatures of a three - dimensional element is established. Similar equations are formulated for finite-difference and MWR, in particular for collocation and Galerkin techniques. The critical time-steps are derived by Von Neumann method for various numerical methods and include 1-D, 2-D, and 3-D transient heat conduction problems. A comparison shows a drastic need for robust and efficient codes due to not only an increase in band width, number of nodes (by several orders of magnitude), etc. but also a decrease in critical time-step (by more than order of magnitude) for multi-dimensional problems.

INTRODUCTION. Army Research Office recognized, aerodynamic heating of hypervelocity projectiles, as a very complicated problem with many facets, each of which offers formidable challenges. This project is initiated in FY 90. The motivation for this paper comes from the needs of this project. Department of Defense prepared a list of twenty-two critical technologies for Congress. The following are pertinent to this project: Hypervelocity Projectiles where USSR is ahead; Computational Fluid Dynamics (CFD), Parallel Computers; Simulation and Modeling; Software Producibility; and Composite Materials. The ultimate goal, to be

*County College of Morris/Rutgers University, College of Engineering, Piscataway, NJ

accomplished, is to prepare a self contained package of simulation from the time a weapon is fired until the time the hypervelocity projectile reaches the target.

The thermal package involves convection, conduction and radiation in addition to other complications such as unsteady, three dimensional and hypervelocity (viscous-inviscid interaction and non-equilibrium flow) effects. For example, convective heat transfer may be modeled from CFD package which is governed by full three dimensional Navier Stokes equations. Typical computer run may take more than 20 hours on a cray supercomputer in order to simulate one supersonic flow filed around a projectile: Transient 3-D Navier-Stokes Solver, 500K nodes, 10 seconds per iteration, 8000 iterations.

The transient three dimensional heat conduction model will provide a means to determine the temperature distribution as a function of location (3 dimensions) and time for any given initial and boundary conditions. The boundary conditions are usually obtained from CFD. There are occasions where there is a strong compiling between convective flow and conduction. No matter what method is utilized, all require great computer storage and large amounts of computer time. The solution process is not only subjected to these restrictions but also bound to blow-up, in the middle, if accuracy, stability, and monoscillation characteristics are not taken into account by proper selection of numerical techniques. If a combined convection and conduction problem is attempted in one-step, the failure in one area can lead to a loosing proposition in both areas. Therefore, a search is initiated to find an accurate, robust and efficient numerical scheme for the solution of transient three dimensional head conduction problems.

Various numerical methods are in use today. The most popular methods are finite-element (FE) finite-difference (FD), and Weighted residuals (MWR). It is not an easy task to single out the best one. In any case, one has to bring all these methods into the same format in order to make any meaningful comparison.

FINITE ELEMENT DIFFERENCE EXPRESSION

Wilson and Nickell, following Gurtin's discussion of variation principles for linear initial value problems, confirmed that the function $T(x,y,z,t)$ which leads to an extremum of functional

$$\Omega_t(T) = 1/2 \int_V \{ \rho C_p T^* T + \nabla T^* K^* \nabla T - 2\rho C_p T_o^* T \} \, dU - \int_S \hat{Q}_i n_i \, ^* T \, dS$$

is the solution of the following transient heat-conduction equation:

$$(K^* T, _i) , _i - \rho C_p * \frac{\partial T}{\partial t} + \rho \, ^* p = 0$$

with the boundary condition $K^* T, i - \hat{Q}i = 0$

Where $T(x,y,z,t)$ = temperature at the spatial point (x,y,z) and at time t

To = Initial temperatures

T = Gradient of T with respect to spatial coordinates

K = Thermal conductivity

ρ = Material density

Cp = Heat capability of the material per unit mass

$$\hat{Q}_i \ (x,y,z,t) = \int_0^t Q_i \ (x,y,z,\tau) \ d\tau$$

U = Volume

* = Convolution symbol defined as:

$$T^* T = \int_0^t T(x,y,z,t-\tau) \ T \ (x,y,z,\tau) \ d\tau$$

$$\nabla T^* \nabla T = \frac{\partial T}{\partial x} * \frac{\partial T}{\partial x} + \frac{\partial T}{\partial y} * \frac{\partial T}{\partial y} + \frac{\partial T}{\partial z} * \frac{\partial T}{\partial z}$$

Divide the three dimensional solid body into I axial elements (nodes 0 to I), J transverse elements (nodes 0 to J) and K normal elements (nodes 0 to K) such that step sizes are the same in all three directions. This

restriction is introduced to simplify algebraic manipulations involved in the analysis. Instead a unit step size can be assumed without any loss of generality and generalize later.

Consider the nodal point (i,j,k), in the range 0 < i < I, 0 < j < J, 0 < k < K) as shown in Figure 1. The temperature of the nodal point will vary as a function of time, t. The temperature distribution in a subregion is a function of spatial coordinates (x,y,z) and surrounding nodal point temperatures. For simplicity, linearity and the same functional distribution are assumed for all elements. The functions $f_1, f_2, f_3, f_4, f_5, f_6, f_7$, and f_8 are functions of nodal point temperatures. These are determined by substitution of the coordinates of nodal points into the equation and by solving the resulting simultaneous equations. The results for region II are as follows:

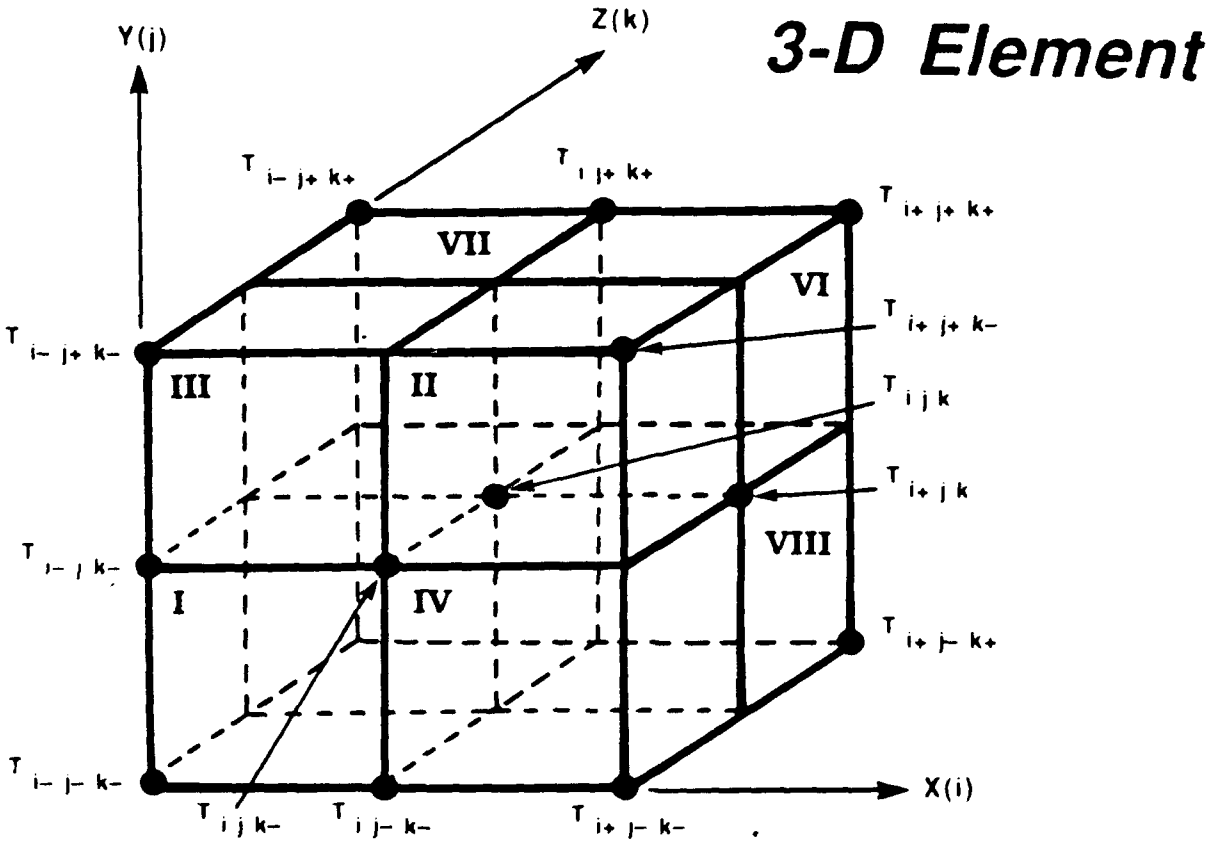


Figure 1

3-D FEA

- 8 Regions (Sides of Length, $\triangle L$)
- Varies Linearly with x, y, z

$$T(x, y, z) = f_1 + f_2 x + f_3 y + f_4 z + f_5 xy + f_6 xz + f_7 yz + f_8 xyz$$

Derive for Unit Length & Generalize Later for $\triangle L$

- Region II

$$f_1 = T_{ijk}$$

$$f_2 = T_{i+jk} - T_{ijk}$$

$$f_3 = T_{ij+k} - T_{ijk}$$

$$f_4 = T_{ijk} - T_{ijk-}$$

$$f_5 = (T_{i+j+k} - T_{i+jk}) - (T_{ij+k} - T_{ijk})$$

$$f_6 = (T_{i+jk} - T_{i+jk-}) - (T_{ijk} - T_{ijk-})$$

3-D FEA (Continued)

$$f_7 = (T_{ij+k} - T_{ij+k-}) - (T_{ijk} - T_{ijk-})$$

$$f_8 = (T_{i+j+k} - T_{i+j+k-}) - (T_{i+jk} - T_{i+jk-}) - (T_{ij+k} - T_{ij+k-})$$

$$T(x,y,z) = z(x+y-1)T_{ijk-} + zx(y-1)T_{i+jk-} + yz(x-1)T_{ij+k-}$$
$$-xyzT_{i+j+k-} + \boxed{(x-1)(y-z-1)-yz}\,T_{ijk} + x(1-y)(z+1)T_{i+jk}$$
$$+(1-x)y(z+1)T_{ij+k} + xy(z+1)T_{i+j+k}$$

ONE CAN DERIVE, THE FOLLOWING PARTIAL DERIVATIVES, FROM THE ABOVE RELATIONSHIP:

$$\frac{\partial T}{\partial X}$$

$$\frac{\partial T}{\partial Y}$$

$$\frac{\partial T}{\partial Z}$$

Similar set of equations can be derived for the remaining seven regions of the 3 dimensional finite element discussed above. It is now time to substitute all equations derived into all three terms of functional, governing equation, and integrate over the volume occupied by region II and take the first variation with respect to $T_{i,j,K}$ at the new time in order to obtain the extremum of the functional. Obtain similar results for the other regions and sum them up. The procedures for the first and third terms of governing equation are same. However, double convolution symbol is involved in the second term. Here the evaluation of the second term in the governing equation involves integration not only over the volume (and the use of first variation with respect to the nodal temperature, $T_{i,j,K}$ at the new time) but also over the time-step due to the additional convolution symbol. Towards this goal, a linear nodal point temperature variation is assumed within each time-step. Summing up the results of all three terms produces an equivalent finite-element equation in a form familiar to finite-difference community.

The details of derivation can be found in the Journal of Heat Transfer, Transactions of ASME, for a two-dimensional case. It is beyond the scope of this paper to provide all those tedious derivations and long equations and may be found in a journal to be published soon for a three dimensional case. The equivalent finite-element equation contains all 27 nodal temperatures of a three-dimensional element, shown above at both old and new times.

FINITE DIFFERENCE EXPRESSION

There are various versions of finite-difference approximations to the transient heat conduction problems. However, all these schemes can be classified as either explicit or implicit type. In the case of explicit scheme, the unknowns are determined, one at a time, without the simultaneous solution of the entire set of algebraic equations. This, in turn, produces tremendous savings in computer time, almost one to three ratio over the implicit counterparts. However, the explicitl schemes are usually conditionally stable and demand small time-steps.

If implicit finite-difference scheme is chosen, one equation for each node is generated in the entire body, and finally simultaneous solution of all these algebraic equations is required. Numerous difference formulas can be formulated for first and second order derivatives and also for three dimensional Laplacian term, depending upon the number of nodes

those nodes. Usually, the difference formulas are constructed for first and second order derivatives by the expansion of the function in Taylor's series and algebraic manipulations. The three dimensional Laplacian is constructed by use of the central difference formulas for the second order derivatives.

The simplest 7 point approximation for three dimensional Laplacian can be written as:

$$(T_{i+jk}+T_{i-jk}+T_{ij+k} + T_{ij-k} + T_{ijk+}+T_{ijk-} -6T_{ijk})/\Delta L^2$$

This is $0(\Delta L^2)$. You can also write a 19-point approximation which is $0(\Delta L^4)$ for three dimensional Laplacian. Similarly, one can also prepare a 27 point approximation which is $0(\Delta L^6)$ for three dimensional Laplacian. There are many more possibilities for a 3-D Laplacian. However, it is suffice to unify or compare various numerical methods for the time being. Remember, apples can't be compared to oranges or vice-versa. They all have to be brought to the same format before a meaningful comparison can be made.

THE METHOD OF WEIGHED-RESIDUALS (MWR) EXPRESSIONS

The method of weighted-residuals unifies many approximate methods of the solution of differential equations that are in use today. Variational principles proposed by several authors are all applications of the MWR. In literature, this technique is commonly called the error distribution principle. The choice of approximating function, in an assumed solution form, is crucial in applying the MWR. No way presently seems to be available to select the approximating functions systematically for all problems. The variation between results obtained by application of different weighting functions to the same approximating solution is much less significant than the variations that can result from the choice of different approximate solution forms. Sometimes, one can obtain the exact solution by use of the MWR if the right choice is made in the selection of the approximate solution form.

The objective of applying the method of weighted residuals is to minimize the error by distribution of it over the interval with the help of a weighting function in such a way that the net error is zero. There are many variations in it The most popular ones are the method of collocation, method of moments, method of Galerkin, and method of least squares. However, this study is limited to method of collocation and Galerkin method. The Dirac-delta function is the weighting function in

method of collocation. The weighting function is same as the distribution function (in approximate solution form) in case of Galerkin method. Mathematically,

$$\iiint R(x,y,z)\ W(x,y,z)\,dxdydz = 0$$

Where W is the weighting function and R, the residual can be written as

$$R(x,y,z) = \phi\left(\frac{\partial^2 T}{\partial x^2}+\frac{\partial^2 T}{\partial y^2}+\frac{\partial^2 T}{\partial z^2}\right)^{t+}_{ijk} + (1-\phi)\left(\frac{\partial^2 T}{\partial x^2}+\frac{\partial^2 T}{\partial y^2}+\frac{\partial^2 T}{\partial z^2}\right)^{t}_{ijk}$$
$$-\left(T^{t+}_{ijk} - T^{t}_{ijk}\right) \Big/ \left(K/(\rho C_p)\,\Delta t\right)$$

The parameter ϕ allows a weighted average of three second order spatial derivatives at two discrete times. For compactness, the commas are omitted in between subscripts. The algebraic sign following the subscript or superscript indicates an increment (+) or decrement (-) in the corresponding step-size. The linear temperature distribution is assumed between the adjacent discrete points in order to apply the Galerkin method. Substituting the residual, weighting functions and 3-D Laplacian over 8 regions and performing the integration process yields equations similar to the finite element and finite difference methods.

CRITICAL TIME STEPS

Various numerical methods are introduced for transient 3-D heat conduction problems from classical to modern approaches. However, all require some guidelines in order to obtain successful numerical solutions subjected to various initial and boundary conditions. We are forced to select finite step sizes, both time and spatial, to satisfy practical considerations. The temperatures and material properties change continuously within each time step. However, it is not uncommon to integrate with respect to time based on the information known only at the beginning of the time-step (Euler) or by utilization of data equally both at the beginning and end of the time-step (Crank-Nicolson). Other variations are possible. Since the techniques vary considerably, the errors can either grow or decay. The errors also vary not only with respect to time but also from location to location. It is possible that less accurate scheme may be more accurate in one location than more accurate scheme and vice versa.

Numerous authors discuss accuracy and more or less understood in the following way: The RMS error and the absolute maximum error can be computed from the difference between the analytical and the numerical solution and include error contributions from every grid point. Double precision arithmatic is usually used to minimize the effects of roundoff error. Since the RMS error and the absolute maximum error behaves similarly, the RMS error is used in determining the accuracy of scheme.

In general, the error is a function of time, time-step, and spatial step-sizes. If sufficient time is allowed to communicate the influence of boundary conditions to the interior points, the error decreases as time increases thereafter. Therefore, the steady state error is smaller than transient errors and the selection of numerical scheme may not be that critical for steady state cases.

It is well known that selection of large time-step can lead to meaningless oscillatory numerical solution. The largest time-step for which an Euler Solution will be stable is called the critical time-step. Of course, if Euler solution is stable, the Crank-Nicolson solution is nonosillatory for the same critical time-step. The critical time-step can be derived either by Von Neumann method or by other techniques. The following table provides information for selection of critical time-steps for various numerical methods and heat conduction problems.

MULTI-DIMENSIONAL EFFECTS

FOURIER NUMBER (F) = $\alpha \Delta t / \Delta L^2$

DIMENSION(S)	FOURIER NUMBER FD	FE
1-D	1/2	1/6
2-D	1/4	1/12
3-D	1/8	1/24

CONCLUSIONS. The critical time-steps, shown above, form the guiding light on use of numerical methods. They help to avoid wastage of precious supercomputer time. The table also demonstrates that the finite-element techniques require smaller time-steps over their finite-difference counterparts. Another interesting fact is drastic reduction in time-step sizes as number of dimensions increases. Overall, more than order of magnitude reduction in time-step is not uncommon depending upon the chosen numerical method and multidimensional effects. Remember that the number

of nodes increases by several orders of magnitude for multi-dimensional problems over their one-dimensional problems. As far as band-width is concerned,the typical one - dimensional problem may contain three non zero diagonal terms whereas three dimensional one may have 27 non-zero diagonal terms. All these adverse effects point to a drastic need for development of robust and efficient codes.

In summary, practical hypervelocity projectile heating problem is analyzed. The principal subproblem is investigated, in detail, for solution of transient three-dimensional heat conduction by various numerical methods. A methodology is established for comparison of several numerical methods, in particular, finite-element, finite-difference, and method of weighted residuals. The critical time-steps are derived by Von Neumann method for several cases.

A comparison shows a drastic need for robust and efficient codes due to not only an increase in band-width and number of nodes by several orders of magnitude but also a decrease in critical time-steps by more than order of magnitude for multi-dimensional problems. Even if the time-step, shown above, is satisfactory for many problems, a further reduction is in order for problems subjected to convection and radiation boundary conditions as suggested by Professor Meyers of The University of Wisconsin-Madison, based on analysis of 1-D and 2-D problems.

Symbolic Computation: Pure Computer Mathematics

Bruno Buchberger
RISC-LINZ

—

Research Institute for Symbolic Computation
Johannes Kepler University, A4040 Linz, Austria (Europe)
Tel: +43 (7236) 3231-41, Fax: +43 (7236) 3338-30
Bitnet: k313370 at aearn

—

October 24, 1990

Abstract

This paper is a survey on Symbolic Computation. We first propose *a new definition* of the term "Symbolic Computation": Symbolic Computation is computation with objects that are "symbolic" in the sense that they are finite representations (symbols) for infinite, abstract ojects in the domains of pure mathematics. In the second part of the paper we summarize the *facilities available in modern symbolic computation software systems.* In the third part we give some *examples of recent mathematical research* results on which improved symbolic computation algorithms can be based. Finally, we compile the most important *literature and software systems references* for encouraging newcomers to access these powerful new problem solving techniques.

The main message of this paper is twofold:

- Applied mathematicians are encouraged to experiment with the available Symbolic Computation software systems and, by doing so, will experience an enormous expansion of their problem solving potential.
- Pure mathematicians are encouraged to consider the "algorithmization" of their respective field of interest and will experience the enormous intellectual and mathematical challenge of such a project that often goes far beyond the degree of difficulty found in "traditional" pure mathematics.

1 A New Definition of Symbolic Computation

Pragmatically, one could "define" Symbolic Computation (symbolic mathematics, computer algebra, formula manipulation) to encompass everything that is available in present-

day symbolic computation systems like MACSYMA, Maple, Mathematica or Scratchpad. Of course, a definition by characterizing properties of the field is more desirable.

Sometimes, Symbolic Computation is simply characterized as "Non-Numerical Computation". This characterization, however, is too wide. Many non-numerical algorithms, for example, graph theoretical algorithms are not considered to be symbolic. Similarly, a definition that characterizes Symbolic Computation simply as "computation with symbols" is too wide because, for example, word processing considers symbols without having the flavor of symbolic computation.

In an attempt to characterize the flavor of computations in existing "Symbolic Computation" software systems with a view to predict and challenge what should and could be added to theses systems in the future, we belive that the fundamental feature of Symbolic Computation is that

Symbolic Computation =
computation with finite (concrete) objects
having infinite (abstract) semantics.

Example: The string "$\sin(2x)$" is a finite object. It represents an infinite object, namely, a mathematical function consisting, in terms of set theory, of infinitely many pairs of real numbers. In other words, the symbolic, finite, concrete expression "$\sin(2x)$" has an infinite semantics (meaning) in an abstract domain of mathematics.

Example: The string "$((x,y),(2^8,3^3))$" is a finite object that may be considered as a concise representation of a very complex ("practically infinite") abstract mathematical object, namely, a finite group with 6912 elements and a 6912×6912 multiplication table.

Example: A formula of the theory of real closed fields, for example, $\forall a_0, a_1, a_2 \exists y (x^3 + a_2 x^2 + a_1 x + a_0 = 0)$ is a finite string object. It says something about infinitely many abstract objects, namely, real numbers. If, by finite manipulation on such finite formula objects, the formula can be proven it says something about an infinite abstract domain, i.e. the formula has an infinite semantics.

Problem solving in Symbolic Computation, then, proceeds in the following steps:

- We want to solve a problem in a domain of infinite, complex finite, "abstract", objects (a domain of *"pure" mathematics*).
- We represent (a subset of) these abstract by finite representations ("symbolic" representations).
- We try to solve the problem for the abstract domain by solving, by an *algorithm*, the corresponding problem for the finite representations.

Example: The famous problem of "symbolic integration" is specified as follows: Given a symbol string s find a symbol string t such that the function represented by t is the antiderivative of the function represented by s. Note that the problem specification necessarily involves the *semantics* of the string objects, i.e. the notion "the function represented by ...".

As a pun, we could also say that

Symbolic Computation = Pure Computer Mathematics

with two possible parsings: Symbolic Computation is computerization of pure mathematics (i.e. mathematics in abtract domains) and Symbolic Computation is computer mathematics in its purest form.

It is a common misunderstanding that

Symbolic Computation =
Trivial Mathematics
Repeated in Loops on a Computer

and therefore is the domain of "mathematicians" who want to escape the intrinsic difficulties of inventing proofs by, instead, "experimenting" with well-known existing mathematics on examples using computers.

I dare to assert that, rather,

> Pure Computer Mathematics typically needs more sophisticated mathematics (i.e. more sophisticated proofs) than Pure Mathematics !

This is so because problem solving in mathematics essentially proceeds by proving theorems that show how a problem can be reduced to other (hopefully easier) problems.

- In Pure Mathematics the problem reduction (proof of the theorem) may involve very powerful, non-algorithmic operators (e.g. "choose an x such that $P(x)$").
- In Pure Computer Mathematics (Symbolic Computation) the problem reduction must involve only algorithmic operators (e.g. "while $P(x)$ do $x := f(x)$").

Pure Computer Mathematics, therefore, often is more demanding than Pure Mathematics because fewer tools are available in the algorithmic reduction methods of constructive proofs (that can be translated, one-to-one, to computer algorithms).

(On the other hand, one can also argue that Pure Mathematics is more demanding than Pure Computer Mathematics because, by results of algorithm theory, Pure Computer Mathematics can sometimes represent only relatively modest subdomains of the abstract general domains of Pure Mathematics and, thus, the problem reductions of Pure Mathematics are more general and therefore sometimes more difficult.)

Summarizing,

- Symbolic Computation is an exciting and challenging future area for pure mathematicians from all areas (because more sophisticated proofs are needed).
- Symbolic Computation becomes more and more powerful for all application areas (because the more sophisticated proofs result in "better" problem solving methods).
- Symbolic Computation combines the elegance and "insight" of pure mathematics with the practical efficiency of computer mathematics.

2 Symbolic Computation at Work

In order to be able to assess the practical problem solving power of recent symbolic computation software systems it is best to experiment with at least one of them, typically in interactive mode on a workstation. In the final section of the paper we compile the information necessary to obtain these systems from the academic or professional vendors. Also, it may help to read the papers (Arney et al. 1990) and (Wang 1990) in the same proceedings.

In this section we can only enumerate the most important facilities these systems typically provide.

Arithmetic in Basic Domains:
Symbolic computation software systems provide

- "long" integers and rationals,
- high precision floating point numbers,
- other number domains (e.g. finite fields, algebraic number fields),
- basic number theoretic functions (e.g. Extended GCD, Moebius function etc.),
- basic combinatorial function (e.g. Stirling numbers, Bernoulli numbers etc.),
- polys and rational functions,
- classes of orthogonal polys (e.g. Legendre etc.),
- elementary transcendental functions and mathematical constants,
- special functions (e.g. Bessel, Gamma, exponential integral etc.),
- simplification of expressions involving all of the above,
- coercion and transformation between different number domains.

Algebraic Systems:
Symbolic computation software systems provide

- exact solution of algebraic equations (with symbolic coefficients, multivariate, higher degree),
- use of equations as "algebraic rules",
- transformation between various representations of curves and surfaces,
- complete solution to systems of multivariate, non-linear inequalities,
- topologically correct decomposition of n-space w.r.t polynomial inequalities.

Example: The command `Reduce[ a x^2 + b x + c == 0, x]` in Mathematica will result in a complete symbolic analysis of the various possible solutions dependent on the values of the parameters a, b, c.

Example: The command `AlgebraicRules[ { a == x + y, b == x y}, {x, y, a, b}]` in Mathematica will result in the set of rewrite rules `{ - y ^2 -> b - a y, -x -> -a + y }` that completely rewrites any polynomial in x, y in terms of the polynomials a and b, if this is possible, or decides that this is not possible at all. This powerful simplification mechanism is based on the author's Gröbner bases method, see (Buchberger 1985).

Example: The command `gbasis( [ c1 * c2 - cf * ct * cp + sf * sp, ...], [sp, st, sf, cp, ct, cf, py, s2, s1, c2, c1] )`, where `[ c1 * c2 - cf * ct * cp + sf * sp, ...]` is the system of algebraic equations that describes the inverse kinematic of a certain class of robots, produces a fully triangularized version of the system

$$
\begin{aligned}
& c_1^2 + \frac{px^2}{pz^2 - 2 \cdot l_1 \cdot pz - l_2^2 + l_1^2} = 0, \\
& c_2 + \frac{pz^2 - 2 \cdot l_1 \cdot pz - l_2^2 + l_1^2}{l_2} \cdot px \cdot c_1 = 0, \\
& s_1^2 - \frac{pz^2 - 2 \cdot l_1 \cdot pz + px^2 - l_2^2 + l_1^2}{pz^2 - 2 \cdot l_1 \cdot pz - l_2^2 + l_1^2} = 0, \\
& s_2 - \frac{pz - l_1}{l_2} = 0, \\
& py + \frac{pz^2 - 2 \cdot l_1 \cdot pz - l_2^2 + l_1^2}{px} \cdot c_1 \cdot s_1 = 0, \\
& cf^2 - \frac{pz^2 - 2 \cdot l_1 \cdot pz + px^2 - l_2^2 + l_1^2}{pz^2 - 2 \cdot l_1 \cdot pz - l_2^2 + l_1^2} = 0, \\
& ct = 0, \\
& cp + \frac{pz^3 - 3 \cdot l_1 \cdot pz^2 - l_2^2 \cdot pz + 3 \cdot l_1^2 \cdot pz + l_1 \cdot l_2^2 - l_1^3}{l_2 \cdot pz^2 - 2 \cdot l_1 \cdot l_2 \cdot pz + l_2 \cdot px^2 - l_2^3 + l_1^2 \cdot l_2} \cdot s_1 \cdot cf = 0, \\
& sf + \frac{pz^2 - 2 \cdot l_1 \cdot pz - l_2^2 + l_1^2}{pz^2 - 2 \cdot l_1 \cdot pz + px^2 - l_2^2 + l_1^2} \cdot c_1 \cdot s_1 \cdot cf = 0, \\
& st + \frac{pz^2 - 2 \cdot l_1 \cdot pz - l_2^2 + l_1^2}{pz^2 - 2 \cdot l_1 \cdot pz + px^2 - l_2^2 + l_1^2} \cdot s_1 \cdot cf = 0, \\
& sp + \frac{pz^4 - 4 \cdot l_1 \cdot pz^3 - 2 \cdot l_2^2 \cdot pz^2 + 6 \cdot l_1^2 \cdot pz^2 + 4 \cdot l_1 \cdot l_2^2 \cdot pz - 4 \cdot l_1^3 \cdot pz + l_2^4 - 2 \cdot l_1^2 \cdot l_2^2 + l_1^4}{l_2 \cdot px \cdot pz^2 - 2 \cdot l_1 \cdot l_2 \cdot px \cdot pz + l_2 \cdot px^3 - l_2^3 \cdot px + l_1^2 \cdot l_2 \cdot px} \cdot c_1 \cdot s_1 \cdot cf = 0.
\end{aligned}
$$

where the dependent quantities $c1, c2, s1, s2, py, cf, ct, cp, sf, st, sp$ are decoupled and expressed in terms of the independent quantities px, pz and symbolic parameters $l1, l2$ (the length of the robot arms). For a detailed description of this example see (Buchberger 1987).

Example: Again, a call to a Gröbner basis package (which is meanwhile availabe in most symbolic computation software systems) with a parameter presentation `[x - r t, y - r t^2, z - r^2]` of a surface in 3D space as input, will automatically produce the implicit presentation `x^4 - y^2 z` of the same surface, see (Buchberger 1987).

Computer Analysis

Symbolic computation software systems provide

- limites of symbolic expressions,
- calculations with finite and infinite power series,
- derivatives of symbolic expressions,

- indefinite and definite integral of symbolic expression,
- decision about existence of integrals in certain domains,
- Laplace transforms etc.
- symbolic solution of certain classes of differential equations,
- symbolic simplification of sums and products,

Example: The command `powerseries( log(sin(x)/x), x, 0)` in MACSYMA will produce the infinite symbolic sum presentation of the input function:

$$-(\sum_{i=0}^{\infty} \frac{2^{2i}\mathtt{bern}(2i)}{(2i)!})\log(x) - \log(x)$$

Example: The commands `eq1: 3 * 'diff(f(x),x,2) - 2 * 'diff(g(x),x) = sin(x);` and `eq2: a * 'diff(g(x),x,2) + 'diff(f(x),x) = a * cos(x);` in MACSYMA define two ordinary second order differential equations `eq1` and `eq2`. The commands `eq1: laplace(eq1,x,s)` and `eq2: laplace(eq2,x,s)` then compute the Laplace transform of `eq1` and `eq2`. A call of `linsolve` will solve the linear equations in the unknowns `laplace(f(x),x,s)` and `laplace(g(x),x,s)` and a call of `ilt` (inverse Laplace transform) will finally give the symbolic result

$$\mathtt{f(x)} = \frac{\frac{27a^{5/2}\sin(\frac{\sqrt{6}x}{3\sqrt{a}})}{\sqrt{6}(3a-2)} - 3a^2\cos(\frac{\sqrt{6}x}{3\sqrt{a}})}{3a} - \frac{3a\sin(x)}{3a-2} + a + f(0)$$

$$\mathtt{g(x)} = \frac{\frac{9a^{3/2}\sin(\frac{\sqrt{6}x}{3\sqrt{a}})}{\sqrt{6}} + \frac{27a^2\cos(\frac{\sqrt{6}x}{3\sqrt{a}})}{6a-4}}{3a} - \frac{(3a+1)\cos(x)}{3a-2} + 1/2.$$

See (Fateman 81) for more details on this example.

Linear Algebra, Tensor Calculus

Symbolic computation software systems provide

- operations on matrices with symbolic entries,
- inversion, linear systems, nullspace, eigenvalues, ...
- operations on tensors.

Numerical Mathematics, Statistics

Symbolic computation software systems provide routines

- for numerical computation (curve fitting, Fourier transform, Newton root finding, numerical integration, Runge Kutta DE solution, ...), and

- for statistics.

Automated Theorem Proving

Symbolic computation software systems provide

- automated proofs of geometrical theorems,
- automated proofs of arbitrary first-order-formulae on the reals,
- proofs of first order equations w.r.t. equational axiom systems,
- general first order theorem proving.

Example: Apollonios' Circle Theorem: "The altitude pedal of the hypotenuse of a right-angled triangle and the midpoints of the three sides of the triangle lie on a circle." After introducing coordinates, a possible algebraic formulation of this problem is as follows:

for all coordinates $a_1, \ldots, a_{10} \in \mathbf{R}$:
if $h_1(a_1, \ldots, a_{10}) = 0, \ldots, h_8(a_1, \ldots, a_{10}) = 0$,
then $c(a_1, \ldots, a_{10}) = 0$,

where the h_i are the polynomials describing the hypotheses of the theorem and c is the polynomial describing the conclusion of the theorem. The algebraic formulation is what is called the radical membership problem "$c \in$ Radical$(\{h_1, \ldots, h_m\})$?". Arbitrary such questions can be decided by deciding "$1 \in$ Gröbner-Basis$(\{h_1, \ldots, h_m, z.c - 1\})$?", where z must be a new indeterminate. More details on this example are contained in (Buchberger 1987). In the system (Kutzler 1988) and similar systems, proofs of the above kind can be carried out in a few seconds.

Algebraic Geometry:

Symbolic computation software systems provide

- analysis of and computation in residue class rings modulo polynomial ideals (by Gröbner bases),
- free resolution of polynomial ideals (determination of the sequence of syzygy modules by Gröbner bases),
- analysis of and computation in commutative and non-commuative associative algebras (Lie, Weyl algebras), group algebras etc.
- analysis of and computation in groups given in various representations.

Example: In the CAYLEY system the following sequence of interactive commands:

```
g:  free(a,b);
g.relations: a ↑ 8, b ↑ 7, (a * b) ↑ 2, (a ↑ −1 * b) ↑ 3;
h =< a ↑ 2, a ↑ −1 * b >;
i = todd coxeter(g,h);
Print i;
```

effects the enumeration of all 448 cosets of the subgroup $< a^2, a^{-1}b >$ of $< a, b \mid a^8, b^7, (ab)^2, (a^{-1}b)^3 >$ by the famous Todd-Coxeter algorithm.

Group Theory Zoomed:

Any of the above areas could be "zoomed" revealing a wealth of problems and solution techniques available in present symbolic computation systems. For example, in computational group theory the following can be computed:

- test for nilpotency, commutativity, solvability etc.
- coset enumeration
- normalizers
- centralizers
- central chains
- series
- lattice of (normal) subgroups
- classes of conjugacy subgroups
- orbits
- test of imprimitivity
- test of isomorphism
- word problems
- automorphism groups.

Different representations of the group theoretical objects lead to different solution algorithms for the above problems with drastically different efficiency. Therefore the study of representations and the conversion between different representations is an important subarea of computational group theory. The main representation methods are:

- generators and relations
- permutations, block permutations

- power commutators
- bases
- strong generating sets
- matrices over **Z**, Galois fields and algebraic number fields,
- Cayley graphs
- character tables.

"Icons" of Other Symbolic Computation Areas

Other important and evolving subareas of Symbolic Computation can only be mentioned as "icons" here:

- Symbolic Computation based computational geometry
- automatic programming
- computational number theory
- computational topology
- ...

Interfaces:

Symbolic computation software systems provide

- advanced 2D and 3D graphics and animation,
- interface to textprocessing,
- output in FORTRAN, C etc. syntax (symbolic computation as preprocessing!)

Programming facilities:

Symbolic computation software systems provide

- functional and procedural programming,
- "generic" programming,
- "rewrite rules" style programming using pattern matching,

Example: Mathematica, for example, provides a convenient style of programming in the form of "rewrite rules" that is particularly convenient for mathematicians who are used present mathematical knowledge in the form of problem reduction rules ("theorems"). For example, an "algorithm" for limes computations could be quickly assembled by formulating Mathematica rules of the following style: `Limes[ a_ + b_] := Limes[ a] + Limes[ b]`. For a given symbolic input expression, Mathematica would check in its rule base whether an expression matching the pattern `a_ + b_` occurs as a subexpression and would effect the corresponding transformation.

3 Examples of Recent Theoretical Results

In this section we give some samples of recent theoretical research results in the area of symbolic computation. These results are taken from papers that appeared or will appear in one of the special issues of the Journal of Symbolic Computation. A complete list of all special issues of this journal that appeared in the last five years and will appear in the next two years is contained in Section 4.8.

These samples of theoretical results should give a flavor of the breadth of ongoing foundational research in symbolic computation and should also demonstrate the depth of mathematics necessary to come up with algorithmic solutions to symbolic computation problems. Each of these results also gives rise to many important and challenging open problems that could attract the attention of mathematicians who are interested in embarking on new directions off the beaten track.

3.1 Proofs of Combinatorial Identies

In (Zeilberger, Takayama 1992) a new approach to the automated proof of combinatorial identities and the determination of definite integrals and sums is developed. It is based on the notion of "holonomic functions". Roughly, a holonomic function is a function f for which an ideal A in the Weyl algebra of differential operators exists such that $Af = 0$ (and A satisfies some other properties). The class of holonomic functions is huge and includes most of the practically interesting functions like polynomials, rational functions, algebraic functions, trigonometric functions, exponential and logarithm, hypergeometric functions, binomial coefficients, Bessel functions, Legendre functions etc.

It is shown how, for an A for $f(x,t)$ an annihilating B for $g = \int_{-\infty}^{\infty} f(x,t)\mathrm{d}t$ can be constructed. Hence, the integral g can be obtained as a solution to the equation $Bg = 0$.

An essential subalgorithm in the construction of B for the elimination of variables and other purposes is a generalized version of the Gröbner bases algorithm (Buchberger 1985).

3.2 Automatic Discovery of Geometric Theorems

In (Sturmfels, Whiteley 1991) "Cayley factorization" of expressions in the Cayley algebra is discussed. The mathematical problem consists in devicing an algorithm that takes arbitrary expressions in the bracket algebra, for example, an expression of the form

$$-[abc][ade][bdf][cef] + [abd][ace][bcf][def] \tag{1}$$

and to generate an equivalent expression in the Cayley algebra, for example, the expression

$$(ab \wedge de) \vee (bc \wedge ef) \vee (cd \wedge fa). \tag{2}$$

The problem is open for the general case. A solution is given in the above paper for the multlinear case.

This solution, however, suffices to "automatically generate" a lot of non-trivial geometrical theorems or to automatically prove geometrical conjectures. For example, the

question of finding a necessary and sufficient condition for 6 points a, b, c, d, e, f in the plane to lie on a quadric, which is equivalent to the statement that

$$\det \begin{pmatrix} a_1^2 & a_2^2 & a_3^2 & a_1a_2 & a_1a_3 & a_2a_3 \\ \vdots & & & & & \vdots \\ f_1^2 & f_2^2 & f_3^2 & f_1f_2 & f_1f_3 & f_2f_3 \end{pmatrix} = 0 \qquad (3)$$

can be answered by transforming (by the well known "straightening algorithm") condition (3) into the equivalent condition that (1) is zero and then, by the new Cayley factorization algorithm, into the equivalent condition that (2) is zero. The latter condition, however, has the immediate geometrical interpretation that the 6 points lie on a quadric if and only if certain intersection points of connection lines between the lines lie on one common straight line (Pascal's Theorem). This means that the theorem was automatically produced.

3.3 Point Location Problems Solved With Cylindrical Algebraic Decomposition

In (Chazelle, Sharir 1990) it is shown how a very general class of point location problems can be solved by using Collins' well known algorithm for producing "cylindrical algebraic decompositions" of the n-space.

Given n polynomials in m variables, the m-space is naturally partitioned into a finite set of "cells". Given a point in m-space the point location problem consists in locating the "cell" in which the point lies. (Collins 1975) showed how the decomposition of the m-space can effectively be computed by algebraic algorithms. *In (Chazelle, Sharir 1990)* it is shown how Collins' decomposition can be used to solve the general point location problem in $O(\log n)$ time after $O(n^{2^d})$ preprocessing operations for building up a suitable data structure.

3.4 Black Box Algorithms

In (Kaltofen, Trager 1990) a new type of algorithms is introduced: "black box algorithms" for problems having polynomials as input and output. Their method starts from the observation that for a polynomial to be "known" it suffices to have a method that produces, for any argument, the corresponding value of the polynomial. It is not actually necessary that we know the coefficients of the polynomial.

Let us consider, for example, the factorization problem for multivariate polynomials over the integers. Given a polynomial f (i.e. given some method to compute $f(a_1, \ldots, a_n)$ for arbitrary arguments), the factorization problem is solved if we know a method that, given any $(x_1, \ldots, x_n)$, produces the values $h_1(x_1, \ldots, x_n), \ldots, h_m(x_1, \ldots, x_n)$ of the factors $h_1, \ldots, h_m$ of f. This method may use calls of f for various $(a_1, \ldots, a_m)$ as "oracles". In the paper, such a method is developed for the factorization problem and related problems and it is shown that polynomial time complexity can be achieved.

3.5 Polycyclic Quotient Algorithm

In (Sims 1990) an important problem of computational group theory is treated: the computation of polycylcic quotient groups for a group given by a finite presentation. The paper starts from an earlier method by Baumslag-Cannonito-Miller that did not fully cover all the algorithmic subproblems. (Sims 1990) fills the gaps by giving a method how to completely reveal the structure of two residue class rings that appear in the construction. This new method is a modification of the Gröbner basis technique.

3.6 Linear Diophantine Equations and Unification

The problem of unification can be considered to be the most general formulation of the problem of solving equations in arbitrary first order theories. It is well known that the unification problem in "AC"-theories (theories involving only equational axioms for associative-commutative function symbols) can be reduced to the problem of solving systems of linear diophantine equations over the natural numbers.

Earlier solutions of the linear diophantine equations problems relied on systematic enumeration. In (Clausen, Fortenbacher 1989) a new method is presented that uses certain "completion" steps that can be interpreted as "walks" in certain labeled digraphs. Significant speed-up can be achieved by this new approach.

3.7 Gröbner Fans of Ideals

The method of Gröbner bases has numerous applications in polynomial ideal theory and related areas (algebraic geometry, geometric modeling). Gröbner bases depend on an underlying "admissible" ordering of power products. There are infinitely many "admissible" orderings. In (Mora, Robbiano 1988) it is shown that, for a given polynomial ideal, the infinitely many admissible orderings fall into finitely many classes. One class corresponds to exactly one Gröbner basis. Furthermore, it is possible to compute, for a given polynomial ideal, one "universal" Gröbner basis, i.e. a set of polynomials that is a Gröbner basis for the given ideal under all possible admissible orderings.

3.8 Cluster-Based Cylindrical Algebraic Decomposition

As explained above, Collins' algorithm constructs, for n polynomials in m variables, a decomposition of the n-space into "sign-invariant cells". The computation proceeds by, first, reducing the problem for the n-space to the problem for the $(n-1)$-space and then showing how to construct a solution for the n-space from a solution for the $(n-1)$-space. The latter construction proceeds by building a "stack" of n-cells over each $(n-1)$-cell.

In (Arnon 1988) it is shown how the cells of the $(n-1)$-space can be automatically clustered into bigger sign-invariant blocks. In Collins' algorithm it then suffices to erect stacks over these bigger, and fewer, blocks. By recursion, this may save enormous time in the construction of the n-space decomposition.

4 A Guide to the Literature and Software Systems

4.1 References to Papers Cited in the Text

(Arney et al. 1990) David C. Arney, Jeffrey Misner,
Derive as a Research Tool, this conference.

(Arnon 1988) D.S. Arnon,
A Cluster-Based Cylindrical Algebraic Decomposition Algorithm, Journal of Symbolic Computation 5/1-2, February/April 1988.

(Buchberger 1985) Bruno Buchberger,
Gröbner Bases: An Algorithmic Method in Polynomial Ideal Theory, Chapter 6, pp. 184-232 in: N.K. Bose: Multidimensional Systems Theory, D. Reidel Publishing Company.

(Buchberger 1987) Bruno Buchberger,
Proceedings Workshop on Scientific Software, IMA, Minneapolis, USA, March 23-26, 1987, pp.59-88, IMA Volums in Mathematics and its Applications, Volume 14, Springer.

(Chazelle, Sharir 1990) B. Chazelle, M.Sharir,
An Algorithm for Generalized Point Location and its Applications, Journal of Symbolic Computation 10/3-4, September 1990.

(Clausen, Fortenbacher 1989) M. Clausen, A. Fortenbacher,
Efficient Solutions of Linear Diophantine Equations, Journal of Symbolic Computation 8/1-2, July/August 1989.

(Collins 1975) G.E. Collins,
Quantifier Elimination for Real Closed Fields by Cylindric Algebraic Decomposition, Proceedings 2nd GI Conference on Automata Theory and Formal Languages, Springer Verlag. LNCS 33, Berlin, 1975, pp. 515-532.

(Fateman 1981) Richard J. Fateman,
Symbolic and Algebraic Computer Programming Systems, ACM SIGSAM Bulletin. Volume 15, Number 1, 2/81, pp.21-32.

(Kaltofen, Trager 1990) E. Kaltofen, B. Trager,
Computing with Polynomials Given by Black Boxes for their Evaluations: Greatest Common Divisors, Factorization, Seperation of Numerators and Denominators, Journal of Symbolic Computation 9/3, March 1990.

(Kutzler 1989) Bernhard Kutzler
Algebraic Approaches to Automated Geometry Theorem Proving, PhD Thesis, RISC-Institute, University of Linz, Austria, November 1988.

