(4)

AD-A213 349

AL-TR-89-011

AD:

Final Report
for the Period
July 1988 to
March1989

Gas Flows in Rocket Motors

Volume 2. Appendix C. Time Iterative Solution of Viscous Supersonic Flow

August 1989

Authors: Science Applications International Corporation
21151 Western Avenue
Torrance CA 90501

Y. Kronzon
C. Lyan-Chang
C.L. Merkle
Pennsylvania State University
Mechanical Engineering Department
University Park PA 16802

F04611-88-C-0014

Approved for Public Release

Distribution is unlimited. The AL Technical Services Office has reviewed this report, and it is releasable to the National Technical Information Service, where it will be available to the general public, including foreign nationals.

Prepared for the **Astronautics Laboratory (AFSC)**

Air Force Space Technology Center
Space Systems Division
Air Force Systems Command
Edwards Air Force Base, California 93523-500(

NOTICE

When U.S. Government drawings, specifications, or other data are used for any purpose other than a definitely related Government procurement operation, the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise, or in any way licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may be related thereto.

FOREWORD

This is the final report for Task 1, Navier Stokes Analysis of Rocket Nozzles, for SETA contract F04611-88-C-0014 with the Astronautics Laboratory (AFSC), Edwards AFB CA. This work was performed by Pennsylvania State University as a subcontractor to Science Applications International Corporation (the SETA contractor). Dr Philip A. Kessel was the project manager for this analysis task.

This report has been reviewed and is approved for release and distribution in accordance with the distribution statement on the cover and on the DD Form 1473.

PHILIP A. KESSEL, Ph.D.
Project Manager

LAWRENCE P. QUINN, PH.D.
Chief, Aerothermochemistry Branch

FOR THE DIRECTOR

ROBERT C. CORLEY
Deputy Director, Astronautical Sciences Division

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for Public Release; Distribution is unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S) AL-TR-89-011
6a. NAME OF PERFORMING ORGANIZATION Pennsylvania State University 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION Astronautics Laboratory (AFSC)
6c. ADDRESS (City, State, and ZIP Code) Mechanical Engineering Department University Park PA 16802	7b. ADDRESS (City, State, and ZIP Code) AL/LSCF Edwards AFB CA 93523-5000
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F04611-88-C-0014
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. 62302F; PROJECT NO. 3058; TASK NO 00; WORK UNIT ACCESSION NO. SD

11. TITLE (Include Security Classification)
Gas Flows in Rocket Nozzles. Volume 2: Appendix C - Time-Iterative Solutions of Viscous Supersonic Flow

12. PERSONAL AUTHOR(S)
Kronzon, Yigal; Lyan-Chang, Chau; and Merkle, Charles L.

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Final	FROM 88/07 TO 89/03	89/06	

16. SUPPLEMENTARY NOTATION
This is volume 2 of a 3 volume report

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	nozzle analysis, Navier-Stokes, turbulent flow, equilibrium chemistry
20	04		

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Detailed descriptions of the governing equations, the method of solution, and the computer code for calculating perfect gas or real gas flow in axisymmetric nozzles are given. The codes permit calculation of a perfect gas or a real gas for inviscid flow by solving the Euler equations, and for viscous flow by solving the thin layer Navier-Stokes equations. These equations are written in a conservative form and solved implicitly in body-fitted coordinates. The solution obtained by the conservative variables is expressed in terms of the density, ρ, the momentum parallel to the axis of symmetry (ρu), the momentum perpendicular to the axis (ρv), and the total internal energy, e_o. These variables are then used to calculate the noncon.ervative primitive variables, the velocity components, u and v, the pressure, p, and the temperature, T. The nozzle performance including the rate of mass flow, $\dot{m}$, the thrust, $\mathcal{T}$, and the specific impulse are also computed.

The codes were written in FORTRAN V and ran on the CYBER 180/840, NOS/BE system which limited the number of grid points to 20 x 44 for solving the Euler equations and 60 x 40 for

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	UNCLASSIFIED
22a NAME OF RESPONSIBLE INDIVIDUAL Dr Philip A. Kessel	22b. TELEPHONE (Include Area Code) (805) 275-5591
	22c. OFFICE SYMBOL AL/LSCF

DD Form 1473, JUN 86 *Previous editions are obsolete.* SECURITY CLASSIFICATION OF THIS PAGE

AL-TR-89-011 Vol 1 (continued)

Block 19.

the TLNS equations. The results obtained for the nozzle flowfield showed maximum global mass flux errors of lwss than $\pm 1\%$ for the Euler equations and less tha $\pm 2\%$ for the TLNS equations. Solutions with more dense grids (typically 100 x 50 or higher) consistently showed global mass conservation of better than one percent.