(Mora, Robbiano 1988) T. Mora, L. Robbiano,
The Gröbner Fan of an Ideal, Journal of Symbolic Computation 6/2-3, October/December 1988.

(Sims 1990) C. Sims,
Implementing the Baumschlag-Cannonito-Miller Polycyclic Quotient Algorithm, Journal of Symbolic Computation 9/5-6, May/June 1990.

(Sturmfels, Whiteley 1991) B. Sturmfels, W. Whiteley,
On the Synthetic Factorization of Projectively Invariant Polynomials, Journal of Symbolic Computation 12/4-5, October 1991.

(Wang 1990) Paul S. Wang,
Advances in integrating Symbolic, Numeric and Graphics Computing, this conference.

(Zeilberger, Takayama 1992) D. Zeilberger, N. Takayama
Computerized Proofs of Combinatorial and Special Function Identities, Journal of Symbolic Computation 13/5-6, June 1992.

4.2 Symbolic Computation Software Systems

Some of the more important symbolic computation software systems are listed here in alphabetic order with some characterizing information in the order

- developed since which year
- address for ordering the system
- available on which machines
- some characteristic features.

CAYLEY V4

- since the early 1970's, steadily expanded and improved
- Prof. John Cannon
  Department of Pure Mathematics
  University of Sidney
  Sidney NSW 2006
- Workstations of Apollo, DEC, IBM, SUN.
  IBM machines running VM/CMS and VAX machines running VMS.

- Cayley V4 is a system designed for solving problems in algebra, number theory and algebraic combinatorics, with strong emphasis on structural questions. A user language based on the concepts of set, mapping and algebraic structure provides a natural notation for algorithms in this area.

DERIVE

- 1990 (Version 1.62)
- Soft Warehouse, Inc.
  3615 Harding Avenue, Suite 505
  Honolulu, HI 96816
  Phone: (808) 734-5801
- IBM PC compatible machines under MS-DOS: approximately 200 $.
- The computer algebra system Derive is designed for use on microcomputers. A very compact system. A menu-driven interface makes Derive easy and natural to use. Derive has great potential as a teaching aid.

MACSYMA

- 1988 (Version 412.6)
- Computer Aided Mathematics Group
  Symbolics Inc.
  New England Executive Park East
  Burlington, MA 01803
  Phone: (617) 221-1250
  Fax: (617) 221-1099
- 386-based MS-DOS systems.
  Workstations of Apollo, DEC, Symbolics, SUN.
- Macsyma is a general system for numerical, symbolic and graphical computation. It offers over 1300 documented commands for solving a wide range of mathematical problems.

MAPLE

- 1989 (Version 4.3)

- Waterloo Maple Software
  160 Columbia Street West
  Waterloo, Ontario, Canada
  N2L 3L3
  Phone: (519) 747-2373
  Fax: (519) 747-5284

- 386-based: approximately 680 $. Atari ST, Apple MacII, MacSE, MacPlus: approximately 380.00 $.
  Workstations of Apollo, DEC, HP, IBM, MIPS, Silicon Graphics, SUN and many others: approximately 2400 $.

- Maple is a powerful and user-friendly system for numerical, symbolic and graphical computation that incorporates a high-level programming language. It delivers a large library of about 2000 mathematical functions. Maple requires remarkably little memory and is therefore an ideal multi-user system.

MATHEMATICA

- 1989 (Version 1.2)

- Wolfram Research, Inc.
  P.O. Box 6059
  Champaign, Illinois 61821
  Phone: 217-398-0700
  Fax: 217-398-0747

- 386-based MS-DOS systems: approx. 695 $.
  Apple MacII, MacSE, MacPlus: approx.495 $.
  Workstations of Apollo, DEC, HP, IBM, MIPS, Silicon Graphics, SUN and many others: approx. 2,250 $.

- Mathematica is a general system for numerical, symbolic and graphical computation. The system produces excellent 2D and 3D PostScript graphics and incorporates a modern high-level programming language with pattern-match and rewriting style programming. On the Macintosh, Mathematica has a sophisticated user interface which supports interactive textbooks.

REDUCE

- 1988 (Version 3.3)
- Dr. A.C. Hearn
  The RAND Corporation
  P.O. Box 2138
  Santa Monica, CA 90406-2138
- Reduce has been implemented on many different computers ranging in power from the IBM PC to the Cray X-MP.
- Reduce is a general-purpose computer algebra system designed for physicists, mathematicians and engineers. Since, at present, Reduce is the most widely-used computer algebra system in a number of countries many application packages are available. Users with access to any of the major research computer networks can obtain newly released packages and other material from a digital library.

SAC-2

- Since the early 1970'2, continuously expanding and improving.
- Prof. George Collins
  Computer Science Department
  Ohio State University
  Columbus, Ohio
- Available on all machines having a FORTRAN or C compiler.
- SAC-2 is a collection of carefully designed algebraic algorithms mainly for algorithms on polynomials including Collins' decision algorithm for the logical theory of real closed fields. The algorithms are the basis for many implementations in other systems. ALDES, the programming language of the system, is compiled into FORTRAN or C and, thus, makes the system widely available and fast. All algorithms are availabe in source code and are scientifically documented.

SCRATCHPAD

- Prototyp
- Richard D. Jenks
  Computer Algebra Group
  IBM Research Division

T.J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

- IBM mainframes under VM/CMS.
  RT/PC and PS/2 under AIX.

- Scratchpad is a general computer tool for mathematics. Its design is based upon abstract datatypes which are organised into programmable algebraic hierarchies. Scratchpad's modular library has over 200 datatypes. In addition, there is a growing library of algebraic functions. Scratchpad has an interactive language for easy access to library facilities and simple programming tasks.

4.3 Text Books and Survey Books on Computer Algebra

Akritas, A.G. (1989).
Elements of Computer Algebra with Applications.
John Wiley and Sons.

Buchberger, B. (ed.) (1985).
Computer Algebra.
Proceedings of EUROCAL 85, Vol. 1 (Invited Lectures). Springer LNCS 203.
(A collection of survey lectures on some main topics in computer algebra.)

Buchberger, B., Collins, G.E., Loos, R. (Eds.) (1982).
Computer Algebra: Symbolic and Algebraic Computation.
Springer Verlag, Wien - New York.
(This is not really a text book but a collection of survey articles on the main subareas of computer algebra. Some of the articles, in addition, contain details of theorems and algorithms not available in other text books.)

Davenport, J.H., Siret, Y., Tournier, E. (1988).
Computer Algebra: Systems and Algorithms for Algebraic Computation.
Academic Press, London.
(The first real text book on computer algebra. For some of the subjects treated, no proof details are given. Instead, references to the original literature are provided.)

Della Dora, J., Fitch, J. (1989).
Computer Algebra and Parallelism.
Academic Press.

Knuth, D.E. (1981).
The Art of Computer Programming. Vol. 2: Seminumerical Algorithms.
Addison-Wesley, Reading.

(The algorithms, together with their mathematical foundation, for arithmetic in the most important algebraic domains are given in this book.)

Janßen, R. (1988).
Trends in Computer Algebra..
Springer.
(A collection of survey lectures at an international sympolsium in Bad Neuenahr, May 1987.)

Lipson, J.D. (1981).
Elements of Algebra and Algebraic Computing.
(This text book is a systematic introduction to basic areas of algebra together with algorithms for the algebraic problems discussed. However, it covers only a part of what today is computer algebra.)

Mignotte, M. (1989).
Mathematiques pour le calcul formel.
Presses Universitaires de France.

4.4 Survey Articles on Computer Algebra:

Buchberger, B., Kutzler, B. (1986).
Computer-Algebra für den Ingenieur.
In: *Rechner-Orientierte Verfahren* (Buchberger et al. eds.). Teubner, Stuttgart, pp.11-64.

Caviness, B.F. (1985).
Computer Algebra: Past and Future.
J of Symbolic Computation 2/3, 217-236.

Kaltofen, E. (1987).
Computer Algebra Algorithms.
In: *Annual Review of Computer Science* Vol.2, 91-118 (J.F. Traub ed.), Annual Review Inc., Palo Alto, California.

Yun, D.Y.Y., Stoutemyer, R.D. (1980).
Symbolic Mathematical Computation.
In: *Encyclopedia of Computer Science and Technology* (J. Belzer et al. eds.), Vol. 15, 235-310, M. Dekker, New York - Basel.

4.5 Books on Subareas of Computer Algebra:

Davenport, J.H. (1981).
On the Integration of Algebraic Functions.
Springer LNCS 102.
(Published version of a Ph.D. thesis.)

Atkinson, M.D. (ed.) (1984).
Computational Group Theory.
Academic Press, London.
(Proceedings of the 1983 LMS Symposium in Durham. There is no text book on computational group theory. These proceedings may serve as a possible starting point for obtaining an overview on the subject.)

Stauffer, D., Hehl, F.W., Winkelmann, V., Zabolitzky, J.G. (1988).
Computer Simulation and Computer Algebra.
Springer.

Tournier, E. (1989).
Computer Algebra and Differential Equations.
Academic Press.

4.6 Books on Applications of Computer Algebra:

Caviness, B.F., Gilbert, R.P., Shtokhamer, R. (19??).
An Introduction to Applied Symbolic Computation Using MACSYMA.
(In preparation).

Howard, J.C. (1979).
Practical Applications of Symbolic Computation.
IPC Science and Technology Press, Guildford, England.
(Some application examples, using FORMAC, are given in much detail. Somewhat out of date.)

Klimov, D.M., Rudenko, V.M. (1989).
Computer Algebra Methods for Mechanics Problems.
Moskva.

Pavelle, R. (Ed.) (1985).
Applications of Computer Algebra.
Kluwer Academic Publisher, Boston - Dordrecht - Lancaster.
(A collection of application papers using MACSYMA).

Rand, R.H. (1984).
Computer Algebra in Applied Mathematics: An Introduction to MACSYMA.
Pitman Publishing Inc., Marshfield MA.
(Examples of using MACSYMA. Not so much a primer of MACSYMA.)

Rand, R.H., Armbruster, D. (1987).
Perturbation Methods, Bifurcation Theory and Computer Algebra.
Springer.
(Application of computer algebra in a special area.)

4.7 Books on Specific Computer Algebra Systems:

Rayna, G. (1987).
REDUCE: Software for Algebraic Computation. Springer, New York.
(Introduction to REDUCE with case studies.)

Symbolics Inc. (1987).
MACSYMA User's Guide.
(A primer for getting acquainted with MACSYMA.)

Wooff, C., Hodgkinson, D. (1987).
muMATH: A Microcomputer Algebra System.
Academic Press, London.
(A tutorial on muMATH.)

4.8 Journals on Computer Algebra:

[JSC] *Journal of Symbolic Computation (B. Buchberger et al. eds.).*
Academic Press London, 1985- .
(The first refereed journal on computer algebra and all other areas of symbolic computation. In addition to the regular issues it frequently publishes special issues on certain subareas of computer algebra. For example,

- Vol. 3/1-2 (Feb 1987). *Rewriting Techniques and Applications.* (J.P. Jouannoud ed.)
- Vol. 4/1 (August 1987). *Algorithmic Methods in Algebra and Number Theory.* (M. Pohst ed.)
- Vol. 5/1-2 (April 1988). *Algorithms in Real Algebraic Geometry.* (D. Arnon and B. Buchberger eds.)
- Vol. 6/2-3 (Oct 1988). *Computational Aspects of Commutative Algebra.* (L. Robbiano ed.)

- Vol. 7/3-4 (March - April 1989). *Unification: Part 1.* (C. Kirchner ed.)
- Vol. 8/1-2 (July - Aug. 1989). *Unification: Part 2.* (C. Kirchner ed.)
- Vol. 9/3 (March 1990). *Complexity of Algebraic Algorithms.* (E. Kaltofen ed.)
- Vol. 9/5-6 (May - June 1990). *Computational Group Theory.* (J. Cannon ed.)
- Vol. 10/3-4 (Sept. - Oct. 1990). *Computational Geometry.* (E. Welzl and R. Seidel eds.)
- Vol. 11/1-2 (Jan. - Feb. 1991). *Rewriting Techniques for Theorem Proving.* (J. Hsiang and L. Bachmair eds.)
- Vol. 12/4-5 (Oct. - Nov. 1991). *Invariant Theoretic Algorithms in Geometry.* (B. Sturmfels and N. White eds.)
- Vol. ? (June - July 1992). *Symbolic Algorithms for Combinatoric Identities.* (P. Paule and D. Zeilberger eds.))

Journal of Applicable Algebra and Error Correcting Codes.
Springer, New York-Heidelberg, 1990- .
(A recent journal emphasizing the application of Computer Algebra to coding theory and related subjects.)

Journal of Automated Reasoning.
Reidel Publishing Company, 1985- .
(This journal specializes in Symbolic Comutation algorithms in the area of automated theorem proving.)

ACM SIGSAM Bulletin.
Distributed by the ACM Special Interest Group on Symbolic and Algebraic Manipulation.
(This is a non-refereed informal bulletin for the fast dissemination of papers, implementation notes, announcements, bibliographies etc. in the area of computer algebra.)

Acknowledgement: This paper was written in the frame of the project "Gröbner Bases" sponsored by the "Österreichisches Ministerium für Wissenschaft und Forschung". I am grateful to my student W. Windsteiger who helped in the production of the manuscript.

Phase Transitions and Maximally Dissipative Dynamic Solutions in the Riemann Problem for Impact*

Thomas J. Pence**

Dept. of Metallurgy, Mechanics and Materials Science
Michigan State University
East Lansing, MI 48824-1226

Introduction

Nonlinearly elastic materials whose strain energy density is a nonconvex function of strain can undergo phase transitions in which displacement gradients are discontinuous across internal surfaces. In a dynamic setting, this gives rise to both conventional shock waves and travelling phase boundaries. Physically such phase boundaries separate states which involve different microstructure, even though they are the same compositionally. These type of phase transitions can be induced by the mechanical impact of solid bodies [1]-[4] or even by the shock due to an intense laser pulse [5].

The one-dimensional theory of fully dynamical isothermal phase transitions has been utilized in the investigation of phase boundary propagation [6]-[8]. In this setting the material response mirrors the anomalous pressure-volume relation of a van der Waals fluid and corresponding mathematical issues arise [9]-[16]. In particular, it is often the case that the equations of mechanical motion are not sufficient to ensure a unique solution to various boundary value problems. A number of criteria for selecting physically admissible solutions have been proposed [9]-[16] and many of their solid mechanical analogues are discussed in [6].

We consider an impact problem for the case where one material (that of the target) admits the possibility of phase transitions and the other material (that of the impactor) does not. The goal is to determine the nature of the wave packet that is generated upon impact at the interface of contact for various values of impact velocity $\bar{v}$. Attention is restricted to longitudinal compressive waves; interactions with additional surfaces of the target or impactor are not considered. As this study is motivated by considerations of high velocity impact, the impact velocity $\bar{v}$ may be large. Thus a mathematical treatment can not utilize techniques based on small initial data, as for example utilized in [12] in the parallel setting of a van der Waals fluid.

The purpose of this communication is to outline a method for treating this problem. This treatment relies on a mapping from candidate solutions for both impactor and target to a plane (the (σ^*,v^*)-plane) of contact interface stress σ^* vs. contact interface velocity v^*. For certain values of impact velocity $\bar{v}$ we find that the problem under consideration yields a unique solution on the basis of the equations of motion alone. However for values of impact velocity $\bar{v}$ within a finite interval $(\bar{v}_1,\bar{v}_2)$ we find that uniqueness does not hold. In this case we seek solutions that are maximally dissipative. The maximum dissipation criterion was proposed by Dafermos [16] for resolving uniqueness, although in [16] it is called the entropy rate admissibility criterion. Here we find that this criterion delivers uniqueness for a certain proper subinterval of $(\bar{v}_1,\bar{v}_2)$. In addition we find that this uniqueness will

*Supported by the U.S. Army Research Office under DAAL03-89-G-0089.
**The author of this paper presented it at the Seventh Army Conference on Applied Mathematics and Computing.

extend to the whole interval, and hence all impact velocities $\bar{v} > 0$, if certain additional constitutive restrictions are assumed of the two-phase material.

The purpose of this communication is not to give an exhaustive treatment of these results. Instead, a number of theorems are presented, many without proof, which summarize the basic methodology used in arriving at these conclusions. Some of the proofs rely on fairly elaborate algebraic calculations, which have been performed with the aid of MACSYMA. A complete exposition is given in [17].

Preliminaries

We consider the impact between an initially stationary target (occupying $x > 0$), and a moving impactor (which occupies $x < 0$ at the instant of contact $t = 0$). Since impact will give rise to compressive stress it will be convenient to define compressive strain γ and compressive stress σ as positive. The constitutive response is given by

$$\sigma = \begin{cases} \zeta(\gamma), & \text{in } x < 0, \\ \tau(\gamma), & \text{in } x > 0, \end{cases} \tag{1}$$

where $\zeta(\gamma)$, $\tau(\gamma)$ are nonlinear elastic (compressive) stress response functions for the impactor and target respectively. By definition, the compressive strain γ is confined to the interval $0 \le \gamma < 1$ and diverges as $\gamma \to 1$. For the purpose of the pilot problem presented here it will be convenient to assume that γ can take on all values obeying $\gamma \ge 0$ and that $\zeta(\gamma)$ and $\tau(\gamma)$ are defined for all $\gamma \ge 0$. The modifications to the argument to be given here appropriate for the case $0 \le \gamma < 1$ can be found in [17].

In a Lagrangian frame, the one dimensional propagation of longitudinal compression waves in an isothermal setting are described by the equations:

$$\begin{aligned} &\gamma_t + v_x = 0, \qquad v_t + \zeta'(\gamma)\,\gamma_x = 0, \qquad \text{in } x < 0, \\ &\gamma_t + v_x = 0, \qquad v_t + \tau'(\gamma)\,\gamma_x = 0, \qquad \text{in } x > 0, \end{aligned} \tag{2}$$

and discontinuity conditions

$$\begin{aligned} &\dot{s}\,[|\gamma|] - [|v|] = 0, \quad \dot{s}\,[|v|] - [|\zeta|] = 0, \quad \text{in } x < 0, \\ &[|v|] = 0, \qquad \zeta = \tau \qquad \text{on } x = 0, \\ &\dot{s}\,[|\gamma|] - [|v|] = 0, \quad \dot{s}\,[|v|] - [|\tau|] = 0, \quad \text{in } x > 0, \end{aligned} \tag{3}$$

where $\dot{s}$ represents the time derivative of a generic discontinuity curve $x = s(t)$, and $[|\ |]$ denotes the jump in the enclosed quantity across $x = s(t)$.

Let $\bar{v}$ denote the initial velocity of the impactor. Both impactor and target are assumed to be intially stress free. In view of the absence of length and time scales in the problem formulation, we introduce the similarity variable $\lambda = x/t$ and assume that

$$v(x,t) = \tilde{v}(\lambda), \quad \gamma(x,t) = \tilde{\gamma}(\lambda). \tag{4}$$

The initial conditions then give

$$\tilde{v}(-\infty) = \dot{v}, \quad \tilde{\gamma}(-\infty) = 0,$$
$$\tilde{v}(\infty) = 0, \quad \tilde{\gamma}(\infty) = 0. \tag{5}$$

In view of (4) the equations of motion (2) now become ordinary differential equations, while the discontinuity conditions (3) become a set of algebraic restrictions, across any rays of discontinuity $\lambda = \lambda_i$.

As is well known, the motion of a discontinuity interface will in general give rise to a change in the total mechanical energy of the dynamical fields. The rate of mechanical energy change is given as $-\Sigma\, d_i$ where the summation is over the total number of discontinuity interfaces and d_i is the dissipation rate of the i-th discontinuity interface. Consider the i-th discontinuity interface and let $\gamma-$ and $\gamma+$ obeying $0 \le \gamma- \le \gamma+$ be the strains adjacent to this interface. Then

$$d_i = D(\gamma-,\gamma+), \tag{6}$$

where $D:[0,\infty)X[0,\infty)\to(-\infty,\infty)$ is the dissipation function. If the interface in question occurs in the target, i.e. $x > 0$, then this dissipation function D is given by

$$D(y,z) = |S(y,z)|\, A(y,z), \tag{7}$$

where

$$S(y,z) = \left((\tau(z)-\tau(y))/(z-y)\right)^{\frac{1}{2}},$$
$$A(y,z) = (\tau(z)-\tau(y))*(z-y)/2 - \int_y^z \tau(q)\, dq. \tag{8}$$

The function S gives the velocity of the discontinuity interface. If the interface in question occurs in the impactor, i.e. $x < 0$, then (7),(8) continue to hold with τ replaced by ζ. If the interface in quesiton is the interface of contact $x = 0$, then $D = 0$ since this interface is stationary in a Lagrangian frame.

A pair of piecewise smooth functions $\left(\tilde{v}_L(\lambda),\ \tilde{\gamma}_L(\lambda)\right)$ defined on $\lambda \le 0$ which satisfy $(2)_1$, $(3)_1$, (4) and in addition obey $(5)_1$ will be called a *candidate dynamical state for the impactor*. A pair of piecewise smooth functions $\left(\tilde{v}_R(\lambda),\ \tilde{\gamma}_R(\lambda)\right)$ defined on $\lambda \ge 0$ which satisfy $(2)_2$,$(3)_3$, (4) and in addition obey $(5)_2$ will be called a *candidate dynamical state for the target*. For each candidate dynamical state for the impactor, denote the values of strain, velocity and stress on the interface of contact by γ_L^*, v_L^* and ζ^*, that is

$$\gamma_L^* = \tilde{\gamma}_L(0), \quad v_L^* = \tilde{v}_L(0), \quad \zeta^* = \zeta(\gamma_L^*). \tag{9}$$

For each candidate dynamical state in the target denote the corresponding contact interface strain, velocity and stress values by

$$\gamma_R^* = \tilde{\gamma}_R(0), \quad v_R^* = \tilde{v}_R(0), \quad \tau^* = \tau(\gamma_R^*). \tag{10}$$

A pair of candidate dynamical states, $\left\{\left[\bar{v}_L(\lambda),\ \bar{\gamma}_L(\lambda)\right],\ \left[\bar{v}_R(\lambda),\ \bar{\gamma}_R(\lambda)\right]\right\}$, will be said to be a *dynamical state solution* for the impact problem if

$$v_R^* = v_L^* = v^*, \quad \varsigma^* = \tau^* = \sigma^*, \tag{11}$$

where we have defined common interface values v^* and σ^* for the velocity and stress. Condition (11) follows from $(3)_2$. A dynamical state solution gives rise to a pair of functions $\left[\bar{v}(\lambda),\ \bar{\gamma}(\lambda)\right]$ defined on $-\infty < \lambda < \infty$ that satisfy all the conditions to be a solution to the impact problem.

It shall henceforth be assumed that the target admits the possibility of phase transitions in compression and that the impactor does not. In particular, the response function $\varsigma(\gamma)$ for the single phase impactor material is assumed to obey:

HI1) $\varsigma(0) = 0$, and $\varsigma(\gamma) > 0, \quad \varsigma'(\gamma) > 0, \quad \varsigma''(\gamma) < 0$ for $\gamma > 0$.

The response function $\tau(\gamma)$ for the two phase target material is assumed to obey

HT1) $\tau(0) = 0$, and $\tau(\gamma) > 0$ for $\gamma > 0$.

In addition it is assumed that there exist two distinguished values of compressive strain γ_M and γ_m obeying $0 < \gamma_M < \gamma_m$ such that $\tau'(\gamma_M) = \tau'(\gamma_m) = 0$, and

HT2) $\tau'(\gamma) > 0, \quad \tau''(\gamma) < 0,$ for $0 < \gamma < \gamma_M$,

HT3) $\tau'(\gamma) < 0,$ for $\gamma_M < \gamma < \gamma_m$,

HT4) $\tau'(\gamma) > 0, \quad \tau''(\gamma) \geq 0,$ for $\gamma > \gamma_m$.

HT5) $\lim_{\gamma \to \infty} \tau'(\gamma) > \tau'(0)$.

The strain regions $(0,\gamma_M)$ and (γ_m,∞) will respectively be called the low-strain or I-phase and the high-strain or II-phase. The interval (γ_M,γ_m) will be called the unstable phase by virtue of the well known instability properties connected with such an interval in either an equilibrium or a dynamic setting.

We shall restrict attention to candidate dynamical states and hence dynamical state solutions which obey two admissibility conditions, both of which involve additional requirements on discontinuity interfaces. The first admissibility condition that we require of all discontinuity interfaces is:

A1) $D(\gamma-,\gamma+) \geq 0$,

corresponding to the idea that no discontinuity interface should ever be a source of mechanical energy. The second admissibility condition that we require pertains only to phase boundaries. It is required that all phase boundaries connect phase-I to phase-II. This has the physical consequence of precluding the existence of dynamical states that involve constant strain and

velocity regions with strains corresponding to the unstable phase. We write this condition thus:

A2) For phase boundaries, $0 \le \gamma- \le \gamma_M$, and $\gamma+ \ge \gamma_m$.

Admissible Candidate Dynamical States for the Impactor and the associated Locus of Contact Values

It can be shown that (A1) forbids discontinuity interfaces in admissible candidate dynamical states for the impactor. Consequently all such dynamical states in the (x,t)-quarter plane (x<0, t>0) consist of a continuous compression wave bounded on each side by a constant strain and velocity region. Moreover it is found that the value γ_L^* completely parametrizes all of these admissible candidate dynamical states in the impactor. The values of ς^* and v_L^* for this parametrization are given by

$$\varsigma^* = \varsigma(\gamma_L^*), \qquad v_L^* = \bar{v} - \int_0^{\gamma_L^*} (\varsigma'(z))^{\frac{1}{2}} dz. \tag{12}$$

Let Γ_1 denote the locus of (admissible) contact values for the impactor generated by (12) in a (σ^*, v^*)-plane. The following result is immediate from (12) and (HI1):

Theorem 1: Γ_1 is a connected semi-infinite curve which is concave down and monotonically decreasing from the initial data point $(\sigma^*, v^*) = (0, \bar{v})$. Changing the value of $\bar{v}$ merely translates Γ_1 vertically. There is a one-to-one correspondence between points on this locus and admissible candidate dynamical states for the impactor.

■

Admissible Candidate Dynamical States for the Target and the associated Locus of Contact Values

It can be shown that the only candidate dynamical states for the target that obey (A1), (A2) fall into one of two possible families. The first family, the i-family, involves dynamical states in the (x,t)-quarter plane (x>0, t>0) that consist of a continuous compression wave bounded on each side by a constant strain and velocity region, all of which are in the I-phase. This one parameter family can be parametrized by γ_R^* on the interval $[0, \gamma_M]$. Let $\Gamma_{2,i}$ denote the locus of admissible contact value pairs (σ^*, v_R^*) generated by the i-family in the (σ^*, v^*)-plane. The main features of the locus $\Gamma_{2,i}$ are summarized in the following theorem.

Theorem 2: $\Gamma_{2,i}$ is a connected curve of finite length that passes through the initial data point $(\sigma^*, v^*) = (0,0)$ and which subsequently is monotonically increasing. Moreover this locus is concave up in the (σ^*, v^*)-plane. There is a one-to-one correspondence between points on $\Gamma_{2,i}$ and the i-family of admissible candidate dynamical states for the target.

■

The second family, the ii-family, gives rise to dynamical states in the (x,t)-quarter plane (x>0, t>0) that involve at most five distinct sectors: S1 - a constant strain and velocity region in the I-phase, S2 - a continuous compression wave in the I-phase, S3 - a constant strain and velocity region in the I-phase, S4 - a phase boundary separating the I and II-phases, and S5 - a constant strain and velocity region in the II-phase. The sectors S1 and S4 are present in all members of this family, whereas sectors S2, S3 and S5 may or may not be present. The absence of S5 indicates that the phase boundary is on the interface of contact x = 0 and hence stationary.

For this family γ_R^* is also the value of the strain adjacent to the phase boundary in the II-phase. Let γ_s denote the value of the strain adjacent to the phase boundary in the I-phase. It can then be shown that pairs (γ_s, γ_R^*) parametrize the ii-family of admissible candidate dynamical states for the target.

In order to characterize this parametrization we introduce the distinguished strain values γ_q, γ_a, γ_b, γ_o, γ_f as follows (see also Figure 1). The value γ_q is given as the unique solution of

$$\tau(\gamma_q) = \tau(\gamma_M), \qquad \gamma_q > \gamma_m. \tag{13}$$

Let γ_a, γ_b denote the well known phase-I and phase-II values of Maxwell strain. That is γ_a, γ_b are the unique roots of

$$\tau(\gamma_a) = \tau(\gamma_b), \qquad D(\gamma_a, \gamma_b) = 0, \qquad 0 < \gamma_a < \gamma_M < \gamma_m < \gamma_b. \tag{14}$$

Let γ_f denote the solution of

$$\tau'(0)\,\gamma_f = \tau(\gamma_f), \qquad \gamma_f > \gamma_m. \tag{15}$$

The existence and uniqueness of γ_f is ensured by (HT5). Finally let γ_o denote the unique solution of

$$D(0,\gamma_o) = 0, \qquad \gamma_o > \gamma_m. \tag{16}$$

It is easily seen that

$$\gamma_b < \gamma_q < \gamma_f, \qquad \gamma_b < \gamma_o < \gamma_f. \tag{17}$$

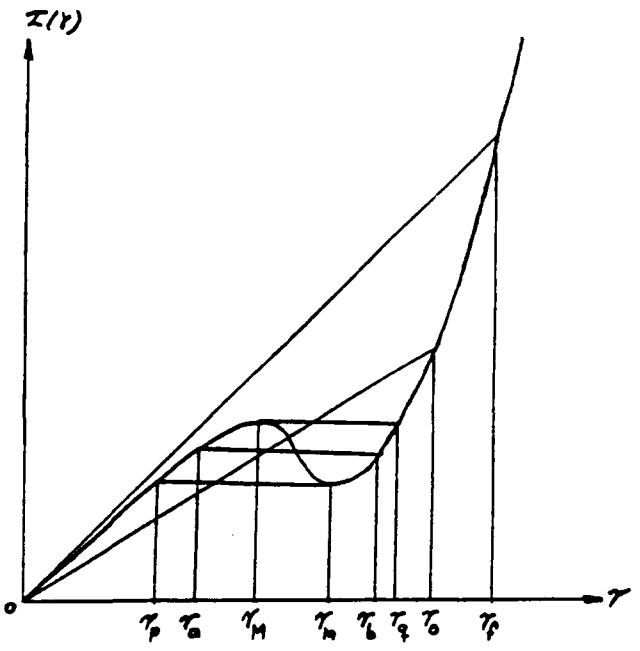


Figure 1: The stress response function $\tau(\gamma)$ for the multi-phase target material and the associated distinguished strain values.

We shall let $\mathbf{T}$ denote the domain of the ordered pairs (γ_s, γ_R^*) for the parametrization of the ii-family. It can then be shown that

$$\mathbf{T} = \{(\gamma_s, \gamma_R^*) \mid \gamma_R^* \geq \gamma_m,\ \gamma_s \in [\mu_s(\gamma_R^*), \mu^s(\gamma_R^*)]\ \}, \tag{18}$$

where the functions μ_s, μ^s are each C^0: $[\gamma_m, \infty) \to [0, \gamma_M]$ (see also Figure 2). The function μ^s is uniquely determined via:

$$\begin{aligned} &\tau(\mu^s(\gamma)) = \tau(\gamma), && \text{for } \gamma_m \leq \gamma \leq \gamma_q, \\ &\tau'(\mu^s(\gamma))\ (\gamma - \mu^s(\gamma)) = \tau(\gamma) - \tau(\mu^s(\gamma)), && \text{for } \gamma_q \leq \gamma \leq \gamma_f, \\ &\mu^s(\gamma) = 0, && \text{for } \gamma \geq \gamma_f. \end{aligned} \tag{19}$$

The function μ_s is uniquely determined via:

$$\begin{aligned} &\tau(\mu_s(\gamma)) = \tau(\gamma), && \text{for } \gamma_m \leq \gamma \leq \gamma_b, \\ &D(\mu_s(\gamma), \gamma) = 0, && \text{for } \gamma_b \leq \gamma \leq \gamma_o, \\ &\mu_s(\gamma) = 0, && \text{for } \gamma \geq \gamma_o. \end{aligned} \tag{20}$$

■

Let $\Gamma_{2,ii}$ denote the locus of (admissible) contact value pairs (τ^*, v_R^*) generated by ii-family in the (σ^*, v^*)-plane . Then the mapping from $\mathbf{T}$ to $\Gamma_{2,ii}$ which takes $(\gamma_s, \gamma_R^*) \to (\tau^*, v_R^*)$ will be denoted by $\mathbf{F} = (F_1, F_2)$. It is

given by

$$\tau^* = F_1(\gamma_R^*) = \tau(\gamma_R^*), \tag{21}$$

$$v_R^* = F_2(\gamma_s, \gamma_R^*) = \int_0^{\gamma_s} (\tau'(z))^{\frac{1}{2}} dz + \left\{ (\tau(\gamma_R^*) - \tau(\gamma_s)) (\gamma_R^* - \gamma_s) \right\}^{\frac{1}{2}}.$$

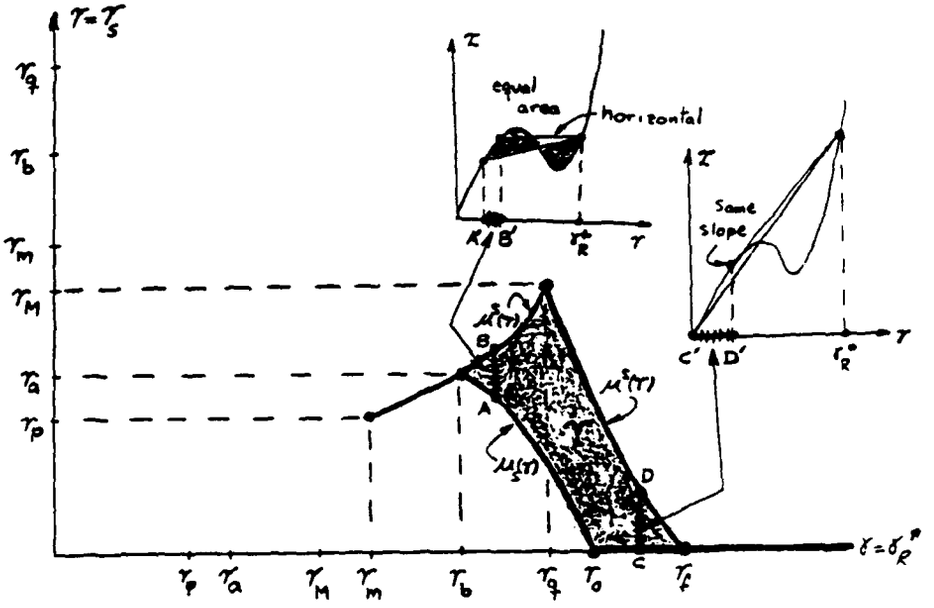


Figure 2: The locus T. The functions $\mu^s(\gamma)$ and $\mu_s(\gamma)$ defined in (19) & (20) can each be constructed graphically with the aid of the stress response function $\tau(\gamma)$.

The locus $\Gamma_{2,ii}$ will consist of curves for $\gamma_m \le \gamma_R^* \le \gamma_b$ and $\gamma_R^* \ge \gamma_f$. These curves will be denoted by $\Gamma_{2,ii}^A$ and $\Gamma_{2,ii}^C$ respectively. For $\gamma_b < \gamma_R^* < \gamma_f$, the locus $\Gamma_{2,ii}$ will consist of a region which will be denoted by $\Gamma_{2,ii}^B$. The boundary of this region is given by curves φ_s and φ^s where

$$\varphi_s = \{(\tau^*, v_R^*) \mid (\tau^*, v_R^*) = F(\mu_s(\gamma_R^*), \gamma_R^*), \quad \gamma_R^* \in (\gamma_b, \gamma_f)\},$$
$$\varphi^s = \{(\tau^*, v_R^*) \mid (\tau^*, v_R^*) = F(\mu^s(\gamma_R^*), \gamma_R^*), \quad \gamma_R^* \in (\gamma_b, \gamma_f)\}. \tag{22}$$

It will also be convenient to define the following distinguished points in the (σ^*, v^*)-plane:

$$P_1 = F(\mu(\gamma_m), \gamma_m), \quad P_2 = F(\mu(\gamma_b), \gamma_b), \quad P_3 = F(\mu^s(\gamma_q), \gamma_q),$$
$$P_4 = F(\mu_s(\gamma_o), \gamma_o), \quad P_5 = F(\mu(\gamma_f), \gamma_f) \tag{23}$$

Salient properties of $\Gamma_{2,ii}$ in the (σ^*, v^*)-plane are given in the following theorem.

Theorem 3: $\Gamma_{2,ii}$ is a connected locus consisting of the union of the finite curve $\Gamma^A_{2,ii}$, the bounded region $\Gamma^B_{2,ii}$ and the semi-infinite curve $\Gamma^C_{2,ii}$. The former curve is monotonically increasing from P_1 to P_2. The latter curve is concave up and monotonically increasing from P_5 to (∞,∞). The boundary of $\Gamma^B_{2,ii}$ is given by $\varphi_s \cup \varphi^s$. Each of the curves φ_s and φ^s originate at P_2 and terminate at P_5. Between these points both φ_s and φ^s are monotonically increasing with φ_s strictly above φ^s in the (σ^*,v^*)-plane. There is a one-to-one correspondence between points on the locus $\Gamma_{2,ii}$ and admissible candidate dynamical states for the target from the ii-family.

■

We note that this theorem ensures that that F possesses a unique inverse on $\Gamma_{2,ii}$ which we shall subsequently denote as F^{-1}. It is found that $\Gamma_{2,i} \cap \Gamma_{2,ii} = \Gamma^A_{2,ii} \cup \varphi^s|_{[P_2\text{-}P_3]}$. The candidate dynamical states of $\Gamma_{2,ii}$ corresponding to this intersection involve a phase boundary which is stationary, i.e. confined to the interface of contact, and thus dissipation free. Other than this stationary phase boundary, these dynamical states are identical to the corresponding candidate dynamical state of $\Gamma_{2,i}$. In what follows these corresponding candidate dynamical states will be regarded as identical. Thus, with this view, *there is a one-to-one correspondence between (admissible) candidate dynamical states in the target and points in the locus* Γ_2.

Admissible Dynamical State Solutions of the Riemann Problem for Impact

From the immediately preceding remark, Theorem 1 and (11), it follows that there is a one-to-one mapping from admissible dynamical state solutions for the impact problem to the set $\Gamma_1 \cap \Gamma_2 = \Pi(\bar{v})$. The notation $\Pi(\bar{v})$ acknowledges the dependence of this set on the impact velocity $\bar{v}$, which in this treatment enters through Γ_1 (viz Theorem 1). We now turn to examine this dependence in more detail. Suppose that P is an arbitrary point in the first quadrant of the (σ^*,v^*)-plane. Theorem 1 then ensures that there is exactly one value of $\bar{v}$ such that $P \in \Gamma_1$. Let θ denote this mapping, i.e. $\bar{v} = \theta(P)$. We shall use this mapping to define two distinguished values of impact velocity $\bar{v}_1$ and $\bar{v}_2$:

$$\bar{v}_1 = \theta(P_2), \quad \bar{v}_2 = \theta(P_5). \tag{24}$$

The significance of these two values for the solution to the impact problem is given in the following theorem.

Theorem 4: The set $\Pi(\bar{v})$ consists of a single point for $\bar{v} \in (0,\bar{v}_1] \cup [\bar{v}_2,\infty)$. However for $\bar{v} \in (\bar{v}_1,\bar{v}_2)$ the set $\Pi(\bar{v})$ consists of a curve coinciding with Γ_1 within the region $\Gamma^B_{2,ii}$. Thus for $\bar{v} \in (0,\bar{v}_1] \cup [\bar{v}_2,\infty)$ there exists a unique admissible dynamical state solution for the impact problem, but for $\bar{v} \in (\bar{v}_1,\bar{v}_2)$ there exists a one-parameter family of admissible dynamical state solutions for the impact problem.

■

The endpoints of $\Pi(\bar{v})$ will be denoted by $P_s(\bar{v}) = \Pi(\bar{v}) \cap \varphi_s$ and $P^s(\bar{v}) = \Pi(\bar{v}) \cap \varphi^s$. Theorem 1 ensures that $\Pi(\bar{v})$ is monontonically decreasing and concave down between these two points (see Figure 3).

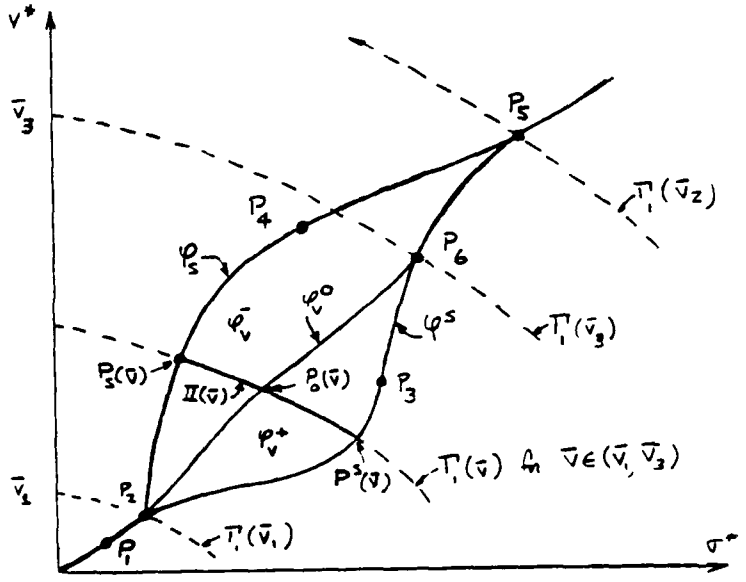


Figure 3: The locus of contact values for the target Γ_2 showing intersections with the locus of contact values for the impactor $\Gamma_1 = \Gamma_1(\bar{v})$ for various impact velocities $\bar{v}$.

Maximally Dissipative Admissible Dynamical State Solutions

Since there is no dissipation associated with the dynamical states Γ_1, the rate of mechanical energy change for an admissible dynamical state solution is that associated with the travelling phase boundaries (if any) in the dynamical state from Γ_2. In particular, for $\bar{v} \in (0,\bar{v}_1]$ the unique admissible dynamical state solution to the impact problem involves conservation of mechanical energy. Whenever $\bar{v} \in [\bar{v}_2,\infty)$, the unique admissible dynamical state solution to the impact problem involves a finite rate of mechanical energy loss (dissipation). For $\bar{v} \in (\bar{v}_1,\bar{v}_2)$, each of the admissible dynamical state solutions in $\Pi(\bar{v})$ will have associated with it a finite value of dissipation. Let $D^*: \Gamma^B_{2,ii} \to [0,\infty)$ be defined as $D^* = D \circ F^{-1}$. Since $\Pi(\bar{v}) \subset \Gamma^B_{2,ii}$ is closed, and D^* is bounded and continuous on the subdomain $\Gamma^B_{2,ii}$, the function D^* will assume a finite maximum at one or more points of $\Pi(\bar{v})$. For $\bar{v} \in (\bar{v}_1,\bar{v}_2)$ let

$\Omega(\bar{v}) \subset \Pi(\bar{v})$ denote those points at which D^* assumes this maximum value. In addition we shall extend the definition of $\Omega(\bar{v})$ to $\bar{v} \in (0,\infty)$ via $\Omega(\bar{v})=\Pi(\bar{v})$ whenever $\bar{v} \in (0,\bar{v}_1] \cup [\bar{v}_2,\infty)$.

Each dynamical state solution corresponding to a point in $\Omega(\bar{v})$ is a *maximally dissipative solution* to the impact problem at impact velocity $\bar{v}$. It is thus immediate from this definition and Theorem 4, that *there is a unique maximally dissipative solution to the impact problem for all impact velocities obeying* $\bar{v} \in (0,\bar{v}_1] \cup [\bar{v}_2,\infty)$. We now seek to determine under what circumstances, if any, there will be a unique maximally dissipative dynamical solution to the impact problem for impact velocities obeying $\bar{v} \in (\bar{v}_1,\bar{v}_2)$.

To determine $\Omega(\bar{v})$, we define loci of constant dissipation on $\Gamma^B_{2,ii}$ for each dissipation value $d \geq 0$ via

$$\Xi(d) = \{(\sigma^*,v^*) \mid (\sigma^*,v^*) \in \Gamma^B_{2,ii},\ D^*(\sigma^*,v^*) = d\}. \tag{25}$$

One then finds that

$$\Xi(0) = \varphi_s|_{[P_2,P_4]} \cup \varphi^s|_{[P_2,P_3]}. \tag{26}$$

The behavior of D^* on the remaining portion of the boundary of $\Gamma^B_{2,ii}$ is given in

Theorem 5: $D^*(\sigma^*,v^*)$ is monotonically increasing from zero to $D(0,\gamma_f)$ on $\varphi_s|_{[P_4,P_5]}$ as $P = (\sigma^*,v^*)$ proceeds from P_4 to P_5. $D^*(\sigma^*,v^*)$ is monotonically increasing from zero to $D(0,\gamma_f)$ on $\varphi^s|_{[P_3,P_5]}$ as $P = (\sigma^*,v^*)$ proceeds from P_3 to P_5.