ABSTRACT

Numerical solutions of supersonic viscous flows are studied by applying an implicit time-dependent scheme to the thin-layer Navier-Stokes(TLNS) equations. The alternating direction implicit(ADI) scheme is first formulated to solve transonic viscous axisymmetric flows in two dimensions. The results indicate that the ADI scheme is not efficient enough for supersonic viscous calculations.

Accordingly, a spatial discretization scheme using upwind flux-vector split differencing in the streamwise direction and central differencing in the cross-stream direction is chosen. Three approximate factorization schemes and one fully implicit direct solver are considered. Of them, the diagonally dominant ADI(DDADI) and the parabolized ADI are found to be much faster than the standard ADI procedure. The optimum CFL number for the DDADI method is about 5000 and it provides competitive convergence with direct solvers. In terms of CPU time requirements, the parabolized ADI procedure is as fast as the DDADI method.

These numerical algorithms are applied to solve supersonic flows through conical and high expansion ratio contoured nozzles for different Reynoids numbers, wall temperatures, and back pressures. Proper downstream boundary conditions for the subsonic portion of the outflow are shown to allow variations of the boundary layer thickness at the exit plane and recirculating separated flows for sufficiently high back pressure. Excellent global mass conservations are demonstrated by using the fully conservative form, while quasi-conservative formulations lead to unacceptably large mass conservation errors.

Along with the investigations of Navier-Stokes algorithms, parabolized Navier-Stokes(PNS) procedures are also studied. The PNS algorithms are devised from generalized flux split TLNS equations which include both the traditional pressure gradient split procedure and a characteristics split system. Comparisons with TLNS

results show that the characteristics-based PNS system gives results that are as accurate as pressure-gradient-split PNS procedures. The use of a safety factor in the pressure gradient splitting is shown to cause inaccuracies and should be avoided.

The global pressure iteration for the PNS algorithm is interpreted as an alternating-direction procedure for the TLNS equations. This global procedure is shown to be mathematically well-posed and numerically efficient.

Swirling viscous flows in transonic and supersonic propulsive nozzles have been investigated numerically. The central-difference ADI and the flux-vector split algorithms are utilized to solve the thin-layer Navier-Stokes equations for axisymmetric two-dimensional flow with swirl. The effects of swirl on viscous flow are studied for nozzles with mild to high expansion ratios. Both flowfield detail and integral nozzle performance are compared to previously published inviscid calculations. The results show that the presence of swirl has a significant effect on the flowfield and integral nozzle performance, especially for plug nozzle and high expansion ratio nozzles.

Finally, the algorithms developed for axisymmetric two-dimensional flows are extended to solve the three-dimensional TLNS equations. The algorithms are based upon DDADI splitting for the streamwise flux vector and additional approximate factorization of the operators on the cross-stream plane. The optimum CFL number reduces to the order of 10 and it gives slower convergence as compared to corresponding two-dimensional algorithms due to the approximate factorization error.

TABLE OF CONTENTS

Page

LIST OF FIGURES viii

LIST OF SYMBOLS xiii

ACKNOWLEDGEMENTS xvi

Chapter 1 INTRODUCTION 1

Chapter 2 THE APPLICATION OF TIME-ITERATIVE SCHEMES TO THE ONE-DIMENSIONAL EULER EQUATIONS 10

2.1 Governing Equations 10
2.2 The Central-Differencing Algorithm 13
2.2.1 Boundary Conditions 14
2.2.2 Stability Analysis 17
2.2.3 Computational Results 23
2.3 The Upwind-Differencing Algorithm 26
2.3.1 Boundary Conditions 32
2.3.2 Stability Analysis 33
2.3.3 Computational Results 37

Chapter 3 THE APPLICATION OF TIME-ITERATIVE SCHEMES TO THE TWO-DIMENSIONAL NAVIER-STOKES EQUATIONS 41

3.1 Governing Equations 41
3.2 The Implicit ADI Scheme 47
3.2.1 Boundary Conditions for the ADI Scheme 51
3.2.2 Stability Analysis of the ADI Scheme 54
3.2.3 Results and Discussion 59
3.3 Stability Consideration of Upwind Algorithms 66
3.3.1 The Fully Implicit Scheme 67
3.3.2 The Standard ADI Scheme 68
3.3.3 The Diagonally Dominant ADI Scheme 69
3.3.4 The Parabolized ADI Scheme 70
3.3.5 Algorithm Comparisons 71
3.4 Algorithms for Viscous Supersonic Flow 76
3.4.1 The Standard ADI Procedure 77
3.4.2 The Diagonally Dominant ADI Procedure 78
3.4.3 The PNS-ADI Procedure 80

3.4.4 Direct Solution by Flowfield Partitioning 80
3.5 Algorithm Comparisons 83
3.6 Nozzle Flowfield Predictions 87
3.6.1 Verification of Solution Accuracy 91
3.6.2 Effects of Downstream Boundary Conditions 94
3.6.3 Effects of Reynolds Number and Wall Temperature 104
3.6.4 Effects of Coupled Wall Cooling and Nozzle Flows 107
3.6.5 Nozzles with Subsonic Inflow 111
3.6.6 Verification of Global Conservation 113

Chapter 4 THE APPLICATION OF TIME-ITERATIVE SCHEMES TO THE PARABOLIZED NAVIER-STOKES EQUATIONS 117

4.1 General Flux-Vector Splitting Navier-Stokes Equations 118
4.1.1 Splitting Based on Characteristics 119
4.1.2 Splitting Based on Pressure Gradient 120
4.2 Obtaining the PNS Procedure from Navier-Stokes Algorithms 123
4.2.1 Pressure Gradient Splitting 124
4.2.2 Characteristics Splitting 125
4.2.3 Non-Iterative PNS Procedure 125
4.2.4 Comparisons of Time-Iterative and Space-Marching Algorithms 127
4.2.5 Stability Analysis of Time-Iterative PNS Algorithms 130
4.2.6 Results and Discussion 130
4.3 Global PNS Procedures 142

Chapter 5 THE APPLICATION OF TIME-ITERATIVE SCHEMES TO VISCOUS SWIRLING NOZZLE FLOWS 147

5.1 Governing Equations and Numerical Algorithms 147
5.2 Boundary Conditions 150
5.3 Nozzle Flowfield Predictions 151
5.3.1 Convergent-Divergent Nozzle 152
5.3.2 Convergent Nozzle and Plug Nozzle 159
5.3.3 High Expansion Nozzle 162
5.3.4 Verification of Global Conservation 168

Chapter 6 THREE-DIMENSIONAL NOZZLE FLOWS 172

6.1 Governing Equations 172
6.2 Three-Dimensional Supersonic Algorithms 176
6.2.1 Stability Analysis for the Scalar Equation 176
6.2.2 Numerical Algorithms for the TLNS equations 182
6.3 Flowfield Predictions 189

Chapter 7 SUMMARY 206

BIBLIOGRAPHY 211

LIST OF FIGURES

Page

Figure 1. Stability results of 1-D implicit central-difference scheme without artificial viscosity 21

Figure 2. Stability results of 1-D implicit central-difference scheme with artificial viscosity 22

Figure 3. Nozzle geometry for 1-D calculations 24

Figure 4. Convergence for 1-D implicit central-difference scheme 25

Figure 5. Mach number distribution of transonic 1-D calculation, central-difference solutions 27

Figure 6. Pressure distribution of transonic 1-D calculation, central-difference solutions 28

Figure 7. Stability results of 1-D implicit first-order upwind scheme, approximate Jacobian, $M = 0.5$ 34

Figure 8. Stability results of 1-D implicit first-order upwind scheme, true Jacobian, $M = 0.5$ 35

Figure 9. Stability results of 1-D implicit first-order upwind scheme, supersonic flow, $M = 2.0$ 36

Figure 10. Convergence for 1-D implicit first-order upwind scheme 38

Figure 11. Mach number distribution of supersonic 1-D calculation, upwind solutions 39

Figure 12. Pressure distribution of supersonic 1-D calculation, upwind solutions 40

Figure 13. Stability results of 2-D fully implicit scheme without approximate factorization 56

Figure 14. Stability results of 2-D implicit ADI scheme with approximate factorization 57

Figure 15. Stability results of 2-D implicit ADI scheme with approximate factorization for various Mach numbers 58

Figure 16. Nozzle geometry for 2-D transonic calculations 60

Figure 17. Convergence of 2-D ADI scheme on uniform grid 61

Figure 18. Convergence of 2-D ADI scheme on stretched grid 62

Figure 19. Mach number contours of transonic solutions for ADI scheme 63

Figure 20. Convergence of 2-D ADI scheme, inviscid & viscous calculations for supersonic flows 65

Figure 21. Stability results of first-order upwind-central scheme, $\sigma_x^+ = -\sigma_x^- = \sigma_y = \nu = 10$ 72

Figure 22. Stability results of first-order upwind-central scheme, $\sigma_x^+ = \sigma_y = \nu = 10, \sigma_x^- = 0$ 73

Figure 23. Stability results of first-order upwind-central scheme, varying CFL, $\sigma_x^- \neq 0$ 75

Figure 24. Partitioning of physical and computational domains 81

Figure 25. 15° conical nozzle for convergence comparison with grid size 21 × 40, Area ratio 30 85

Figure 26. Convergence in terms of number of iteration of 2-D supersonic algorithms with true Jacobians 86

Figure 27. Convergence in terms of CPU time of 2-D supersonic algorithms with true Jacobians 88

Figure 28. Convergence in terms of number of iteration of 2-D supersonic algorithms with approximate Jacobians 89