■

The mathematical dificulty in determining $\Xi(d)$ and hence $\Omega(\bar{v})$ for $\bar{v} \in (\bar{v}_1,\bar{v}_2)$ stems from the fact that F cannot be inverted explicitly. Thus the funciton $D^* = D \circ F^{-1}$ must be investigated by implicit means. Let us introduce notation for the derivatives of the functions D and F which are defined for $(\gamma_s,\gamma_R^*) \in T$:

$$D_s = \frac{\partial D}{\partial \gamma_s},\quad D_r = \frac{\partial D}{\partial \gamma_R^*},\quad F_{1,s} = \frac{\partial F_1}{\partial \gamma_s},\quad F_{1,r} = \frac{\partial F_1}{\partial \gamma_R^*},\quad F_{2,s} = \frac{\partial F_2}{\partial \gamma_s},\quad F_{2,r} = \frac{\partial F_2}{\partial \gamma_R^*}. \tag{27}$$

Expressions for these quantities can be found from (7),(8),(21). It thus follows that

$$D_r \geq 0, \quad F_{1,s} = 0, \quad F_{1,r} \geq 0, \quad F_{2,s} \leq 0, \quad F_{2,r} \geq 0, \qquad (\gamma_s, \gamma_R^*) \in \mathbf{T} . \tag{28}$$

The four inequalities in (28) can be shown to be strict for $(\gamma_s, \gamma_R^*) \in \text{interior}(\mathbf{T})$. In addition we introduce notation for the derivative of the function D^* which we recall is defined for $(\sigma^*, v^*) \in \Gamma^B_{2,ii}$:

$$D_v^* = \frac{\partial D^*}{\partial v^*}, \qquad D_\sigma^* = \frac{\partial D^*}{\partial \sigma^*} . \tag{29}$$

Differentiating (7),(8),(21) yields

$$D_v^* \circ \mathbf{F} = \frac{\left[\tau'(\gamma_s)\,(\gamma_R^* - \gamma_s) - (\tau(\gamma_R^*) - \tau(\gamma_s)) \right] \left[A(\gamma_s, \gamma_R^*) - (\tau(\gamma_R^*) - \tau(\gamma_s))(\gamma_R^* - \gamma_s) \right]}{(\gamma_R^* - \gamma_s) \left[(\tau'(\gamma_s)\,(\gamma_R^* - \gamma_s))^{1/2} - (\tau(\gamma_R^*) - \tau(\gamma_s))^{1/2} \right]^2} , \tag{30}$$

while the definition of D^* gives directly

$$D_\sigma^* \circ \mathbf{F} = - \left[(D_v^* \circ \mathbf{F})\, F_{2,r} - D_r \right] / F_{1,r} , \tag{31}$$

for $(\gamma_s, \gamma_R^*) \in \mathbf{T}$. Define

$$\begin{aligned} \varphi_v^- &= \{ (\sigma^*, v^*) \mid (\sigma^*, v^*) \in \Gamma^B_{2,11}, \ D_v^*(\sigma^*, v^*) < 0 \}, \\ \varphi_v^o &= \{ (\sigma^*, v^*) \mid (\sigma^*, v^*) \in \Gamma^B_{2,11}, \ D_v^*(\sigma^*, v^*) = 0 \}, \\ \varphi_v^+ &= \{ (\sigma^*, v^*) \mid (\sigma^*, v^*) \in \Gamma^B_{2,11}, \ D_v^*(\sigma^*, v^*) > 0 \}. \end{aligned} \tag{32}$$

For the purpose of classifying points $\mathbf{P} \in \partial\Gamma^B_{2,11}$, the boundary of $\Gamma^B_{2,11}$, the function D_v^* is defined in terms of limiting values. In a similar fashion, by replacing $D_v^*(\sigma^*, v^*)$ with $D_\sigma^*(\sigma^*, v^*)$ in (32), we define φ_σ^-, φ_σ^o and φ_σ^+.

We now turn to examine the loci φ_v^-, φ_v^o and φ_v^+. It can be shown that the equation

$$A(\mu^s(\gamma), \gamma) - (\tau(\gamma) - \tau(\mu^s(\gamma)))\,(\gamma - \mu^s(\gamma)) = 0, \tag{33}$$

has a unique root obeying $\gamma \in (\gamma_q, \gamma_f)$. This root shall be denoted γ_w. Let

$$\mathbf{P}_6 = \mathbf{F}(\mu^s(\gamma_w), \gamma_w), \qquad \bar{v}_3 = \theta(\mathbf{P}_6), \tag{34}$$

and note that $\mathbf{P}_6 \in \varphi^s|_{(\mathbf{P}_3 - \mathbf{P}_5)}$ and $\bar{v}_1 < \bar{v}_3 < \bar{v}_2$. The main properties of the

loci φ_v^-, φ_v^o and φ_v^+ in the (σ^*,v^*)-plane are now summarized in

Theorem 6: φ_v^o is a finite connected curve with endpoints P_2 and P_6. Between these points it is montonically increasing and confined to interior($\Gamma^B_{2,ii}$). Both φ_v^- and φ_v^+ are simply connected regions. The boundary $\partial\varphi_v^-$ of φ_v^- is given by $\varphi_v^o \cup \varphi^s|_{[P_6,P_5]} \cup \varphi_s|_{[P_2,P_5]}$. The boundary $\partial\varphi_v^+$ of φ_v^+ is given by $\varphi_v^o \cup \varphi^s|_{[P_2,P_6]}$.

■

Uniqueness of Maximally Dissipative Solutions within the one-parameter family of Dynamical State Solutions for $\bar{v} \in [\bar{v}_3\ \bar{v}_2)$

If $\bar{v} \in (\bar{v}_1,\bar{v}_2)$, theorems 1, 4 and 6 indicate that the curve $\Pi(\bar{v})$ will intersect φ_v^o if and only if $\bar{v} \in (\bar{v}_1,\bar{v}_3]$, and moreover that this intersection is then unique. Accordingly we define $P_o(\bar{v}) = \Pi(\bar{v}) \cap \varphi_v^o$ for $\bar{v} \in (\bar{v}_1,\bar{v}_3]$. Thus for $\bar{v} \in (\bar{v}_1,\bar{v}_3)$ the curve $\Pi(\bar{v})$ is contained in φ_v^- between $P_s(\bar{v})$ and $P_o(\bar{v})$, and is contained in φ_v^+ between $P_o(\bar{v})$ and $P^s(\bar{v})$. However for $\bar{v} \in (\bar{v}_3,\bar{v}_2)$ the curve $\Pi(\bar{v})$ is confined to φ_v^- everywhere between $P_s(\bar{v})$ and $P^s(\bar{v})$.

An immediate consequence of (28) and (31) is that

$$\text{interior}(\varphi_v^-) \subset \text{interior}(\varphi_\sigma^+), \tag{35}$$

and

$$\varphi_v^o \cap \varphi_\sigma^o \cap \text{interior}(\Gamma^B_{2,ii}) = \emptyset. \tag{36}$$

The latter result precludes the possibility of any locus $\Xi(d)$ containing an isolated point or a limit point within interior($\Gamma^B_{2,ii}$). These results, in conjunction with Theorem 5, can be shown to yield

Theorem 7: For $0 < d < D(0,\gamma_f)$, each locus $\Xi(d)$ consists of a finite curve with one endpoint on $\varphi_s|_{(P_4,P_5]}$ and the other endpoint on $\varphi^s|_{(P_3,P_5]}$. Moreover

$$\Gamma^B_{2,ii} = \bigcup_{d \in [0,D(0,\gamma_f)]} \Xi(d) \quad . \tag{37}$$

■

It is convenient to define a slope function ν in the (σ^*,v^*)-plane as follows. If ξ is any curve in the (σ^*,v^*)-plane and if $P \in \xi$, then the slope $\frac{dv^*}{d\sigma^*}$ of the curve ξ at P will be denoted by $\nu(\xi,P)$. In particular if $P \in \Xi(d)$ for some $d \in (0,D(0,\gamma_f))$ then

$$\nu(\Xi(d),P) = -\frac{D^*_\sigma}{D^*_v} \quad . \tag{38}$$

Hence (35) indicates that the curves $\Xi(d)$ are monotonically increasing in the region φ^-_v. The following theorem can be established:

Theorem 8:
$$\begin{aligned} &\Omega(\bar{v}) = P^s(\bar{v}), && \text{for } \bar{v} \in [\bar{v}_3,\bar{v}_2), \\ &\Omega(\bar{v}) \subset \varphi^+_\sigma \cap \text{interior}(\varphi^+_v), && \text{for } \bar{v} \in (\bar{v}_1,\bar{v}_3). \end{aligned} \tag{39}$$

■

Theorem 8 indicates that *maximally dissipative solutions are unique for* $\bar{v} \in [\bar{v}_3,\bar{v}_2)$ and that *nonuniqueness of maximally dissipative solutions remains at issue only for* $\bar{v} \in (\bar{v}_1,\bar{v}_3)$.

On the uniqueness of Maximally Dissipative Dynamical Solutions for $\bar{v} \in (\bar{v}_1,\bar{v}_3)$

In view of theorem 8 it follows that

$$\begin{aligned} &\nu(\Xi(d),P) = \nu(\Pi(\bar{v}),P), \qquad d = D^*(\sigma^*,v^*), \\ &\text{if } P = (\sigma^*,v^*) \in \Omega(\bar{v}) \text{ and } \bar{v} \in (\bar{v}_1,\bar{v}_3). \end{aligned} \tag{40}$$

Conditions (40) are necessary, but not sufficient, to establish that a particular point $P = (\sigma^*,v^*)$ is a maximally dissipative dynamical solution when $\bar{v} \in (\bar{v}_1,\bar{v}_3)$. For this range of impact velocity, one must in addition examine the convexity of the curves $\Xi(d)$ in the (σ^*,v^*)-plane. To this end we

shall define a "curvature" function $\nu\nu$ in the obvious way. Namely if ξ is any curve in the (σ^*,v^*)-plane and if $P \in \xi$, then the "curvature" $\frac{d^2v^*}{d\sigma^{*2}}$ of the curve ξ at P will be denoted by $\nu\nu(\xi,P)$. It can be shown that points P obeying (40) will be unique if condition (Q1) is satisfied where

Q1) $\quad \nu\nu(\Xi(d),P) \geq 0, \quad d = D^*(\sigma^*,v^*),$

for all $P = (\sigma^*,v^*) \in \varphi_\sigma^+ \cap \text{interior}(\varphi_v^+)$.

Thus (Q1) is sufficient to ensure the uniqueness of maximally dissipative dynamical state solutions to the impact problem for the range $\bar{v} \in (\bar{v}_1,\bar{v}_3)$, and hence for the full range of impact velocities $\bar{v} > 0$.

Determining conditions that are necessary and sufficient for (Q1) to hold has remained an elusive goal. Accordingly we shall here introduce two additional constitutive hypotheses on the stress response for the target material:

HT6) $\quad \tau'(\gamma_b) \geq \tau'(0),$

HT7) $\quad \tau''(\gamma) = 0 \quad$ for $\gamma \in (\gamma_b,\gamma_w)$.

The calculation of $\nu\nu(\Xi(d),P)$ followed by a fairly elaborate analysis can now be shown to yield

Theorem 9: The additional constitutive hypotheses (HT6) and (HT7) ensure (Q1). ■

Hence the constitutive hypotheses (HI1) for the impactor, and (HT1)-(HT7) for the target are sufficient to ensure that the impact problem studied here has a unique maximally dissipative solution for all impact velocities $\bar{v} > 0$.

References

1. D.E. Grady, R.E. Hollenbach and K.W. Schuler, Compression wave studies in calcite rock, J Geophys. Res. 83 (1978) 2839-2849.

2. G.E. Duvall and R.A. Grahm, Phase Transitions under shock wave loading, Reviews of Modern Physics 49 (1977) 523-579.

3. J.R. Asay and G.I. Kerley, The response of materials to dynamic loading, Int. J. Impact Engng 5 (1987) 69-99.

4. L. Davison and R.A. Grahm, Shock compression of solids, Physics Reports (Review Section of Physics Letters) 55 (1979) 255-379.

5. K. Mukherjee, T.H. Kim and W.T. Walter, Shock deformation and microstructural effects associated with pulse laser-induced damage in metals, in Lasers in Metallurgy, conference proceedings of the Metallurgical Society of AIME, 110th annual meeting, Feb. 22-26, 1981, K. Mukherjee and J. Mazumder eds., 137-150.

6. R.D. James, The propagation of phase boundaries in elastic bars, Arch. Rational Mech. Anal. 73 (1980) 125-150.

7. T.J. Pence, On the emergence and propagation of a phase boundary in an elastic bar with a suddenly applied end load, J. Elasticity 16 (1986) 3-42.

8. T.J. Pence, Formulation and analysis of a functional equation describing a moving one-dimensional elastic phase boundary, Quart. Appl. Math. 45 (1987) 293-304.

9. J. Serrin, Phase transitions and interfacial layers for van der Waals fluids, Proceedings of SAFA IV Conference, Recent Methods in Nonlinear Analysis and Applications, Naples, March 21-28, 1980, A. Canfora, S. Rionero, C. Sbordone, C. Trombetti, eds.

10. M. Slemrod, Admissibility criteria for propagating phase boundaries in a van der Waals fluid, Arch. Rational Mech. Anal 82 (1983) 301-315.

11. R. Hagan and M. Slemrod, The viscosity-capillarity criterion for shocks and phase transitions, Arch. Rational Mech. Anal. 83 (1983) 333-361.

12. H. Hattori, The Riemann problem for a van der Waals fluid with entropy rate admissibility criterion - isothermal case, Arch. Rational Mech. Anal. 92 (1986) 247-263.

13. M. Shearer, Nonuniqueness of admissible solutions of Riemann initial value problems for a system of conservation laws of mixed type, Arch. Rational Mech. and Anal. 93 (1986) 45-59.

14. M. Shearer, D.G. Schaeffer, D. Marchesin & P.L. Paes-Leme, Solution of the Riemann problem for a prototype 2 x 2 system of non-strictly hyperbolic conservation laws, Arch. Rational Mech. and Anal. 97 (1987) 299-320.

15. M. Shearer, Dynamic phase transitions in a van der Waals gas, Q. Appl. Math 4, (1988) 631-636.

16. C. Dafermos, The entropy rate admissibility criterion for solutions of hyperbolic conservation laws, J. Differential Equations 14 (1973) 202-212.

17. T.J. Pence, On the mechanical dissipation of solutions to the Riemann problem for impact involving a two-phase elastic material, submitted.

An Inverse Riemann Problem for Impact involving Phase Transitions*

Thomas J. Pence

Dept. of Metallurgy, Mechanics and Materials Science
Michigan State University
East Lansing, MI 48824-1226

Introduction

Phase transitions in solids can be modelled in a continuum mechanical framework through the consideration of nonlinearly elastic materials whose strain energy density is a nonconvex function of strain. Different regions in strain space can then be identified with different material phases. For such materials, displacement gradients can, in certain situations, become discontinuous across internal surfaces. These surfaces are viewed as phase boundaries whenever they separate regions involving strains corresponding to different material phases. In an earlier paper which also appears in these proceedings [1],** the Riemann problem for impact has been investigated for a case of impact between a target composed of a two phase material and an impactor composed of a single phase material. As is well known, solutions to Riemann problems involving phase transitions lead to nonunique solutions even after imposition of admissibility conditions due to Lax; and the problem under study here is no exception. A methodology is outlined in [1] for determining the families of solutions associated with any such Riemann problem; it involves mapping candidate dynamical states for both impactor and target to a (σ^*, v^*)-plane of contact interface stress vs. contact interface velocity. Each intersection of the contact locus for the impactor with the contact locus for the target corresponds to a solution of the impact problem. For the materials under study in [1], the target locus consists of a region from which emanate two curves, while the impactor locus consists of only a curve. There are two features of this methodology that are quite useful: the technique is global so that assumptions on the size of the data characterizing the Riemann problem are not necessary, and admissibility criteria for selecting from among the possibly nonunique solutions can be studied within this context.

The admissiblility criterion adopted in [1], and to be considered here as well, is the *maximum dissipation (rate) criterion;* this criterion was first introduced by Dafermos [2] under the different name of the entropy rate criterion. By investigating the dissipation topography associated with the target locus, conditions on the strain energy density of the two phase target material have been found which ensure that there is exactly one maximally dissipative solution to the Riemann problem for impact with any single phase impactor material at any impact velocity $\bar{v}$.

This gives rise to an inverse problem that can be stated roughly as follows: "Suppose that a target is composed of such a two phase material. It is desired to determine whether a given candidate dynamical state in the associated contact locus can be generated under the the maximum dissipation criterion by judiciously choosing a simple single phase material for the impactor as well as the impact velocity $\bar{v}$." This type of *inverse Riemann problem*, which bears directly on the extent to which dynamical states in phase transforming materials can be controlled, is the focus of this paper. It will be shown that only a certain subfamily of candidate dynamical states for the

*Supported by the U.S. Army Research Office under DAAL03-89-G-0089.
**See page 817.

target (the maximal dynamical states and the semimaximal dynamical states) can be generated in this fashion. Finally, the extent to which whole families of dynamical states can be attained as the impact velocity $\bar{v}$ is varied will be addressed.

For the sake of expediency, this paper will be regarded as a continuation of [1]. Consequently, notation and results will be carried over from [1] without further elaboration. The reader is also urged to consult [1] for a list of associated references. In addition, equations appearing in [1] shall be refered to directly by the equation number followed by a superscript #, e.g. $(12)^{\#}$ is to be read as (12) in [1].

For ease of reference, we shall define a *simple single phase material* to be one for which $\zeta(\gamma)$ obeys $(HI1)^{\#}$; we shall also define a *simple two phase material* to be one for which $\tau(\gamma)$ obeys $(HT1)^{\#}$-$(HT5)^{\#}$. In [1] it is shown that if a simple two phase target material gives rise to a contact locus Γ_2 such that $(Q1)^{\#}$ holds, then the Riemann problem for impact will have a unique maximally dissipative solution (modulo solutions with stationary phase boundary) irregardless of the choice of simple single phase impactor material and irregardless as well of the impact velocity $\bar{v}$. Let us thus also agree to define an *elementary two phase material* to be a simple two phase material for which the associated Γ_2 obeys $(Q1)^{\#}$. A *non-elementary* simple two phase material is then, of course, a simple two phase material for which the associated Γ_2 does not obey $(Q1)^{\#}$. As shown in [1], $(HT6)^{\#}$ and $(HT7)^{\#}$ are sufficient conditions to guarentee that a simple two-phase material is in fact elementary. However, more general results regarding necessary and sufficient conditions to distinguish between elementary and non-elementary simple two phase materials have yet to be determined; in fact the existence of non-elementary simple two phase materials remains an open question.

Definition of the Inverse Riemann Problem

The purpose of the present work is to explore the following Inverse Riemann Problem (IRP):

Given a simple two phase target material, choose a candidate dynamical state for the target (i.e. a point $\mathbf{P}_o = (\sigma_o^*, v_o^*) \in \Gamma_2$*), then we seek to determine whether this dynamical state in the target can be generated as a (maximally dissipative) solution to the Riemann problem for impact by judicious choice of either* (IRP)

(C1) simple single phase impactor material $\zeta(\gamma)$*,*

and/or

(C2) impact velocity $\bar{v}$*?*

For the remainder of this paper unless otherwise noted, the term *"generate"* is to be understood in the above sense, namely to *"generate using the maximum dissipation criterion in the event of multiple solutions."* If the simple two phase material is elementary, then the phrase *"a* (maximally dissipative) solution" in (IRP) may be replaced by *"the* (maximally dissipative) solution".

The set of all points $\mathbf{P}$ which can be so generated by judicious choice of either (C1) or (C2) will be said to constitute *the well-posed set for (IRP)*; this set will be denoted Γ_2^{wp}. Two questions are now immediate:

(Q1) Can the set Γ_2^{wp} be determined precisely?

(Q2) Given $\mathbf{P}_o \in \Gamma_2^{wp}$, what is the totality of choices for (C1) and (C2) that succeed in generating $\mathbf{P}_o$?

To discuss these issues, imagine for the moment that one is presented with both a simple two phase target material *and* a simple single phase impactor material. Let $\hat{\Omega}$ be the locus of points generated in the (σ^*, v^*)-plane as the impact velocity $\bar{v}$ varies from 0 to ∞, hence

$$\hat{\Omega} = \Omega(\bar{v})\big|_{\bar{v} \in (0,\bar{v}_1]} \cup \Omega(\bar{v})\big|_{\bar{v} \in (\bar{v}_1,\bar{v}_3)} \cup \Omega(\bar{v})\big|_{\bar{v} \in [\bar{v}_3,\bar{v}_2]} \cup \Omega(\bar{v})\big|_{\bar{v} \in (\bar{v}_2,\infty)}, \tag{1}$$

where $\bar{v}_1$, $\bar{v}_3$ and $\bar{v}_2$ depend upon both the simple two phase target material and the simple single phase impactor material (c.f. $(24)^{\#}$, $(34)^{\#}$). Then

$$\begin{aligned}
&\Omega(\bar{v})\big|_{\bar{v} \in (0,\bar{v}_1]} = \Gamma_{2,i}\big|_{(\mathbf{0},\mathbf{P}_2]}, \\
&\Omega(\bar{v})\big|_{\bar{v} \in (\bar{v}_1,\bar{v}_3)} \subset \text{interior}(\boldsymbol{\varphi}_\sigma^+ \cap \boldsymbol{\varphi}_v^+), \\
&\Omega(\bar{v})\big|_{\bar{v} \in [\bar{v}_3,\bar{v}_2]} = \boldsymbol{\varphi}^s\big|_{[\mathbf{P}_6,\mathbf{P}_5]}, \\
&\Omega(\bar{v})\big|_{\bar{v} \in (\bar{v}_2,\infty)} = \Gamma_{2,ii}^C.
\end{aligned} \tag{2}$$

It is to be noted that $(2)_2$ is a slightly stronger result than $(39)_2^{\#}$. To verify $(2)_2$ observe that

$$\partial(\boldsymbol{\varphi}_\sigma^+ \cap \boldsymbol{\varphi}_v^+) = \boldsymbol{\varphi}_\sigma^o \cup \boldsymbol{\varphi}_v^o = \text{interior}(\boldsymbol{\varphi}_\sigma^o) \cup \mathbf{P}_2 \cup \text{interior}(\boldsymbol{\varphi}_\sigma^o) \cup \mathbf{P}_6, \tag{3}$$

since (see Figure 1)

$$\boldsymbol{\varphi}_\sigma^o \cap \boldsymbol{\varphi}_v^o = \partial\boldsymbol{\varphi}_\sigma^o = \partial\boldsymbol{\varphi}_v^o = \mathbf{P}_2 \cup \mathbf{P}_6. \tag{4}$$

Now, by virtue of $(38)^{\#}$, the constant dissipation loci $\Xi(\cdot)$ possess horizontal tangents on interior$(\boldsymbol{\varphi}_\sigma^o)$ and possess vertical tangents on interior$(\boldsymbol{\varphi}_v^o)$. Since Theorem 1 of [1] indicates that the slope of Γ_1 is restricted to finite negative values it follows from $(40)^{\#}$ and the definition of $\mathbb{I}(\bar{v})$ that candidate dynamical states for the target corresponding to points in either

interior(φ_σ^o) or interior(φ_v^o) cannot be associated with a maximally dissipative solution for the Riemann problem under study.

It follows from (2) that the three subloci $\Omega(\bar{v})\big|_{\bar{v} \in (0,\bar{v}_1]}$, $\Omega(\bar{v})\big|_{\bar{v} \in [\bar{v}_3,\bar{v}_2]}$, and $\Omega(\bar{v})\big|_{\bar{v} \in (\bar{v}_2,\infty)}$ are each a connected curve. In the event that the simple two phase material is elementary, then the construction utilizing (40)# which determines the sublocus $\Omega(\bar{v})\big|_{\bar{v} \in (\bar{v}_1,\bar{v}_3)}$ ensures that this sublocus is also a connected curve initiating at P_2 and terminiating at P_6. Thus for an elementary two phase material the complete locus $\hat{\Omega}$ is a connected curve (Figure 1). We shall comment later on the possibilities for $\Omega(\bar{v})\big|_{\bar{v} \in (\bar{v}_1,\bar{v}_3)}$ in the event that the simple two phase material is not elementary.

It is enlightening to consider how the locus $\hat{\Omega}$ changes as one varies the associated simple single phase impactor material, while retaining the same simple two phase target material. Then the values of $\bar{v}_1$, $\bar{v}_3$ and $\bar{v}_2$ will in general change, whereas the right hand side of (2) remains the same. Thus the dynamical states associated with the three subloci $\Omega(\bar{v})\big|_{\bar{v} \in (0,\bar{v}_1]}$, $\Omega(\bar{v})\big|_{\bar{v} \in [\bar{v}_3,\bar{v}_2]}$, and $\Omega(\bar{v})\big|_{\bar{v} \in (\bar{v}_2,\infty)}$ will remain the same, eventhough the value of impact velocity corresponding to each individual dynamical state in the above list of sets would in general be expected to change. In contrast, for $\Omega(\bar{v})\big|_{\bar{v} \in (\bar{v}_1,\bar{v}_3)}$ the dynamical states themselves will in general change. Hence if the simple two phase target material is elementary, then different simple single phase impactor materials would be expected to give rise to different curves connecting P_2 to P_6.

Maximal, Semi-maximal and Non-maximal Dynamical States

In view of the above discussion it is useful to partition Γ_2 into the following three distinct sets:

$$\Gamma_2^{max} = \Gamma_{2,i}\big|_{(0,P_2]} \cup \varphi^s\big|_{[P_6,P_5]} \cup \Gamma_{2,ii}^C,$$

$$\Gamma_2^{sem} = \text{interior}(\varphi_\sigma^+ \cap \varphi_v^+), \tag{5}$$

$$\Gamma_2^{non} = \Gamma_2 - \{ \Gamma_2^{max} \cup \Gamma_2^{sem} \}.$$

It then follows immediately from the considerations given above that

$$\Gamma_2^{max} \subset \Gamma_2^{wp}, \qquad \Gamma_2^{non} \not\subset \Gamma_2^{wp}. \tag{6}$$

Moreover each $P_o \in \Gamma_2^{max}$, is generated for *every* simple single phase impactor material provided that the impact velocity for a given simple single phase impactor material is chosen to be $\bar{v} = \theta(P_o)$ (c.f. $(24)^{\#}$). For this reason, points $P \in \Gamma_2^{max}$ will be said to be *maximal dynamical states* in the sense that (IRP) has a solution for all such points irregardless of the choice of simple single phase impactor material. On the other hand, points $P \in \Gamma_2^{non}$ will be said to be *non-maximal dynamical states* in the sense that (IRP) has no solutions for all such points.

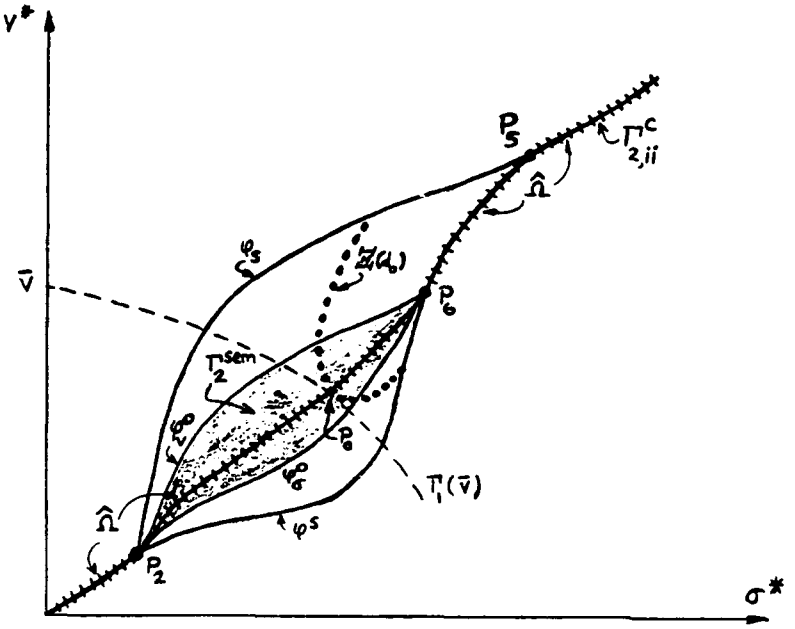


Figure 1: The connected curve $\hat{\Omega}$ for an elementary two phase target material. The determination of $\hat{\Omega}$ at $P_o \in \Gamma_2^{sem}$ by the osculation condition $(40)^{\#}$ is indicated.

The remaining locus Γ_2^{sem} will be said to comprise the *semi-maximal dynamical states*. Notice from (5) and $(38)^{\#}$ that Γ_2^{sem} consists of points in Γ_2 through which the associated constant dissipation locus $\Xi(\cdot)$ has negative slope. One may conclude from the previous discussion that at least some points in Γ_2^{sem} are also contained in Γ_2^{wp}, but that any such points in $\Gamma_2^{sem} \cap \Gamma_2^{wp}$ will certainly not be generated for every choice of simple single phase impactor material. Now according to $(40)^{\#}$, a necessary condition for a point $P_o \in \Gamma_2^{sem}$ to be generated as a solution to the Riemann problem for a given simple single phase material and impact velocity $\bar{v}$ is that $\Pi(\bar{v})$ oscullate the associated constant dissipation locus at P_o (Figure 1). Furthermore, this condition is sufficient provided that the simple two phase material is in fact elementary. Thus in order to determine whether or not a point $P_o \in \Gamma_2^{sem}$ is contained in Γ_2^{wp}, it is necessary to determine the extent to which $\Pi(\bar{v})$ and hence $\Gamma_1 = \Gamma_1(\bar{v})$ can be made, through selection of the stress response function $\zeta(\gamma)$ and impact velocity $\bar{v}$, to assume a given form in the (σ^*, v^*)-plane. Now Theorem 1 of [1] serves to impose certain limitations on any such form. The following theorem, which is a converse to Theorem 1 of [1], shows that *any* form obeying the restrictions set forth in Theorem 1 of [1] is, in fact, attainable.

Theorem 1 - On the existence of a simple single phase material with a given contact locus form:

Let $I(\cdot):[0,\infty)\to[0,-\infty)$ obey $I(0)=0$, $I'(\cdot)<0$, $I''(\cdot)<0$. For $\bar{v} > 0$, let $\Psi(\bar{v}) = \{(\sigma^*,v^*) \mid v^* = \bar{v} + I(\sigma^*),\ \sigma^*\geq 0\}$. Then there exists a stress response function $\zeta(\gamma)$ corresponding to a simple single phase material such that the associated $\Gamma_1(\bar{v}) = \Psi(\bar{v})$.

Proof. For $s \geq 0$ define

$$H(s) = \int_0^s (I'(z))^2 dz, \tag{7}$$

so that $H(0)=0$ and $H'(s)>0$. Thus $H(s)$ is invertible with inverse function $H^{(-1)}(\cdot)$ obeying $H^{(-1)}(0) = 0$. Let $\zeta(\gamma) = H^{(-1)}(\gamma)$ for $\gamma\geq 0$. Then one may verify directly from (7) and the properties of $I(\cdot)$ that $(HI)^{\#}$ holds. In addition, since

$$\begin{aligned} &-\int_0^{\gamma} (H^{(-1)\prime}(z))^{\frac{1}{2}}\, dz = -\int_0^{\gamma} (H'(H^{(-1)}(z)))^{-\frac{1}{2}}\, dz \\ &= -\int_0^{H^{(-1)}(\gamma)} (H'(y))^{\frac{1}{2}}\, dy = \int_0^{H^{(-1)}(\gamma)} I'(y)\, dy = I(H^{(-1)}(\gamma)), \end{aligned} \tag{8}$$

the identification $\Gamma_1(\bar{v}) = \Psi(\bar{v})$ follows from $(12)^{\#}$.

■

Fix now a point $\mathbf{P}_o = (\sigma_o^*, v_o^*) \in \Gamma_2^{sem}$, then by taking any function $I(\cdot):[0,\infty)\to[0,-\infty)$ such that

$$I(0)=0,\quad I'(\cdot)<0,\quad I''(\cdot)<0,\quad I'(\sigma_o^*) = \nu(\Xi(d_o),\mathbf{P}_o),\quad d_o = D^*(\sigma_o^*,v_o^*), \tag{9}$$

it follows that $\zeta(\gamma)$ constructed on the basis of Theorem 1 will lead to the satisfaction of $(40)^{\#}$ provided that the impact velocity is then taken to be $\Theta(\mathbf{P}_o)$. (Notice that an infinity of functions $I(\cdot)$ exist which satisfy (9).) Consequently it may be concluded that

$$\Gamma_2^{sem} \subset \Gamma_2^{wp}, \tag{10}$$

provided that the simple two phase target material is an elementary two-phase material. For a non-elementary two phase target material we may only conclude that the necessary condition $(40)^{\#}$ can be met.

Prescribing the Impact Velocity

Until further notice, let us agree to confine attention to elementary two phase target materials. For $\mathbf{P}_o = (\sigma_o^*, v_o^*) \in \Gamma_2^{sem}$, we shall define

$$\bar{v}_m = v_o^*, \qquad \bar{v}_M = v_o^* - \sigma_o^* \times \nu(\Xi(d_o), \mathbf{P}_o), \qquad d_o = D^*(\sigma_o^*, v_o^*). \tag{11}$$

It then follows from the convexity of the loci Γ_1 in conjunction with $(40)^\#$ that if $\mathbf{P}_o$ is generated as the solution to (IRP), then the impact velocity $\bar{v}$ must obey

$$\bar{v}_m < \bar{v} < \bar{v}_M \ . \tag{12}$$

Furthermore, Theorem 1 ensures that for any $\bar{v}$ obeying (12) there will exist an infinity of simple single phase impactor materials that will generate $\mathbf{P}_o$ as the solution to the impact problem with impact velocity $\bar{v}$. A simple single phase material stress response $\zeta(\gamma)$ which generates $\mathbf{P}_o$ at an impact velocity near $\bar{v}_m$ must then have a locus Γ_1 that is essentially horizontal until just before σ_o^* where it must radically bend in order meet $(40)^\#$. Conversely, a simple single phase material stress response $\zeta(\gamma)$ which generates $\mathbf{P}_o$ at an impact velocity near $\bar{v}_M$ must have a locus Γ_1 that is essentially linear from $(\sigma^*, v^*) = (0, \bar{v})$ to $(\sigma^*, v^*) = (\sigma_o^*, v_o^*)$ (see Figure 2). According to $(12)^\#$, the absolute value of the slope $\nu(\Gamma_1, (\sigma^*, v^*))$ goes like the reciprocol of the square root of the material stiffness at stress value σ^*. Consequently, generating $\mathbf{P}_o$ near the lower bound impact velocity $\bar{v}_m$ requires a simple single phase impactor material with a very high stiffness (i.e. essentially rigid response) for all stresses ζ obeying $0 \le \zeta < \sigma_o^*$. On the other hand, generating $\mathbf{P}_o$ near the upper bound impact velocity $\bar{v}_M$ requires a material with an essentially linear stress response for all stresses ζ obeying $0 \le \zeta \le \sigma_o^*$, where the stiffness is determined by the slope of the constant dissipation locus at $\mathbf{P}_o$.

One should note that the absence of an oscullation condition like $(40)^\#$ for dynamical states in $\mathbf{P}_o = (\sigma_o^*, v_o^*) \in \Gamma_2^{max}$, indicates that the associated range of impact velocities $\bar{v}$ that succeed in generating $\mathbf{P}_o \in \Gamma_2^{max}$ (by suitable choice of simple single phase impactor material) is simply $\bar{v} > v_o^*$.

If we monitor the values of $\bar{v}_m$ and $\bar{v}_M$ as P_o roams within Γ_2^{sem}, one may then conclude from (11) and (38)# that

$$\begin{aligned} &\bar{v}_M \to \bar{v}_m, && \text{if } P_o \to P \in \varphi_\sigma^o \ , \\ &\bar{v}_M \to \infty, && \text{if } P_o \to P \in \varphi_v^o \ , \end{aligned} \tag{13}$$

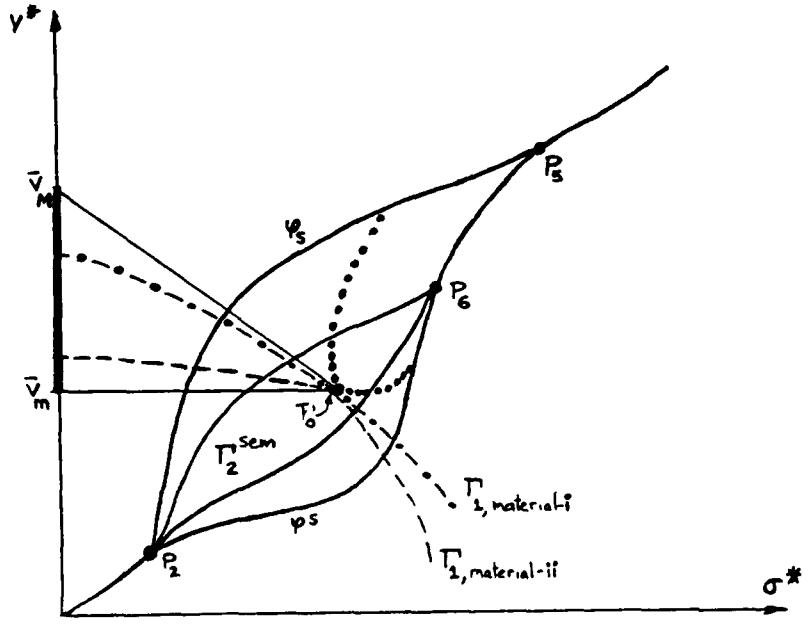


Figure 2: The window of impact velocities (12) for $P_o = (\sigma_o, v_o) \in \Gamma_2^{sem}$. Also shown are contact loci for two different simple single phase impactor materials that generate this P_o. In both cases the oscullation condition (40)# must hold. Here material-i is more compliant than material-ii at stresses less than σ_o and so requires a greater impact velocity to generate P_o.

Let $P_2 = (\sigma_2^*, v_2^*)$, $P_5 = (\sigma_5^*, v_5^*)$, so that $\sigma_2^* = r(\gamma_b)$ and $\sigma_5^* = r(\gamma_f)$ (c.f. (14)# and (15)#). Suppose that σ^* obeys $\sigma_2^* < \sigma^* < \sigma_5^*$. It can then be shown that if one travels from φ^s to φ_s along the vertical line segment associated with the value σ^*, then the corresponding dynamical states involve phase boundaries with progressively greater phase boundary velocities. Thus, in this sense, points P_o near φ_σ^o correspond to dynamical states with a relatively low phase boundary velocity while points P_o near φ_v^o correspond to dynamical states with a relatively high phase boundary velocity. It is thus interesting to note that the window of possible impact velocities (12) is very small for dynamical states of the former type, and that generating these dynamical states neccesitates the use of simple single phase impactor materials that are essentially rigid.

The Riemann Problem for Target Materials that are not Elementary

We now turn to consider in more detail certain aspects of the Riemann problem for simple two phase target materials that are not elementary. Perhaps the most interesting aspect of such problems is that the possibility exists that the locus $\Omega(\bar{v})\big|_{\bar{v} \in (\bar{v}_1, \bar{v}_3)}$ will not be connected! To see this, let d_o be a value for the dissipation such that the constant dissipation locus $\Xi(d_o)$ intersects Γ_2^{sem}. Assume that a contact locus Γ_1 associated with impact velocity $\bar{v}_a$ oscullates $\Xi(d_o)$ at a point, say P_a, at which $\Xi(d_o)$ has positive

curvature. This then assures that the dynamical state associated with $\mathbf{P}_a$ provides a local maximum to the dissipation on the (pre-criterion) solution locus $\mathbf{\Pi}(\bar{\mathbf{v}}_a)$. The question then arises as to whether or not $\mathbf{\Pi}(\bar{\mathbf{v}}_a)$ might also oscullate $\Xi(d_o)$ at some other point, say $\mathbf{P}_{aa}$, at which $\Xi(d_o)$ also has positive curvature. Of course the convexity condition on Γ_1 embodied in Theorem 1 of [1] indicates that this is not possible unless $\nu\nu(\Xi(d_o),\mathbf{P}_o) < 0$ for some point $\mathbf{P}_o \in \Xi(d_o)$ between $\mathbf{P}_a$ and $\mathbf{P}_{aa}$. Now, by definition, a non-elementary simple two phase material is one for which there exists a dissipation value d_o and a point $\mathbf{P}_o \in \Gamma_2^{sem} \cap \Xi(d_o)$ such that the anomolous convexity condition $\nu\nu(\Xi(d_o),\mathbf{P}_o) < 0$ is met. In fact, given such a d_o and $\mathbf{P}_o$, Theorem 1 suggests that a simple single phase impactor material exists such that the associated Γ_1 oscullates two points $\mathbf{P}_a$ and $\mathbf{P}_{aa}$ in the manner described above (Figure 3).

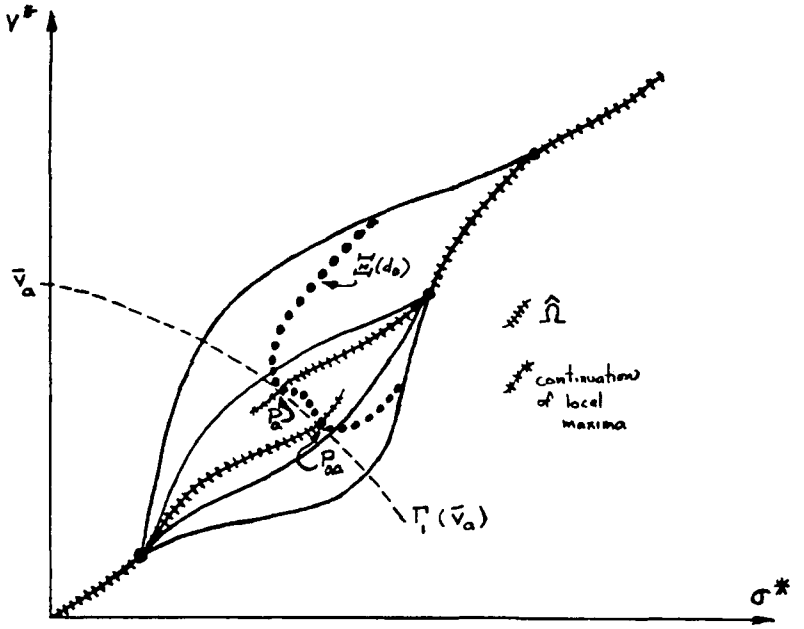


Figure 3: The anomolous convexity of the dissipation loci $\Xi(\cdot)$ for a non-elementary two phase target material may give rise to a locus $\hat{\Omega}$ that is disconnected. The continuations represent dynamical states that are locally maximally dissipative on $\Gamma_1(\bar{v})$ but not globally maximally dissipative.

Consequently, let us assume that we have identified a simple single phase impactor material $\zeta(\gamma)$ and an impact velocity $\bar{v}_a$, such that the above scenario obtains. Then the dynamical states associated with $\mathbf{P}_a$ and $\mathbf{P}_{aa}$ provide local maxima to the dissipation on the solution locus $\mathbf{\Pi}(\bar{\mathbf{v}}_a)$. In fact, these maxima each have the same dissipation value d_o. If, as may well be the case, these local maxima are in fact global maxima, then the dynamical states associated with $\mathbf{P}_a$ and $\mathbf{P}_{aa}$ each provide a maximally dissipative solution to the impact problem at impact velocity $\bar{v}_a$. Now within the above scenario the generic case would have the following features:

(i) $\mathbf{P}_a$ and $\mathbf{P}_{aa}$ will provide the only oscullation points of $\mathbf{\Pi}(\bar{\mathbf{v}}_a)$ with $\Xi(d_o)$,

(ii) as $\bar{v}$ departs from $\bar{v}_a$, the local maxima that "evolve" from $\mathbf{P}_a$ and $\mathbf{P}_{aa}$ will not have identical dissipation values.

Hence the generic case is that the impact problem will have a unique maximally dissipative solution for each impact velocity near $\bar{v}_a$, with the special impact velocity $\bar{v}_a$ giving rise to two maximally dissipative solutions because it induces an exchange in global maxima of dissipation on the $\bar{v}$-parametrized solution sets $\mathbf{\Pi}(\bar{v})$. For the situation that we that we have just described, the locus $\mathbf{\Omega}(v)\big|_{\bar{v} \in (\bar{v}_1, \bar{v}_3)}$ will be discontinuous at $\bar{v} = \bar{v}_a$ (Figure 3). Furthermore, if a norm is placed upon dynamical state solutions to Riemann problems, then this norm is also likely to suffer a discontinuity on $\mathbf{\Omega}(v)\big|_{\bar{v} \in (\bar{v}_1, \bar{v}_3)}$ at the impact velocity $\bar{v}_a$.

Extended versions of the Inverse Riemann Problem

In view of the preceding discussion, *we shall henceforth confine attention to elementary two phase target materials so that the locus* $\hat{\mathbf{\Omega}}$ *is guarenteed to be connected.* Then the inverse Riemann problem (IPP) can be regarded as determining whether or not certain choices for the simple single phase impactor material (C1) will result in a given point $\mathbf{P}_o \in \mathbf{\Gamma}_2$ being contained in the associated connected curve $\hat{\mathbf{\Omega}}$. If such is indeed the case, then the associated impact velocity is necessarily given by $\bar{v} = \theta(\mathbf{P}_o)$. As already shown, the choice of (C1) is crucial only if $\mathbf{P}_o \in \mathbf{\Gamma}_2^{sem}$.

It is thus natural to consider an extended inverse problem of the following type: given $\underline{n}$ distinct points in $\mathbf{\Gamma}_2^{sem}$, is it possible to select a simple single phase material such that the associated $\hat{\mathbf{\Omega}}$ will contain all n points? In fact, one can enquire into the general extent to which the curve $\hat{\mathbf{\Omega}}$ can be prescribed within the region $\mathbf{\Gamma}_2^{sem}$.

Let us agree to denote the inverse Riemann problem in which n points in in $\mathbf{\Gamma}_2^{sem}$ are prescribed as a problem of type (IRP-n). It has already been shown that a problem of type (IRP-1) is always well posed. For n>1 there do, of course, exist problems of type (IRP-n) that are also well posed (e.g. choose n points on the $\hat{\mathbf{\Omega}}$ associated with some fixed $\zeta(\gamma)$). Conversely, for any n>1, there exist problems of type (IRP-n) that are *not* well posed. To verify this let us suppose that two points in the prescription of such a problem are $\mathbf{P}_o=(\sigma_o^*, v_o^*)$, $\mathbf{P}_{oo}=(\sigma_{oo}^*, v_{oo}^*) \in \mathbf{\Gamma}_2^{sem}$. Let us first of all assume that $\sigma_{oo}^* > \sigma_o^*$. It then follows from $(40)^{\#}$ and Theorem 1 of [1] that a necessary condition for the problem to be well posed is that

$$\nu(\Xi(d_{oo}), \mathbf{P}_{oo}) < \nu(\Xi(d_o), \mathbf{P}_o), \quad d_{oo} = D^*(\sigma_{oo}^*, v_{oo}^*), \quad d_o = D^*(\sigma_o^*, v_o^*). \tag{14}$$

Secondly, if we assume that $\sigma_{oo}^* = \sigma_o^*$, then (14) is to be replaced by

$$\nu(\Xi(d_{oo}), \mathbf{P}_{oo}) = \nu(\Xi(d_o), \mathbf{P}_o), \quad d_{oo} = D^*(\sigma_{oo}^*, v_{oo}^*), \quad d_o = D^*(\sigma_o^*, v_o^*), \tag{15}$$

as a necessary condition for the well-posedness of (IRP-n). Since it is certainly possible to chose points $\mathbf{P}_o$ and $\mathbf{P}_{oo}$ violating either (14) or (15), this establishes the existence of problems of type (IRP-n) that are not well posed for each $n \geq 2$.

Clearly (15) is also a severe restriction on the possibility that curves $\hat{\mathbf{\Omega}}$ are not mapped bijectively to the σ^*-axis. Letting $\mathbf{P}_{oo}$ approach $\mathbf{P}_o$ in (15) one may obtain the following necessary condition for a point $\mathbf{P}_o \in \Gamma_2^{sem}$ to be a location of vertical tangency for the curve $\hat{\mathbf{\Omega}}$:

$$D^*_{vv}\, D^*_{\sigma} - D^*_{\sigma v}\, D^*_{v} = 0, \tag{16}$$

where D^*_{vv}, $D^*_{\sigma v}$ (and $D^*_{\sigma\sigma}$) are second derivitives of $D^*(\sigma^*, v^*)$ defined analogously to D^*_v and D^*_σ (c.f. $(29)^{\#}$). It shall be shown shortly that (16) is also a sufficient condition for any smooth curve $\hat{\mathbf{\Omega}}$ to suffer a vertical tangency at a point $\mathbf{P}_o \in \Gamma_2^{sem}$. *Consequently if $\hat{\mathbf{\Omega}}$ is a smooth curve passing through a point $\mathbf{P}_o \in \Gamma_2^{sem}$ obeying (16), then $\mathbf{P}_o$ is necessarily a point of vertical tangency for $\hat{\mathbf{\Omega}}$.* Whether or not such points exist, however, remains an open question.

A problem in which a *curve segment* within Γ_2^{sem} is prescribed for an inverse Riemann problem will be said to be a problem of type (IRP-∞). Let us agree to limit attention to such problems for cases in which the prescribed curve is mapped bijectively to the σ^*-axis so that the curve segment can be parametrized by $v^* = \chi(\sigma^*)$ on some σ^*-subinterval of the interval $\tau(\gamma_b) \leq \sigma^* \leq \tau(\gamma_w)$ (c.f. $(34)^{\#}$), say $\sigma^*_o \leq \sigma^* \leq \sigma^*_{oo}$. Then the considerations leading to (14) in conjunction with $(38)^{\#}$ indicate that a necessary condition for the problem to be well posed is that

$$\frac{d}{d\sigma^*}\left\{ \frac{-D^*_\sigma(\sigma^*, \chi(\sigma^*))}{D^*_v(\sigma^*, \chi(\sigma^*))} \right\} < 0, \qquad \sigma^*_o \leq \sigma^* \leq \sigma^*_{oo}. \tag{17}$$

In fact, (17) is also a sufficient condition for the well-posedness of the problem as can be seen from the following construction for the stress response function $\zeta(\gamma)$ of a simple single phase impactor material which solves the problem. First let

$$\eta(\sigma^*) \equiv \left\{ \frac{-D^*_\sigma(\sigma^*, \chi(\sigma^*))}{D^*_v(\sigma^*, \chi(\sigma^*))} \right\}, \qquad \sigma^*_o \leq \sigma^* \leq \sigma^*_{oo}, \tag{18}$$

and then extend this function to the range $\sigma^* \geq 0$ in such a fashion that $\eta(\sigma^*) < 0$, $\eta'(\sigma^*) < 0$. Now define the function $I(\sigma^*)$ to be

$$I(\sigma^*) = \int_o^{\sigma^*} \eta(z)\, dz. \tag{19}$$

Finally since the function $I(\sigma^*)$ so constructed satisfies the hypotheses of Theorem 1, the stress response function $\zeta(\gamma)$ can be constructed as the inverse function to H(s) as given in (7).

Since verification of (17) is an easy task (if the function $D^*(\sigma^*,v^*)$ is known), it is a simple matter to determine the well-posedness of any problem of type (IRP-∞) for a curve $(\sigma^*,\chi(\sigma^*)) \in \Gamma_2^{sem}$ on $\sigma_o^* \le \sigma^* \le \sigma_{oo}^*$ where $\tau(\gamma_b) \le \sigma_o^* \le \sigma_{oo}^* \le \tau(\gamma_w)$. In particular (17) can be employed as a necessary and sufficient condition to test well-posedness for the case when this curve extends all the way from $\mathbf{P}_2$ to $\mathbf{P}_6$ $\left\{ \text{i.e. } \sigma_o^* = \tau(\gamma_b),\ \sigma_{oo}^* = \tau(\gamma_w),\right.$ $\left.\mathbf{P}_2 = (\sigma_o^*,\chi(\sigma_o^*)),\ \mathbf{P}_6 = (\sigma_{oo}^*,\chi(\sigma_{oo}^*)) \right\}$.

All of the extended versions of the inverse Riemann problems discussed thus far can be thought of as *global inverse Riemann problems* in that their specification is given in terms of properties of the prescribed locus $\hat{\Omega}$ at more than one point in Γ_2^{sem}. In contrast a *local inverse Riemann problem* will be one in which only properties at a single point of $\hat{\Omega}$ are prescribed. The most immediate such problem is one in which the curve $\hat{\Omega}$ is required to pass through a given point $P_o = (\sigma_o^*,v_o^*) \in \Gamma_2^{sem}$ *with a prescribed slope, say* ν_o. Such problems will be said to be of type (IRP-ν). In the absence of a slope specification, the problem, now simply of type (IRP), is well posed (c.f. (10)) and can be solved by means of the algorithm developed in connection with Theorem 1. The immediate questions which come to mind are thus:

(Q3) For a given $\mathbf{P}_o \in \Gamma_2^{sem}$, what values for the prescribed slope ν_o give rise to a well posed problem?

(Q4) Given a well posed problem of type (IRP-ν), what is an algorithm for determining a simple single phase impactor material which solves the problem?

To address these questions, consider a standard Riemann problem in which the simple single phase impactor material stress response function $\zeta(\gamma)$ is given. Let the corresponding contact locus Γ_1 be given by $\Gamma_1(\bar{v}) = \{ (\sigma^*,v^*) \mid v^* = \bar{v} + I(\sigma^*),\ \sigma^* \ge 0 \}$ (c.f. Theorem 1). A calculation based upon $(38)^{\#}$ and $(40)^{\#}$ then yields

$$\nu_o = \nu(\hat{\Omega},\mathbf{P}_o) = \frac{D_{\sigma\sigma}^* D_v^* - D_{\sigma v}^* D_\sigma^* + I''(\sigma_o^*) D_v^{*2}}{D_{vv}^* D_\sigma^* - D_{\sigma v}^* D_v^*}, \qquad (20)$$

where all of the derivatives D_σ^*, ... D_{vv}^* are to be evaluated at the point $\mathbf{P}_o \in \Gamma_2^{sem}$ under consideration. In view of Theorem 1, it may be concluded that a problem of type (IRP-ν) is well posed for and only for those values of ν_o which can be obtained from (20) by chosing finite negative values for $I''(\sigma_o^*)$. It thus follows that $|\nu_o| \to \infty$ gives rise to a well posed problem of type (IRP-ν)

at and only at points $\mathbf{P}_o$ at which (16) holds; moreover at all such points the value of $I''(\sigma_o^*)$ has no bearing on the solution to the problem. Thus it only remains to consider points $\mathbf{P}_o$ at which (16) does not hold. Letting $I''(\sigma^*)$ approach in turn both zero and negative infinity in (20) one thus finds that (IRP-ν) is well posed if and only if

$$\nu_o < \frac{D^*_{\sigma\sigma}\, D^*_v - D^*_{\sigma v}\, D^*_\sigma}{D^*_{vv}\, D^*_\sigma - D^*_{\sigma v}\, D^*_v}, \quad \text{whenever} \quad D^*_{vv}\, D^*_\sigma - D^*_{\sigma v}\, D^*_v > 0,$$

$$\nu_o > \frac{D^*_{\sigma\sigma}\, D^*_v - D^*_{\sigma v}\, D^*_\sigma}{D^*_{vv}\, D^*_\sigma - D^*_{\sigma v}\, D^*_v}, \quad \text{whenever} \quad D^*_{vv}\, D^*_\sigma - D^*_{\sigma v}\, D^*_v < 0. \tag{21}$$

Hence at each point $\mathbf{P}_o \in \Gamma_2^{sem}$ one may sector the (σ^*, v^*)-plane according to (21); all possible curves $\hat{\Omega}$ through $\mathbf{P}_o$ are then confined to the sector (21) in a neighborhood of $\mathbf{P}_o$. The algorithm for determining a simple single phase impactor material which solves a well posed problem of type (IRP-ν) is now essentially the same algorithm used in solving (IRP) given in connection with Theorem 1; the only difference is that the value of $I''(\sigma_o^*)$ is no longer allowed to be arbitrary, but instead must be given by the value

$$I''(\sigma_o^*) = \left\{ (D^*_{vv}\, D^*_\sigma - D^*_{\sigma v}\, D^*_v)\, \nu_o - (D^*_{\sigma\sigma}\, D^*_v - D^*_{\sigma v}\, D^*_\sigma) \right\} / D^{*2}_v. \tag{22}$$

In view of (16) and (21) it would be useful to partition Γ_2^{sem} into regions based on the sign of $D^*_{vv}\, D^*_\sigma - D^*_{\sigma v}\, D^*_v$. Although I have not yet examined this issue, an interesting result does follow from the observation that if $d_o = D^*(\sigma_o^*, v_o^*)$, then a direct calculation gives

$$\nu\nu(\Xi(d_o), \mathbf{P}_o) = -\left\{ D^*_{\sigma\sigma}\, D^{*2}_v - 2\, D^*_{\sigma v}\, D^*_\sigma\, D^*_v + D^*_{vv}\, D^{*2}_\sigma \right\} / D^{*3}_v. \tag{23}$$

Consequently $(Q1)^{\#}$ allows us to draw

$$(D^*_{\sigma\sigma}\, D^*_v - D^*_{\sigma v}\, D^*_\sigma)\, D^*_v + (D^*_{vv}\, D^*_\sigma - D^*_{\sigma v}\, D^*_v)\, D^*_\sigma \le 0 \tag{24}$$

for all $\mathbf{P}_o \in \Gamma_2^{sem}$. In view of (24) and $(38)^{\#}$ we may hence augment (21) to

$$\nu_o < \frac{D^*_{\sigma\sigma}\, D^*_v - D^*_{\sigma v}\, D^*_\sigma}{D^*_{vv}\, D^*_\sigma - D^*_{\sigma v}\, D^*_v} \le \nu(\Xi(d_o), \mathbf{P}_o), \quad \text{whenever} \quad D^*_{vv}\, D^*_\sigma - D^*_{\sigma v}\, D^*_v > 0,$$

$$\nu_o > \frac{D^*_{\sigma\sigma}\, D^*_v - D^*_{\sigma v}\, D^*_\sigma}{D^*_{vv}\, D^*_\sigma - D^*_{\sigma v}\, D^*_v} \ge \nu(\Xi(d_o), \mathbf{P}_o), \quad \text{whenever} \quad D^*_{vv}\, D^*_\sigma - D^*_{\sigma v}\, D^*_v < 0. \tag{25}$$

In particular (25) indicates that if $\mathbf{P}_o$ appraches a point at which (16) holds, then the corresponding sector in which $\hat{\Omega}$ must locally lie will collapse onto a vertical line.

As a closing remark, it may be noted that we have here considered an extended local inverse Riemann problem in which only the first derivative of the locus $\hat{\Omega}$ is prescribed at a point $\mathbf{P}_o = (\sigma_o^*, v_o^*) \in \Gamma_2^{sem}$. One could certainly consider an extended local inverse Riemann problem in which the *first k derivatives* of the locus $\hat{\Omega}$ are prescribed at a point $\mathbf{P}_o = (\sigma_o^*, v_o^*) \in \Gamma_2^{sem}$. On the basis of the above development, one would expect that the solution to any such problem would involve requiring definite values for the first k+1 derivatives of the function $I(\sigma^*)$ at $\sigma^* = \sigma_o^*$.

References

1. T.J. Pence, Phase transitions and maximally dissipative dynamic solutions in the Riemann problem for impact, these proceedings.

2. C. Dafermos, The entropy rate admissibility criterion for solutions of hyperbolic conservation laws, J. Differential Equations 14 (1973) 202-212.

THE INFLATION AND DEFLATION OF A THICK WALLED VISCO-HYPERELASTIC SPHERE

A. R. Johnson, C. J. Quigley, K. D. Weight and C. Cavallaro
U. S. Army Materials Technology Laboratory
Watertown, MA 02172-0001

and

D. L. Cox
Naval Underwater Systems Center
New London, CT 06320-55594

ABSTRACT

Large time dependent deformations of styrene-butadiene copolymers (SBR) and polyurethanes show both plastic and viscoelastic effects. Such materials are also often nearly incompressible and have nonlinear elastic constitutive laws. The development of finite element methods to analyze these materials is an active area of research. In this effort a recently developed finite element algorithm for the analysis of viscoelastic behavior in rubberlike materials is applied to the inflation and deflation of a thick walled visco-hyperelastic sphere under internal pressure. The material is assumed to have been previously worked so that damage effects do not significantly contribute to the deformations. A one dimensional finite element analysis is constructed for incompressible materials. Pressure vs time curves are developed for both the elastic and loss solids during an inflation and deflation cycle.

INTRODUCTION

In a recent paper Johnson et al[1] demonstrated that the standard linear solid could be generalized to a nonlinear solid and used to model large viscoelastic deformations of elastomers. The model proposed in Ref. 1 consists of a hyperelastic solid in parallel with a nonlinear internal loss solid. The elastic component of the loss solid has a hyperelastic constitutive law and the dash pot is proportional to the rate of change of the shear stresses in the elastic component of the loss solid. This paper extends the nonlinear solid model[1] so that deformations resulting from time dependent loadings can be conveniently computed. First we present some information needed[1] to define the nonlinear relaxation model. Then the model is modified and used to simulate the inflation and deflation of a visco-hyperelastic thick walled sphere. A finite element model for the thick walled sphere is derived which includes a nonlinear internal solid and a nonlinear dash pot in parallel with the sphere's hyperelastic model. The equilibrium equations are coupled in such a way that the creep problem can be solved using a standard integration method. Numerical tests are

performed by simulating the inflation and deflation of a visco-hyperelastic sphere to demonstrate the finite element algorithm.

BACKGROUND

There are numerous papers on the viscoelastic behavior of elastomers. We highlight Refs. 2-7 as a source of background information. These papers indicate that the general theory of Green and Rivlin[3] can be used with only two or three constitutive parameters to model viscous effects in many elastomers (typical visco-hyperelastic solids). That result is significant since it implies that the less general but computationally more attractive nonlinear solid network model may also be useful. Next, we define a nonlinear solid, similar to the one presented in Ref. 1, which we later extend to the case of inflation and deflation of thick walled visco-hyperelastic spheres.

Consider the network model of a simple visco-hyperelastic solid given in Figure 1. The energy in the solid is described using total Lagrangian kinematics. Nodal locations are given by x_E for the elastic (storage) solid and x_L for the loss solid. Computation of forces due to the loss solid is made assuming that the loss solid's current shape, $x_L(t)$, is its relaxed (unloaded) shape and its deformed shape, $x_E(t)$, is the current deformed shape of the hyperelastic solid. Rivlin expansions in the strain invariants are used for the energy density functions, W_E (elastic) and W_L (loss), which are given by

$$W_m = \sum_i \sum_j C_{ij}^m (I_1 - 3)^i (I_2 - 3)^j \qquad (1)$$

where $m = E,L$ $i,j = 0,1,2,\ldots$; $C_{00} = 0$,

$$I_1 = \lambda_1^2 + \lambda_2^2 + \lambda_3^2$$

$$I_2 = \lambda_1^2\lambda_2^2 + \lambda_2^2\lambda_3^2 + \lambda_3^2\lambda_1^2 \qquad \text{(strain invariants)}$$

$$I_3 = \lambda_1^2\lambda_2^2\lambda_3^2 = 1$$

and λ_1, λ_2, λ_3 are the principal stretch ratios determined from the deformed and undeformed kinematics. At any time, t, the forces deforming the loss solid are given by

$$f_L(t) = \frac{\partial W_L(x_E(t),\, x_L(t))}{\partial x_L} \qquad (2)$$

Our model assumes that the viscous forces which resist the deformation of the loss solid are proportional to the time rate of change of the

deformation forces. That is, the equilibrium equation for the loss solid (this equation also defines the dash pot) is

$$-\eta \frac{d}{dt}\left(\frac{\partial W_L}{\partial x_L}\right) = -\eta \frac{\partial^2 W_L}{\partial x_L^2}\dot{x}_L = \frac{\partial W_L}{\partial x_L} \tag{3}$$

where $\dot{x} = \frac{dx}{dt}$, η = the viscous proportionality constant. If $x_E(t)$ is specified, then equation (3) is a nonlinear differential equation which can be integrated by discretizing $x_E(t)$ with a time dependent step function and then solving a series of relaxation problems by integrating equation(3) in each of the discretized step function intervals. Both scalar and axisymmetric finite element versions of this method were formulated and used to compute uniaxial cyclic deformations in Ref. 1.

THICK WALLED VISCO-HYPERELASTIC SPHERE: SCALAR MODEL

The classical pressure - radius equation for the inflation of a thick walled incompressible hyperelastic sphere is presented in Ref. 3. The geometric definitions for this static inflation problem are shown in Figure 2, and the pressure - internal radius relationship can be summarized as follows.

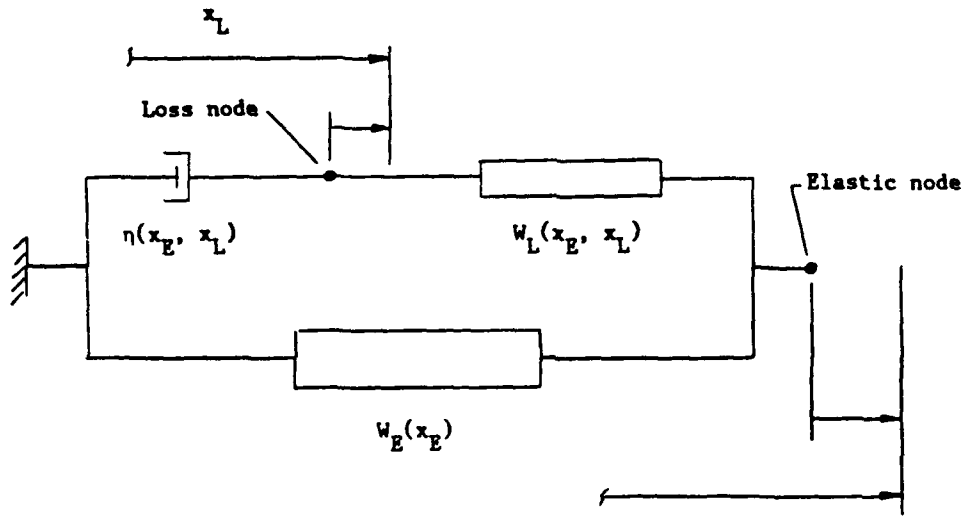
$$p = p(r_1, R_1) = 2\int_{Q_1}^{Q_2}[(Q^3 + 1)\,\Phi + (Q + \frac{1}{Q^2})\,\Psi\,]\,dQ \tag{4}$$

where $Q = \frac{r}{R}$; $Q_i = \frac{r_i}{R_i}$ $i = 1,2$; $\Phi = 2\frac{\partial W}{\partial I_1}$; $\Psi = 2\frac{\partial W}{\partial I_2}$;

$W = W(I_1, I_2)$ = the hyperelastic energy function of the material

and p = internal pressure.

To demonstrate how viscous effects with simple memory can be modeled on this classical hyperelastic inflation problem, we used the network model shown in Figure 3. The equilibrium equations are written by balancing pressure at the internal surface of the sphere in the deformed geometry. With the first variable in the list of arguments representing the undeformed geometry and the second being the deformed geometry, we can write the following equilibrium equations for the network in Figure 3.



W_E, W_L = hyperelastic energy functions.

η = nonlinear viscous element.

x_E, x_L = nodal location of coordinates.

Figure 1. A simple visco-hyperelastic solid.

$$r_2 = \left(r_1^3 + \frac{3V_o}{4\pi} \right)^{1/3}$$

$$R_2 = \left(R_1^3 + \frac{3V_o}{4\pi} \right)^{1/3}$$



Undeformed
r_1, r_2

Deformed
R_1, R_2

p = internal pressure

V_o = initial volume of material = constant (incompressible).

Figure 2. Classical inflation on an incompressible hyperelastic sphere.

$$\frac{dR_{1L}}{dt} = -\frac{1}{\eta} P_L(R_{1L}, R_{1E})$$

$$\frac{dR_{1E}}{dt} = -\frac{1}{\eta^*} [p(t) - P_E(r_1, R_{1E}) + P_L(R_{1L}, R_{1E})] \qquad (5)$$

where $P_i(r_j, R_k)$ = the pressure at the internal surface of the sphere,

$i = E, L$,

$p(t)$ = the applied internal pressure at time t,

η = the damping proportionality constant(see eq. 3),

η^* = a small damping constant selected so that the force generated in the dash pot parallel with the elastic sphere is small when compared to the other forces in the network.

Equation (5) was integrated for the case of a Neohookean material. The form of the applied pressure is shown in Figure 4 along with the resulting forces in the elastic and loss solids.

THICK WALLED HYPERELASTIC SPHERE: FINITE ELEMENT MODEL

Consider the symmetric deformation of a thick walled sphere and let λ_1 = the radial stretch and $\lambda_2 = \lambda_3$ = the hoop stretches. To demonstrate the use of internal hyperelastic solids for rubber viscoelasticity we construct a potential energy model for the inflation of a thick walled sphere in which the incompressibility constraint is enforced with a penalty. The incompressibility constraint requires $\lambda_2 = \lambda_3 = 1/\sqrt{\lambda_1}$. With $\lambda = \lambda_1$ as the only unknown variable, the strain invariants become

$$I_1 = \lambda^2 + \frac{2}{\lambda}$$

$$I_2 = 2\lambda + \frac{1}{\lambda^2} \qquad (6)$$

Equation (6) is used in equation (1) to express the internal strain energy in terms of radial stretch.

Next, we discretize the undeformed, r, and deformed, R, radial coordinates and map them both to $\xi \subset (0,1)$ as follows[9-12].

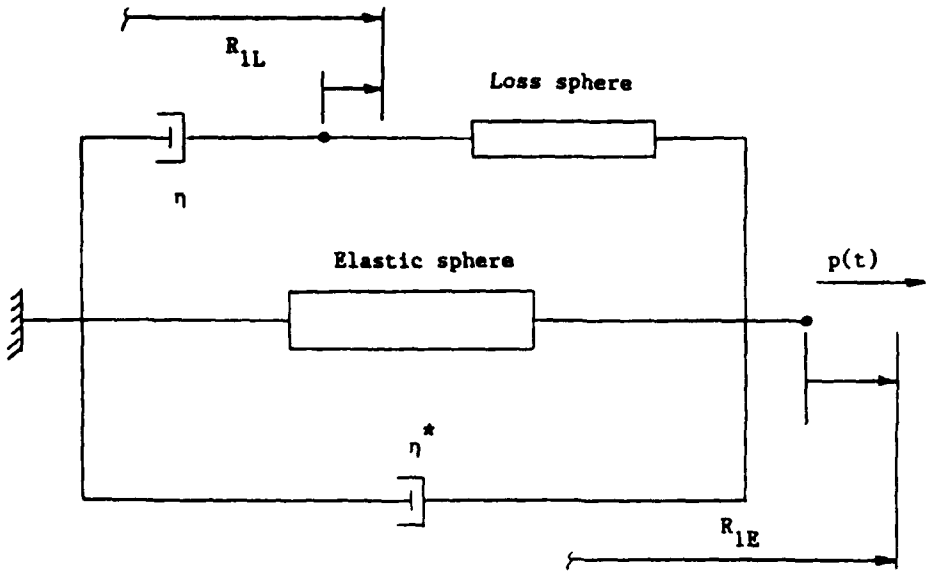


Figure 3. Network model for a scalar pressurized visco-hyperelastic sphere with scalar dash pots.

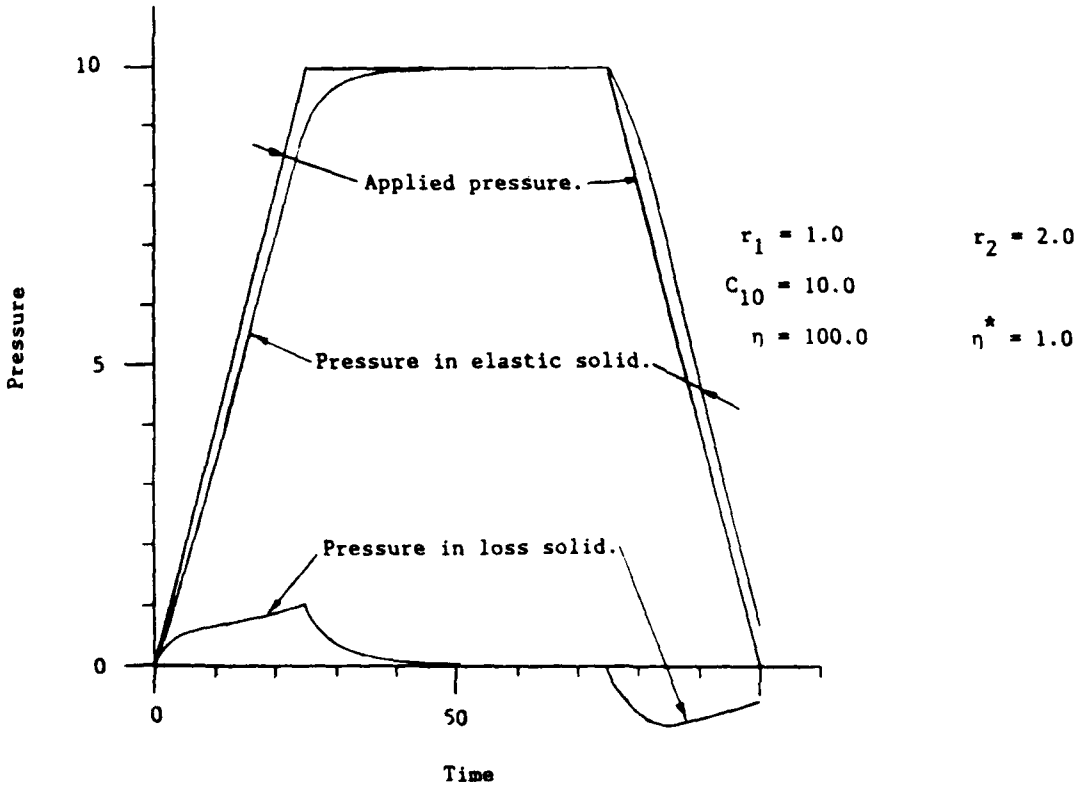


Figure 4. Pressure vs time curves for a viscous Neohookean sphere being inflated and deflated.

$$r(\xi) = (r_i \quad r_{i+1}) \begin{pmatrix} 1 - \xi \\ \xi \end{pmatrix} = \mathbf{r}^T \boldsymbol{\phi} \tag{7}$$

$$R(\xi) = (R_i \quad R_{i+1}) \begin{pmatrix} 1 - \xi \\ \xi \end{pmatrix} = \mathbf{R}^T \boldsymbol{\phi}$$

Then the internal energy is given by

$$U = \sum_{\text{Elements}} 4\pi \int_0^1 W(r,R)\, r^2 \frac{dr}{d\xi}\, d\xi \tag{8}$$

We have, from the geometry and volume constraints,

$$\lambda_1 = \left(\frac{r}{R}\right)^2 = \lambda \tag{9}$$

and $\lambda_2 = \lambda_3 = \frac{R}{r} = \frac{1}{\sqrt{\lambda}}$

Then,

$$W(r,R) = W(\lambda) \tag{10}$$

and the element gradient, with respect to the unknown deformed coordinates R_e (where e = element number) for the element becomes,

$$\mathbf{g}_e = \frac{\partial U_e}{\partial \mathbf{R}_e^T} = 4\pi \int_{\xi=0}^1 \frac{\partial W}{\partial \lambda} \frac{\partial \lambda}{\partial \mathbf{R}_e^T} r^2 \frac{dr}{d\xi}\, d\xi \tag{11}$$

Then,

$$\mathbf{g}_e = 8\pi \int_{\xi=0}^1 \begin{pmatrix} -R\lambda^2(r_1 - r_2) \frac{\partial W}{\partial \lambda} \end{pmatrix} \boldsymbol{\phi}\, d\xi \tag{12}$$

Similarly, the element tangent matrix becomes

$$\mathbf{k}_e = 8\pi \int_{\xi=0}^{1} \left(3\lambda^2 \frac{\partial W}{\partial \lambda} + 2\lambda^3 \frac{\partial W}{\partial \lambda^2} \right) (r_2 - r_1) \bullet \bar{\bullet}^T d\xi \tag{13}$$

The element gradients and tangents were evaluated using three point numerical integration. We added the square of the volume error times a penalty number (1000 times C_{10}) to enforce incompressibility as follows.

$$U_e^v = \frac{\hat{\lambda}}{2} \left((R_{i+1}^3 - R_i^3) - (r_{i+1}^3 - r_i^3) \right)^2 \tag{14}$$

where $\hat{\lambda}$ = penalty parameter(1000 * C_{10})
The terms to be added to the gradient and tangent become

$$\mathbf{g}_e^v = \hat{\lambda} \left((R_{i+1}^3 - R_i^3) - (r_{i+1}^3 - r_i^3) \right) \begin{pmatrix} -3R_i^2 \\ 3R_{i+1}^2 \end{pmatrix} \tag{15}$$

and $$\mathbf{k}_e^v = \hat{\lambda} \begin{pmatrix} 9R_i^4 - 6R_i G & -9R_{i+1}^2 R_i^2 \\ -9R_{i+1}^2 R_i^2 & 9R_{i+i}^4 + 6R_{i+1} G \end{pmatrix}$$

where $G = (R_{i+1}^3 - R_i^3) - (r_{i+1}^3 - r_i^3)$

Finally, the work done by the internal pressure is

$$W^P = \frac{4}{3} \pi (R_1^3 - r_1^3) \tag{16}$$

with contributions to the global gradient and tangent of

$$\mathbf{g}^P = \begin{pmatrix} 4\pi R_1^2 \\ 0 \\ 0 \\ \vdots \\ \vdots \end{pmatrix} \tag{17}$$

and

$$\mathbf{k}^P = \begin{pmatrix} 8\pi R_1 & 0 & 0 & . & . & . \\ 0 & 0 & 0 & . & . & . \\ . & . & . & . & . & . \end{pmatrix}$$

The above model was used to numerically simulate the inflation of a thick walled sphere and is compared graphically to the exact solution (using finite elements) in Figure 5.

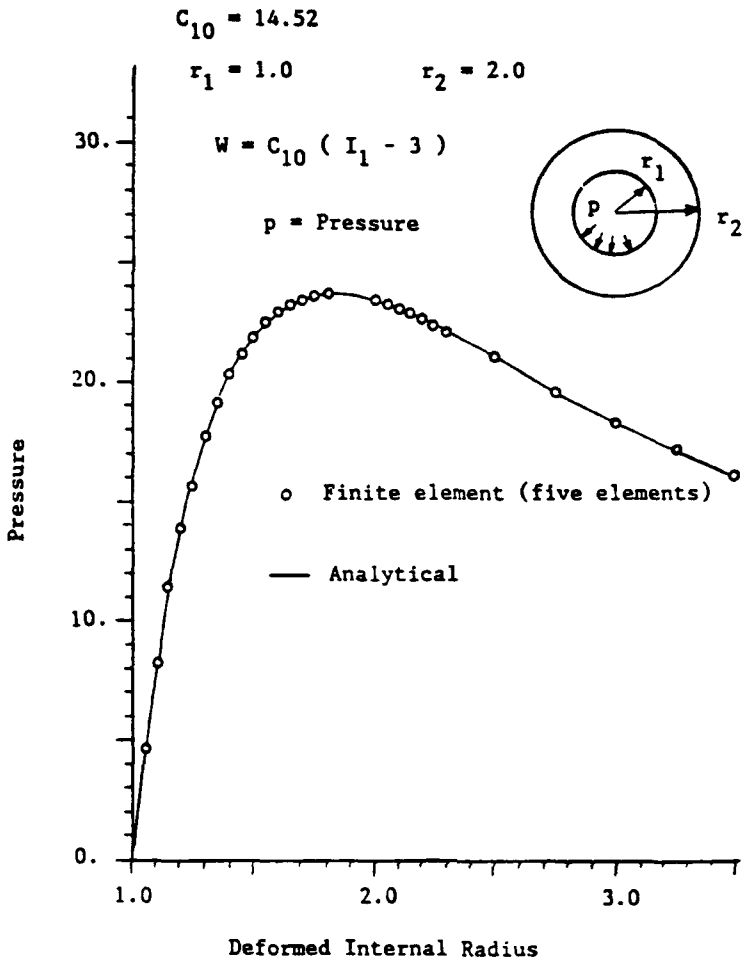


Figure 5. Finite element solution vs analytical solution for inflation of a thick walled Neohookean sphere.

INFLATION AND DEFLATION OF A VISCO-HYPERELASTIC SPHERE

Using the network model in Figure 3 with configuration vectors at the nodes and dashpots proportional to the tangent strain energy matrices, we can write equilibrium equations for the finite element model as follows.

$$\begin{Bmatrix} \eta \dfrac{d^2U_L}{d\mathbf{R}_L d\mathbf{R}_L^T} & 0 \\ 0 & \eta^* \dfrac{d^2U_E}{d\mathbf{R}_E d\mathbf{R}_E^T} \end{Bmatrix} \begin{Bmatrix} \dot{\mathbf{R}}_L \\ \dot{\mathbf{R}}_E \end{Bmatrix} = \begin{Bmatrix} \mathbf{g}_L \\ -\mathbf{g}_E - \mathbf{g}_L \end{Bmatrix} \tag{18}$$

In general, the selection of η and W_L should be made by matching hysteretic or relaxation data after the quasi-static elastic engery function, W_E, has been determined. As in the scalar model, we are using η^* here so that equation (18) maintains a form which can be easily integrated (eg., using the Runge-Kutta method). Obviously, additional internal solids can be added to increase the accuracy of this method (a generalized nonlinear Maxwell model). To demonstrate the model, we computed one inflation and deflation of a visco-NeoHookean sphere. The pressure vs time curves for the loss and elastic solids are shown in Figure 6.

CONCLUSION

A nonlinear visco-hyperelastic solid model was proposed and used to simulate the inflation and deflation of a visco-Neohookean sphere. The method was easy to program and no difficulties were encountered integrating the differential equations. As pointed out in Ref. 1., the Runge-Kutta updates for the elastic solid consist of Newton-Raphson steps generated from a potential energy function. These steps must be small to be useful. Further research is being done to investigate the sensitivity of the numerical computations to the values of the parameters in the model.

REFERENCES

1. JOHNSON, A.R., QUIGLEY, C.J., CAVALLARO, C. and WEIGHT, K.D., A Large Deformation Viscoelastic Finite Element Model for Elastomers, MAFELAP 1990, Ed. J.R. Whiteman, J. Wiley and Sons.

2. TRELOAR, L.R.G., Stress-Strain Data for Vulcanized Rubber Under Various Types of Deformation, Trans. Faraday Soc. 40, 59-70 (1940).

3. GENT, A.N., Relaxation Processes in Vulcanized Rubber Under. I. Relations Between Stress, Relaxation, Creep Recovery and Hysteresis, J. Appl. Poly. Sci. 6, 433-441 (1962).

4. MULLINS, L., Softening of Rubber by Deformation, Rubber Chem. Technol. 42, 339-362 (1969).

5. GREEN, A.E., and RIVLIN, R.S., The Mechanics of Nonlinear Materials with Memory, Arch. Rational Mech. Anal. 1, 1-21 (1957).

6. BERNSTEIN, B., KEARSLY, E.A., and ZAPAS, L.J., A Study of Stress Relaxation with Finite Strain, Rubber Chem. Technol. 38, No. 1, 76-89 (1965).

7. SMITH, T.L., and DICKIE, R.A., Effect of Finite Extensibility on the Viscoelastic Properties of a Styrene-Butadiene Rubber Vulcanizate in Simple Deformations up to Rupture, J. of Polymer Sci. 7, 635-658 (1969).

8. GREEN, A. E., and ZERNA, W., Theoretical Elasticity, Oxford University Press, London (1968).

9. ODEN, J.T., Finite Elements of Nonlinear Continua, McGraw-Hill, New York (1971).

10. MALKUS, D.S., Finite Elements for Penalties in Nonlinear Elasticity, Int. J. Numer. Meth. Eng. 16, 121-136,(1980).

11. FRIED,I. and JOHNSON, A.R.,Nonlinear Computation of Axisymmetric Solid Rubber Deformation, Comput. Meths. Appl. Mech. Engng. 67, 241-253 (1988).

12. FRIED, I. and JOHNSON, A.R., A Note on Elastic Energy Density Functions for Largely Deformed Compressible Rubber Solids, Comput. Meths. Appl. Mech. Engng. 69, 53-64 (1988).

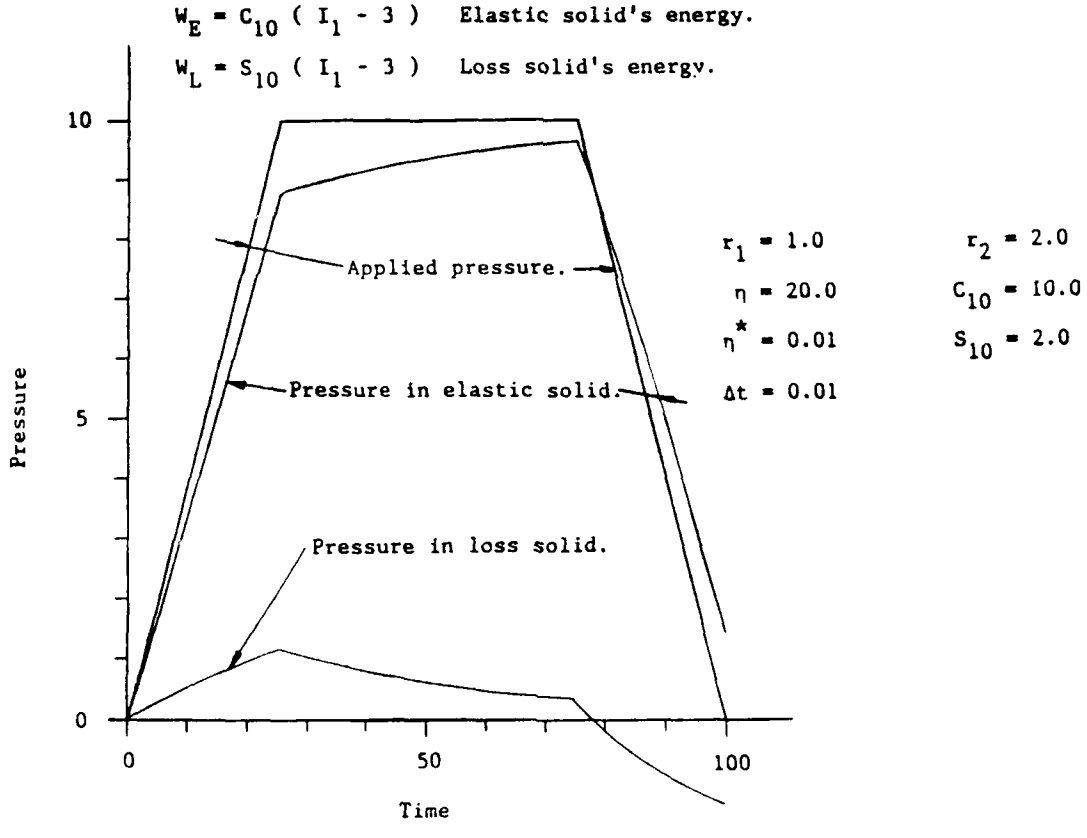


Figure 6. Pressure vs time curves for a viscous Neohookean sphere being inflated and deflated - finite element solution with nonlinear dash pots.

EVALUATION OF DRAG LOADING MODELS IN OVERTURNING RESPONSE CODES

Aaron Das Gupta
Research Mechanical Engineer
US Army Ballistic Research Laboratory, US Army Laboratory Command
Aberdeen Proving Ground, MD 21005-5066

ABSTRACT

The dynamic overturning response of Army vehicles with shelters and trailers as well as overturn susceptible structures has been of considerable interest to the defense community since critical structures and internal equipments can be damaged resulting in system malfunction and reduction of vehicle performance jeopardizing the primary mission objective. To overcome vulnerability of vehicles and associated equipments to overturning, the Army is actively engaged in a hardening program which will result in overturn mitigation and increased survivability. Flexible multibody dynamics programs can be used to predict overturning response of structures due to transient overpressure loading and facilitate evaluation of mitigation devices such as outriggers, cables and guy wires. However, accuracy of overturning prediction is dominated by the loading model during the drag phase when the structure becomes unstable. To assess validity of the loading model, accuracy of drag coefficients as a function of the roll angle and flow velocity should be evaluated. The current investigation is devoted to a comparative evaluation of drag coefficients used in loading models in overturning response codes as well as any experimental data that may be available.

INTRODUCTION

Research in vehicle dynamics started with Olley [1] and Segal [2] in the area of linear vehicle dynamics which was followed by several others [3-5]. The first vehicle models allowed between two and four degrees of freedom. Closed form analytical solutions for critical speed were obtained for these models for certain maneuvers. However, the solution techniques could not be extended to real vehicles due to complexity of the vehicle models and the transient response characteristics.

With increased computational capabilities, the vehicle models became increasingly complex. Nonlinear tire models replaced linear models to describe the tire forces more accurately [6]. Eventually, very complex nonlinear vehicle models could be generated to simulate the dynamic response of vehicles with additional degrees of freedom [7]. An overview of these developments is given by Ellis [8]. However, these studies were devoted to a rather limited class of vehicles. The equations of motion for these vehicles were derived manually and then transformed to computer code to be numerically integrated.

Although a simple vehicle could be simulated in this manner, creating larger representative models and computation of response are rather difficult and laborious.

Recently, flexible multibody dynamics software have been developed which automatically formulate the equations of motion for many types of mechanical systems. These programs require the user to define the mass and inertial properties of all rigid as well as flexible bodies in the system and the type of joints that connect each body relative to one another. In addition, force elements can be prescribed to simulate springs, dampers and actuators between rigid or flexible bodies. This involves defining the spring mass and suspension system elements connected with ideal joints or bushings. The springs and shocks are modeled as force elements.

Multibody dynamics programs can be divided into two seperate groups. The first type numerically formulates the equations of motion at each time step during the numerical integration process. ADAMS [9] and DADS [10-12] are multibody dynamics programs which numerically formulate and solve the equations of motion. The second group of programs symbolically formulates the equations of motions which can then be numerically integrated. NEWEUL [13-15], and MESA VERDE [16] codes belong to this group. Other differences exist between all of these programs including coordinate selection, integration method, and output format. Schiehlen et al [14] obtained a good comparison of the performances of these programs as related to the vehicle dynamics field.

Although many of these programs have a high potential for modeling wheeled vehicle dynamics, their capabilities are currently limited due to a lack of appropriate forcing functions to which the vehicle may be subjected in a battlefield. Majority of flexible multibody dynamics codes that are commercially available addresses the needs of the vehicle industries which are interested primarily in dynamic response due to sudden change in road friction due to rain, snow and accidental spillage or the influence of a rough terrain with bumps or discontinuities upon vehicle stability. Researchers at the University of Iowa have modeled the ride characteristics of several vehicle combinations [11,12]. Majority of these models did not contain any lateral response dynamics.

Lee, Hobbs and Atkinson [17] have developed a computer program TRUCK 3.1 to find the dynamic response of a vehicle subjected to different types of loads including such intensive loads as blast waves from conventional or nuclear explosives. The code yields gross motions of the vehicle body, and of tires, axles, shelters, and racks relative to the vehicle body. Large motions, including sliding and overturning of the vehicle, are permitted. However, each individual element like an axle, shelter or rack is treated as a rigid body. Recently, the governing equations of motion used in the TRUCK code that model the rigid body motions of the vehicle assembly and its components and modelling of the frictional force between the tires and the ground has been validated by Batra [18] under Ballistic Research Laboratory (BRL) sponsorship.