Figure 29. Convergence in terms of CPU time of 2-D supersonic algorithms with approximate Jacobians 90

Figure 30. Contoured nozzle geometry with 75 × 50 grid 92

Figure 31. Comparison of Mach number contours for MOC and upwind-central difference algorithm 94

Figure 32. Comparison of wall pressure distribution for MOC and upwind-central difference algorithm 95

Figure 33. Converged Mach number contours for conical nozzle using various back pressures 98

Figure 34. Velocity profiles at nozzle exit plane for conical nozzle using various back pressures 99

Figure 35. Turbulent Mach number contours in 272:1 contoured nozzle using various back pressure 100

Figure 36. Laminar Mach number contours in 272:1 contoured nozzle using various back pressure 101

Figure 37. Velocity profiles at exit plane for turbulent flow in 272:1 contoured nozzle 102

Figure 38. Velocity profiles at exit plane for laminar flow in 272:1 contoured nozzle 103

Figure 39. Converged Mach number contours for conical nozzle at lower Reynolds number and colder wall temperature 105

Figure 40. Converged Mach number contours for high Reynolds number Extrapolated boundary conditions 106

Figure 41. Converged Mach number contours for 272:1 nozzle at lower Reynolds number 108

Figure 42. Schematic of nozzle with wall cooling 109

Figure 43. Mach number contours for various wall cooling conditions 112

Figure 44. Mach number contours for nozzle with subsonic inflow 114

Figure 45. Comparison of global mass conservation for quasi-conservative and strongly conservative schemes 116

Figure 46. Stability results for the time-iterative PNS scheme with pressure gradient splitting 131

Figure 47. Convergence rate of the time-iterative PNS procedure at typical axial location 133

Figure 48. Comparison of Mach number contours computed by PNS algorithm based on pressure gradient splitting and the TLNS algorithm 135

Figure 49. Comparison of cross-stream pressure profile at exit plane for various PNS results with TLNS calculations 136

Figure 50. Comparison of cross-stream velocity profile at exit plane for various PNS results with TLNS calculations 137

Figure 51. Mach number contours computed by PNS algorithm based on pressure gradient splitting and using a safety factor of 0.85 139

Figure 52. Pressure distribution on the wall for TLNS and PNS procedures . 140

Figure 53. Width of subsonic layer as a function of axial distance along the wall, showing results of TLNS and various PNS calculations 141

Figure 54. Mach number contours computed by PNS algorithm based on pressure gradient splitting, first-order upwind results. 143

Figure 55. Convergence of PNS-ADI algorithm for the TLNS equations for 272:1 contoured nozzle 146

Figure 56. Inviscid grid for 35°-18.5° convergent-divergent nozzle 153

Figure 57. Viscous grid for 35°-18.5° convergent-divergent nozzle 154

Figure 58. Convergence rates of C-D nozzle with swirl, inviscid calculation .. 155

Figure 59. Convergence rates of C-D nozzle with swirl, viscous calculation ... 156

Figure 60. Inviscid Mach number contours for C-D nozzle, with $S_i = 0$ and $S_i = .382$ 157

Figure 61. Viscous Mach number contours for C-D nozzle with $Re = 7,000$, $S_i = 0$ and $S_i = .464$ 158

Figure 62. Dependence of integral performance on inlet swirl number for C-D nozzle 160

Figure 63. Dependence of nozzle performance on Reynolds number for $S_i = 0$ and $S_i = .361$ 161

Figure 64. Viscous Mach number contours for convergent nozzle 163

Figure 65. Integral performance of transonic flow through convergent nozzle . 164

Figure 66. Viscous Mach number contour for annular nozzle 165

Figure 67. Integral performance of transonic flow through annular plug nozzle 166

Figure 68. Convergence rates of contoured nozzle with swirl for viscous calculations 167

Figure 69. Laminar Mach number contour for high expansion contoured nozzle for $S_i = 0$ and $S_i = .521$ 169

Figure 70. Turbulent Mach number contour for high expansion contoured nozzle for $S_i = 0$ and $S_i = .551$ 170

Figure 71. Integral performance for supersonic flow through contoured nozzle 171

Figure 72. Stability results of 3-D Burger's equation based on unfactored scheme for subsonic case 180

Figure 73. Stability results of 3-D Burger's equation based on factored scheme for subsonic case 181

Figure 74. Stability results of 3-D Burger's equation based on unfactored scheme for supersonic case 183

Figure 75. Stability results of 3-D Burger's equation based on factored scheme for supersonic case 184

Figure 76. Geometry of 15^0 expanding 3-D nozzle for convergence test 188

Figure 77. $5 \times 30 \times 30$ grid of 15^0 expanding 3-D nozzle for convergence test 190