This paper is devoted to an evaluation of drag coefficients in the drag phase loading models currently installed in the MINITRUCK code as well as the Overturning Code [19] developed at BRL and validation with respect to some experimental data for two specific Army vehicles obtained by S-CUBED [20] Inc. under BRL sponsorship. Commercially available flexible multibody dynamics programs do not include blast overpressure loading at present. However, capability exists in these codes to interface with user-supplied loading routines. Accurate prediction of vehicle overturning response from these codes will depend on the development of reliable blast loading models for incorporation into these codes and response validation using simple generic problems with known solutions.

STRUCTURAL MODELING

The complexity of the mathematical model selected to represent a structure is necessitated by the need to obtain accurate information to be derived from the model. Accurate analyses of wheeled vehicles require adequate representation of tire and suspension systems which dominate vehicle response. Additionally, accurate loading functions need to be developed with capability to impose these loads upon the vehicle model in a realistic manner. Since most multibody dynamics programs do not currently address these problems in sufficient detail, it is imperative upon the user to produce loading subroutines and interface them with an existing program. Complex kinematic models suited for multibody modeling require a large amount of geometric and physical data. Any gains due to increased model complexity may be lost due to lack of accurate input and loading data. In spite of these drawbacks, flexible multibody dynamics programs have several advantages for vehicle modeling applications where large number of kinematically constrained members and complex kinematic relationships for suspension systems and joints are involved.

CODE DESCRIPTION

The MINITRUCK [21] code is a simplified two-dimensional version of the TRUCK code developed by Kaman AviDyne to predict the response of a variety of Army vehicles to blast waves. The program models the five major degrees-of-freedom which include roll, sideslip, heave, and two suspension related degrees-of-freedom. The vehicle is assumed to be initially at rest on level ground. Only a side-on blast encounter from the roadside is allowed for the 2-D version. Capabilities include modeling regular or independent suspension, flexible or rigid outriggers and guy wires. Equivalent properties are used for characterizing many vehicle components. Thus stiffnesses of all axle springs on one side of a vehicle are represented by one equivalent spring In the same manner, masses are represented by equivalent lumped mass models. This is due to the 2-D MINITRUCK structural model restriction. This representation can best be visualized by compressing or collapsing a vehicle in the fore-and-aft direction until it is entirely compressed in a vertical plane, normal to the shock front. During the response, all motions are confined within this plane. The three-dimensional aerdynamic representation used in the TRUCK code was retained.

The current MINITRUCK version allows parking the vehicle on a simple horizontal ground which excludes special positioning of a loaded field vehicle on an inclined ground at the time of a blast encounter. However, such effects of inclined slopes on vehicle overturning due to a blast load is rarely encountered and require complex ground modeling. The assumption of horizontal ground simplifies considerably the theoretical formulation. The primary coordinate system is a body axis system attached to the vehicle body with its origin at the center of gravity of the entire system.

The Overturning Code [19] developed at BRL included an airblast loading model for closed cubical box type structures which were used to model various targets. The targets were subdivided into a convenient number of closed boxes which were attached to each other to represent the shapes of the targets. To allow estimation of overturning moments, values of the rolling moment coefficients measured in the wind tunnel for specific truck-shelter combinations were fitted as a function of the angle of rotation and incorporated in the code.

AERODYNAMIC LOADING

The loading on the vehicle results from the blast wave from a nuclear or conventional explosion. Associated with the blast wave are increased pressure density and material velocity. Blast wave characteristics for two heights of burst are considered in these codes. The first one corresponds to a height of burst of zero representing the sea level and the second corresponds to a burst height of $60W^{1/3}$ m based upon BRL data. The two blast wave characteristics are specified in tabular form for a 1 KT weapon yield at sea level and the results are applied for other weapon yields and conditions other than sea level using the built-in modified Sachs scaling laws. The nominal range of the structure from the burst point or the peak overpressure behind the shock needs to be specified.

The blast wave front is assumed to be normal to the ground surface, and the vehicle is assumed initially normal to the ground; shock encounter can be from any orientation. As the blast wave envelops the vehicle, the wave reflects from the vehicle surface and rarefaction waves emanate from free edges to relieve the reflected pressure. At later times, the loading is essentially a drag type loading, resulting from the material velocity associated with the blast wave. The development of the aerodynamic loading is seperated into two phases namely, diffraction loading and drag loading.

Diffraction Loading

The diffraction loading phase begins with the shock impinging upon the vehicle. As the vehicle is enveloped, the shock wave reflects from the vehicle surfaces and rarefaction waves emanate from free edges to relieve the reflected pressure. The diffraction loading is based upon shock tube experiments in which front and back face pressures on the structural component were measured for a shock front normally incident on the front face [22]. For MINITRUCK only front face normal incidence is considered and diffraction loadings for arbitrary incidence angles are excluded.

It is assumed that the vehicle moves very little during the diffraction period, so that the shock wave and the vehicle is expected to remain normal to the ground during this period of time. This assumption limits the number of configurations that can be addressed.

The basic model for the diffraction loading describes waves emanating from the free edges of the aerodynamic boxes representing the exposed vehicle surfaces. When the undisturbed shock front reaches an edge point, a relief wave emanates which leads to an exponential decay in the time-varying pressure loading applied to the faces.

Drag Loading

When the blast wave first encounters and engulfs the target, the overturning moment due to shock reflection and diffraction process is large, and drag loading is not considered. After the shock front traverses the target, the drag loading moment relative to the diffraction loading moment becomes significant. Thus following the diffraction phase, the pressure loading becomes a drag type of loading. Drag loading depends primarily on the dynamic pressure associated with the material velocity behind the shock front. When the drag loading moment becomes equal to or larger than the diffraction loading moment, the diffraction loading phase is terminated, and only the drag loading moment is used in calculating further response of the target.

The pressure loading during the drag phase for a non-decaying shock of strength P_s, is given by

$$p(t) = P_s + C_p q \tag{1}$$

where P_s and q are the overpressure and the dynamic pressure associated with the blast wave. The task of estimating the drag-phase loading thus becomes a matter of determining the pressure coefficient, C_p. The pressure coefficient must be determined for arbitrary vehicle orientations for various rectangular areas which make up the exterior surfaces of the box-like configuration while the shock is assumed to be travelling parallel to the ground.

a. MINITRUCK Drag Model

There are many factors which can influence the value of C_p . In addition to alpha, the angle between the flow velocity and the surface normal, these dependent variables may include Reynolds number, Mach number, aspect ratio (i.e., ratio of the two dimensions of the rectangular surface), etc. In MINITRUCK, expressions for the drag coefficient in a generalized form are given as

$$C_p = d\cos(9/8\alpha) \tag{2}$$

where, $0 \leq \alpha \leq \alpha_1$. However, if $\alpha_1 \leq \alpha \leq \pi$,then C_p = -0.4 and,

$$d = (1.00 - 0.15R) + (0.20 + 0.15R)(\psi/3)^2 \tag{3}$$

The above equations account only for the α-variation and R in equation (3) refers to the aspect ratio. The Mach number M is related to the shock strength ratio ψ according to the Rankine-Hugoniot relation

$$M = 5\psi/\sqrt{(7 + 7\psi)(7 + \psi)} \tag{4}$$

At low Mach numbers and aspect ratio of unity the equations are in general agreement with experimental data of Hankins [23] and Hoerner [24] which results in a drag coefficient of 1.25 for a cuboid. In MINITRUCK code the generalized drag coefficient model is necessitated by a lack of extensive pressure distribution data for various surfaces at higher Mach number obtained from wind-tunnel tests.

b. Overturning Code Drag Load

Drag loading model used in the Overturning Code [18] assumes that the target is a collection of cuboids. The drag loading on each cuboid is calculated independently of the others, and the total drag moment is calculated by summing the moments calculated for each cuboid. The drag loading on a cuboid is assumed to be zero until the shock front arrives at the center line of the cuboid. Loading is then computed using appropriate areas corresponding to the angle of rotation of the target. The characteristics of the blast wave are assumed to remain unchanged over the depth of the target, so that the same free-field blast waveform can be used for all cuboids.

The drag loading is divided into a horizontal and a vertical component which in turn depends upon the dynamic pressure, horizontal or vertical area, constants corresponding to drag coefficient of horizontal or vertical area at zero flow velocity, and the roll angle of rotation. The code multiplies each component with the corresponding moment arm and the product is summed over all cuboids to calculate the overturning moment due to drag. The increase in horizontal and vertical drag with Mach number is based on the approach of Hoerner [24], who describes the drag in subsonic flow of bluff forms with entirely seperated from the rear in terms of two components. These are a positive front surface pressure proportional to the incident pressure, and a negative base pressure assumed to be independent of Mach number. The relation which was obtained by fitting wind tunnel data on block configurations can be expressed as:

$$C_{Drag} = C_0(0.375 + 0.625(1 + 0.25M^2)) \tag{5}$$

where,C_0 = 1.26 for a cube on a ground plane and C_0 = 1.15 for a rectangular parallelipiped on a ground plane, with width to height ratios of 2 or 4.

c. Modified Drag Load

It is rather difficult to obtain a generalized drag equation which will accurately predict drag coefficients as a function of the roll angle and the Mach number for all types of structural configurations since these coefficients are very sensitive to geometrical configurations, surface smoothness and lateral dimensions of the target body. Although diffraction loading influences initial response of structures, the onset of overturning in most cases occurs at a later stage which is largely dominated by loading during the drag phase. As a result reliable prediction of overturning is directly dependent on accurate prediction of drag coefficients.

Computational fluid dynamics codes have been developed to reliably predict transient structural loading due to a propagating wave during the diffraction phase. However, serious numerical difficulties occur during the drag phase at later times which precludes the use of these codes for drag loading prediction. A preferred alternative is to generate a scaled mechanical model for each structure or vehicle and experimentally obtain drag force and moment coefficients from wind tunnel tests at various orientations along roll, pitch and yaw directions relative to the wave front. This method is expensive since the experiment has to be repeated several times at various combinations of Mach numbers and roll, pitch and yaw angles until overturning is ensured. However, it is the only reliable and fully acceptable technique of drag loading assessment at present.

In the absence of experimental data for most recently developed Army vehicles and equipments, a modified drag loading equation based on a combination of useful features of the previous two approaches can be deployed as shown below:

$$C_{Drag} = C_1 Cos(9/8\alpha)[0.375 + 0.625(1 + 0.25M^2)] \tag{6}$$

where, C_1 = 1.2 for a rectangular parallelipiped on a ground plane with width to height ratios of 2-4.

The equation above has a constraint condition on the roll angle (alpha) such that $0 \leq \alpha \leq 7\pi$ /12. However, beyond this range the coefficient is determined to be a constant, C_{Drag} = -0.4 provided

we have $7\pi/12 \leq \alpha \leq \pi$.

d. Experimental Drag Load

The aerodynamic force and moment coefficients are essential input to the solution of any airblast loading and response problem. Recently S-CUBED Inc. [20] derived the drag, lift and moment coefficients from experimental data from a series of wind tunnel tests on scale models of two Army vehicles.

The roll (or overturn) angle dependency and to a lesser extent, the Mach Number effect can be seen when the moment coefficients due to side-on exposures to a number of steady state flow field environments are obtained as a function of the normalized angle of attack for a particular vehicle.

The aerodynamic drag, lift and moment coefficients were shown to be quite sensitive to changes in both the overturn angle and the aspect ratio. This sensitivity is, to a large extent, introduced artificially as a consequence of using a body-fixed coordinate system and could be reduced by performing two operations on the data. First, the dependence on overturn angle is suppressed by transforming the coefficients to a non-rotating coordinate system. The geometry effects are reduced by normalizing the overturn angle to an angle defined by the roll position of the system when the projected side-on area is a maximum.

As shown in Figure 1, the aerodynamic drag coefficient from wind tunnel tests on a 1-1/4 ton truck scale model subjected to a side-on blast overpressure typically exhibit an initial increase at small roll angles followed by rapid decrement at higher angles of attack beyond 20 degrees. The drag appears to decrease with increased Mach number until the roll angle equals or exceeds 60 degrees when Mach number dependency of the drag coefficient is diminished considerably. For accurate estimation of the drag coefficient, the experiment has to be repeated for each vehicle model at various flow velocites at both low and high Mach numbers and roll angles until occurrence of overturning is ensured.

COMPARISON OF DRAG LOADING MODELS

The variation of drag coefficient with roll or overturning angle currently used in the MINITRUCK code has been compared with the modified drag loading model from the Overturning Code as shown in Figure 2. Additionally, the dependence of drag on flow velocity and roll angle in the form of desensitized aerodynamic coefficient data from S-CUBED Inc. obtained by transforming the coefficients to a non-rotating coordinate system for two specific Army Vehicles has been compared with the MINITRUCK drag model as shown in Figure 1.

The MINITRUCK drag model in both Figure 1 and 2 has no dependence on flow velocity since the simplified semi-empirical drag equations described earlier ignores the influence of Mach number on drag coefficients. In Figure 2 both models describe gradual decrease of drag coefficients from initial configuration to a roll angle of approximately 80 degrees. Beyond this angle drag coefficients appear to have small negative values for both models. However, when the roll angle exceeds 100 degrees, the drag is assumed to be a constant since at these large angles overturning is fully ensured in most cases.

Comparison of numerical values of the drag coefficient at small roll angles shows considerable difference between the two models. At the initial configuration with a null value for the roll angle, the coefficient varies from 1.33 to 1.2 due to Mach number variation of .1 to .8 for the modified

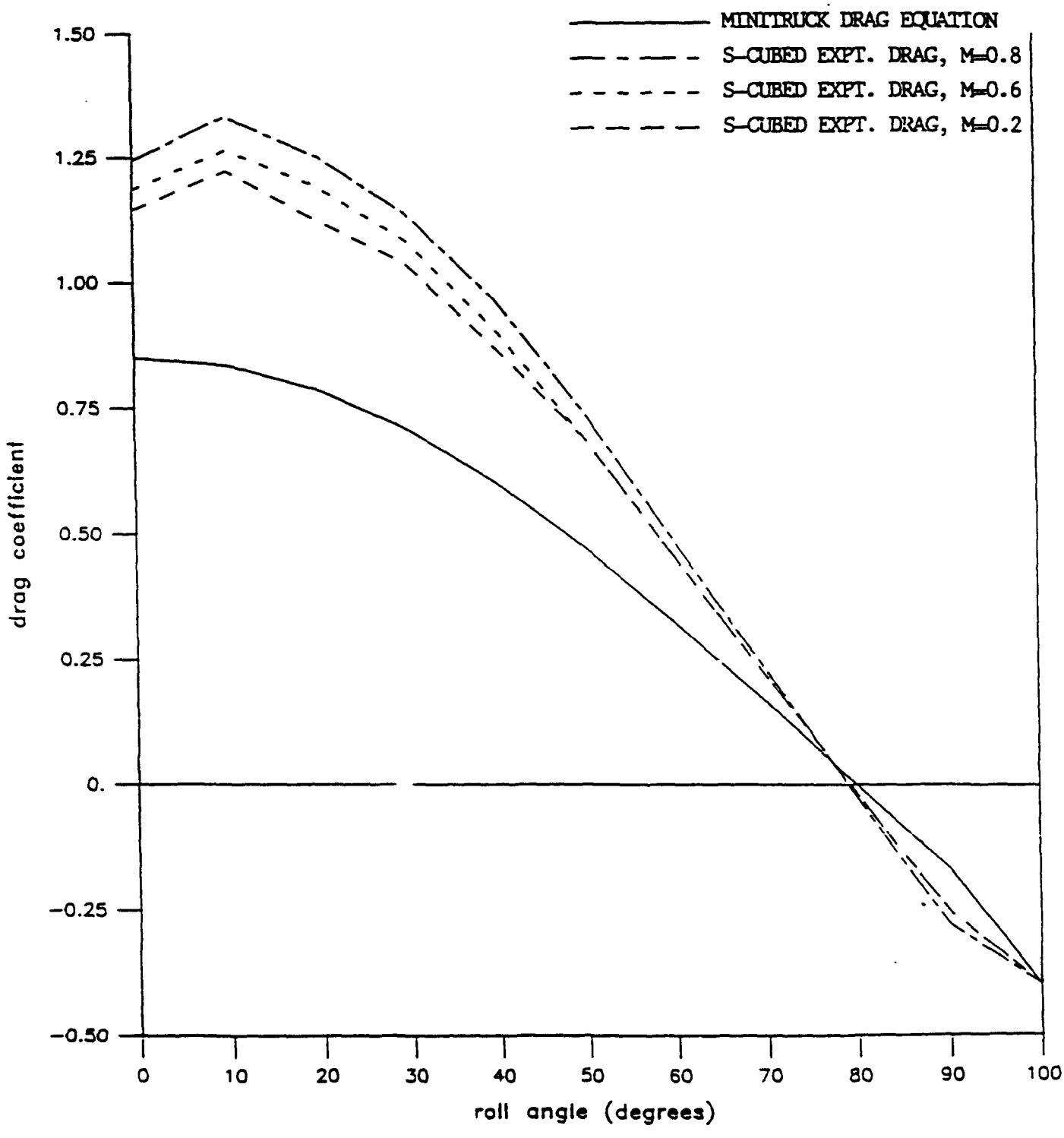


Figure 1: Comparison of S-CUBED experimental drag data with MINITRUCK drag model.

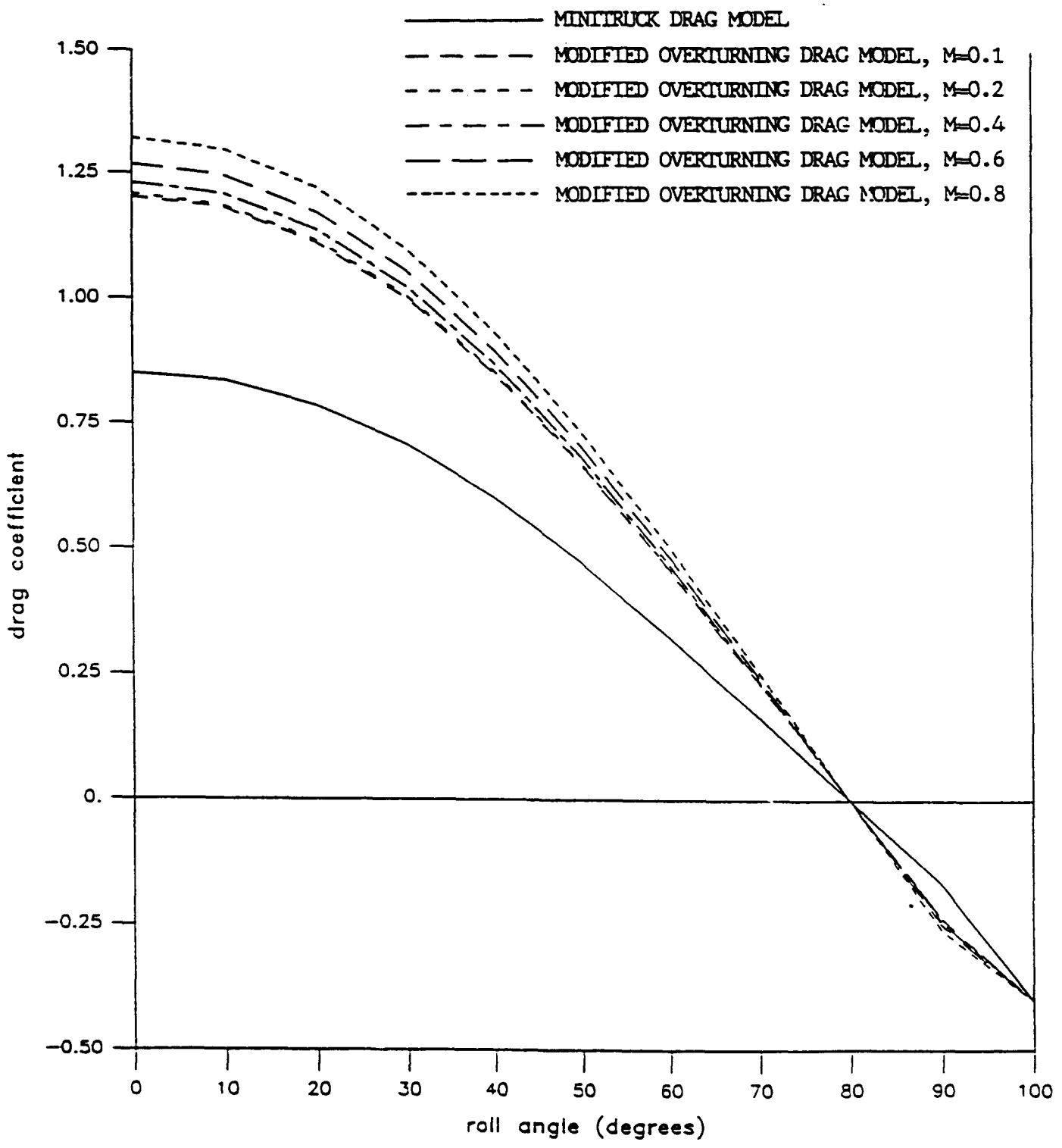


Figure 2: Comparison of MINITRUCK drag with modified Overturning Code drag model.

overturning code drag model while the corresponding MINITRUCK drag value is approximately 0.85. This difference is rather large and it can exert significant influence upon the overturning response prediction. As a result, the large difference is of concern even though the difference diminishes rapidly with increasing roll angle until approximately 80 degrees when the data appear to converge at zero drag.

Comparison of drag in Figure 1 between the semi-empirical model in the MINITRUCK code and the experimental data from S-CUBED shows large difference in magnitude of drag coefficients between the two sets of curves similar to observation in Figure 2. However, comparison of drag between the experimental data and the modified Overturning Code loading model shows very satisfactory agreement with an exception that initially the drag data tends to increase with roll angle until the drag reaches a peak value at an angle of 10 degrees beyond which it decays gradually as expected while the modified model does not exhibit such a trend. The MINITRUCK drag loading model is in substantial disagreement with available data and other predictive models indicating a probable lack of validity and a need for improvement of the currently installed MINITRUCK loading capability.

The agreement between the experimental data and the modified model is rather unexpected in view of the fact that the data pertains to two specific Army vehicles while the modified model is a generalized predictive model. Validity of such a model cannot be made without conducting extensive number of experiments upon various vehicles and structures and comparing the data with the modified model. Such a study will facilitate evaluation of an experimentally based correlation factor which can improve the capability of the modified loading model. However, interaction effect of various components of the structure upon the drag loading cannot be predicted accurately by a generalized model and experimental drag loading data are deemed to be vital for overturning response computation of complicated structures and vehicles which can be dominated by such effects.

DISCUSSION AND RECOMMENDATION

Comparison of various drag loading models with experimental data is necessary to determine accuracy of prediction of vehicle overturning to airblast loading. In most cases the drag loading on each rectangular paralleliped or cuboid aerodynamic configuration is calculated independently of the others and the total drag moment is calculated by summing the moments calculated as a function of the dynamic pressure, facing area, the moment arm and the angle of rotation for each block. As a result, the interaction effects of adjacent blocks upon one another are ignored during the computation. In structural assemblies when some blocks are located behind others along the direction of wave propagation, significant error in loading calculation will occur due to exclusion of overlapped regions. The linear summation method of loading assessment can result in rather excessive load estimation contributing to incorrect overturning prediction. Any proposed improvement in prediction of overturning response must include accurate estimation of drag loading that accounts for mutual interaction effects of adjacent structural components upon the drag coefficient.

In summary, a major shortcoming in computional modeling and accurate prediction of overturning response is the lack of a reliable drag function to use in flexible multibody dynamic response programs currently available. Due to inability of current computational fluid dynamics codes to predict accurate drag functions of structures subjected to side-on overpressures at late response times, there is a need to conduct an experimental effort to obtain drag functions of various structures susceptible to overturning. The effort is expected to include model studies of vulnerable structures from generators to tanks in wind tunnels at various roll and pitch angles at small as well large flow velocities in the subsonic

range. The data obtained from such experiments on a variety of structures must be reduced to a form suitable for direct input to one or more of the currently available flexible body dynamics codes and may constitute part of a library of standardized drag loading functions for a variety of frequently encountered structures and vehicle configurations commonly used by the Army.

Additionally, whole body motion of targets in terms of sliding and overturning could be studied in open-ended shock tubes. This should also include the case where the loading function consists of a rounded, relatively nonreflecting shock front. It will have very low side-on overpressure component combined with a very high stagnation pressure component which is of great interest to the Army. Initially wooden cuboids like those used by Ethridge may be modeled and tested for sliding and overturning, the data from which could be used to validate prediction from currently available flexible multibody dynamics codes. The side-on and stagnation overpressure versus time at open end of the shock tube, at various longitudinal and radial locations can be mapped, for use as data input in appropriate prediction codes.

ACKNOWLEDGEMENTS

Valuable assistance of Drs. John F. Polk and Joseph M. Santiago of the Terminal Ballistics Division during the course of this investigation is gratefully acknowledged.

REFERENCES

1. Olley, M., "Road Manners of the Modern Car," Proceedings of the Institute of Automobile Engineers, Vol.41, 1946-1947,pp. 147-182.

2. Segel, L., "Theoretical Prediction and Experimental Substantiation of the Response of the Automobile to Steering Control," Proceedings of the Institution of Mechanical Engineers Automobile Division, 1956-1957, pp. 310-330.

3. Bergman, W., "The Basic Nature of Vehicle Understeer-Oversteer," SAE Paper 957B, Proceedings of the Society of Automobile Engineers, January 1965.

4. Bundorf, R.T.,"The Influence of Vehicle Design Parameters on Characteristic Speed and Understeer," SAE Paper 670078, Proceedings of the Society of Automobile Engineers, January 1967.

5. Beauvais, F.N., Garelis, C., and Iacovani, D.H., "An Improved Analogue for Vehicle Stability Analysis," SAE Paper 295C, Proceedings of the Society of Automobile Engineers, January 1961.

6. Dugoff, H., Fancher, P.S., and Segel, L., "An Analysis of Traction Tire Properties and Their Influence on Vehicle Dynamic Performance," SAE Paper 700477, Proceedings of the Society of Automobile Engineers, January 1970.

7. McHenry, R., et al., "Vehicle Dynamics in Single Vehicle Accidents," Cornell Aeronautical Laboratories, Report No. VJ-2251-V-3, December 1968.

8. Ellis, J.R., Vehicle Dynamics, Business Books Inc., London, U.K., 1969.

9. Orlandea, N., Chase, M.A., and Callahan D.A., "A Sparsity-Oriented Approach to the Dynamic Analysis and Design of Mechanical Systems, Parts I and II," Journal of Engineering for Industry, Vol. 99, August 1977, pp. 773-784.

10. Haug, E.J., et al., "Computer Aided Analysis of Large Scale, Constrained, Mechanical Systems," Proceedings of the 4th International Symposium on Large Engineering Systems, Calgary, Alberta, Canada, June 1982, pp. 3-15.

11. Wehage, R.A., and Haug. E.J., "Generalized Coordinate Partitioning for Dimension Reduction in Analysis of Constrained Dynamic Systems," Journal of Mechanical Design, Vol. 104, January 1982, pp. 247-255.

12. Nikravesh, P.E., and Chung, I.S., "Application of Euler Parameters to the Dynamic Analysis of Three-Dimensional Constrained Dynamic Systems," Journal of Mechanical Design, Vol.104, October 1982, pp.785-791.

13. Kortum, W., and Schiehlen, W.O., "General Purpose Vehicle System Dynamics Software Based on Multibody Formalisms," Vehicle System Dynamics, Vol.14, No.4-6, June 1985, pp. 229-263.

14. Schiehlen, W.O., "Modeling of Complex Vehicle Systems," Vehicle System Dynamics, Vol.12, No.1-3, July 1983, pp. 12-14.

15. Kreuzer, E.J., and Schiehlen, W.O., "Generation of Symbolic Equations of Motion for Complex Spacecraft using NEWEUL," AIAA paper 83-302, August 1983.

16. Wittenburg, J., et al.,"MESA VERDE: a Symbolic Program for Nonlinear Articulated Rigid-Body Dynamics," Proceedings of the 10th Design Engineering Division Conference on Mechanical Vibration and Noise, Cincinnati, Ohio, September 10-13, 1985.

17. Lee, W.N., Hobbs, N.P., and Atkinson, M., "TRUCK 3.1-An Improved Digital Computer Program for Calculating the Response of Army Vehicles to Blast Waves," Kaman AviDyne, Burlington, MA, 1983.

18. Batra, R.C., "Computations for Truck Sliding with TRUCK 3.1 Code," Contractor Report BRL-CR-616, U.S. Army Ballistic Research Laboratory, APG, Maryland, August 1989.

19. N.H. Ethridge, "Blast Overturning Model for Ground Targets," BRL Report No. 1889, U.S. Army Ballistic Research Laboratory, Aberdeen Proving Ground, MD, June 1976.

20. McKinley, T.K., and Sweet, K., "Nuclear Weapons Effects Test Guidelines Program, Phase I: Methods Development for Airblast," Report No. SSS-R-87- 8378, S-CUBED Inc, La Jolla, CA, December 1986.

21. Atkinson, M., et al., "MINITRUCK User's Manual," Report No. HDL-CR-86-081-3, U.S. Army Laboratory Command, Harry Diamond Laboratories, Adelphi, MD, July 1986.

22. Taylor, W.J., "A Method for Predicting Blast Loads During the Diffraction Phase," Shock And Vibration Bulletin 42, Part 4, p. 135, January 1972.

23. Hankins, D.M., "Experimental Pressure Distributions and Force Coefficients on Block Forms for Varying Mach Number, Reynolds Number, and Yaw angle," SC-42-4(TR), January 1959.

24. Hoerner, S.F., "Fluid Dynamic Drag,", 1965, publication available from the author at 148 Busteed Drive, Midland Park, NJ 07432.

ELASTIC-PLASTIC ANALYSIS OF A STEEL PRESSURE VESSEL WRAPPED WITH MULTILAYERED COMPOSITES

Peter C.T. Chen
U.S. Army Armament, Research, and Engineering Center
Close Combat Armaments Center
Benet Laboratories
Watervliet, NY 12189-4050

ABSTRACT. An elastic-plastic analysis of stresses and strains in an internally pressurized, composite-jacketed cylinder is studied here. Each layer is orthotropic but with different material properties. Analytical expressions are derived for a steel pressure vessel wrapped with multilayered composites. Numerical results are obtained for three types of composite jackets. The interface pressure, hoop strains, and stresses in the liner and jacket are presented.

INTRODUCTION. In recent years there has been increasing emphasis on the use of composite materials in armament structures. A current problem in Army cannon design is to replace a portion of the steel wall thickness with a lighter material. The inner portion, steel liner, maintains the tube projectile interface and shields the composite from the extremely hot gases. The outer portion, composite jacket, is made of single or multilayered graphite-bismaleimide wound and wrapped on the steel liner. Two subscale models have been fabricated and tested [1,2]. An analytical elastic-plastic solution for the model with a single-layered composite jacket has been presented in a recent paper [3]. This paper covers an elastic-plastic analysis for the model with a multilayered composite jacket. Analytical solutions are presented separately for the composite-jacket and steel liner and then for the compound cylinder problem. Numerical results are obtained for loading within and beyond the elastic region up to failure.

COMPOSITE JACKET. The composite jacket is made of n layers bounded by radii $(r_1, r_2, \ldots, r_n, r_{n+1})$. Each layer is elastically orthotropic but with different material properties. The strain-stress relations for the k-th layer in cylindrical coordinates are given by

$$\begin{Bmatrix} \epsilon_r^{(k)} \\ \epsilon_\theta^{(k)} \\ \epsilon_z^{(k)} \end{Bmatrix} = \begin{bmatrix} 1/E_r & -\nu_{\theta r}/E_\theta & -\nu_{zr}/E_z \\ -\nu_{r\theta}/E_r & 1/E_\theta & -\nu_{z\theta}/E_\theta \\ -\nu_{rz}/E_r & -\nu_{\theta z}/E_\theta & 1/E_z \end{bmatrix}^{(k)} \begin{Bmatrix} \sigma_r^{(k)} \\ \sigma_\theta^{(k)} \\ \sigma_z^{(k)} \end{Bmatrix} \tag{1}$$

or

$$\epsilon_i^{(k)} = S_{ij}^{(k)}\, \sigma_j^{(k)} \quad (i,j = r,\theta,z) \tag{2}$$

where $S_{ij}^{(k)}$ are components of the compliance matrix. The superscript k refers to the k-th layer. In plane-strain conditions, the above strain-stress relations modify to

$$\begin{Bmatrix} \epsilon_r^{(k)} \\ \epsilon_\theta^{(k)} \end{Bmatrix} = \begin{bmatrix} \beta_{rr}^{(k)} & \beta_{r\theta}^{(k)} \\ \beta_{r\theta}^{(k)} & \beta_{\theta\theta}^{(k)} \end{bmatrix} \begin{Bmatrix} \sigma_r^{(k)} \\ \sigma_\theta^{(k)} \end{Bmatrix} \tag{3}$$

where

$$\begin{aligned} \beta_{rr}^{(k)} &= (1-\nu_{rz}^{(k)}\nu_{zr}^{(k)})/E_r^{(k)} \\ \beta_{r\theta}^{(k)} &= -(\nu_{\theta r}^{(k)}+\nu_{\theta z}^{(k)}\nu_{zr}^{(k)})/E_\theta^{(k)} \\ \beta_{\theta\theta}^{(k)} &= (1-\nu_{\theta z}^{(k)}\nu_{z\theta}^{(k)})/E_\theta^{(k)} \end{aligned} \tag{4}$$

The normal traction acting on the interface between (k-1)th and k-th layers is denoted by q_k. Then the general elastic solution for the k-th layer bounded by radii (r_k, r_{k+1}) and subjected to interface pressure (q_k, q_{k+1}) is given by [4]

$$\begin{aligned} \sigma_r^{(k)} &= (-a_k q_k + c_k q_{k+1})(r_{k+1}/r)^{g_{k+1}} + (a_k q_k - b_k q_{k+1})(r/r_{k+1})^{g_{k-1}} \\ \sigma_\theta^{(k)} &= g_k(a_k q_k - c_k q_{k+1})(r_{k+1}/r)^{g_{k+1}} + g_k(a_k q_k - b_k q_{k+1})(r/r_{k+1})^{q_{k-1}} \\ u^{(k)} &= r(\beta_{r\theta}^{(k)}\sigma_r^{(k)} + \beta_{\theta\theta}^{(k)}\sigma_\theta^{(k)}) \end{aligned} \tag{5}$$

where

$$\begin{aligned} d_k &= r_{k+1}/r_k \quad , \quad g_k = (\beta_{rr}^{(k)}/\beta_{\theta\theta}^{(k)})^{1/2} \\ c_k &= (d_k^{2g_k}-1)^{-1} \quad , \quad b_k = c_k d_k^{2g_k} \quad , \quad a_k = c_k d_k^{g_{k-1}} \end{aligned} \tag{6}$$

At the two ends of the k-th layer the expressions for the displacements and hoop stresses are

$$\begin{aligned} u_{k+1} &= (A_k q_k - B_k q_{k+1}) r_{k+1} \\ u_k &= (C_k q_k - D_k q_{k+1}) r_k \\ \sigma_\theta^{(k)} &= 2a_k g_k q_k - (b_k + c_k) g_k q_{k+1} \quad \text{at } r_{k+1} \\ \sigma_\theta^{(k)} &= (b_k + c_k) g_k q_k - 2a_k d_k^2 g_k q_{k+1} \quad \text{at } r_k \end{aligned}$$

where

$$\begin{aligned} A_k &= 2a_k g_k \beta_{\theta\theta}^{(k)} \quad , \quad B_k = \beta_{r\theta}^{(k)} + (b_k + c_k) g_k \beta_{\theta\theta}^{(k)} \\ C_k &= -\beta_{r\theta}^{(k)} + (b_k + c_k) g_k \beta_{\theta\theta}^{(k)} \quad , \quad D_k = 2a_k d_k^2 g_k \beta_{\theta\theta}^{(k)} \end{aligned} \tag{8}$$

At the interfaces $(r_k, k=2,\ldots,n)$ the displacements should be continuous and these require

$$A_{k-1} q_{k-1} - B_{k-1} q_k = C_k q_k - D_k q_{k+1} \tag{9}$$

Let $\bar{Q}_k = q_k/q_n$ for all k, then $\bar{Q}_{n+1} = 0$, $\bar{Q}_n = 1$, and we can calculate $\bar{Q}_{k-1}$ backward for k = n to 2 by

$$\bar{Q}_{k-1} = A_{k-1}^{-1}[(B_{k-1}+C_k)\bar{Q}_k - D_k\bar{Q}_{k+1}]$$

Normalizing by $\bar{Q}_1$ leads to

$$\bar{q}_k = q_k/q_1 \quad \text{for } k = 1,2,\ldots,n \tag{10}$$

i.e., the relative values for the interface pressures when $q_1 = 1$. We can also obtain the corresponding displacements $u_1,\ldots,u_n$, u_{n+1} at $r_1,\ldots,r_n,r_{n+1}$.

STEEL LINER. The steel liner of inside radius a and outer radius b is elastic-plastically isotropic and assumed to obey Tresca's yield criterion, the associated flow rule, and linear strain-hardening. The elastic solution for the steel liner subjected to internal pressure p and external pressure q is

$$\begin{matrix}\sigma_r \\ \sigma_\theta\end{matrix} = \{\mp(p-q)(b/r)^2 + p-q\ b^2/a^2\}/(b^2/a^2-1)$$

$$u/r = E^{-1}(1+\nu)[(p-q)(b/r)^2 + (1-2\nu)(p-q\ b^2/a^2)]/(b^2/a^2-1) \tag{11}$$

When the internal pressure p is large enough, part of the steel liner ($a \leq r \leq \rho$) will become plastic and ρ is the elastic-plastic interface. The elastic-plastic solution can be written in the elastic portion ($\rho \leq r \leq b$) as

$$\frac{E}{\sigma_o}\frac{u}{r} = \frac{1+\nu}{2}\frac{\rho^2}{r^2} + (1-\nu-2\nu^2)\left[\frac{1}{2}\frac{\rho^2}{b^2} - \frac{q}{\sigma_o}\right]$$

$$\begin{matrix}\sigma_r/\sigma_o \\ \sigma_\theta/\sigma_o\end{matrix} = \frac{1}{2}\left(\mp\frac{\rho^2}{r^2} + \frac{\rho^2}{b^2}\right) - \frac{q}{\sigma_o}$$

$$\sigma_z/\sigma_o = \nu\ \rho^2/b^2 - 2\nu\ q/\sigma_o \tag{12}$$

and in the plastic portion ($a \leq r \leq \rho$)

$$\frac{E}{\sigma_o}\frac{u}{r} = (1-\nu-2\nu^2)\frac{\sigma_r}{\sigma_o} + (1-\nu^2)\frac{\rho^2}{r^2}$$

$$\begin{matrix}\sigma_r/\sigma_o \\ \sigma_\theta/\sigma_o\end{matrix} = \mp\frac{1}{2}(1-\eta\beta+\eta\beta\frac{\rho^2}{r^2}) + \frac{1}{2}\frac{\rho^2}{b^2} - (1-\eta\beta)\ln\frac{\rho}{r} - \frac{q}{\sigma_o}$$

$$\sigma_z/\sigma_o = \nu\ \rho^2/b^2 - 2\nu(1-\eta\beta)\ln\frac{\rho}{r} - 2\nu\ q/\sigma_o$$

$$\bar{\epsilon}^p = \beta(\rho^2/r^2-1) \quad , \quad \eta\beta = \frac{m}{m + \frac{3}{4}\frac{(1-m)}{(1-\nu^2)}}$$

$$\eta = \frac{2}{\sqrt{3}}\frac{E}{\sigma_o}\frac{m}{1-m} \quad , \quad m = \frac{E_t}{E} \quad , \quad \sigma = \sigma_o(1+\eta\bar{\epsilon}^p) \tag{13}$$

where σ_o is the initial tensile yield stress and E_t is the tangent modulus in the plastic range of the stress-strain curve.

When the internal pressure is further increased, the steel liner will become fully-plastic. Using Tresca's yield criterion, the associated flow rule, and assuming linear strain-hardening, the fully-plastic solution derived in [3] is given below.

Subject to $\sigma_\theta \geq \sigma_z \geq \sigma_r$, the analytical expressions for the stresses and displacement are

$$\sigma_r = -p + \sigma_o(1-\eta\beta)\ell n(\frac{r}{a}) + \frac{1}{2}\frac{\eta\beta}{(1-\nu^2)}[\frac{b^2}{a^2} - \frac{b^2}{r^2}]E\phi$$

$$\sigma_\theta = \sigma_r + \sigma_o(1+\eta\bar{\epsilon}^p$$

$$ru = E^{-1}(1-2\nu)(1+\nu)r^2\sigma_r + \phi\, b^2 \qquad (14)$$

where

$$\phi = u_b/b + (1-2\nu)(1+\nu)E^{-1}q$$

$$\bar{\epsilon}^p = \frac{2}{\sqrt{3}}[\phi\, b^2/r^2 - (1-\nu^2)\sigma_o/E]/[1 + \frac{2}{\sqrt{3}}(1-\nu^2)\eta\sigma_o/E]$$

COMPOUND CYLINDER. The compound cylinder consists of an inner steel liner and an outer composite jacket. The steel liner of inside radius a and outer radius b is wrapped by a multilayered composite jacket. The displacement and normal traction at the interface between the liner and jacket should be continuous, i.e., $q = q_1$ and $u_b = u_1$. From these conditions we can determine the relations between p and q.

When the internal pressure p is small, an explicit functional relation exists.

$$\frac{2p}{q} = \frac{(b^2/a^2-1)}{(1-\nu^2)}[E(C_1-D_1\bar{q}_2) + (1-\nu-2\nu^2)] + 2 \qquad (15)$$

where every term in the right-hand side is known. The displacement at the bore can also be expressed as an explicit function of p,

$$(\frac{b^2}{a^2} - 1)\frac{E}{p}\frac{u_a}{a} = (1+\nu)\frac{b^2}{a^2} + (1-\nu-2\nu^2) - 2(1-\nu^2)\frac{b^2}{a^2}\frac{q}{p} \qquad (16)$$

When the internal pressure is large enough, part of the steel liner will become plastic. The elastic-plastic solution is given in terms of the parameter ρ. The conditions of continuity require

$$\frac{q}{\sigma_o} = \frac{(1-\nu^2)\rho^2/b^2}{(1-\nu-2\nu^2) + E(C_1-D_1\bar{q}_2)} \qquad (17)$$

This, together with

$$\frac{p}{\sigma_o} = \frac{q}{\sigma_o} + \frac{1}{2}\left(1 - \frac{\rho^2}{b^2}\right) + (1-\eta\beta)\ln\frac{\rho}{a} + \frac{\eta\beta}{2}\left(\frac{\rho^2}{a^2} - 1\right) \tag{18}$$

serves to give an implicit relation between p and q. By letting ρ = a and b, we can determine the lower limits p*, q*, u_a*, u_b* and the upper limits p**, q**, u_a**, u_b**, respectively.

When the internal pressure p is further increased, i.e., p > p**, u_a > u_a**, u_b > u_b**, the conditions of continuity lead to

$$\phi = q[(C_1 - D_1\bar{q}_2) + (1-\nu-2\nu^2)/E] \tag{19}$$

and

$$\frac{p}{\sigma_o} = (1-\eta\beta)\ln\frac{b}{a} + \frac{q}{\sigma_o}\left\{1 + \frac{\eta\beta(b^2/a^2-1)}{2(1-\nu^2)}\left[E(C_1 - D_1\bar{q}_2) + (1-\nu-2\nu^2)\right]\right\} \tag{20}$$

It should be pointed out that the pressure q and the displacement u_b at the interface are linear functions of internal pressure p. The bore displacement u_a can be written as

$$\frac{u_a}{a} = -(1-2\nu)(1+\nu)\frac{p}{E} + \frac{b^2}{a^2}\phi \tag{21}$$

which is also a linear function of internal pressure p.

NUMERICAL RESULTS. Given any value of internal pressure, we can obtain numerical results for the stresses and strains in the radial and tangential directions and also for the displacement at any radial position in a steel pressure vessel wrapped with multilayered composites. The steel liner for the subscale test specimens [1] had an inner diameter of 2.0 inches and an outer diameter of 2.34 inches. The steel was 4130 seamless mechanical tubing heat treated to a hardness of 34 to 36 Rockwell "C." A standard ASTM tensile test was conducted to determine the 0.1 percent offset yield strength (120 Ksi) and the ultimate tensile strength (140 Ksi). The composite jacket is a graphite-bismaleimide produced by Fiberite Corporation. Its cure temperature is 450°F and it is wound and wrapped on the steel liner in the same manner as the full-scale gun tube specimen denoted as CTL III. The layup is again approximately half-scale and made up of two longitudinal layers alternating with two circumferential layers. Sixteen layers are applied in this way. Lamina properties for this material are given in Table 1. For the purpose of comparison, numerical results are obtained for four types of composite jackets as shown in Table 2. Cases 3 and 4 represent four hoop-axial and axial-hoop alternating layers, respectively, while cases 1 and 2 represent eight axial and hoop layers, respectively. The total thickness of each composite jacket is 0.12 inch and the steel liner is assumed to be linear strain-hardening with a = 1 inch, b = 1.17 inches, σ_o = 120 Ksi, m = 0.04. In addition to the lower and upper limits (p* and p**) of internal pressure in the elastic-plastic range, we also show in Table 2 two other limits ($P_{0.8}$ and $P_{1.3}$) which correspond to the internal pressure when u_b/b = 0.8 and 1.3 percent, respectively. It should be noted that u_b/b is the maximum hoop strain in the composite. Brittle failure of the composite material is assumed to occur at a maximum strain of 0.8 or 1.3 percent. The limits ($P_{0.8}$ or $P_{1.3}$) will be the maximum values of internal pressure these compound tubes can contain without failure.