Figure 78. Convergence of 3-D PNS algorithm with $CFL = 20$ 191

Figure 79. Convergence of 3-D TLNS algorithm with $CFL = 20$ 192

Figure 80. Nozzle geometry of 3-D flowfield prediction 193
Figure 81. 75 × 30 surface grid on the side wall of the 3-D nozzle 195
Figure 82. 30 × 30 grids on the inlet and exit cross-planes. 196
Figure 83. Locations of flowfield plots for 3-D nozzle 197
Figure 84. Mach number contours at locations A and B 198
Figure 85. Streamwise velocity contours at location C and D 199
Figure 86. Streamwise velocity contours at location E 200
Figure 87. Streamwise velocity contours at location F 201
Figure 88. Cross-stream velocity vector plots at location C and D 202
Figure 89. Cross-stream velocity vector plots at location E 203
Figure 90. Cross-stream velocity vector plots at location F 204

LIST OF SYMBOLS

a	One-Dimensional Nozzle Area
A,B,C	Flux Vector Jacobian Matrices
ADI	Alternating-Directions Implicit
AF	Approximate Factorization
AR	Area Ratio
Bi	Biot number
B_v	Three-Dimensional Viscous Jacobian Matrix in η Direction
B_{v1}, B_{v2}	Viscous Terms Jacobian Matrices
c	Sonic Speed
C_l	Specific Heat of Cooling Liquid
C_p	Constant Pressure Specific Heat
C_v	Three-Dimensional Viscous Jacobian Matrix in ζ Direction
C_ξ	ξ Direction Characteristic Speed$(=\sqrt{\xi_x^2 + \xi_y^2}c)$
C_η	η Direction Characteristic Speed$(=\sqrt{\eta_x^2 + \eta_y^2}c\ )$
CFL	Courant-Friedrichs-Lewy Number
d	Elements of Source Jacobian Matrix
D	Source Jacobian Matrix
D'	Diagonal Matrix in DDADI algorithm
$DDADI$	Diagonally Dominant ADI
e	Total Energy
f	Arbitrary Function for Fourier Transformation
E,F,G	Flux Vectors
$Ext.$	Extrapolation
g	Amplification Factor
G	Amplification Matrix
h	Source Term in Characteristics Equations
H	Source Vector
i	$\sqrt{-1}$
I	Identity Matrix
J	Coordinate-Transformation Jacobian
k	Thermal Conductivity
K	Boundary Constants
L	Length of the Domain
L_1,L_2	Left and Right Operators in Fourier Domain
L^+,L^-	Selection Matrix
M,M_ξ,M_η	Left Eigenvector Matrix (M_ξ is also used as the streamwise Mach number)
$\dot{m}$	Mass Flow Rate
M_ξ	Streamwise Mach Number

MOC	Method of Characteristics
n	Normal Direction of the Wall
p, P	Pressure
P_b	Back Pressure
PNS	Parabolized Navier-Stokes
Q	Dependent Variable Vector
$\tilde{Q}_1, \tilde{Q}_2$	Variables for Separated Viscous Terms
R, R'	Right Hand Side Residual Vectors
R_g	Gas Constant
Re	Reynolds Number
R_1, R_2	Viscous Term Coefficient Matrix
S	Sutherland Constant
S_i	Swirl Number at Inlet
t	Time
T	Temperature
TLNS	Thin-Layer Navier-Stokes
u	Velocity in x-direction
U	ξ-Directional Contravariant Velocity
v	Velocity in y-direction
V	η-Directional Contravariant Velocity
w	Velocity in z-direction
W	ζ-Directional Contravariant Velocity
x, y, z	Coordinates
α	Viscous Terms Constants
γ	Ratio of the Specific Heats
Δ	Increments
ϵ_i	Second-Order Implicit Artificial Viscosity
ϵ_e	Fourth-Order Explicit Artificial Viscosity
ϵ	Internal-Energy per Unit Mass
θ	Normalized Contravariant Velocity
λ	Eigenvalue
Λ	Diagonal Eigenvalue Matrix
Λ_ξ	ξ-Direction Diagonal Eigenvalue Matrix
Λ_η	η-Direction Diagonal Eigenvalue Matrix
Λ_ζ	ζ-Direction Diagonal Eigenvalue Matrix
Λ^+	Flux-Split Positive-Eigenvalue Diagonal Matrix
Λ^-	Flux-Split Negative-Eigenvalue Diagonal Matrix
μ	Molecular Viscosity
ν	von Neumann Number
ρ	Density
σ	CFL number
$\bar{\sigma}$	Safety Factor for PNS Algorithm

Ψ, Φ	Constant in Eigenmatrices
ξ, η, ζ	Generalized Coordinates
ω	Pressure Gradient Coefficient
$\omega_x, \omega_y, \omega_z$	Wavenumbers
ω_ξ	ξ-Direction Wavenumber
ω_η	η-Direction Wavenumber
Ω	Boundary-Condition Vector
Ω_i	Inlet Boundary Constant Vector
Ω_e	Exit Boundary Constant Vector

SUBSCRIPTS

i, j, k	Grid Indices
I	Inviscid Flow
v	Viscous Terms
w	Quantities at Wall

SUPERSCRIPTS

n	Time-Step Index
T	Transpose of Vectors
$+$	Downstream-Propagated Flux-Split Variables
$-$	Upstream-Propagated Flux-Split Variables
$*$	Intermediate Iteration Step
0	Stagnation Quantities

MATH ACCENTS

$\tilde{}$	Fourier Transformed Functions
$\hat{}$	Characteristic Variables
$\bar{}$	Variables in General Coordinate System

ACKNOWLEDGEMENTS

The author would like to express his sincere appreciation to his advisor, Dr. Charles L. Merkle, for his guidance, assistance, and encouragement throughout the course of this study.

This research was sponsored by the Rocket Propulsion Laboratory, Edwards Air Force Base, California. The financial support is acknowledged.

The author wishes to express his gratitude to his parents, and Miss Shu-Yao Hsu, for their devotion and love.

CHAPTER 1

INTRODUCTION

Recent interest in the aerospace plane and hypersonic vehicles has revitalized research on high-speed propulsion systems. In the design of a propulsion system, accurate prediction of viscous supersonic flowfields together with certain physical parameters such as thrust play a critical role. Traditionally, these parameters are obtained from wind tunnel tests or simplified analytical models. The analytical approach is only valid for very simple geometries and flow conditions due to the difficulties in obtaining exact solutions of the complicated governing equations. Consequently, successful design has been reliant upon expensive wind tunnel experiments. With the advancement in computational fluid dynamics (CFD) and computer architectures, numerical computations now can be used as alternatives of experiments for much of the configuration design process. Although wind tunnel tests continue to be important, the trend is clearly toward the computational approach using accurate numerical schemes to enhance the experimental findings. The focus of present research is to develop accurate numerical algorithms for predicting viscous supersonic flowfields that occur in propulsion systems. In particular, the predictions of supersonic flows through high expansion ratio nozzles will be emphasized.

The analysis of viscous supersonic flows would require the solution of the compressible Navier-Stokes equations with proper boundary conditions. It is well known that the compressible Navier-Stokes equations are very difficult to solve because the whole equation set is strongly coupled and highly non-linear. To avoid directly solving this stiff non-linear system, certain degrees of approximations have to be

made. One example for this is the classical Prandtl boundary layer approach. The boundary layer assumptions allow inviscid and viscous flows to be computed independently. For supersonic flows, the governing equations for the inviscid region are rendered hyperbolic by neglecting viscous effects. This hyperbolic equation set can be efficiently solved by the method of characteristics (MOC) [1]. For the viscous region, the pressure gradient normal to the wall is neglected from order of magnitude considerations; thus, the Navier-Stokes equations reduce to boundary layer equations. Numerous attempts have been made to solve the boundary layer equations both analytically and numerically. The analytical technique given by von Karman and Pohlhausen [2] requires assumptions of the velocity profile inside the boundary layer and is only valid for very simple problems. The numerical solutions of boundary layer equations, which can handle more complex problems, have been extensively investigated since the early 1970's. Some representative algorithms are summarized by Anderson et al. [3].

The classical boundary layer approach assumes the interaction between the inviscid region and the viscous region is small; consequently, either region can be solved independently. To take into account this interaction, some sort of inviscid-viscous patching procedure has to be employed. One typical example of this approach is given by Ref. [4]. The patching method is based on the combination of an inviscid MOC procedure and a boundary layer algorithm. An iterative procedure between inviscid and viscous regions is accomplished by interchanging the wall pressure from the MOC procedure and the displacement thickness from the boundary layer procedure until convergence is achieved. This inviscid-viscous interaction technique does provide an efficient algorithm to calculate viscous supersonic flows. However, it is only valid for weak-interaction flows. For flows with strong interaction, the pressure gradient normal to the wall cannot be neglected. Thus,

the boundary layer algorithm as well as the inviscid-viscous interaction procedure are no longer applicable. One typical example of these strong interaction flows is the supersonic flow through a high-expansion nozzle. A recent work done by Kushida [5] indicates that the boundary layer displacement thickness inside the nozzle can be as large as 42% of the nozzle radius at the exit. In this regime, the inviscid-viscous patching procedure fails to describe the pressure variation inside the boundary layer and the realistic mass flow rate, thus numerical solutions of the Navier-Stokes equations are required.