The pressure at the interface between the liner and jacket has been obtained as a function of internal pressure and the results for the first three cases are shown in Figure 1. The results of the hoop strains at the bore, interface between the liner and jacket, and outside surface for three cases are shown in Figures 2, 3, and 4, respectively, as functions of internal pressure. The complete (including elastic, elastic-plastic, and fully-plastic) ranges of loadings up to $P_{0.8}$ have been considered. These numerical results for the strains are presented here for future comparisons with experimental results. The results of hoop stresses in the liner at the bore are shown in Figure 5 as functions of internal pressure. It should be noted that the relation changes drastically when yielding occurs. The results of hoop stresses in the liner at the interface are shown in Figure 6 as functions of internal pressure. The relation changes from linear to nonlinear when yielding sets in and more significant change occurs when the fully-plastic state is reached. The distribution of hoop stresses in the liner and jacket can be obtained at any given value of internal pressure. In Figures 7, 8, and 9 we present the numerical results for three cases of composite jackets at three values of internal pressure, i.e., p = p*, p** and when half of the liner is plastic. The values of internal pressure when half of the liner is plastic are p = 18.61, 23.86, 21.41 Ksi for cases 1, 2, 3, respectively. The values of two limits, p* and p**, are given in Table 2 for all four cases. When the composite jacket is made of axial lamina only, the hoop stresses in the jacket are very small as shown in Figure 7. When the liner is wrapped by hoop lamina only, the hoop stresses in the jacket become larger as the internal pressure increases as shown in Figure 8. When the jacket consists of alternating hoop-axial lamina, the hoop stresses become discontinuous not only at the interface between the liner and jacket but also at all other interfaces between axial and hoop lamina.

REFERENCES

1. M.A. Scavullo, M.D. Witherell, K. Miner, T.E. O'Brien, and W. Yaiser, "Experimental and Analytical Investigation of a Steel Pressure Vessel Overwrapped With Graphite Bismaleimide," ARDEC Technical Report ARCCB-TR-87013, Benet Weapons Laboratory, Watervliet, NY, May 1987.

2. M.D. Witherell and M.A. Scavullo, "An Investigation of Stresses and Strains in an Internally Pressurized, Composite-Jacketed Steel Cylinder," ARDEC Technical Report ARCCB-TR-88042, Benet Laboratories, Watervliet, NY, November 1988.

3. P.C.T. Chen, "Elastic-Plastic Analysis of a Thick-Walled Composite Tube Subjected to Internal Pressure," ARDEC Technical Report ARCCB-TR-89027, Benet Laboratories, Watervliet, NY, October 1989.

4 S.W. Tsai, Composite Design, 3rd Edition, Think Composites, Dayton, OH, 1987.

TABLE 1. ELASTIC CONSTANTS OF STEEL AND COMPOSITE MATERIALS

Material	E_θ x10^6 psi	E_r x10^6 psi	E_z x10^6 psi	ν_{rz}	$\nu_{r\theta}$	$\nu_{z\theta}$
Hoop lamina Im6	21.0	1.0	1.0	0.40	0.02	0.02
Axial lamina G50	1.3	1.3	31.0	0.01	0.39	0.39
Steel 4130	30.8	30.8	30.8	0.30	0.30	0.30

TABLE 2. LIMITS OF INTERNAL PRESSURE FOR FOUR CASES

Case	Layup	p*	p**	$P_{0.8}$	$P_{1.3}$
1	$(90°)_8$	16.49	19.44	21.26	23.34
2	$(0°)_8$	20.95	25.55	35.20	45.99
3	$(0°,90°)_4$	18.87	22.70	28.59	35.25
4	$(90°,0°)_4$	18.80	22.60	28.38	34.90

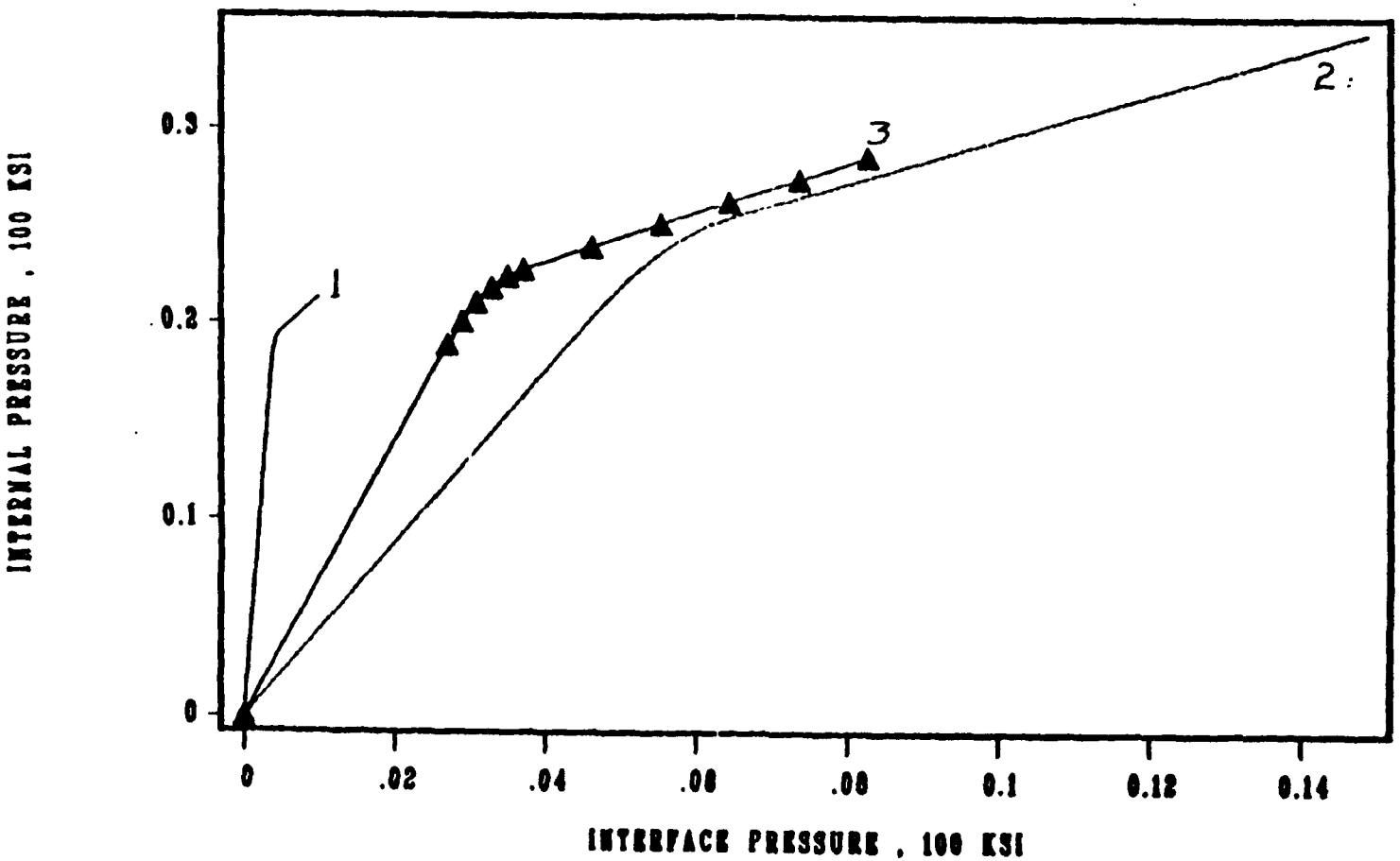


Figure 1. Interface pressure as a function of internal pressure.

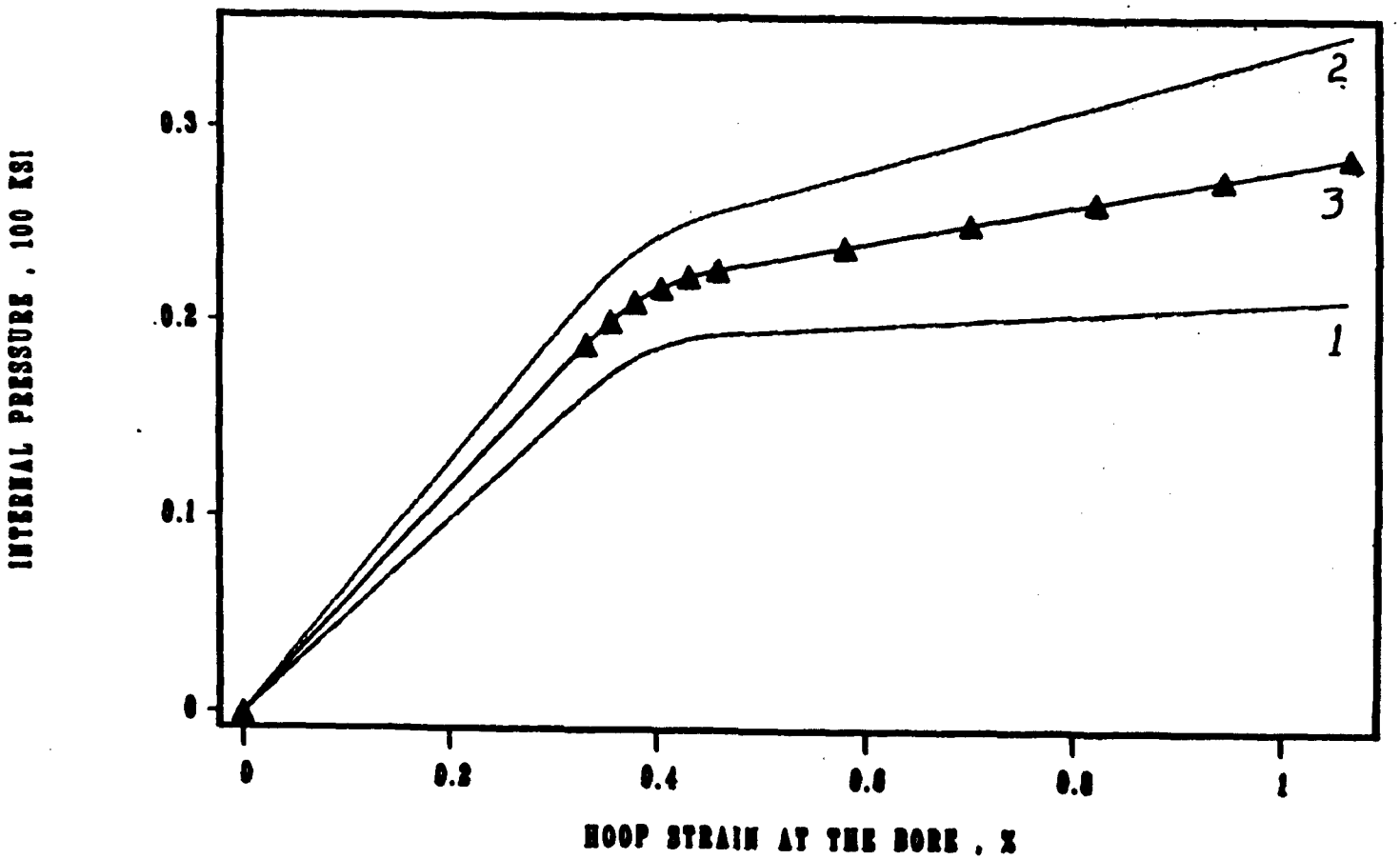


Figure 2. Hoop strain at the bore as a function of internal pressure

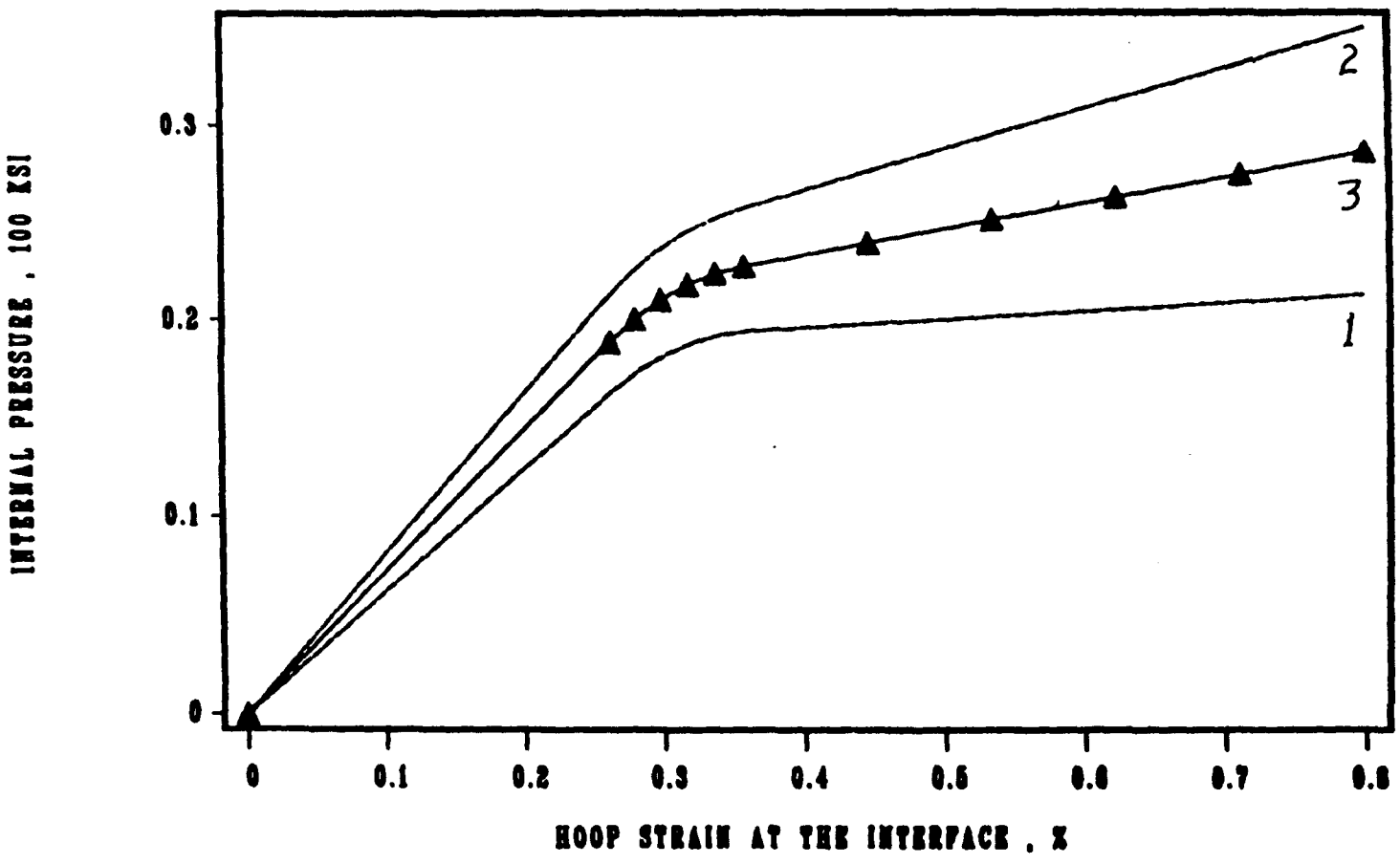


Figure 3. Hoop strain at the interface as a function of internal pressure.

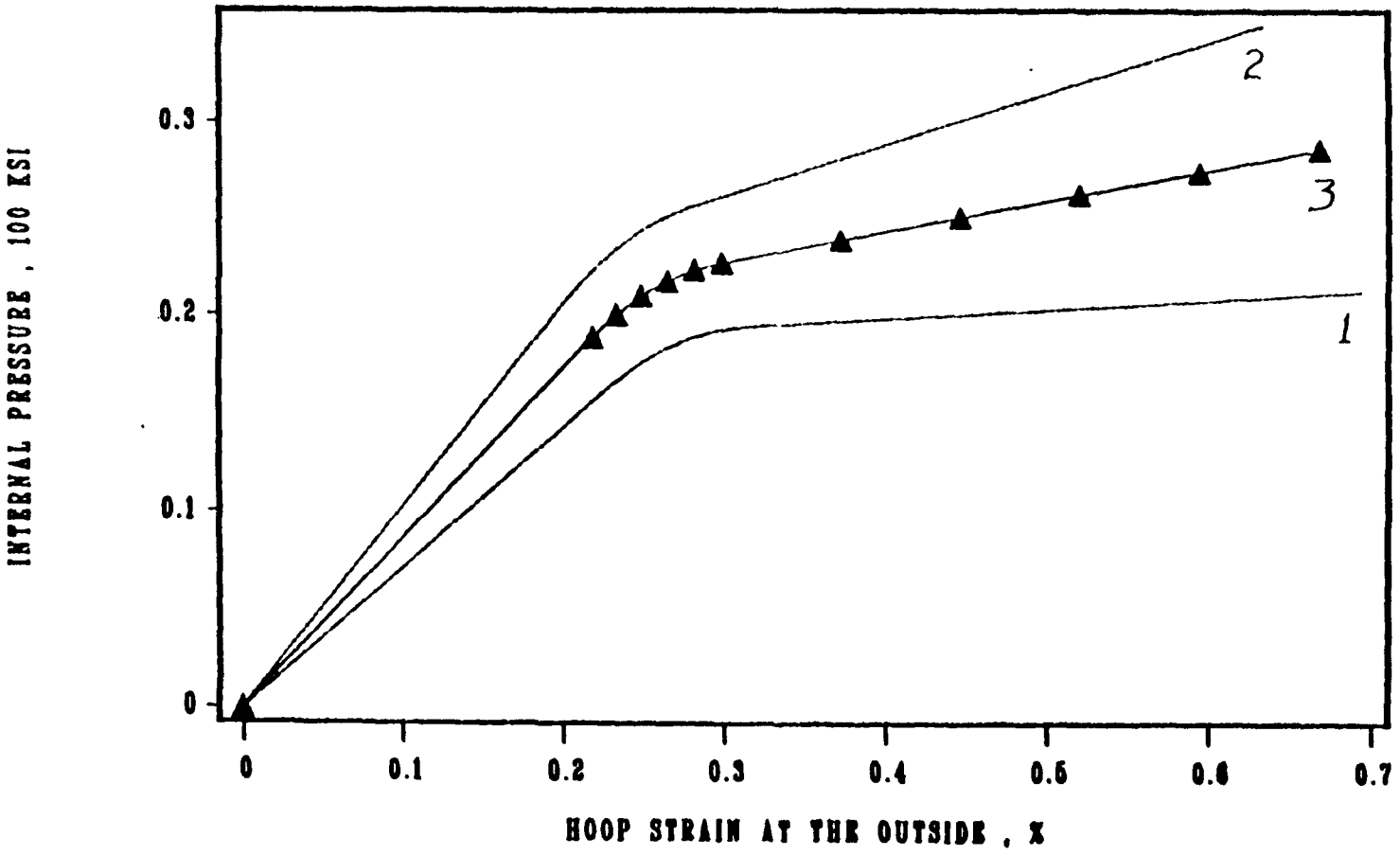


Figure 4. Hoop strain at the outside as a function of internal pressure.

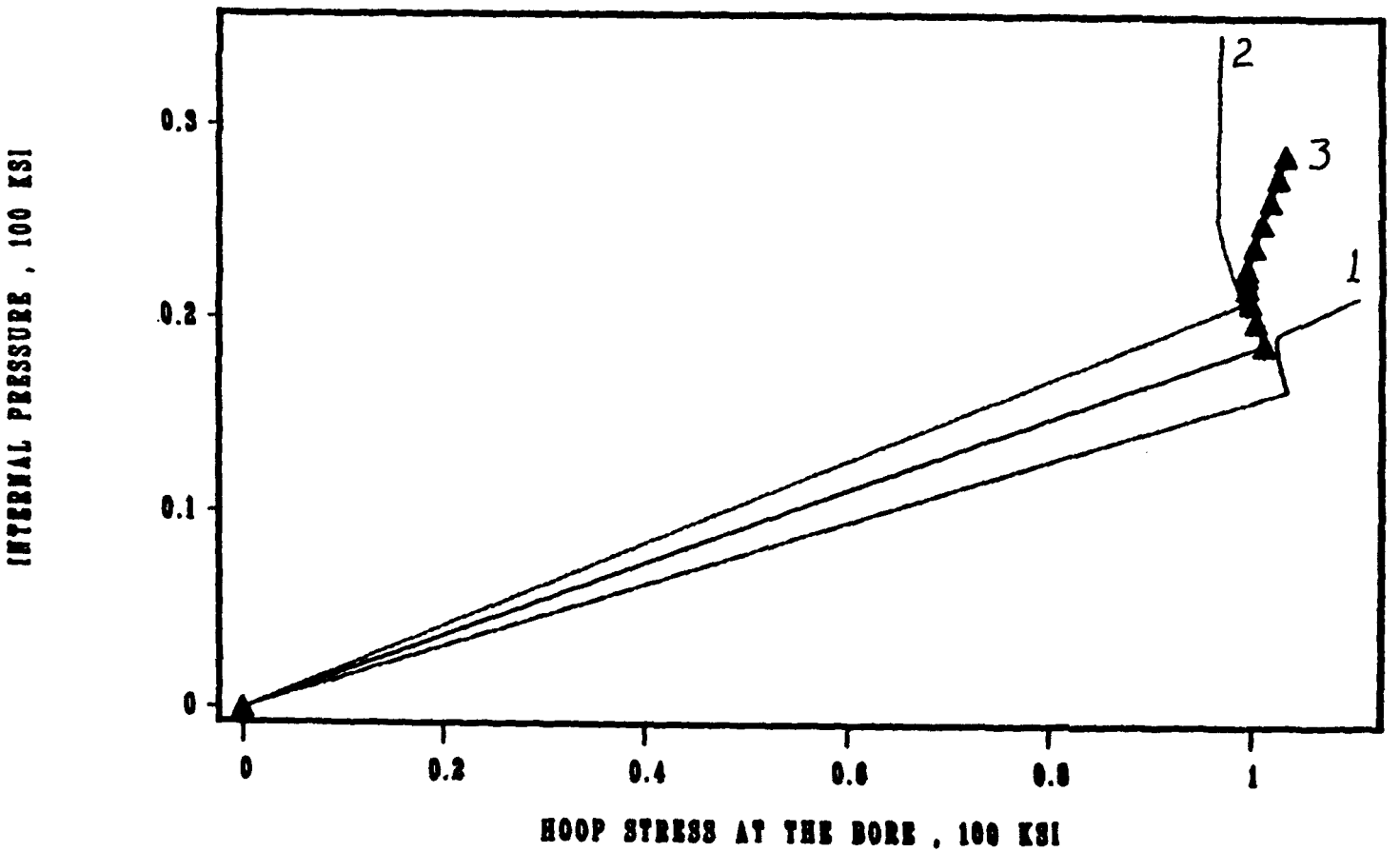


Figure 5. Hoop stress at the bore as a function of internal pressure.

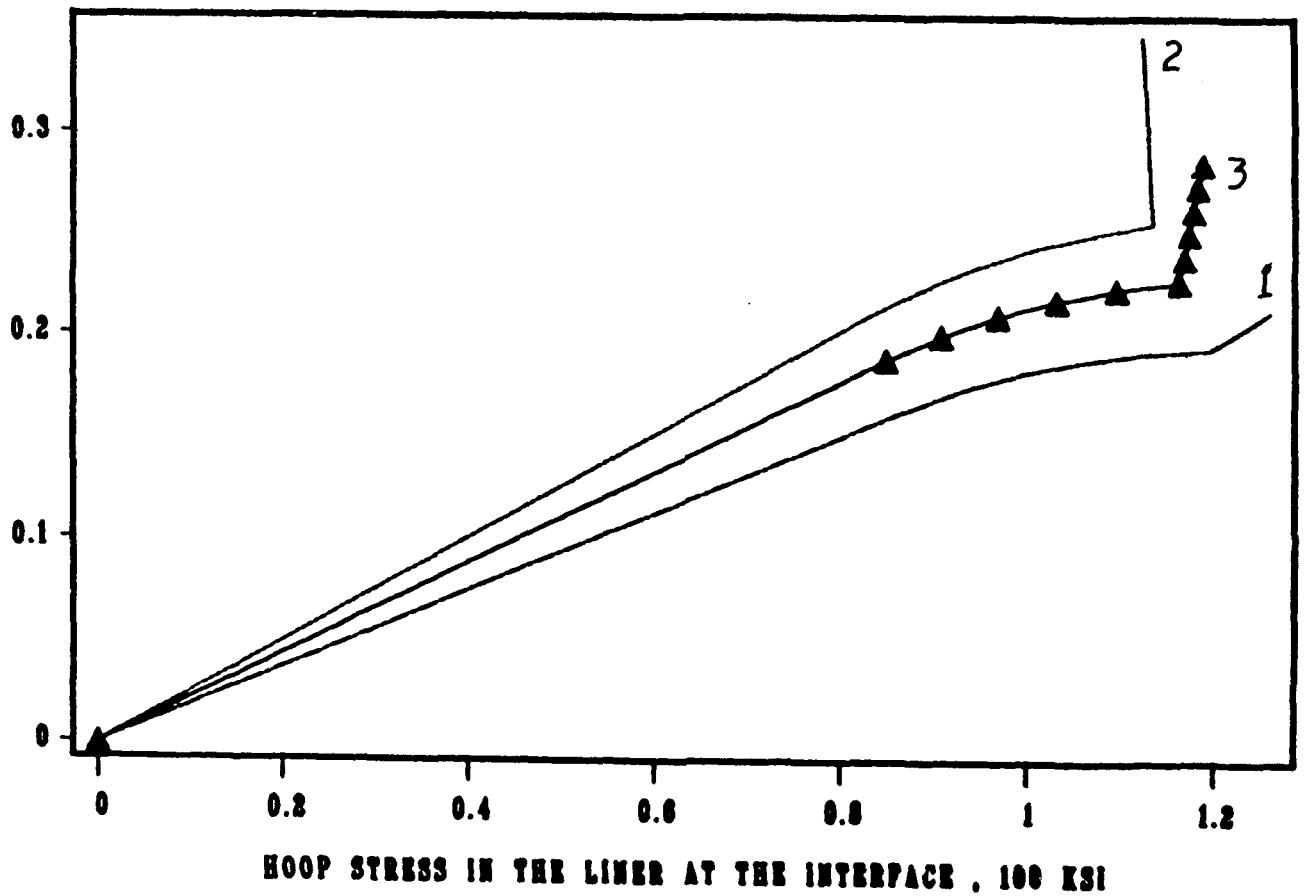


Figure 6. Hoop stress in the liner at the interface as a function of internal pressure.

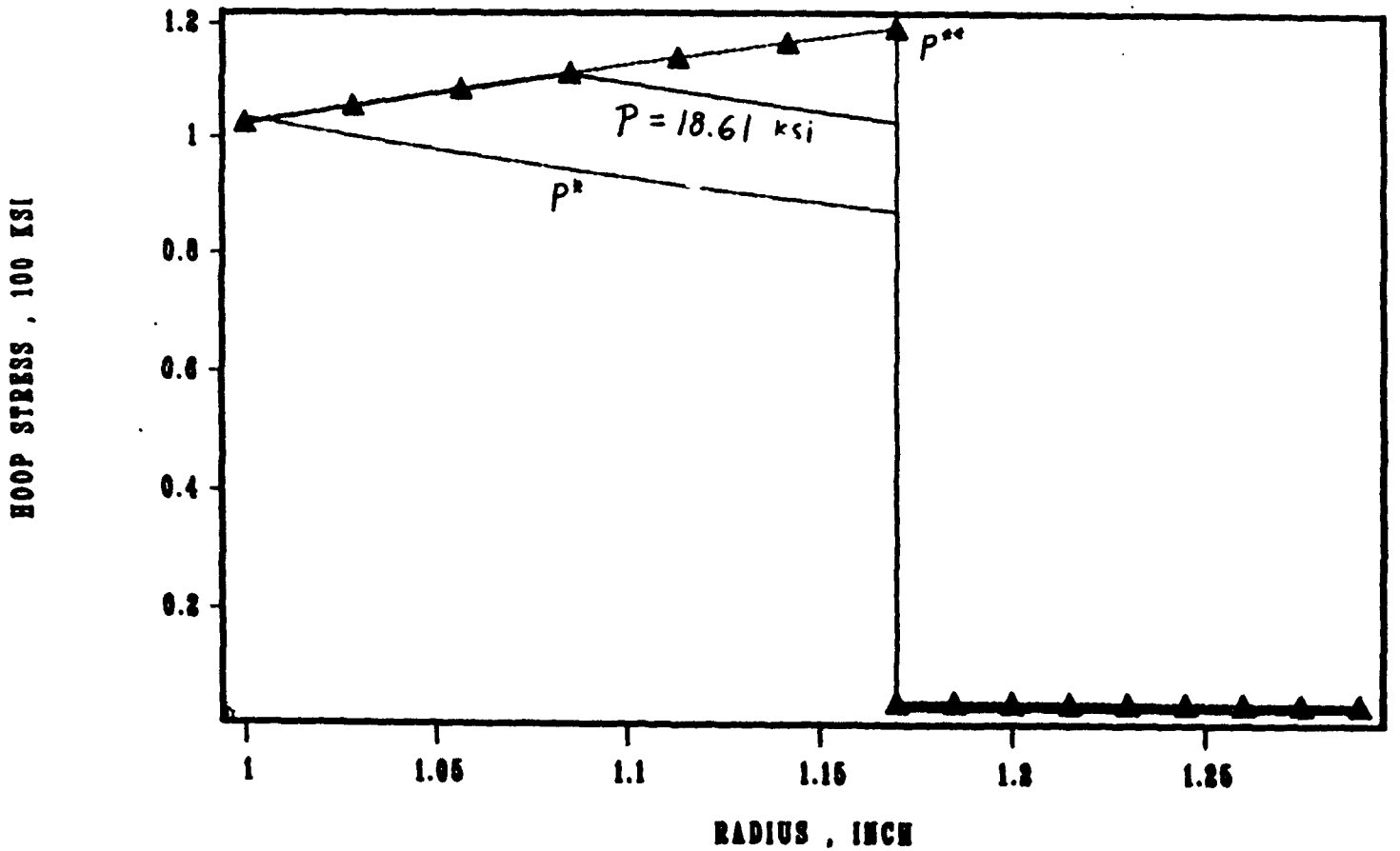


Figure 7. Distribution of hoop stresses in the liner and jacket for case 1.

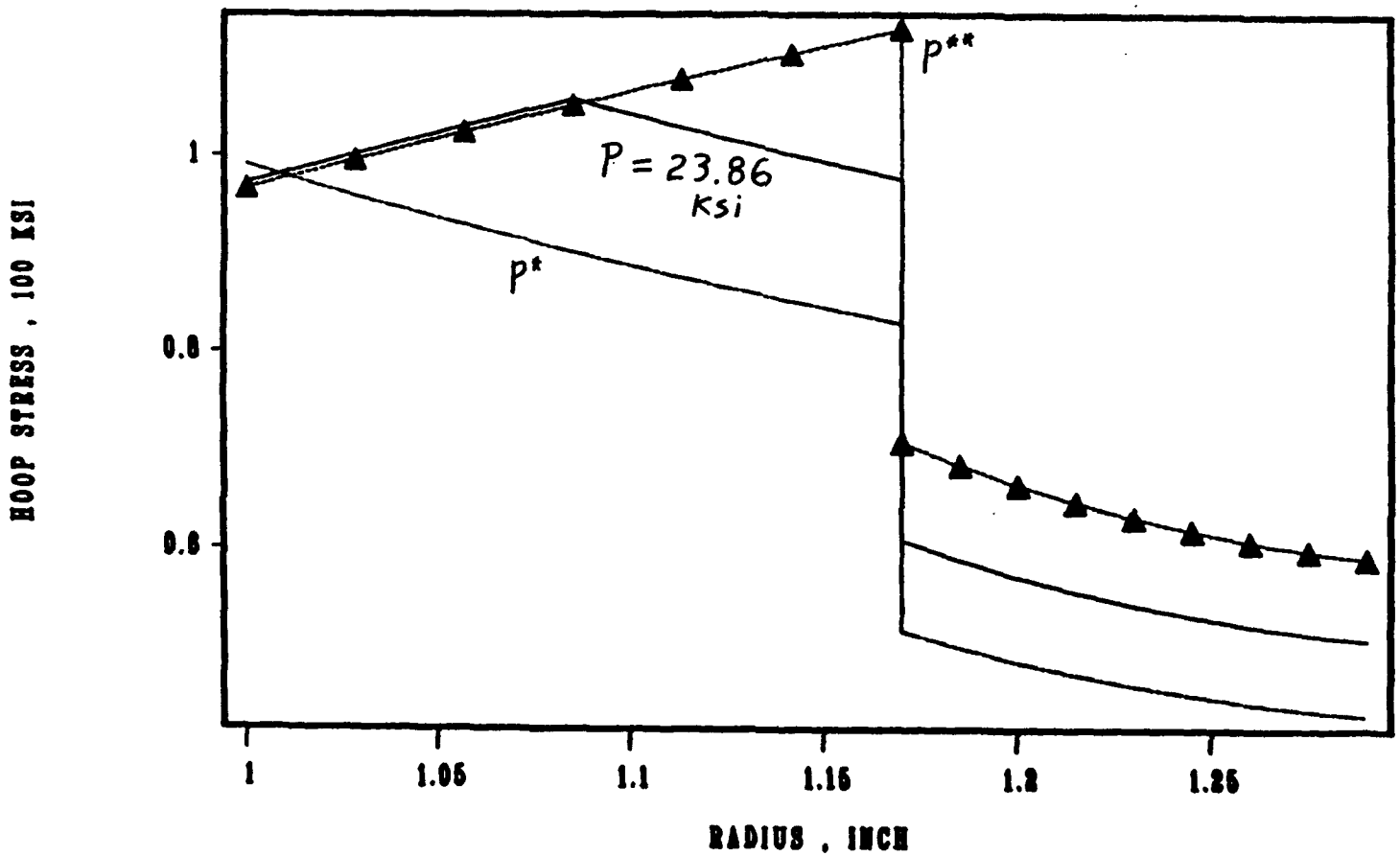


Figure 8. Distribution of hoop stresses in the liner and jacket for case 2.

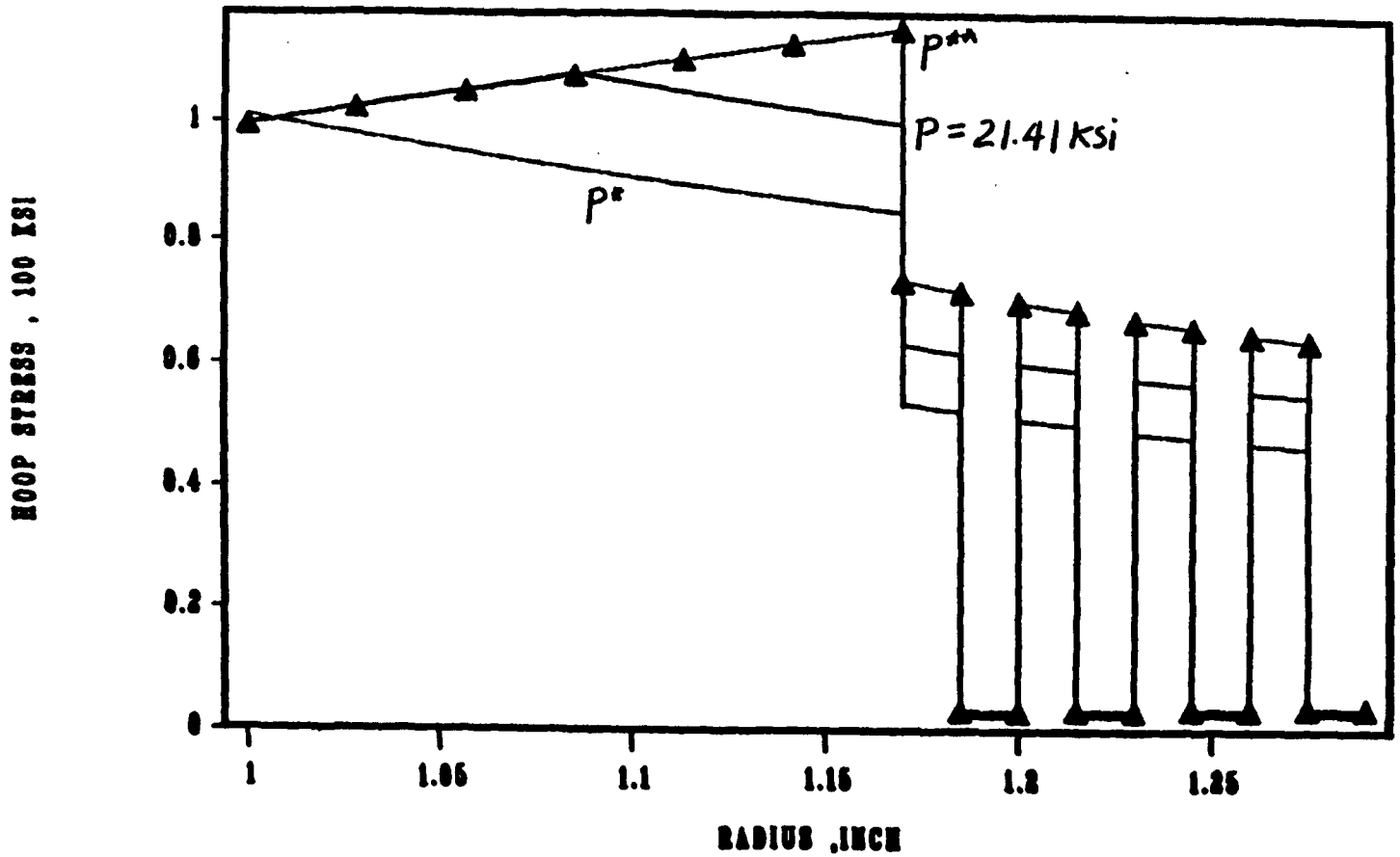


Figure 9. Distribution of hoop stresses in the liner and jacket for case 3.

Expression Swell Analysis of the Computation of Matrix Characteristic Polynomials*

Michael Wester**
Department of Mathematics and Statistics
University of New Mexico
Albuquerque, New Mexico 87131

Abstract

A common problem that occurs when performing exact computations is expression swell, in which the size of expressions involved in a calculation grow dramatically. An important special case of this phenomenon is intermediate expression swell where, during the middle stages of a calculation, intermediate expressions can expand substantially, but the final results of the calculation are comparatively simple. Computing the characteristic polynomial of a matrix is a good calculation for examining the effects of expression swell, which are very striking, even for small matrices. A number of case studies involving matrices consisting of integer and rational entries have been performed. In addition, some worst case theoretical analyses have been done and the results compared with those of the case studies.

1 Introduction

A common phenomenon that occurs when performing exact computations is expression swell, in which the size of numbers and expressions involved in a calculation grow dramatically as the calculation progresses. A typical example of this phenomenon is the calculation of the roots of a third or fourth degree univariate polynomial which does not factor over the rational numbers and so the cubic or quartic formula must be used. As a particular example

*This work is partially sponsored by the Army Research Office and is being done under the direction of Professor Stanly Steinberg as part of the requirements for a Ph.D.

** The author of this paper presented it at the Sixth Army Conference on Applied Mathematics and Computing.

[Gro87], the characteristic polynomial of the 9×9 real symmetric Hankel matrix

$$\mathcal{H} = \begin{pmatrix} -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 \\ -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 \\ -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 \\ -1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 & 1 \\ -1 & 1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 \end{pmatrix}$$

is

$$\lambda^9 + \lambda^8 - 40\lambda^7 - 24\lambda^6 + 240\lambda^5 + 144\lambda^4 ,$$

which can be factored over the rational numbers into

$$\lambda^4(\lambda + 6)(\lambda^4 - 5\lambda^3 - 10\lambda^2 + 36\lambda + 24) .$$

To complete the solution of the eigenvalue problem, the roots of the last factor must be extracted using the quartic formula. One of the roots is shown in Figure 1. An estimate of the size of this expression can be obtained by counting the number of operators and atomic operands involved in its construction. In this case, using the expression's internal representation in MACSYMA, a size of 300 was computed. The other three roots are of the same complexity and also have sizes of 300. Of course, if these roots were to be evaluated numerically, each root would be reduced to a single complex floating point number (which would have a size of 5).

An important special case of the expression swell phenomenon is intermediate expression swell. This refers to a condition where, during the middle stages of a calculation, intermediate expressions can expand substantially, but the final results of the calculation are comparatively simple. A typical example here is the verification of a trigonometric or tensor identity. As an example of the latter, Figure 2 presents the results of MACSYMA computing the left-hand side of the Bianchi identity for a symmetric connection

$$K_j^{\ell}{}_{hk|p} + K_j^{\ell}{}_{kp|h} + K_j^{\ell}{}_{ph|k}$$

in terms of Christoffel symbols of the second kind. Here, K is the Riemann curvature tensor. This sum contains 72 terms, each of which is a product of 2 or 3 Christoffel symbols, for a total of 180 Christoffel symbols. However, upon simplifying this expression by consistently renaming the dummy indices, the simple result of zero is obtained which verifies the identity.

Expression swell is a major problem in symbolic mathematical computations. As an expression grows in size, it takes up more and more memory and/or disk space while also

Figure 1.

taking more and more time to be manipulated. A shortage of either of these resources can cause a computation to fail, even if the final result is known to be *relatively simple*. Sometimes a computation can be reorganized so that it uses less resources (or more of one and less of another) and thus succeeds where previously it failed. Sometimes nothing will help but the acquisition of more space (memory, disk, etc.) and/or an increase in the processing speed (bringing the time of a computation down to a reasonable level).