For typical viscous supersonic flowfields, the governing equation set is hyperbolic in the supersonic region and is elliptic in the subsonic region inside the boundary layer. This mixed hyperbolic/elliptic character makes the steady Navier-Stokes equations extremely difficult to solve because a different numerical algorithm has to be employed in each region, as we have seen in the classical approach. However, if we consider the unsteady Navier-Stokes equations, the equation set becomes hyperbolic in time for both supersonic and subsonic regions. Therefore, given an initial guess of the flowfield, the solutions can be obtained by marching in time until the steady state is reached. This procedure, generally referred to as a time-dependent or time-iterative scheme, enables one numerical algorithm to be used throughout the flowfield. The time-dependent concept was first applied to inviscid calculations for flows over blunt bodies by Moretti and Abbett [6] in 1966. Since then, time-dependent solutions have become an important segment of CFD. The first application in compressible viscous flows was done by MacCormack [7] in 1969. In this early work, an explicit predictor-corrector scheme was proposed to solve the Navier-Stokes equations. This method is very straightforward to program but it suffers from a limitation on the time step size when only steady-state solutions are of interest.

Time-dependent schemes can also be implemented in an implicit fashion. The implicit time-dependent formulation imposes no stability limitation on the size of time steps, hence, in most cases, is superior to the explicit scheme if only steady-state solutions are concerned. One important application of implicit time-dependent algorithms to compressible Navier-Stokes equations is the alternating direction implicit (ADI) scheme suggested by Beam and Warming [8,9], which is also noted as the linearized block implicit (LBI) scheme by Briley and McDonald [10]. The ADI scheme has gained popularity since the mid-70's due to its capability to solve multi-dimensional inviscid as well as viscous flows.

With the progress in CFD during the past decade, numerous well-developed algorithms are now available for compressible Navier-Stokes calculations. These algorithms can in general be divided into two categories according to the the type of spatial discretizations. For those of central-difference type, Steger [11] formulated the ADI scheme in the general coordinate system, Baldwin and Lomax [12] solved the thin-layer Navier-Stokes (TLNS) equations with an algebraic turbulence model, and Pulliam [13] applied the implicit ADI scheme to solve flows over airfoils. For those of upwind-difference type, Lombard et al. [14] proposed a conservative supra-characteristics method (CSCM) based on non-conservative flux-difference splitting, and MacCormack developed a line Gauss-Seidel procedure based on Steger and Warming [16] flux-vector splitting. Similar investigations are also noted by other authors; these include the relaxation scheme by Chakravarthy [17], the LU scheme by Yoon and Jameson [18], the single level scheme by Lombard et al. [19], and the diagonally dominant ADI scheme by Chang et al. [20]. Thomas and Walters [21] used a similar relaxation procedure to solve two-dimensional viscous supersonic flows based on van Leer's flux-vector splitting [22]. More recently, this work has been extended to three dimensions by Newsome et al. [23].

In developing an efficient numerical algorithm that is well suited to the viscous supersonic calculations required for this study, four aspects of solutions of the Navier-Stokes equations are considered:

1. The algorithm should be able to take into account the predominantly supersonic nature of the flowfield, and consequently give rapid convergence in the high Reynolds number, unseparated limit.
2. For lower Reynolds number flows, proper downstream boundary conditions have to be implemented on the subsonic portion of the exit profile so that the flow will respond to downstream environmental changes.
3. To predict thrust with accuracy, global mass conservation has to be ensured. This feature is accomplished by using the strong conservative form of the governing equations.
4. The algorithm can be easily simplified to a certain extent such that a pure space-marching procedure is allowed for high Reynolds number, unseparated flows. For this reason, the parabolized Navier-Stokes (PNS) procedure is also considered in this study.

To begin with, the implicit time-dependent scheme is first applied to solve the quasi one-dimensional Euler equations for spatial discretizations based on both central differencing and upwind differencing. This preliminary work allows the first assessment of algorithms in terms of computational efficiency and accuracy. Some details of the algorithms such as effects of approximate Jacobians, and comparisons of accuracy between first order and second order upwind schemes, will be discussed.

Two-dimensional calculations start with the application of the ADI scheme to the axisymmetric two-dimensional TLNS equations in order to justify the appropriateness of this algorithm for viscous supersonic computations. To encounter predominantly supersonic flows, a hybrid upwind/central differencing scheme is pro-

posed along with its Fourier stability analysis [24]. Accordingly, three approximate factorization algorithms and one direct method are formulated for the solutions of the discretized TLNS equations based on this hybrid differencing scheme. To verify the accuracy of the proposed hybrid scheme, the results computed by using current algorithms will be compared to those by the MOC procedure given in Ref. [4].

For better understanding of the effects of downstream boundary conditions on the flowfield, supersonic flows through a conical nozzle and a high area ratio contoured nozzle are computed by using the proposed algorithms. The variation of flow character is obtained by varying the back pressure level. In particular, back pressure levels that are sufficiently high to produce separation inside the nozzle are considered in order to simulate the classical experimental characteristics that are observed when altitude nozzles are operated on sea-level thrust stands. The flowfield demonstrations include both laminar and turbulent calculations. The turbulent calculations are based on the Baldwin and Lomax model [12,25]. Comparisons of global mass conservation between strong conservative and weak conservative formulations are made.

Parallel to the development of Navier-Stokes algorithms, the applications of the time-dependent scheme on PNS procedures are also studied. Parabolized Navier-Stokes algorithms [26-29] have proven to be very popular because of their accuracy and efficiency. For many flowfields, they give results that are almost identical to those obtained with full Navier-Stokes equations, although the CPU time required is much less than that needed for the complete equations. The basic idea of PNS schemes is to render the steady state Navier-Stokes equations parabolic in the streamwise direction by proper approximations. This parabolic set of equations can then be solved by a space-marching procedure similar to the MOC procedure used for inviscid supersonic flows. The PNS algorithms differ from the classical

boundary layer approach in that the normal pressure gradient inside the boundary layer is retained and coupled to the pressure variation of the inviscid core flow in the parabolized equations. Consequently, PNS schemes can handle strong inviscid-viscous interaction flows without losing accuracy. The drawback of PNS algorithms is that the marching procedure fails if reverse flow is present in the flowfield [3].

The major difference between PNS procedures and Navier-Stokes solvers is that PNS schemes are normally formulated in terms of the steady state equations (see, for example Ref. [26]) while Navier-Stokes schemes are generally formulated in terms of the time-dependent equations. Because of this, it is difficult to extend a PNS algorithm to a Navier-Stokes algorithm. In the present study, PNS algorithms are obtained as a simplification of the time-dependent general flux split Navier-Stokes algorithms. One advantage of this is that a number of PNS approximations can be defined including the traditional Vigneron approach [26] and a new approach based upon the physical characteristics of the equations. Furthermore, the resulting PNS procedure still contains the temporal derivative. This requires the solutions to be obtained by iterations in time at every streamwise station. This time-iterative PNS procedure makes the space-marching problem well-posed and consequently eliminates the necessity of a safety factor that occurs in the traditional approach.

As a further example of the application of Navier-Stokes solvers mentioned above, axisymmetric swirling nozzle flows are studied. Swirling flows ahead of the combustor in ramjet applications have been suggested as a means to reduce the reattachment length of the combustor flowfield. The introduction of swirl generated by fixed vanes located in the inlet of the dump combustor can greatly increase the efficiency of the combustion process and thus reduce the length of the combustor [30]. However, the residual swirling flow in the combustor will enter the exhaust nozzle, resulting in losses in thrust and reducing the mass flow rate. Both of these

decrease the nozzle performance. Therefore, it is important to understand to what degree swirling affects the nozzle flowfield and, subsequently, the overall nozzle performance. Several previous investigations have considered the effects of swirl, but have ignored the effects of viscosity. In this study, we look at the effects of swirl as a function of nozzle Reynolds numbers.

Previous investigations of swirling nozzle flow include both quasi-one-dimensional and axisymmetric two-dimensional analyses. Carpenter et. al. [31] in an early study obtained one-dimensional results by neglecting the radial velocity component. Hoffman and co-workers [32,33] studied swirling flows in annular propulsive nozzles by means of two-dimensional inviscid numerical techniques. To parameterize their studies, they used four different inlet swirl profiles: free vortex, constant angle, forced vortex, and Rankine vortex. Their calculations are based upon the explicit MacCormack scheme [7] for the transonic flowfield, while the method of characteristics was used to compute the supersonic flowfield after the throat. They concluded that for values of swirl often encountered in ramjet and turbojet applications, the effect of swirl on the nozzle performance is small and can probably be neglected.

A recent work by Dutton [34] indicates that significant reductions in the nozzle discharge coefficient and the vacuum stream thrust efficiency may occur for high values of swirl at the inlet of the nozzle. Again, Dutton uses the [illegible] MacCormack scheme to analyze three different nozzles, including a convergent-divergent (C-D) nozzle, an annular nozzle, and a converging nozzle. Several inlet swirl profiles were enforced as inlet boundary conditions, and the corresponding effects of them were identified. He also verified the numerical results by comparing the computed wall static pressure with experiments for a C-D nozzle with an area ratio of .25.

The swirling flow investigations mentioned above are all confined to inviscid calculations. As indicated before, the boundary layer displacement thickness inside high area-ratio nozzles can be very large at the exit. In this regime, the inviscid assumption is inadequate. The present study proceeds with the numerical solutions of viscous swirling nozzle flows by using implicit time-dependent schemes. Viscous calculations are done for a series of Reynolds numbers to identify the effect of the boundary layer on swirling nozzle flows. To place these viscous results in perspective with inviscid calculations appearing in the literature, the results in the inviscid limit are also presented along with those of the viscous calculations. Additional calculations of swirling flows in high expansion nozzles are also given. Both flowfield details