In the past, discussion of expression swell in symbolic mathematical computations has often been anecdotal. In this paper, I will present some quantitative results of the effects of expression swell when computing the characteristic polynomial (and the determinant) of a matrix. These results will be a combination of case studies involving matrices consisting of integer and rational entries and some theoretical worst case (and other) analyses. Many of the findings are quite striking, even for small matrices.

$$
\begin{aligned}
&- \Gamma_{\#19}{}^{\ell}{}_{h}\Gamma_{j}{}^{\#19}{}_{\#27}\Gamma_{p}{}^{\#27}{}_{k} + \Gamma_{j}{}^{\ell}{}_{h,\#25}\Gamma_{p}{}^{\#25}{}_{k} + \Gamma_{\#19}{}^{\ell}{}_{\#22}\Gamma_{j}{}^{\#19}{}_{h}\Gamma_{p}{}^{\#22}{}_{k} - \Gamma_{j}{}^{\ell}{}_{\#20,h}\Gamma_{p}{}^{\#20}{}_{k} \\
&+ \Gamma_{j}{}^{\ell}{}_{\#18,k}\Gamma_{p}{}^{\#18}{}_{h} + \Gamma_{\#10}{}^{\ell}{}_{k}\Gamma_{j}{}^{\#10}{}_{\#16}\Gamma_{p}{}^{\#16}{}_{h} - \Gamma_{\#10}{}^{\ell}{}_{\#12}\Gamma_{j}{}^{\#10}{}_{k}\Gamma_{p}{}^{\#12}{}_{h} - \Gamma_{j}{}^{\ell}{}_{k,\#11}\Gamma_{p}{}^{\#11}{}_{h} \\
&+ \Gamma_{j}{}^{\ell}{}_{\#9,h}\Gamma_{k}{}^{\#9}{}_{p} + \Gamma_{\#1}{}^{\ell}{}_{h}\Gamma_{j}{}^{\#1}{}_{\#7}\Gamma_{k}{}^{\#7}{}_{p} - \Gamma_{\#1}{}^{\ell}{}_{\#3}\Gamma_{j}{}^{\#1}{}_{h}\Gamma_{k}{}^{\#3}{}_{p} - \Gamma_{j}{}^{\ell}{}_{h,\#2}\Gamma_{k}{}^{\#2}{}_{p} + \Gamma_{j}{}^{\ell}{}_{p,\#18}\Gamma_{k}{}^{\#18}{}_{h} \\
&+ \Gamma_{\#10}{}^{\ell}{}_{\#15}\Gamma_{j}{}^{\#10}{}_{p}\Gamma_{k}{}^{\#15}{}_{h} - \Gamma_{\#10}{}^{\ell}{}_{p}\Gamma_{j}{}^{\#10}{}_{\#13}\Gamma_{k}{}^{\#13}{}_{h} - \Gamma_{j}{}^{\ell}{}_{\#11,p}\Gamma_{k}{}^{\#11}{}_{h} - \Gamma_{h}{}^{\#20}{}_{k}\Gamma_{j}{}^{\ell}{}_{p,\#20} \\
&+ \Gamma_{\#9}{}^{\ell}{}_{k,h}\Gamma_{j}{}^{\#9}{}_{p} + \Gamma_{\#1}{}^{\ell}{}_{h}\Gamma_{\#7}{}^{\#1}{}_{k}\Gamma_{j}{}^{\#7}{}_{p} - \Gamma_{\#1}{}^{\ell}{}_{k}\Gamma_{\#4}{}^{\#1}{}_{h}\Gamma_{j}{}^{\#4}{}_{p} + \Gamma_{\#19}{}^{\ell}{}_{h}\Gamma_{\#27}{}^{\#19}{}_{k}\Gamma_{j}{}^{\#27}{}_{p} \\
&+ \Gamma_{\#20}{}^{\ell}{}_{k}\Gamma_{j}{}^{\#20}{}_{p,h} - \Gamma_{\#2}{}^{\ell}{}_{h,k}\Gamma_{j}{}^{\#2}{}_{p} + \Gamma_{\#19}{}^{\ell}{}_{h}\Gamma_{j}{}^{\#19}{}_{p,k} - \Gamma_{\#19}{}^{\ell}{}_{\#26}\Gamma_{h}{}^{\#26}{}_{k}\Gamma_{j}{}^{\#19}{}_{p} \\
&+ \Gamma_{\#19}{}^{\#26}{}_{h}\Gamma_{\#26}{}^{\ell}{}_{k}\Gamma_{j}{}^{\#19}{}_{p} - \Gamma_{\#19}{}^{\#26}{}_{k}\Gamma_{\#26}{}^{\ell}{}_{h}\Gamma_{j}{}^{\#19}{}_{p} + \Gamma_{\#19}{}^{\ell}{}_{h,k}\Gamma_{j}{}^{\#19}{}_{p} - \Gamma_{\#18}{}^{\ell}{}_{h}\Gamma_{j}{}^{\#18}{}_{p,k} \\
&- \Gamma_{\#10}{}^{\ell}{}_{k}\Gamma_{\#16}{}^{\#10}{}_{h}\Gamma_{j}{}^{\#16}{}_{p} - \Gamma_{\#10}{}^{\ell}{}_{k}\Gamma_{j}{}^{\#10}{}_{p,h} + \Gamma_{\#10}{}^{\#15}{}_{h}\Gamma_{\#15}{}^{\ell}{}_{k}\Gamma_{j}{}^{\#10}{}_{p} - \Gamma_{\#10}{}^{\#15}{}_{k}\Gamma_{\#15}{}^{\ell}{}_{h}\Gamma_{j}{}^{\#10}{}_{p} \\
&- \Gamma_{\#10}{}^{\ell}{}_{k,h}\Gamma_{j}{}^{\#10}{}_{p} + \Gamma_{h}{}^{\#9}{}_{p}\Gamma_{j}{}^{\ell}{}_{k,\#9} - \Gamma_{\#9}{}^{\ell}{}_{p}\Gamma_{j}{}^{\#9}{}_{k,h} - \Gamma_{\#1}{}^{\ell}{}_{h}\Gamma_{\#7}{}^{\#1}{}_{p}\Gamma_{j}{}^{\#7}{}_{k} - \Gamma_{\#19}{}^{\ell}{}_{h}\Gamma_{\#27}{}^{\#19}{}_{p}\Gamma_{j}{}^{\#27}{}_{k} \\
&+ \Gamma_{\#25}{}^{\ell}{}_{h,p}\Gamma_{j}{}^{\#25}{}_{k} + \Gamma_{\#19}{}^{\ell}{}_{p}\Gamma_{\#23}{}^{\#19}{}_{h}\Gamma_{j}{}^{\#23}{}_{k} - \Gamma_{\#20}{}^{\ell}{}_{p,h}\Gamma_{j}{}^{\#20}{}_{k} + \Gamma_{\#10}{}^{\ell}{}_{p}\Gamma_{\#13}{}^{\#10}{}_{h}\Gamma_{j}{}^{\#13}{}_{k} \\
&+ \Gamma_{\#11}{}^{\ell}{}_{h}\Gamma_{j}{}^{\#11}{}_{k,p} + \Gamma_{\#10}{}^{\ell}{}_{p}\Gamma_{j}{}^{\#10}{}_{k,h} - \Gamma_{\#10}{}^{\#12}{}_{h}\Gamma_{\#12}{}^{\ell}{}_{p}\Gamma_{j}{}^{\#10}{}_{k} + \Gamma_{\#10}{}^{\#12}{}_{p}\Gamma_{\#12}{}^{\ell}{}_{h}\Gamma_{j}{}^{\#10}{}_{k} \\
&+ \Gamma_{\#10}{}^{\ell}{}_{p,h}\Gamma_{j}{}^{\#10}{}_{k} - \Gamma_{\#1}{}^{\ell}{}_{h}\Gamma_{j}{}^{\#1}{}_{k,p} + \Gamma_{\#1}{}^{\ell}{}_{\#6}\Gamma_{h}{}^{\#6}{}_{p}\Gamma_{j}{}^{\#1}{}_{k} - \Gamma_{\#1}{}^{\#6}{}_{h}\Gamma_{\#6}{}^{\ell}{}_{p}\Gamma_{j}{}^{\#1}{}_{k} \\
&+ \Gamma_{\#1}{}^{\#6}{}_{p}\Gamma_{\#6}{}^{\ell}{}_{h}\Gamma_{j}{}^{\#1}{}_{k} - \Gamma_{\#1}{}^{\ell}{}_{h,p}\Gamma_{j}{}^{\#1}{}_{k} + \Gamma_{\#1}{}^{\ell}{}_{k}\Gamma_{\#4}{}^{\#1}{}_{p}\Gamma_{j}{}^{\#4}{}_{h} - \Gamma_{\#25}{}^{\ell}{}_{k}\Gamma_{j}{}^{\#25}{}_{h,p} \\
&- \Gamma_{\#19}{}^{\ell}{}_{p}\Gamma_{\#23}{}^{\#19}{}_{k}\Gamma_{j}{}^{\#23}{}_{h} + \Gamma_{\#2}{}^{\ell}{}_{p}\Gamma_{j}{}^{\#2}{}_{h,k} - \Gamma_{\#19}{}^{\ell}{}_{p}\Gamma_{j}{}^{\#19}{}_{h,k} + \Gamma_{\#19}{}^{\#22}{}_{k}\Gamma_{\#22}{}^{\ell}{}_{p}\Gamma_{j}{}^{\#19}{}_{h} \\
&- \Gamma_{\#19}{}^{\#22}{}_{p}\Gamma_{\#22}{}^{\ell}{}_{k}\Gamma_{j}{}^{\#19}{}_{h} - \Gamma_{\#19}{}^{\ell}{}_{p,k}\Gamma_{j}{}^{\#19}{}_{h} + \Gamma_{\#18}{}^{\ell}{}_{p,k}\Gamma_{j}{}^{\#18}{}_{h} + \Gamma_{\#10}{}^{\ell}{}_{k}\Gamma_{\#16}{}^{\#10}{}_{p}\Gamma_{j}{}^{\#16}{}_{h} \\
&- \Gamma_{\#10}{}^{\ell}{}_{p}\Gamma_{\#13}{}^{\#10}{}_{k}\Gamma_{j}{}^{\#13}{}_{h} - \Gamma_{\#11}{}^{\ell}{}_{k,p}\Gamma_{j}{}^{\#11}{}_{h} + \Gamma_{\#1}{}^{\ell}{}_{k}\Gamma_{j}{}^{\#1}{}_{h,p} + \Gamma_{\#1}{}^{\#3}{}_{k}\Gamma_{\#3}{}^{\ell}{}_{p}\Gamma_{j}{}^{\#1}{}_{h} \\
&- \Gamma_{\#1}{}^{\#3}{}_{p}\Gamma_{\#3}{}^{\ell}{}_{k}\Gamma_{j}{}^{\#1}{}_{h} + \Gamma_{\#1}{}^{\ell}{}_{k,p}\Gamma_{j}{}^{\#1}{}_{h} - \Gamma_{\#1}{}^{\ell}{}_{k}\Gamma_{h}{}^{\#4}{}_{p}\Gamma_{j}{}^{\#1}{}_{\#4} + \Gamma_{h}{}^{\#25}{}_{k}\Gamma_{j}{}^{\ell}{}_{\#25,p} \\
&+ \Gamma_{\#19}{}^{\ell}{}_{p}\Gamma_{h}{}^{\#23}{}_{k}\Gamma_{j}{}^{\#19}{}_{\#23} - \Gamma_{h}{}^{\#2}{}_{p}\Gamma_{j}{}^{\ell}{}_{\#2,k}
\end{aligned}
$$

Figure 2.

2 Theoretical Expression Swell Analysis

Since expression swell is such an important problem in symbolic mathematical computations, a very useful ability is to be able to predict the extent of this phenomenon during the course of a particular calculation. One way to do this is to set up an algorithm and run an extensive series of calculations under a variety of initial conditions. An analysis of the results will provide an idea (perhaps a good one) of the progress of expression growth during the calculation. In a subsequent section, a statistical survey of determinant and characteristic polynomial calculations under a limited set of initial conditions is presented. The problem with this approach to quantitatively assessing the expression swell inherent in a given calculation is that typically, a large number of sample calculations need to be made, which can be an expensive and time consuming proposition.

Generally, a better way of charting the course of expression growth is to make some kind of theoretical prediction. A common type of theoretical estimate is a bounding calculation. For an expression swell analysis of an algorithm, there are two important kinds of bounding calculations that can be performed. One involves worst case behavior in which the expression size is maximized for the outcome of each mathematical operation (the results of an addition, of applying a function to its arguments, of applying an operator to a function, etc.), while the other involves best case behavior in which the expression size is minimized for the outcome of each mathematical operation. In general, determining worst or best case behavior for expression swell of a general operation on general operands is extremely difficult. Even just the notion of expression size is not clearly defined. How is the size of an expression to be judged? Does it mean the number of terms in the expression, the number of operators and atomic operands, the number of characters in some representation of the expression or some other measure? Or perhaps, different measures are appropriate at different times.

Matters are considerably simplified if only the expression swell analysis of algorithms involving infinite precision integer and rational number expressions is considered. In these cases, expression size can be well-defined. For an integer, the number of digits in its decimal representation is a good indicator of its size. (One could also take the absolute value of the number itself to represent its size but this definition does not provide enough generality to be useful—it is no easier nor much more informative to do an analysis using this definition than it is to just do the calculation with the original integers.) This definition of the size of an integer is essentially a logarithmic measure. Indeed, the number of digits comprising an integer n, $\mathcal{N}(n)$, can alternatively be defined by

$$\mathcal{N}(n) \equiv \begin{cases} \lfloor \log_{10} |n| \rfloor + 1 & , n \neq 0 \\ 0 & , n = 0 \end{cases}$$

where $\lfloor \ \rfloor$ denotes the floor function. (In practice, $\mathcal{N}(n)$ will be computed directly by actually counting the number of digits in n.) Similarly, the size of a rational number (which

need not be in lowest terms) can be defined as the sum of the sizes of its numerator and denominator (or possibly as the maximum of these two values). For example, the size of $-\frac{327}{1955}$ would be $\mathcal{N}(-327) + \mathcal{N}(1955) = 3 + 4 = 7$.

In order to deal only with integer and rational number expressions in an algorithm, the operations permitted on these quantities must be restricted as well. In particular, the expression swell analysis of an algorithm with solely integer inputs can proceed in a simple manner only if all the operations in the algorithm are integer or at least rational number preserving. Therefore, confining the scope of algorithms studied to only those that just use rational arithmetic operations (except perhaps at the last stage, such as when computing a polynomial from its coefficients) will make expression swell analysis manageable for those problems with such algorithms. Most of the calculations from linear algebra fall into this category.

In the following subsections, an expression swell arithmetic is developed. This arithmetic will operate on numbers that represent the number of digits in classes of integers and rational numbers. Integer preserving operations will be treated separately from rational number preserving ones. This division is made since exact rational number arithmetic exhibits a more complex behavior than exact integer arithmetic. After developing this expression swell arithmetic and discussing an implementation of it in MACSYMA, examples of its use and comparisons of its predictions with the results of actual calculations will be presented in later sections.

2.1 Integer Calculations

Here, a worst case expression swell arithmetic will be developed for certain integer preserving operations. These include the rational arithmetic operations of addition, negation, multiplication, exact division and exponentiation to a nonnegative integer power. Also included are absolute values and greatest common divisors (GCDs). Precisely, worst case behavior under an integer preserving operation means that the result will contain the greatest number of digits that are possible (GCDs are a special case and will be discussed below).

To begin this development, note that $\mathcal{N}$ partitions the integers into an infinite set of equivalence classes. Each equivalence class can be designated by a nonnegative integer: $\bar{0}, \bar{1}, \bar{2}, \ldots$. Thus, $\overline{n}$ will represent the set of integers that have exactly n digits. For example, $\bar{1} = \{-9, \ldots, -1, 1, \ldots, 9\}$ (as a special case, $\bar{0} = \{0\}$). The intent behind classifying the integers in this way is to permit the analysis of algorithms where the inputs have specified numbers of digits (are members of specified equivalence classes). The analyses will produce upper bounds on the number of digits in the results for any representatives chosen from the respective equivalence classes that are used as inputs to the algorithms.

Operations on the equivalence classes defined by $\mathcal{N}$ can yield one of two types of results. An operation can produce an integer value. An obvious example is $\mathcal{N}(\overline{n}) \equiv n$. That is, the

number of digits in any representative of $\overline{n}$ is no greater than n (actually, is exactly n in this simple usage). This defines the operation of $\mathcal{N}$ on an equivalence class. The other possibility is that an operation on one or more equivalence classes will itself yield an equivalence class. For example, suppose

$$f(\overline{n}_1, \overline{n}_2, \ldots) \equiv \overline{g(n_1, n_2, \ldots)} \, .$$

Then, for worst case expression swell behavior, this will mean that $g(n_1, n_2, \ldots)$ will bound the number of digits in $f(n_1', n_2', \ldots)$ over all possible choices of $n_1', n_2', \ldots$ such that $\mathcal{N}(n_1') = n_1, \mathcal{N}(n_2') = n_2, \ldots$. Another way of writing this is

$$\mathcal{N}(f(\overline{n}_1, \overline{n}_2, \ldots)) \leq \mathcal{N}(\overline{g(n_1, n_2, \ldots)}) \, .$$

For example, multiplying the equivalence class $\bar{2}$ with itself (i.e. multiplying together all possible pairs of 2-digit integers) will produce a set of 3 and 4-digit integers (ranging from $\pm 10 \cdot \pm 10 = \pm 100$ to $\pm 99 \cdot \pm 99 = \pm 9801$). Therefore, the product of $\bar{2}$ with itself will be defined to be $\bar{4}$ since no element of the product set will contain more than 4 digits. Note that the result could have been defined as $\bar{5}$ or $\bar{6}$ or ... but these choices would not have been as good since they do not yield as strict a bound on the maximum number of digits as does $\bar{4}$.

The best choice for g in the above formulation will be a function that actually takes on the value of $\mathcal{N}(f(\overline{n}_1, \overline{n}_2, \ldots))$ (i.e. that maximizes the number of digits in $f(n_1', n_2', \ldots)$ where $\mathcal{N}(n_i') = n_i$ for $i = 1, 2, \ldots$). For unary and binary operations, *this is not difficult. Consider* the unary operations first.

Definition 1. The unary operations of integer absolute value and negation when applied to the equivalence class $\overline{n}$ are defined by

(i) $|\overline{n}| \equiv \overline{n}$,
(ii) $\ominus \overline{n} \equiv \overline{n}$.

This says that taking the absolute value or the negation of an integer does not change the number of digits it contains. The $\ominus$ is a "maximal expression swell" operator. This notation has been introduced to emphasize the distinction between worst case expression swell arithmetic and traditional integer arithmetic.

Definition 2. The binary operations of integer addition, subtraction, multiplication and exact division when applied to the equivalence classes $\overline{n}_1$ and $\overline{n}_2$, where $n_1 > 0$ and $n_2 > 0$, are defined by

(i) $\overline{n}_1 \oplus \overline{n}_2 \equiv \overline{\max(n_1, n_2) + 1}$,
(ii) $\overline{n}_1 \ominus \overline{n}_2 \equiv \overline{\max(n_1, n_2) + 1}$,
(iii) $\overline{n}_1 \circledcirc \overline{n}_2 \equiv \overline{n_1 + n_2}$,

(iv) $\overline{n}_1 \overset{\text{exact}}{\oslash} \overline{n}_2 \equiv \overline{n_1 - n_2 + 1} \quad (n_1 \geq n_2)$.

If $n_1 = 0$ then the right-hand sides of (i) and (ii) become $\overline{n}_2$ and the right-hand sides of (iii) and (iv) become $\bar{0}$. If $n_2 = 0$ then the right-hand sides of (i)-(iv) are respectively, $\overline{n}_1, \overline{n}_1, \bar{0}$ and undefined.

Worst case expression swell addition and subtraction are equivalent since in the worst case for regular subtraction, the two operands will be of opposite signs and so it will really be an addition of two terms of the same sign, which is the worst case for regular addition. Symbolically, this can be stated

$$\overline{n}_1 \ominus \overline{n}_2 = \overline{n}_1 \oplus (\ominus \overline{n}_2) = \overline{n}_1 \oplus \overline{n}_2 \, .$$

Now, to understand the rest of Definition 2, it is best to consider some examples. For instance, the worst case of adding an n_1-digit number to an n_2-digit number occurs when all the digits in the larger number are 9's. Then adding a smaller or equal sized number of the same sign will at worst (in this particular case, will always) produce a result with one more digit than the original integer. For example, the validity of the assertion $\bar{4} \oplus \bar{3} = \bar{5}$ can be seen by looking at a worst case calculation like $9999 + 999 = 10998$. For multiplication, the worst case of expression size growth can be exhibited when both numbers consist of all 9's. In this case, the multiplication becomes $(10^{n_1} - 1)(10^{n_2} - 1) = 10^{n_1+n_2} - (10^{n_1} + 10^{n_2}) + 1$. The two middle terms will always bring the final number of digits in the result down to $n_1 + n_2$. Thus, the worst case $9999 \cdot 999 = 9989001$ establishes the validity of writing $\bar{4} \odot \bar{3} = \bar{7}$. Finally, the worst case for exact division occurs when the n_1-digit number is as large as possible and the n_2-digit number is as small as possible. For instance, to demonstrate that $\bar{4} \overset{\text{exact}}{\oslash} \bar{3} = \bar{2}$, note that the worst situations are $9900 \div 100 = 99$ and $9999 \div 101 = 99$.

Like their integer arithmetic counterparts, worst case expression swell addition and multiplication are commutative. However, the binary operation $\oplus$ at least is not associative, unlike ordinary addition. A simple example shows this. $(\bar{1} \oplus \bar{2}) \oplus \bar{3} = \bar{3} \oplus \bar{3} = \bar{4}$ while $\bar{1} \oplus (\bar{2} \oplus \bar{3}) = \bar{1} \oplus \bar{4} = \bar{5}$. Remember that the motivation behind developing an expression swell arithmetic is to establish bounds on the rate of expression growth. Therefore, if one way of ordering operations produces a tighter bound on expression swell than other methods, then this way is to be preferred. The above example suggests that the optimal arrangement for adding equivalence classes is to have them ordered in terms of increasing size (i.e. add $\overline{n}_1, \overline{n}_2, \overline{n}_3$ where $n_1 \leq n_2 \leq n_3$ by $(\overline{n}_1 \oplus \overline{n}_2) \oplus \overline{n}_3$). $\odot$ is associative as can be easily seen and so needs no further elaboration.

$\oplus$ and $\odot$ can also be thought of as n-ary operators (operators that can be applied to an indefinite number of operands). $\odot$ can be easily extended to handle an arbitrary number of factors due to its associativity when considered as a binary operator. Extending $\oplus$ is more difficult; however, some properties can be stated.

Definition 3. The n-ary operations of integer multiplication and addition when applied to the equivalence classes $\overline{n}_1, \ldots, \overline{n}_k$ and $\overline{n}$, respectively, where $n_i > 0 \; \forall \; i$ and $n > 0$, are defined by

(i) $\overline{n}_1 \circledcirc \overline{n}_2 \circledcirc \cdots \circledcirc \overline{n}_k \equiv \overline{n_1 + n_2 + \cdots + n_k} \quad (k > 0)$,
(ii) $k \circledcirc \overline{n} \equiv \underbrace{\overline{n} \oplus \overline{n} \oplus \cdots \oplus \overline{n}}_{k \text{ terms}} \equiv \overline{n + \mathcal{N}(k-1)} \quad (k > 0)$.

If any of the $n_i = 0$ then the right-hand side of (i) becomes $\bar{0}$. If $n = 0$ then the right-hand side of (ii) is $\bar{0}$.

The left-hand side of (ii) in the above definition defines a shorthand notation for the n-ary sum on the right. To see the motivation behind this definition of the n-ary sum, consider what happens in the worst case when the n-digit numbers are composed only of 9's. For instance, suppose $n = 2$ then Table 1 will show the results of actual worst case sums for various values of k as well as, in the last column, the theoretical quantities $\mathcal{N}(k \circledcirc \bar{2})$.

k	$k \cdot 99$	$\mathcal{N}(k \cdot 99)$	$\mathcal{N}(k \circledcirc \bar{2})$
1	99	2	2
2	198	3	3
10	990	3	3
11	1089	4	4
100	9900	4	4
101	9999	4	5
102	10098	5	5

Table 1. Actual versus theoretical worst case n-ary addition.

Notice that eventually $\mathcal{N}(k \circledcirc \bar{2})$ will occasionally exceed the actual worst case, but this occurs only for relatively large values of k (starting at $k = 10^n + 1$ in the general case). In a similar vein, note that the results of n-ary expression swell multiplication will also eventually exceed the actual worst cases but again, only for relatively large values of k. For example, if $n_i = 2$, $i = 1, \ldots, k$ then at least $k = 230$ factors are needed before the number of digits predicted exceeds the actual worst case number of digits (by one).

The n-ary $\oplus$ operator introduced above is distinct from the binary version, although both versions do produce the same result of $\overline{n+1}$ for the common case of $\overline{n} \oplus \overline{n}$. For example, using n-ary $\oplus$, $\overline{n} \oplus \overline{n} \oplus \overline{n} = \overline{n+1}$ while under binary $\oplus$, this becomes $(\overline{n} \oplus \overline{n}) \oplus \overline{n} = \overline{n+1} \oplus \overline{n} = \overline{n+2}$. Again, the tighter bound is to be preferred when doing expression swell arithmetic, so when adding like terms, the n-ary definition above will be used. The infix representation of n-ary $\oplus$ is somewhat misleading, hence the alternative notation of $k \circledcirc \overline{n}$ for a sum of k like terms will be adopted whenever possible. This notation has the property that $(k_1 \circledcirc \overline{n}) \oplus (k_2 \circledcirc \overline{n}) = (k_1 + k_2) \circledcirc \overline{n}$ since the parenthesized terms on the left are really

sums and for estimating bounds, it is best to make the whole expression into a single n-ary sum.

Two more worst case expression swell operations need to be defined to complete the theoretical framework which will allow algorithms involving rational arithmetic to be analyzed.

Definition 4. The binary operation of integer exponentiation to a nonnegative integer power and the n-ary operation of integer greatest common divisor when applied to the equivalence classes $\overline{n}$ and $\overline{n}_1, \ldots, \overline{n}_k$, respectively, are defined by

(i) $\overline{n} \textcircled{\uparrow} 0 \equiv \overline{1} \quad (n > 0)$,
(ii) $\overline{n} \textcircled{\uparrow} k \equiv \underbrace{\overline{n} \circledcirc \overline{n} \circledcirc \cdots \circledcirc \overline{n}}_{k \text{ factors}} = \overline{k^n} \quad (k > 0)$,
(iii) $\gcd(\overline{n}_1, \overline{n}_2, \ldots, \overline{n}_k) \equiv \overline{1} \quad (k > 1)$.

If $n = 0$ then the right-hand side of (i) becomes undefined.

The definition for exponentiation follows directly from the definition for n-ary multiplication and is simply a special case of that operation. The definition of the GCD seems at first inconsistent with the other definitions, but it is really quite appropriate for worst case expression swell behavior. Typically in symbolic mathematical calculations, GCDs are used to find the common factors of sets of expressions in order to simplify subsequent computations in some way (e.g. removing the common factors from the numerator and denominator of a rational number, reducing it to lowest terms). Therefore, expression swell will be maximized if the quantities involved in a calculation are all relatively prime with respect to one another. This implies that the GCD is always one. As an example, consider the least common multiple (LCM) of an n_1-digit integer and an n_2-digit integer. In the worst case, the LCM will become

$$\operatorname{lcm}(\overline{n}_1, \overline{n}_2) \equiv (\overline{n}_1 \circledcirc \overline{n}_2) \overset{\text{exact}}{\oslash} \gcd(\overline{n}_1, \overline{n}_2) = \overline{n_1 + n_2} \overset{\text{exact}}{\oslash} \overline{1} = \overline{n_1 + n_2}\,.$$

In a manner similar to the above, a best case expression swell arithmetic for integer preserving operations can be developed. Best case behavior under an integer preserving operation means that the result will contain the least number of digits that are possible. Hence, this arithmetic will provide a lower bound on expression growth during a computation. However, the best case for addition and subtraction of integers with the same number of digits is zero (catastrophic cancellation) so the results of an analysis may be quite a bit less interesting than for worst case behavior. In this paper, no analysis of best case expression swell behavior will be attempted.

2.2 Rational Number Calculations

Worst case expression swell arithmetic can also be developed for rational number operations. Essentially, rational expression swell arithmetic can be considered as an extension of

integer expression swell arithmetic to ordered pairs of integer equivalence classes. These ordered pairs will be denoted like $\frac{\overline{m}}{\overline{n}}$, which represents the set of rational numbers with m-digit numerators and n-digit denominators. Since this is worst case arithmetic, the components of the ordered pairs will be assumed to be relatively prime, implying that the corresponding set of rational numbers are in lowest terms and cannot be reduced in size. This is consistent with defining integer GCDs to be one as was done in the previous subsection. Note that a given rational number (e.g. $\frac{2}{4} \in \frac{\overline{1}}{\overline{1}}$) need not actually be in lowest terms but for worst case arithmetic operations, this assumption will always be made nevertheless. With these definitions, as with the integers, $\mathcal{N}$ can be seen to partition the rational numbers into a countably infinite set of equivalence classes. $\frac{\overline{m}}{\overline{n}}$ will then be the general notation for one of these equivalence classes where m and n can take on any integer value satisfying $m \geq 0$ and $n > 0$ (if $m = 0$ then n is only allowed to be 1).

Now, worst case rational expression swell arithmetic can be defined by analogy with ordinary rational arithmetic in terms of the operations of the previously defined worst case integer expression swell arithmetic. This is done as follows:

Definition 5. The unary operations of rational absolute value and negation when applied to the equivalence class $\frac{\overline{m}}{\overline{n}}$ are defined by

(i) $\left|\frac{\overline{m}}{\overline{n}}\right| \equiv \frac{|\overline{m}|}{|\overline{n}|} = \frac{\overline{m}}{\overline{n}}$,
(ii) $\ominus \frac{\overline{m}}{\overline{n}} \equiv \frac{\ominus \overline{m}}{\overline{n}} = \frac{\overline{m}}{\overline{n}}$.

Definition 6. The binary operations of rational addition, subtraction, multiplication, division, exact division and exponentiation to an integer power when applied to the equivalence classes $\frac{\overline{m_1}}{\overline{n_1}}$, $\frac{\overline{m_2}}{\overline{n_2}}$ and $\frac{\overline{m}}{\overline{n}}$ where $m_1 > 0$ and $m_2 > 0$, are defined by

(i) $\frac{\overline{m_1}}{\overline{n_1}} \oplus \frac{\overline{m_2}}{\overline{n_2}} \equiv \frac{(\overline{m_1} \circledcirc \overline{n_2}) \oplus (\overline{n_1} \circledcirc \overline{m_2})}{\overline{n_1} \circledcirc \overline{n_2}} = \frac{\overline{m_1 + n_2} \oplus \overline{n_1 + m_2}}{\overline{n_1 + n_2}}$,
(ii) $\frac{\overline{m_1}}{\overline{n_1}} \ominus \frac{\overline{m_2}}{\overline{n_2}} \equiv \frac{(\overline{m_1} \circledcirc \overline{n_2}) \ominus (\overline{n_1} \circledcirc \overline{m_2})}{\overline{n_1} \circledcirc \overline{n_2}} = \frac{\overline{m_1 + n_2} \ominus \overline{n_1 + m_2}}{\overline{n_1 + n_2}}$,
(iii) $\frac{\overline{m_1}}{\overline{n_1}} \circledcirc \frac{\overline{m_2}}{\overline{n_2}} \equiv \frac{\overline{m_1} \circledcirc \overline{m_2}}{\overline{n_1} \circledcirc \overline{n_2}} = \frac{\overline{m_1 + m_2}}{\overline{n_1 + n_2}}$,
(iv) $\frac{\overline{m_1}}{\overline{n_1}} \oslash \frac{\overline{m_2}}{\overline{n_2}} \equiv \frac{\overline{m_1}}{\overline{n_1}} \circledcirc \frac{\overline{n_2}}{\overline{m_2}} = \frac{\overline{m_1 + n_2}}{\overline{n_1 + m_2}}$,
(v) $\frac{\overline{m_1}}{\overline{n_1}} \overset{\text{exact}}{\oslash} \frac{\overline{m_2}}{\overline{n_2}} \equiv \frac{\overline{m_1} \overset{\text{exact}}{\ominus} \overline{m_2}}{\overline{n_1} \overset{\text{exact}}{\ominus} \overline{n_2}} = \frac{\overline{m_1 - m_2 + 1}}{\overline{n_1 - n_2 + 1}}$,
(vi) $\left(\frac{\overline{m}}{\overline{n}}\right) \uparrow 0 \equiv \frac{\overline{m} \bigcirc 0}{\overline{n} \bigcirc 0} = \frac{\overline{1}}{\overline{1}} \quad (m > 0)$,
(vii) $\left(\frac{\overline{m}}{\overline{n}}\right) \uparrow k \equiv \frac{\overline{m} \bigcirc k}{\overline{n} \bigcirc k} = \frac{\overline{km}}{\overline{kn}} \quad (k > 0)$,
(viii) $\left(\frac{\overline{m}}{\overline{n}}\right) \uparrow (-k) \equiv \left(\frac{\overline{n}}{\overline{m}}\right) \uparrow k = \frac{\overline{kn}}{\overline{km}} \quad (k > 0, m > 0)$.

If $m_1 = 0$ then the right-hand sides of (i) and (ii) become $\frac{\overline{m_2}}{\overline{n_2}}$ and the right-hand sides of (iii)-(v) become $\frac{\overline{0}}{\overline{1}}$. If $m_2 = 0$ then the right-hand sides of (i)-(v) are respectively, $\frac{\overline{m_1}}{\overline{n_1}}$, $\frac{\overline{m_1}}{\overline{n_1}}$, $\frac{\overline{0}}{\overline{1}}$, undefined and undefined.

A new operation, non-exact division, has been introduced but like the rest of the rational operations, it is adapted directly from ordinary rational arithmetic and so should come as no surprise. Finally, n-ary addition and multiplication for rational expression swell arithmetic are simple extensions of the binary versions.

Definition 7. The n-ary operations of rational multiplication and addition when applied to the equivalence classes $\frac{\overline{m}_1}{\overline{n}_1}, \ldots, \frac{\overline{m}_k}{\overline{n}_k}$ $(k > 0)$ where $m_i > 0 \; \forall \, i$ are defined by

(i) $\frac{\overline{m}_1}{\overline{n}_1} \circledcirc \frac{\overline{m}_2}{\overline{n}_2} \circledcirc \cdots \circledcirc \frac{\overline{m}_k}{\overline{n}_k} \equiv \frac{\overline{m}_1 \circledcirc \overline{m}_2 \circledcirc \cdots \circledcirc \overline{m}_k}{\overline{n}_1 \circledcirc \overline{n}_2 \circledcirc \cdots \circledcirc \overline{n}_k} = \frac{\overline{m_1 + m_2 + \cdots + m_k}}{n_1 + n_2 + \cdots + n_k}$,

(ii) $\frac{\overline{m}_1}{\overline{n}_1} \oplus \frac{\overline{m}_2}{\overline{n}_2} \oplus \cdots \oplus \frac{\overline{m}_k}{\overline{n}_k} \equiv \frac{(\overline{m}_1 \circledcirc \overline{n}_2 \circledcirc \cdots \circledcirc \overline{n}_k) \oplus (\overline{n}_1 \circledcirc \overline{m}_2 \circledcirc \cdots \circledcirc \overline{n}_k) \oplus \cdots \oplus (\overline{n}_1 \circledcirc \overline{n}_2 \circledcirc \cdots \circledcirc \overline{m}_k)}{\overline{n}_1 \circledcirc \overline{n}_2 \circledcirc \cdots \circledcirc \overline{n}_k}$

$= \frac{\overline{n - n_1 + m_1} \oplus \overline{n - n_2 + m_2} \oplus \cdots \oplus \overline{n - n_k + m_k}}{\overline{n}}$ where $n \equiv \sum_{i=1}^{k} n_i$.

If any of the $m_i = 0$ then the right-hand side of (i) becomes $\frac{\overline{0}}{\overline{1}}$ and those terms with $m_i = 0$ are excluded from the sum in (ii).

2.3 MACSYMA Implementation

To obtain some practical experience with the above concepts, both integer and rational number worst case expression swell arithmetic have been implemented in MACSYMA. The top-level interface to expression swell arithmetic operations (there also exists a LISP-level interface which is accessed slightly differently) is provided by the functions **abs_**(x), **neg_**(x), **add_**$(term_1, term_2, \ldots)$, **sub_**(x,y), **mul_**$(factor_1, factor_2, \ldots)$, **div_**$(x,y)$, **ediv_**$(x,y)$ [Exact DIVision], **power_**(x,y) and **gcd_**(x,y) ($\mathcal{N}$ is performed by **ndigits**(e)). If the global variable **exprswell** is **false** (its default value) then these functions will perform ordinary arithmetic. However, if **exprswell** is set to **true** then these functions will treat their arguments as equivalence classes of $\mathcal{N}$ (when appropriate) and perform worst case expression swell arithmetic on them (no mixing of integer and rational number equivalence classes will be permitted). Thus, an algorithm can be implemented using this set of functions in place of the normal MACSYMA arithmetic operators except where expression swell operations are never appropriate, such as index calculations or incrementing loop indices. Then, with **exprswell** set to its default value, the algorithm can be run in a normal manner. However, by simply changing the value of **exprswell**, an expression swell analysis of the algorithm for inputs from a given set of equivalence classes can be performed.

The MACSYMA implementation of general integer worst case expression swell n-ary addition is an extension of the n-ary addition of like terms defined previously. The terms are first sorted into ascending order, then after eliminating any zeroes, the leading set of like terms (which may be a single term) is combined using the n-ary addition of Definition 3. The result is tacked onto the beginning of the list containing the remainder of the terms, which is then resorted, if necessary, to maintain the terms in ascending order. Next, the leading set

of like terms are combined with perhaps an initial non-like term, once again using the n-ary addition of Definition 3 (the non-like term, if there is one, will be treated as just another like term in the n-ary addition). These last two steps will then repeat until the list contains a single term. For example, $\bar{1} \oplus \bar{1} \oplus \bar{2} \oplus \bar{2} \oplus \bar{4} \oplus \bar{4} = \bar{2} \oplus \bar{2} \oplus \bar{2} \oplus \bar{4} \oplus \bar{4} = \bar{3} \oplus \bar{4} \oplus \bar{4} = \bar{5}$. This algorithm was chosen so as to try to minimize the results of general n-ary addition and thus provide as tight a bound as possible.

3 Matrix Determinant Computation

There are two major algorithms for computing the determinant of a square matrix that have been generally implemented in symbolic math systems. The first method is expansion by cofactors (or expansion by minors). For example, if

$$A = \begin{pmatrix} 4 & 3 & 2 \\ 7 & 5 & 1 \\ 6 & 2 & 9 \end{pmatrix}$$

then expanding the cofactors along the third column will yield

$$\det A = 2(7 \cdot 2 - 5 \cdot 6) - 1(4 \cdot 2 - 3 \cdot 6) + 9(4 \cdot 5 - 3 \cdot 7) = -31 \, .$$

Worst case expression swell analysis can be performed on the above method of computing the determinant. To simplify this analysis, it is helpful to expand out all the products in the cofactor expansion. For an $n \times n$ matrix (B), this will produce a sum of $n!$ products of n factors each, half of which are added and the rest subtracted. Therefore, if the entries of the matrix are all m-digit integers, then

$$\begin{aligned} \mathcal{N}(\det B) &\leq \mathcal{N}\left(\left[\frac{n!}{2} \odot (\overline{m} \uparrow n)\right] \ominus \left[\frac{n!}{2} \odot (\overline{m} \uparrow n)\right]\right) \\ &\leq \mathcal{N}\left(\left[\frac{n!}{2} \odot (\overline{m} \uparrow n)\right] \oplus \left[\frac{n!}{2} \odot (\overline{m} \uparrow n)\right]\right) \\ &\leq \mathcal{N}(n! \odot (\overline{m} \uparrow n)) \leq \mathcal{N}(n! \odot \overline{nm}) \leq nm + \mathcal{N}(n! - 1) \, . \end{aligned} \tag{1}$$

The other major algorithm generally implemented in symbolic math systems for computing determinants begins with some variant of a fraction free reduction of a matrix to triangular form (for example, see [Bar66]). This Gaussian elimination algorithm may be either one-step or multi-step where the number of steps indicates the number of iterations performed in a single pass through the matrix. The determinant will then be proportional to the bottom rightmost element of the reduced matrix. For example, a one-step fraction

free reduction to triangular form of [adapted from Fox65]

$$C = \begin{pmatrix} 7 & 9 & -1 & 2 \\ 4 & -5 & 2 & -7 \\ 1 & 6 & -3 & -4 \\ 3 & -2 & -1 & -5 \end{pmatrix}$$

proceeds as

$$\rightarrow \begin{pmatrix} 7 & 9 & -1 & 2 \\ 0 & -71 & 18 & -57 \\ 0 & 33 & -20 & -30 \\ 0 & -41 & -4 & -41 \end{pmatrix} \rightarrow \begin{pmatrix} 7 & 9 & -1 & 2 \\ 0 & -71 & 18 & -57 \\ 0 & 0 & 118 & 573 \\ 0 & 0 & 146 & 82 \end{pmatrix} \rightarrow \begin{pmatrix} 7 & 9 & -1 & 2 \\ 0 & -71 & 18 & -57 \\ 0 & 0 & 118 & 573 \\ 0 & 0 & 0 & 1042 \end{pmatrix}$$

where, during the k^{th} stage, the elements $c_{ij}^{(k)}$ in the $(n-k) \times (n-k+1)$ lower right block are computed by

$$c_{ij}^{(k)} = \frac{c_{kk}^{(k-1)} c_{ij}^{(k-1)} - c_{ik}^{(k-1)} c_{kj}^{(k-1)}}{c_{k-1,k-1}^{(k-1)}}$$

(with $c_{00}^{(0)}$ defined to be one). Thus, $\det C$ is 1042. In general, some kind of strategy will be used to choose a "good" pivot at each stage and also to take care of the case of a forced zero pivot which indicates a singular matrix.

For an $n \times n$ matrix (call it D) consisting of m-digit integer elements, worst case expression swell analysis of the above algorithm reveals that

$$\mathcal{N}(\det D) \leq nm + (n-1)^2 \,. \tag{2}$$

This can be seen by performing the elimination on a representative matrix using expression swell arithmetic. In particular, for a 4×4 matrix of m-digit integers, the reduction proceeds as follows:

$$\begin{pmatrix} m & m & m & m \\ m & m & m & m \\ m & m & m & m \\ m & m & m & m \end{pmatrix} \rightarrow \begin{pmatrix} m & m & m & m \\ 0 & 2m+1 & 2m+1 & 2m+1 \\ 0 & 2m+1 & 2m+1 & 2m+1 \\ 0 & 2m+1 & 2m+1 & 2m+1 \end{pmatrix}$$

$$\rightarrow \begin{pmatrix} m & m & m & m \\ 0 & 2m+1 & 2m+1 & 2m+1 \\ 0 & 0 & 3m+4 & 3m+4 \\ 0 & 0 & 3m+4 & 3m+4 \end{pmatrix} \rightarrow \begin{pmatrix} m & m & m & m \\ 0 & 2m+1 & 2m+1 & 2m+1 \\ 0 & 0 & 3m+4 & 3m+4 \\ 0 & 0 & 0 & 4m+9 \end{pmatrix}.$$

For more detail, consider the second stage calculation of the (3,4) element:

$$\overline{d_{34}^{(2)}} \;=\; [(\overline{d_{22}^{(1)}} \circledcirc \overline{d_{34}^{(1)}}) \ominus (\overline{d_{32}^{(1)}} \circledcirc \overline{d_{24}^{(1)}})] \overset{\text{exact}}{\oslash} \overline{d_{11}^{(1)}}$$

$$= [(\overline{2m+1} \circledcirc \overline{2m+1}) \ominus (\overline{2m+1} \circledcirc \overline{2m+1})] \overset{\text{exact}}{\oslash} \overline{m}$$
$$= \overline{[4m+2 \ominus 4m+2]} \overset{\text{exact}}{\oslash} \overline{m} = \overline{4m+3} \overset{\text{exact}}{\oslash} \overline{m} = \overline{3m+4} .$$

A comparison of the above two theoretical worst case determinations of the size of the determinant of an $n \times n$ matrix with m-digit integer entries shows that the former yields a tighter and thus a better bound. This can be seen by examining the difference

$$\begin{aligned}
\mathcal{N}(\det D) - \mathcal{N}(\det B) &= (n-1)^2 - \mathcal{N}(n! - 1) \\
&= (n-1)^2 - \lfloor \log_{10}(n! - 1) \rfloor - 1 \qquad (n > 1) \\
&\approx n^2 - 2n - \left\lfloor \log_{10} \sqrt{2\pi n} \left(\frac{n}{e}\right)^n \right\rfloor \qquad (n \to \infty) \\
&\approx n^2 - \left(n + \frac{1}{2}\right) \log_{10} n \qquad (n \to \infty) .
\end{aligned}$$

The second equality comes from the definition of $\mathcal{N}$ and the third line is a result of approximating $n!$ (and hence $n! - 1$) for n large by Stirling's formula. Indeed, $\mathcal{N}(\det D)$ will strictly dominate $\mathcal{N}(\det B)$ for all $n > 2$. The primary reason for this behavior is that worst case expression swell analysis is performed one stage at a time while following the Gaussian elimination algorithm but all at once for the expansion by cofactors. The latter analysis takes full advantage of the stricter bounds yielded by combining as many n-ary arithmetical operations as possible at one time and so produces the better estimate. The former analysis does produce bounds on the size of the matrix elements at each stage of the elimination but any overestimates from previous stages tend to accumulate.

The determinant of a matrix can also be bounded using Hadamard's inequality. This relationship states that given an $n \times n$ matrix E,

$$|\det E|^2 \leq \prod_{i=1}^{n} \left(\sum_{j=1}^{n} e_{ij}^2 \right) .$$

If all the elements of E are m-digit integers, then worst case expression swell analysis gives

$$\begin{aligned}
\mathcal{N}(|\det E|^2) &\leq \mathcal{N}([n \circledcirc (\overline{m} \uparrow 2)] \uparrow n) \leq \mathcal{N}([n \circledcirc \overline{2m}] \uparrow n) \\
&\leq \mathcal{N}(\overline{2m + \mathcal{N}(n-1)} \uparrow n) \leq 2nm + n\mathcal{N}(n-1) .
\end{aligned}$$

Hence,

$$\mathcal{N}(\det E) \leq \left\lceil nm + \frac{1}{2} n \mathcal{N}(n-1) \right\rceil \tag{3}$$

carefully applying Definition 4(ii) in reverse. This estimate is of about the same asymptotic order as the one for the cofactor expansion.

Rational expression swell arithmetic can also be applied fruitfully to Hadamard's inequality. Suppose F is an $n \times n$ matrix with rational number entries consisting of m-digit numerators and denominators, then

$$\mathcal{N}(|\det F|^2) \leq \mathcal{N}\left(\left[n \circledcirc \left(\frac{\overline{m}}{\overline{m}} \textcircled{\uparrow} 2\right)\right] \textcircled{\uparrow} n\right) \leq \mathcal{N}\left(\left[n \circledcirc \frac{\overline{2m}}{\overline{2m}}\right] \textcircled{\uparrow} n\right) .$$

Now,

$$k \circledcirc \frac{\overline{m}}{\overline{m}} \equiv \underbrace{\frac{\overline{m}}{\overline{m}} \oplus \cdots \oplus \frac{\overline{m}}{\overline{m}}}_{k \text{ terms}} = \frac{\overbrace{\overbrace{(\overline{m} \circledcirc \cdots \circledcirc \overline{m})}^{k \text{ factors}} \oplus \cdots \oplus \overbrace{(\overline{m} \circledcirc \cdots \circledcirc \overline{m})}^{k \text{ factors}}}^{k \text{ terms}}}{\underbrace{\overline{m} \circledcirc \cdots \circledcirc \overline{m}}_{k \text{ factors}}}$$

$$= \frac{k \circledcirc \overline{km}}{\overline{km}} = \frac{\overline{km + \mathcal{N}(k-1)}}{\overline{km}} ,$$

therefore,

$$\mathcal{N}(|\det F|^2) \leq \mathcal{N}\left(\frac{\overline{2nm + \mathcal{N}(n-1)}}{\overline{2nm}} \textcircled{\uparrow} n\right) \leq \mathcal{N}\left(\frac{\overline{2n^2m + n\mathcal{N}(n-1)}}{\overline{2n^2m}}\right) .$$

Carefully taking an exact square root once again,

$$\mathcal{N}(\det F) \leq \mathcal{N}\left(\frac{\overline{\lceil n^2m + n\mathcal{N}(n-1)/2 \rceil}}{\overline{n^2m}}\right) \leq \left\lceil 2n^2m + \frac{1}{2}n\mathcal{N}(n-1) \right\rceil .$$

4 Matrix Characteristic Polynomial Computation

Generally in symbolic math systems, the characteristic polynomial of a matrix A is calculated by forming the matrix $A - \lambda I$, where λ is a scalar variable, and then computing its determinant. The determinant will always involve polynomial arithmetic, even for a purely numerical matrix, so it is difficult to perform expression swell analysis on this method. However, in the previous section, expression swell analysis was performed on the determinant calculation of certain integer matrices. Since the determinant is plus or minus the constant term of the characteristic polynomial, which for nonsingular matrices is typically the largest coefficient of the polynomial in gross size, these analyses should give a good bound on the size of the largest coefficient of the characteristic polynomial for integer matrices with entries of the same size.

A second method, which computes the coefficients of the characteristic polynomial of a purely numerical matrix using no polynomial arithmetic, involves taking the trace of powers

of the matrix. The procedure takes advantage of the fact that for an $n \times n$ matrix A with eigenvalues $\{\lambda_i\}_{i=1,\ldots,n}$,

$$\text{trace } A^k = \sum_{i=1}^{n} \lambda_i^k \ .$$

The coefficients of the characteristic polynomial are then computed from the symmetric functions of the traces of $A, A^2, \ldots, A^n$ [Sto72].

The above method requires $O(n^4)$ operations and so is not practical for large matrices. Nevertheless, it is instructive to perform worst case expression swell analysis on this algorithm. Initially, consider the first portion of the algorithm, which computes the various traces. If the entries of A are m-digit integers then the number of digits in the entries of $AA = A^2$ will be bounded by $2m + \mathcal{N}(n-1)$. For $AA^2 = A^3$, the number of digits in the entries will be bounded by $3m + 2\mathcal{N}(n-1)$ and in general, the number of digits in the entries of $AA^{k-1} = A^k$ will be bounded by $km + (k-1)\mathcal{N}(n-1)$. Hence,

$$\begin{aligned} \mathcal{N}(\text{trace } A^k) &= \mathcal{N}\left(\sum_{i=1}^{n} a_{ii}^{\langle k \rangle}\right) \\ &\leq \mathcal{N}(n \circledcirc \overline{km + (k-1)\mathcal{N}(n-1)}) \leq k[m + \mathcal{N}(n-1)] \ . \end{aligned}$$

Now, the coefficient c_k of λ^k $(0 \leq k \leq n-1)$ in the characteristic polynomial $P(\lambda)$ of A will be composed of $p(n-k)$ terms, each of which will be proportional to a product of traces such that the total sum of powers involved is $n-k$. Here, $p(n)$ is the number of partitions of the integer n (i.e. the number of ways n can be written as a sum of positive integers where order does not matter) and is computed by Hardy and Ramanujan's formula

$$p(n) \approx \frac{1}{4\sqrt{3}n} e^{\pi\sqrt{2n/3}} \ .$$

For example, the coefficient of λ for A, 4×4, is

$$c_1 = \frac{\text{trace } A^3}{3} - \frac{(\text{trace } A^2)(\text{trace } A)}{2} + \frac{(\text{trace } A)^3}{6}$$

which consists of $p(4-1) = p(3) = 3$ terms. It is easily seen that $\mathcal{N}([\text{trace } A]^k) = \mathcal{N}(\text{trace } A^k)$, therefore,

$$\begin{aligned} \mathcal{N}(c_k) &\leq \mathcal{N}(p(n-k) \circledcirc \overline{(n-k)[m + \mathcal{N}(n-1)]}) \\ &\leq (n-k)[m + \mathcal{N}(n-1)] + \mathcal{N}(p(n-k) - 1) \ . \end{aligned}$$

In particular,

$$\mathcal{N}(\det A) = \mathcal{N}(c_0) \leq nm + n\mathcal{N}(n-1) + \mathcal{N}(p(n) - 1) \qquad (4)$$

hence

$$\begin{aligned}\mathcal{N}(\det A) &\approx nm + n(\lfloor \log_{10}(n-1) \rfloor + 1) + \left\lfloor \log_{10}\left(\frac{1}{4\sqrt{3n}} e^{\pi\sqrt{2n/3}} - 1\right)\right\rfloor + 1 \qquad (n > 1) \\ &\approx nm + n\log_{10} n \qquad (n \to \infty)\,.\end{aligned}$$

This estimate compares favorably to the one derived from the cofactor expansion of the determinant (although it will be seen in the next section that it is somewhat higher).

A third way to compute the characteristic polynomial of a matrix is to transform the matrix into upper Hessenberg form via a similarity reduction (which will preserve the characteristic polynomial) and then compute the characteristic polynomial of the transformed matrix using a computationally cheap algorithm. A matrix is upper Hessenberg if all the entries below the subdiagonal are zero (i.e. H is upper Hessenberg if $h_{ij} = 0$ whenever $i > j + 1$). The reduction of a general matrix into upper Hessenberg form proceeds via a series of elementary similarity transformations. For example, performing the first stage of the reduction on the matrix C of the previous section yields

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -\frac{1}{4} & 1 & 0 \\ 0 & -\frac{3}{4} & 0 & 1 \end{pmatrix} \begin{pmatrix} 7 & 9 & -1 & 2 \\ 4 & -5 & 2 & -7 \\ 1 & 6 & -3 & -4 \\ 3 & -2 & -1 & -5 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & \frac{1}{4} & 1 & 0 \\ 0 & \frac{3}{4} & 0 & 1 \end{pmatrix} = \begin{pmatrix} 7 & \frac{41}{4} & -1 & 2 \\ 4 & -\frac{39}{4} & 2 & -7 \\ 0 & \frac{75}{16} & -\frac{7}{2} & -\frac{9}{4} \\ 0 & \frac{21}{16} & -\frac{5}{2} & \frac{1}{4} \end{pmatrix}.$$

Once an upper Hessenberg matrix, H, similar to the original matrix, has been constructed, the characteristic polynomial of the upper Hessenberg matrix (and thus of the original matrix) is calculated next. The basic algorithm is to create a triangular system of equations using H, which can then be easily solved for the coefficients of the characteristic polynomial. In particular, let H be an $n \times n$ standard upper Hessenberg matrix (one with no zero subdiagonal elements). Now, define the vectors $\mathbf{w}_1, \ldots, \mathbf{w}_n$ by $\mathbf{w}_{i+1} = H\mathbf{w}_i$ for $i = 0, \ldots, n-1$ where $\mathbf{w}_0 = (1, 0, \ldots, 0)^T$ is the $n \times 1$ unit vector $\mathbf{e}_1$. Then the upper triangular system

$$(\mathbf{w}_0 | \cdots | \mathbf{w}_{n-1})\mathbf{a} = -\mathbf{w}_n$$

will have a unique solution for $\mathbf{a} \equiv (a_0, \ldots, a_{n-1})^T$ and the characteristic polynomial of H will be given by $\lambda^n + a_{n-1}\lambda^{n-1} + \cdots + a_1\lambda + a_0$. If H has subdiagonal elements that are zero then its characteristic polynomial will simply be the product of the characteristic polynomials of each standard upper Hessenberg block found along H's diagonal.

Worst case expression swell analysis of the above algorithm is difficult to generalize and so will not be attempted. In the following section, results from the analyses of specific examples will be presented. These analyses were performed using the MACSYMA implementation of worst case expression swell arithmetic.

5 Results of Case Studies

It is all very well to derive theoretical bounds on the possible expression swell in a calculation, but when compared to the results of actual computations, how good of an estimate of real behavior do these bounds really provide? In order to answer this question, a number of case studies were performed involving matrices of various sizes with initial integer or rational number entries. Some of these results provide quite striking examples of intermediate expression swell in action.

To begin with, Table 2 presents the four worst case bounds (relations (3), (1), (4) and (2), respectively, of Sections 3 and 4), derived for the determinant of an $n \times n$ matrix containing m-digit integer entries, for a variety of values of n with $m = 4$. Hadamard's inequality yields the lowest upper estimate on the number of digits in the determinant, although the bounds derived from the algorithms involving expansion by cofactors and taking sums of powers of traces fall within the same approximate asymptotic order. As noted before, worst case bounds derived from the fraction free Gaussian elimination algorithm grow much faster than for the other methods.

Table 3 presents the results of actual calculations performed on $n \times n$ matrices for values of n ranging from 3 through 10, where the initial matrix elements were 4-digit integers. The table is divided into three sections. The first section (a) exhibits the greatest number of digits encountered in the elements of the upper Hessenberg matrix produced by a similarity transformation of the original matrix using a division free version of the algorithm presented previously. The column labeled "exprswell" shows the results produced by setting the global variable **exprswell** to **true** in MACSYMA. The next column displays the outcome of using an initial matrix composed of the n^2 largest 4-digit primes. The last three columns in this table present the results of a statistical sampling in which matrices consisting initially of random 4-digit integers were used. The first pair of numbers is the mean and standard deviation obtained from a series of calculations whose number is given by N_{samples}.[1] The last column gives the minimum and maximum values for the largest number of digits attained.

Section (b) of Table 3 shows the number of digits contained in the largest coefficient of the characteristic polynomial that was computed directly from the corresponding upper Hessenberg matrix of the previous section. Due to the division free nature of the similarity reduction to upper Hessenberg form that was used, a straightforward computation of the characteristic polynomial will produce, in general, a non-monic result. However, the leading coefficient of this polynomial will exactly divide all the other coefficients, producing a monic polynomial whose constant term will be plus or minus the determinant of the original matrix.

[1] In some cases, when a sequence of calculations was performed as here, later calculations were done fewer times due to time constraints and/or random problems occurring when running large MACSYMA jobs continuously for days at a time (such as occasional memory corruption resulting from the growth of a MACSYMA job).

Essentially, this particular procedure for computing the monic characteristic polynomial of an integer matrix reserves all divisions until the final step of the calculation. The maximum number of digits found in a coefficient (nearly always the constant term) of the final, simplified characteristic polynomial is presented in the final section (c) of Table 3.

The trend displayed in Table 3 is quite typical of calculations involving intermediate expression swell. Even though the final numbers have relatively few digits, intermediate computations in this particular algorithm are already creating integers whose size is approaching 5 digits when n is just 10. It is interesting to note that the matrices with prime entries pretty much provide the smallest actual results and this seems to be true as well for the other calculations surveyed here that have started with integer matrices.

The theoretical limits on the size of the determinant, tabulated in Table 2, bound the results in Table 3(c) nicely. The "exprswell" calculations tend to greatly overestimate the maximum number of digits actually produced in the first two phases of the computation except for the lowest values of n. However, after the final exact divisions, these bounds drop down to much more reasonable estimates. One has to be careful, though, in interpreting the results of the exprswell computations for this last calculation. The coefficients of the non-monic characteristic polynomials in Table 3(b) were determined by a series of calculations that maximized the number of digits at each step. In order to truely maximize the number of digits in the coefficients of the final monic polynomials, though, the leading coefficients of the non-monic polynomials (which act as divisors) should really have been minimized in the course of their calculation (using a best case expression swell arithmetic). This action, however, could have impacted the results of the worst case expression swell arithmetic in earlier phases of the calculations. Thus, the occurrence of exact divisions in an algorithm such as this one can lead to theoretical uncertainties in verifying whether the analysis does indeed produce a bounding calculation, although in practice, no exceptions have been found so far.

Table 4 presents the maximum number of digits found in coefficients of the characteristic polynomials of upper Hessenberg matrices whose initial nonzero entries were 4-digit integers. The exprswell calculations and the results from the statistical survey for $n = 3$ through 10 are remarkably similar to those exhibited in Table 3(c). These similarities appear to imply that the results of computing characteristic polynomials of integer Hessenberg matrices will give a very good indication of the trends to be expected when computing monic characteristic polynomials of integer general matrices. This is useful since there are no theoretical uncertainties here as there were above because this algorithm involves no divisions. Also, the computation of the characteristic polynomial of an integer Hessenberg matrix is reasonably

fast as well as conservative of memory, so it was possible to perform calculations up through $n = 100$.[2]

The trends of Table 3(c) become more pronounced for $n > 10$ in Table 4. The worst case maximums derived from Hadamard's inequality continue to provide good bounds on the actual results, although they become less good for increasing n. For $n > 10$, the exprswell bounds, which are growing arithmetically, begin to leave the real recorded maximums, which are growing slightly slower than linearly, further and further behind. Generalizing these data tendencies (as well as those found for $m = 5$ and 6, the data for which are not presented here), a good empirical bound on the maximum number of digits in the determinant of an $n \times n$ integer matrix (A) which has entries consisting of no greater than m digits appears to be given, simply by

$$\mathcal{N}(\det A) \leq nm \ .$$

Table 5 shows what happens when the Hessenberg matrix is initially filled with 4-digit rational numbers (the numerators and denominators are both 4-digit integers) and the matrix is first "derationalized." Derationalization is the process of converting a matrix of rational numbers into a matrix of integers by multiplying the matrix by the least common multiple of the denominators of its rational number entries. This number (call it d) is an implicit divisor of the integer matrix and if the matrix is involved in subsequent calculations, d needs to be taken into account. In this particular example, the characteristic polynomial will again, in general, be non-monic but in this case, dividing the coefficients by the leading coefficient is not guaranteed to be exact (since this is really the characteristic polynomial of a rational number matrix) and so is not performed.

In Table 5, the "prime" rational numbers (rational numbers whose numerators and denominators are prime) provided the worst actual results of expression swell. The observation that "prime" rational numbers typically generate the greatest expression growth can also be noted in other calculations that start off with matrices of rational numbers, completely contrary to what was observed earlier for calculations that started with integer matrices. A nice discovery is that the exprswell calculations produced good bounds on the worst case expression swell. This may be due to a couple of considerations. The prime computations will be particularly large since the denominators of the initial rational number entries will all be big and relatively prime to each other so that derationalization will create nearly the maximum possible integer entries for a matrix of a given size undergoing this transformation. The exprswell calculations will act as if they are computing the characteristic polynomials of integer Hessenberg matrices with large effective values of m (essentially, the original m times the number of nonzero elements in the matrix). Thus, any excesses in the exprswell

[2]All the calculations in this paper were performed on Sun 3/160 workstations with 4 megabytes of memory running under a version of the UNIX operating system (Sun OS 3.4).

computations (which seem to depend only on n) will be overwhelmed (at least, at these values of n) by the large effective values of m.

In Table 6 is presented the results of another characteristic polynomial computation via an intermediate Hessenberg transformation. This time, the entries of the initial general matrices were 4-digit rational numbers and all calculations were done in rational arithmetic. Note that the numbers in this table represent the number of digits in the maximal rational number (i.e. the largest value obtained by summing the number of digits in the numerator and denominator of each rational number under consideration).

Again, the calculations that started off with "prime" rational number matrices produced the greatest growth in expression size. Also, like the computations whose results were shown in Table 2, expression swell decreased dramatically between the intermediate results found in the entries of the Hessenberg matrices and the final numbers that comprised the coefficients of the characteristic polynomials. These latter numbers, upon examination of additional data for $m = 5$ and 6 (not [illegible] n here), appear to be quite nicely bounded by $2n^2m$, which is very similar to the bound derived from Hadamard's inequality. The bounds computed from the MACSYMA implementation of rational worst case expression swell arithmetic are huge and provide virtually no useful information.

As an experiment to see what effect GCDs have on rational number computations, the above calculations on "prime" rational number matrices were repeated with the GCD function forced always to return one, thus allowing no cancellation of common factors. These results are compared with those from Table 6, in which GCDs were taken freely, in Table 7. It is quite clear from this comparison of maximal numbers of digits that GCDs, which are taken every time a new rational number is formed in MACSYMA, have important effects in minimizing expression swell which, in the samples reviewed in Table 7, is growing exponentially for the "no GCDs" cases. The reason that the results for "no GCDs" are complete only through $n = 6$ is because the MACSYMA computations for bigger cases ran out of available memory.

n	Hadamard's inequality	Cofactor expansion	Sums of powers of traces	Gaussian elimination
3	14	13	16	16
4	18	18	21	25
5	23	23	26	36
6	27	27	32	49
7	32	32	37	64
8	36	37	42	81
9	41	42	47	100
10	45	47	52	121
20	100	99	123	441
30	150	153	184	961
40	200	208	245	1681
50	250	265	306	2601
60	300	322	367	3721
70	350	381	427	5041
80	400	439	488	6561
90	450	499	548	8281
100	500	558	609	10201

Table 2. Determinant of an integer general matrix ($m = 4$).

n	exprswell	primes	random numbers	N_{samples}	min → max
3	14	13	11.42 ± 0.75	100	9 → 12
4	43	27	30.73 ± 2.10	100	24 → 35
5	130	59	74.98 ± 2.66	100	68 → 81
6	391	98	142.38 ± 4.26	100	130 → 152
7	1174	155	249.75 ± 6.75	100	233 → 262
8	3523	240	366.50 ± 9.53	100	343 → 388
9	10570	334	585.30 ± 11.79	100	556 → 612
10	31711	469	825.92 ± 14.22	13	797 → 844

Table 3(a). Division free transformation of an integer general matrix into Hessenberg form ($m = 4$).

n	exprswell	primes	random numbers	N_{samples}	min → max
3	43	31	30.09 ± 2.28	100	22 → 34
4	172	92	115.60 ± 7.90	100	90 → 128
5	648	277	364.62 ± 13.36	100	331 → 393
6	2341	561	840.63 ± 25.23	100	765 → 898
7	8209	1058	1731.99 ± 47.74	100	1616 → 1824
8	28170	1882	2911.08 ± 77.09	100	2720 → 3081
9	95110	2959	5243.04 ± 106.39	100	4977 → 5483
10	317083	4644	8243.42 ± 147.58	12	7942 → 8422

Table 3(b). Characteristic polynomial of the Hessenberg matrix.

n	exprswell	primes	random numbers	N_{samples}	min → max
3	17	7	11.40 ± 0.84	10	10 → 12
4	25	7	15.20 ± 0.63	10	14 → 16
5	34	8	18.70 ± 0.67	10	18 → 20
6	44	8	22.70 ± 0.95	10	21 → 24
7	55	11	26.50 ± 0.97	10	24 → 27
8	67	11	30.60 ± 0.52	10	30 → 31
9	80	13	34.20 ± 0.63	10	33 → 35
10	94	14	37.90 ± 0.57	10	37 → 39

Table 3(c). Above divided by its leading coefficient.

n	exprswell	primes	random numbers	N_{samples}	min → max
3	17	9	11.46 ± 0.54	100	10 → 12
4	25	14	15.27 ± 0.60	100	13 → 16
5	34	15	19.13 ± 0.68	100	17 → 20
6	44	21	23.00 ± 0.70	100	21 → 24
7	55	23	26.51 ± 0.75	100	24 → 28
8	67	29	30.35 ± 0.73	100	28 → 32
9	80	33	34.12 ± 0.76	100	33 → 36
10	94	38	37.91 ± 0.79	100	36 → 39
20	289	—	75.40 ± 0.89	5	74 → 76
30	584	—	113.40 ± 1.14	5	112 → 115
40	979	—	150.60 ± 1.34	5	150 → 153
50	1474	—	187.60 ± 1.14	5	186 → 189
60	2069	—	227.60 ± 1.52	5	226 → 230
70	2764	—	264.40 ± 1.14	5	263 → 266
80	3559	—	301.80 ± 1.30	5	301 → 304
90	4454	—	341.60 ± 1.95	5	339 → 344
100	5449	—	378.00 ± 2.83	5	374 → 380

Table 4. Characteristic polynomial of an integer Hessenberg matrix ($m = 4$).

n	exprswell	primes	random numbers	N_{samples}	min → max
3	107	93	78.07 ± 5.23	100	65 → 89
4	225	205	162.47 ± 11.43	100	123 → 187
5	404	376	283.81 ± 19.35	100	236 → 331
6	656	620	451.50 ± 30.81	100	346 → 507
7	993	946	657.26 ± 45.02	100	536 → 746
8	1427	1368	934.24 ± 56.08	100	804 → 1066
9	1970	1895	1245.57 ± 80.66	100	1080 → 1426
10	2634	2540	1612.45 ± 112.96	100	1245 → 1932

Table 5. Characteristic polynomial of a "derationalized" Hessenberg matrix, initially filled with 4-digit rational numbers.

n	exprswell	primes	random numbers	N_{samples}	min → max
3	82	61	50.15 ± 4.84	100	36 → 60
4	911	216	176.29 ± 11.92	100	137 → 202
5	13534	622	504.29 ± 27.15	100	438 → 564
6	255497	1369	1096.85 ± 46.77	93	983 → 1223
7	5852292	2588	2013.28 ± 98.13	100	1722 → 2206
8	157563119	4313	3275.51 ± 130.41	57	3027 → 3584
9	4874278234	6671	4971.33 ± 170.71	12	4730 → 5310
10	170327525637	9717	7194.50 ± 198.44	6	6932 → 7488

Table 6(a). Transformation of a general matrix, initially filled with 4-digit rational numbers, into Hessenberg form.

n	exprswell	primes	random numbers	N_{samples}	min → max
3	711	69	60.34 ± 4.05	100	46 → 69
4	15812	125	105.24 ± 5.19	100	91 → 117
5	493670	198	163.02 ± 6.45	100	145 → 178
6	19254949	286	231.30 ± 8.34	93	208 → 250
7	898326065	390	309.65 ± 11.57	100	281 → 334
8	48860659874	508	399.82 ± 14.61	57	372 → 440
9	3039845981556	642	498.25 ± 11.41	12	484 → 518
10	213099621017855	791	611.50 ± 11.43	6	596 → 631

Table 6(b). Characteristic polynomial of the Hessenberg matrix.

	Hessenberg matrix		Characteristic polynomial	
n	GCDs	no GCDs	GCDs	no GCDs
3	61	61	69	122
4	216	379	125	565
5	622	2360	198	3129
6	1369	14203	286	18365
7	2588	82576	390	----

Table 7. Transformation of a general matrix, initially filled with 4-digit "prime" rational numbers, into Hessenberg form and the characteristic polynomial of the Hessenberg matrix.

6 Concluding Remarks

Expression swell is a significant and in the long run an inevitable problem of symbolic computation. Since expression swell is ultimately unavoidable, one goal should be to make it at least manageable. One way to accomplish this is to develop a collection of procedures which will allow a user to be able to predict the progress of expression growth for a given calculation. In this paper, an attempt has been made to develop some tools to pursue the above goal for a certain class of computations. The application of these tools has had mixed success.

One tool for charting expression growth is to perform a variety of calculations with varying initial conditions and measure the sizes of the final and various intermediate results. This procedure, although tedious, has produced some interesting conclusions about the maximum size of coefficients in the characteristic polynomials of certain integer and rational number matrices where the nonzero entries were initially uniform in size.

A second tool is worst case (and best case) expression swell arithmetic. This can be used both for deriving general bounds on expression swell over a class of calculations and also for determining such bounds for specific computations. Some of the bounds looked at above have produced good estimates on real maximal expression size while others have given outrageous overestimates. At this preliminary stage in the development of a useful expression swell arithmetic, some observations can be made based on the experience gained here.

For integer worst case expression swell arithmetic, the best results (the tightest bounds) have come about when analyses have been performed on algorithms or inequalities that have minimized the mix of operations needed to obtain an estimate. For example, the analysis of the Gaussian elimination algorithm of Section 3 and many of the computed bounds in the previous section required a complex, multi-stage calculation and in many cases, this resulted in excessive estimates, even for small values of the matrix dimension. On the other hand, simple formulas like Hadamard's inequality, where the various arithmetical operations are reasonably consolidated gave good bounds on the expression swell. A major reason for this behavior is that expression swell arithmetic ignores past history. Thus, if a calculation produces a worst case outcome that just barely exceeds n digits (e.g. $11 \circledcirc \overline{n-2}$), the result will normally be treated as if it was the largest possible n-digit integer in the very next calculation, effectively neglecting the number's origin. Thus, the more steps there are in a computation, the more often these jumps in (interpreted) value can occur. Of course, this phenomenon also allows the analysis to be greatly simplified.

Defining $\mathcal{N}(n)$ in terms of an arbitrary integer base b instead of 10 (the latter was chosen only for convenience) could help to soften the effect of jumps in value if a base smaller than 10 was selected. This is because integers having the same number of base b digits will form more and smaller sets as b decreases, thus producing finer divisions of the integers. Therefore,

expression swell bounds can be made more precise and jumps in value can be lessened (for instance, 1111_{10} would jump to 9999_{10} using a decimal definition of $\mathcal{N}(n)$, but it would only jump to 2047_{10} with a binary definition, where $\mathcal{N}$ would now count the number of bits in n).

Another factor in why the expression swell arithmetic developed here sometimes tends to greatly overestimate expression growth is that no provision has been (or is easily able to be) made to account for the effects of subtractive cancellation (which can be significant for procedures such as the solution of a triangular system of equations) and cancellation of common factors in non-exact quotients. A theorem from number theory due to Ernesto Cesàro states that the probability that two integers chosen at random are relatively prime is $6/\pi^2$ or about 61% [Knu69]. Performing rational arithmetic operations again and again on elements of a given matrix involves numbers that are, after a while, far from random and so GCDs play a significant role, as was seen in Table 7.

The other major reason why the rational worst case expression swell arithmetic did so poorly is that rational operations involve extensive integer arithmetic so that the jump in value effect is greatly magnified. Nonetheless, this arithmetic did provide bounds on the size of other smaller expressions involved in the calculations which were more reasonable, as well as indicating the relative ranking by size of the various expressions present at a given stage (e.g. matrix elements or polynomial coefficients). These last remarks also apply to integer expression swell arithmetic.

Worst case expression swell arithmetic is an attempt to effectively systematize asymptotic analysis for a certain class of problems while providing a greater flexibility in its usage (e.g. the elements in a matrix need not be considered initially uniform in size in order to obtain a bound). In this paper, only numerical calculations were considered, but these or like techniques could also be applied to the coefficients and exponents of polynomial or other computations (as has occasionally, asymptotic analysis). One has to be careful with worst (or best) case analyses for complicated algorithms as what is worst case at one stage may be best case (or more usually, a mix) at another. Finally, these analyses are also useful for not only bounding expression swell (and thus computer memory usage) but also for setting limits on CPU time consumption. For example, the time required to multiply n_1 by n_2 using the simplest algorithm is $O(\mathcal{N}(n_1) \cdot \mathcal{N}(n_2))$ [Akr88].

Acknowledgements

I wish to thank Stanly Steinberg for his general helpfulness in making this research possible and meaningful and Jackie Damrau for the wonderful job she did in making this paper look elegant.

References

[Akr88] Alkiviadis G. Akritas, "A New Method for Computing Polynomial Greatest Common Divisors and Polynomial Remainder Sequences", *Numerische Mathematik*, Volume 52, 1988.

[Bar66] Erwin H. Bareiss, *Multistep Integer-Preserving Gaussian Elimination*, ANL-7213, Argonne National Laboratory, Argonne, Illinois, May 1966.

[Fox65] L. Fox, *An Introduction to Numerical Linear Algebra*, Oxford University Press, 1965.

[Gro87] Eric Grosse and Cleve Moler, "Underflow Can Hurt", *SIAM News*, Volume 20, Number 6, November 1987.

[Knu69] Donald E. Knuth, *The Art of Computer Programming, Volume 2: Seminumerical Algorithms*, Addison-Wesley Publishing Company, 1969.

[Sto72] A. P. Stone, "Induced Transformations on Exterior Product Spaces", *Tensor, N.S.*, Volume 23, 1972.

A VARIATIONAL METHOD FOR FINDING HOMOCLINIC ORBITS IN THE LARGE.

I. EKELAND
CEREMADE, UNIVERSITÉ DE PARIS-DAUPHINE.

§1. A simple equation.

Let us start form the following one-dimensional equation :

$$\ddot{q} - q + q^3 = 0 \quad , \quad q(t) \in \mathbf{R} \tag{1}$$

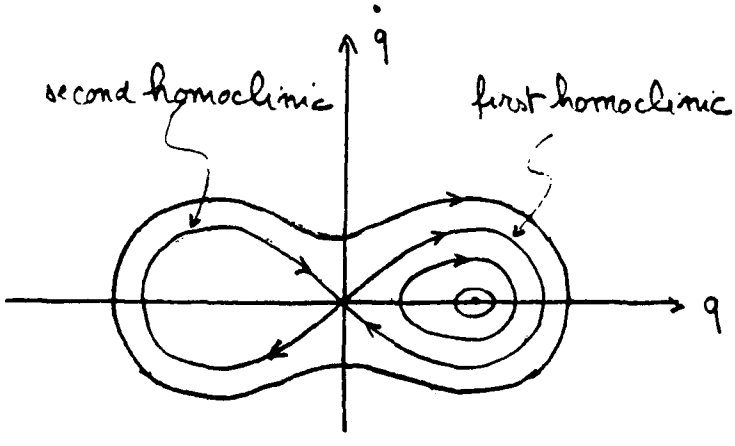
Solving it is a simple task. We know that the total energy is preserved:

$$\frac{1}{2}\dot{q}(t)^2 - \frac{1}{2}q(t)^2 + \frac{1}{4}q(t)^4 = \text{constant}$$

so that the trajectories of equation (1), in $(\dot{q}, q)$-space, are just the level curves of the function $\frac{1}{2}\dot{q}^2 - \frac{1}{2}q^2 + \frac{1}{4}q^4$. Figure 1 shows the existence of two (by symmetry) continuous families of closed level curves, corresponding to periodic solutions of equation (1). The boundary curves correspond to solutions $q(t)$ with the property that :

$$q(t) \to 0 \quad , \quad \dot{q}(t) \to 0 \quad \text{as} \quad t \to \pm\infty$$

Such solutions are nowadays called *homoclinic*, although I would prefer to call them *doubly asymptotic to the origin*, according to Poincaré's original terminology.



For future generalization, it will be convenient to rephrase equation (1) in the Hamiltonian formalism. Introducing the function

$$H(p,q) = \frac{1}{2}p^2 - \frac{1}{2}q^2 + \frac{1}{4}q^4$$

we rewrite equation (1) as a system :

$$\begin{cases} \dot{q} = p = \frac{\partial H}{\partial p} \\ \dot{p} = q - q^3 = -\frac{\partial H}{\partial q} \end{cases}$$

§2. First extension : more dimensions.

We would like to find homoclinic orbits in higher-dimensional situations. The preceding argument breaks down because the level sets $H(p,q) = h$ no longer are trajectories. In fact, no general result was known until the recent paper by V. Coti Zelati ; I. Ekeland and E. Séré (ref. [1]) which I am now proceeding to describe.

Consider a smooth Hamiltonian $H : \mathbf{R}^{2n} \to \mathbf{R}$ of the following form :

$$H(x) = \frac{1}{2}(Ax,x) + R(x)$$

under the assumptions that

(H1) $A^* = A$ and JA is hyperbolic
(no eigenvalue on the unit circle)

(H2) R is strictly convex

(H3) R is superquadratic, that is, for some $\alpha > 2$
and for suitable constants $K > k > 0$, we have :

$$R(x) \leq \frac{1}{\alpha}(R'(x),x)$$

$$k|x|^\alpha \leq R(x) \leq K|x|\alpha .$$

Define $J \in \mathcal{L}(\mathbf{R}^{2n})$ by

$$J = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix}$$

so that $J^* = -J = J^T$. We are interested in the Hamiltonian system

$$\dot{x} = JH'(x) \tag{2}$$

It has the constant (equilibrium) solution $x \equiv 0$. Periodic solutions can be found by the duality methods described in [2]. We want to find homoclinic solutions, i.e.

$$x(t) \to 0 \quad \text{when} \quad t \to \pm\infty \tag{3}$$

One approach is through the classical action principle. Associated with equation (2) is the action integral :

$$\begin{aligned} \Phi(x) &= \int_{-\infty}^{+\infty} \left[\frac{1}{2}(J\dot{x}, x) + H(x)\right] dt \\ &= \int_{-\infty}^{+\infty} \left[\frac{1}{2}(J\dot{x} + ax, x) + R(x)\right] dt \end{aligned}$$

and the solutions of the boundary-value problem (2)-(3) are the extremals of the integral $\Phi(x)$ over a suitable space of curves. However, as we already noted in [2], the functional Φ does not readily lend itself to analysis. Taking advantage of the convexity of R, we will replace Φ by a more tractable functional.

Note first that the equation

$$J\dot{x} + Ax = u$$

has a unique solution x such that $x(t) \to 0$ when $t \to \pm\infty$, for any $u \in L^\beta$. This fact crucially uses the assumption that JA is hyperbolic, and enables us to define a continuous linear map

$$\mathcal{L} : L^\beta(\mathbf{R}) \to L^\alpha(\mathbf{R}) \cap W^{1,\beta}(\mathbf{R})$$

by $x = \mathcal{L}u$. Henre $\beta = \frac{\alpha}{\alpha - 1}$ is the conjugate exponent ($1 < \beta < 2$) and

$$W^{1,\beta} = \{x \in L^\beta \mid \dot{x} \in L^\beta\} .$$

It should be noted that $\mathcal{L}$ is not a compact operator (as would happen on any finite interval). This is the reason why the Palais-Smale condition fails in this problem, as we will see later on.

Introduce the Fenchel conjugate R^* of the convex function R :

$$R^*(y) = \max \{xy - R(x) \mid x \in \mathbf{R}^n\}$$

and define a functional ψ on L^β by :

$$\psi(u) = \int_{-\infty}^{+\infty} [(\mathcal{L}u, u) + R^*(u)] \, dt$$

PROPOSITION 1. *ψ is well-defined and C^1. If u is a critical point of ψ, i.e. if $\psi'(u) = 0$, then $x = \mathcal{L}(u)$ is a solution of (2)-(3).*

FORMAL PROOF: Write $0 = \psi'(u) = \mathcal{L}u + \nabla R^*(u)$. Hence :

$$\nabla R^*(u) \;=\; -\mathcal{L}u \;.$$

By the Legendre reciprocity formula, this can be rewritten as :

$$u \;=\; \nabla R(-\mathcal{L}u)$$

or, introducing $x = -\mathcal{L}u$:

$$-J\dot{x} - Ax \;=\; \nabla R(x) \quad \square$$

The question now is to find critical points of ψ. Simple estimates show that ψ has a local minimum at the origin, with $\psi(0) = 0$, while points $u \in L^\beta$ can be found, with $\|u\|$ arbitrarily large, such that $\psi(u) < 0$. By the Ambrosetti-Rabinowitz theorem (see [2]), we conclude that there is a sequence $u_n \in L^\beta$ with

$$\psi(u_n) \;\to\; c \;>\; 0 \tag{4}$$

$$\psi'(u_n) \;\to\; 0 \;. \tag{5}$$

At this stage, one would like to conclude that u_n has a convergent subsequence, $u_{n-k} \to \overline{u}$ with $\psi'(\overline{u}) = 0$. This is the so-called Palais-Smale condition. Unfortunately it does not hold in this problem. The analytical reason is that the operator $\mathcal{L}$ is not compact, the underlying geometrical reason is the fact that the problem is translation-invariant, and the symmetry group $\mathbf{R}$ is not compact.

To see what could happen, suppose there actually is some critical point $\overline{u} \neq 0$, corresponding to a homoclinic orbit $\overline{x} = -\mathcal{L}\overline{u}$. Set :

$$(\theta * u)(t) \;=\; u(t+\theta) \quad , \quad \text{for } \theta \in \mathbf{R} \;.$$

Define a sequence u_n in L^β by :

$$u_n \;=\; u + n * u \quad , \quad n \to +\infty \;.$$

In other words, u_n is the sum of two bumps which split apart. It is not difficult to see that $u(t) \to 0$ exponentially fast when $t \to \pm\infty$, so that, as n increases there is less and less interaction between u and $n*u$, each of which solves $\psi' = 0$. In the limit, we get :

$$\psi'(u_n) \to 0 \qquad \text{when} \quad n \to +\infty \, .$$

The concentration-compactness lemma of Pierre-Louis Lions (see [3], [4]) tells us that this is exactly what does happen. In fact, from (4)-(5) we conclude that there is some subsequence u_{n_k}, finitely many critical points $u^1, \ldots, u^N$ of ψ, and corresponding sequences $p_k^1, \ldots, p_k^N$ in $\mathbb{Z}$ such that :

$$\left\| u_{n_k} - \sum_{i=1}^{N} p_k^i * u^i \right\|_\beta \to 0$$

$$|p_k^i - p_k^j| \to \infty \quad \text{if } i \neq j$$

$$c = \sum_{i=1}^{N} \psi(u^i)$$

Any of the u^i solves $\psi' = 0$, so that the corresponding $x^i = -\mathcal{L}u^i$ solves (2)-(3). We have proved :

THEOREM 1. *Under assumptions (H1) to (H3), the system has at least one homoclinic orbit.*

§3. Second extension : non-autonomous case.

Let us now consider the equation :

$$\dot{x} = JH'(x) = JAx + JR'(t,x) \tag{6}$$

under assumptions (H1) to (H3). The latter has to hold uniformly with respect to t :

$$\text{(H3)} \qquad \begin{cases} R(t,x) \leq (R'(t,x),x) \\ k|x|^\alpha \leq R(t,x) \leq K|x|^\alpha \, . \end{cases}$$

Assume in addition the R is time-periodic :

$$\text{(H4)} \qquad \exists\, T : \qquad R(t+T,x) \;=\; R(t,x) \qquad \forall\,(t,x)\,.$$

We may then introduce the time-map in phase space. This is the map $f : \mathbf{R}^{2n} \to \mathbf{R}^{2n}$ which associates $x(T)$ with $x(0)$

$$\left.\begin{aligned} x(0) \;&=\; x_0 \\ x(T) \;&=\; x_1 \end{aligned}\right\} \Longrightarrow\; x_1 \;=\; f(x_0)\,.$$

Because of (H3), the origin is a fixed point for f :

$$f(0) \;=\; 0\,.$$

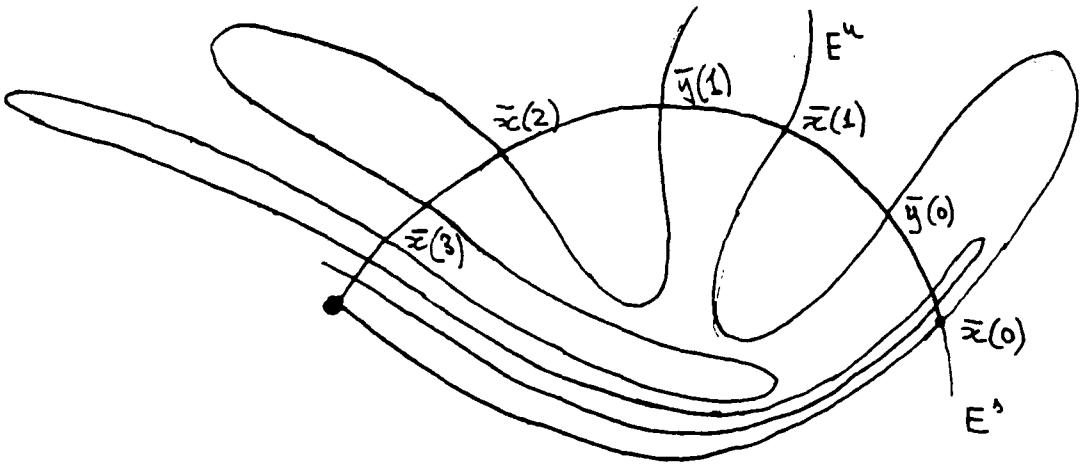
Assumption (H1) tells us that 0 is a hyperbolic fixed point. By the standard theory of dynamical systems, there are two n -dimensional manifolds branching off from 0, the stable one E^s and the unstable one E^u :

$$\begin{aligned} E^s \;&=\; \left\{x \in \mathbf{R}^{2n} \mid f^k(x) \to 0 \text{ when } k \to \infty\right\} \\ E^u \;&=\; \left\{x \in \mathbf{R}^{2n} \mid f^k(x) \to 0 \text{ when } k \to -\infty\right\}\,. \end{aligned}$$

Any homoclinic orbit $\overline{x}(t)$ gives us a sequence of homoclinic points $\overline{x}(nT)$, $n \in \mathbf{Z}$:

$$\forall\, n \in \mathbf{Z}\,, \qquad \overline{x}(nT) \;\in\; E^s \cap E^u\,.$$

The manifolds E^s and E^u therefore have infinitely many intersection points. The first to notice this fact was Poincaré (see [6]), and the intricate picture which arises from his analysis its now classical :



One homoclinic solution $\overline{x}(t)$ is displayed
Another one $\overline{y}(t)$ is found by inspection

It turns out that, if the intersection of E^s and E^u is transversal, there are many more points in this intersection than the $\overline{x}(nT)$. In other words, if there is one homoclinic orbit, there should be many more.

This argument relies on transversality of the intersection $E^s \cap E^u$, a fact that cannot be checked except in very special situations (e.g. Melnikov theory, see [5]) and can at best be hoped to be generic. We have therefore wondered whether we can do better by variational methods. It turns out to be the case :

THEOREM 2. *(Coti-Zelati, Ekeland, Séré [1])*
Under assumptions (H1) to (H4), there are at least two homoclinic orbits $\overline{x}_1$ and $\overline{x}_2$ which are geometrically distinct :

$$\forall\, n \in \mathbf{Z}\,, \qquad (nT) * \overline{x}_1 \neq x_2 \qquad \square$$

THEOREM 3. *(Séré [7])*
Under assumptions (H1) to (H4) there are infinitely many homoclinic orbits $\overline{x}_n$, $n \in \mathbf{N}$, pairwise geometrically distinct :

$$i \neq j \quad \Longrightarrow \quad \forall\, n \in \mathbf{Z}\,,\ (nT) * \overline{x}_i \neq \overline{x}_j \qquad \square$$

Clearly theorem 3 contains theorem 2. The proofs, however, are distinct, betraying the fact (which already arises from a careful investigation of Poincaré's argument) that the second solution is somehow more fundamental than the remaining ones. The second solution is found by a min-max argument around level $2c$. The other solutions are found by perturbation arguments around much higher levels.

The main theoretical advance in both situations is the introduction of a new condition, termed (PSS) (Palais-Smale-Séré) which goes as follows.

DEFINITION 4. *A function ψ is said to satisfy (PSS) if any sequence u_n such that :*

$$\psi(u_n) \to c \quad , \quad \psi'(u_n) \to 0 \quad , \quad \|u_{n+1} - u_n\| \to 0$$

has a convergent subsequence. $\square$

It is remarquable that condition (PSS) (instead of (PS)) is enough for the deformation lemma to hold.

PROPOSITION 5. *Define $\psi : L^\beta \to \mathbf{R}$ as above, and assume there are finitely many critical points of ψ (up to a time translation by some multiple of T). Then (PSS) holds for ψ.* □

The idea behind the proof is that a splitting such as the one we described, with two bumps separating and going away from each other while ψ' goes to zero, cannot occur continuously. In fact, if $\psi'(\overline{u}) = 0$, and if we set :

$$u_\theta \;=\; u + \theta * u$$

with $\theta \to \infty$, we will have $\psi'(u_{nT}) \to 0$ as $n \to \infty$, since the equation is T-periodic, but $\psi'(u_\theta)$ will remain bounded away from 0 as long as θ is bounded away from multiples of T.

We now describe the two min-max procedures that enable us to find the two first homoclinic orbits.

The first one, already alluded to in the preceding section, consists in choosing some point $v \in L^\beta$ with $\|v\|$ large such that $\psi(v) < 0$, and considering all continuous paths connecting 0 and v.

$$\Gamma \;=\; \left\{\gamma \in C^0([0,1]; L^\beta) \mid \gamma(0) = 0\ ,\ \gamma(1) = v\right\}$$

and in defining :

$$c \;=\; \inf_{\gamma\in\Gamma} \max\left\{\psi \circ \gamma(t) \mid 0 \le t \le 1\right\}\ .$$

Then $c > 0$ and (PSS) implies that it is a critical value (no splitting occurs)

$$\exists\, \overline{u}\ :\qquad \psi'(\overline{u}) = 0 \quad \text{and} \quad \psi(\overline{u}) = c\ .$$

The second one consists in introducing a set of continuous maps from the square $K = [0,1]^2$ into L^β :

$$\Sigma \;=\; \left\{ \sigma \in C^0(K; L^\beta) \;\middle|\; \begin{array}{r} \gamma(s,0 = 0 \quad \forall\, s \\ \gamma(0,1) = v = \gamma(1,1) \\ \gamma(s+1,t) = T * \gamma(s,t) \end{array} \right\}$$

and in defining :

$$d = \inf_{\gamma \in \Sigma} \max \left\{ \psi \circ \sigma(s,t) \mid (s,t) \in K \right\} .$$

It can be shown that either $d = c$ (in which case there are infinitely many critical points on that level) or $d > c$ (in which case, by (PSS), there is a critical point $\overline{v}$ with $\psi(\overline{v}) = d \neq c = \psi(\overline{u})$, su $\overline{u} \neq \overline{v}$). In both cases there is a second solution.

BIBLIOGRAPHIE

[1] V. Coti-Zelati, I. Ekeland, E. Séré, *"A variational approach to homoclinic orbits in Hamiltonian systems"*, to appear, Mathematische Annalen.

[2] I. Ekeland, *"Convexity methods in Hamiltonian mechanics"*, Springer 1990.

[3] P.L. Lions, *"The concentration-compactness principle in the calculus of variations"*, Annales de l'IHP "Analyse non linéaire".

[4] P.L. Lions, *"The concentration-compactness principle in the calculus of variations"*, Revista Matematica Iberoamericana 1 (1985), p. 145-201.

[5] V.K. Melnikov, *"On the stability of the center for periodic perturbations"*, Trans. Moscow Math. Soc. 12 (1963) p. 1-57.

[6] H. Poincaré, *"Les Méthodes nouvelles de la mécanique céleste"*, Gauthier-Villars, 1899.

[7] E. Séré, *"Existence of infinitely many homoclinic orbits in Hamiltonian systems"*, preprint CEREMADE, 1990.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188
Exp Date Jun 30, 1986

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release: Distribution unlimited
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) ARO Report 91-1	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Army Research Office 6b. OFFICE SYMBOL (If applicable) SLCRO-MA	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State, and ZIP Code) P.O. Box 12211 Research Triangle Park, NC 27709-2211	7b. ADDRESS (City, State, and ZIP Code)
8a. NAME OF FUNDING / SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO / TASK NO / WORK UNIT ACCESSION NO

11. TITLE (Include Security Classification)

Transactions of the Eighth Army Conference on Applied Mathematics and Computing

12. PERSONAL AUTHOR(S)

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Technical Report	FROM Jan 90 TO Feb 91	1991 February	925

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Fluid and solid mechanics, mathematical physics and numerical methods, symbolic computation, control theory, and stochastic techniques.

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

(U) This is a technical report resulting from the Eighth Army Conference on Applied Mathematics and Computing. It contains most of the papers in the agenda of this meeting. These treat many Army applied mathematical problems.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION
22a. NAME OF RESPONSIBLE INDIVIDUAL Dr. Francis G. Dressel	22b. TELEPHONE (Include Area Code) 919-549-4319 22c. OFFICE SYMBOL SLCRO-MA

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted
All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED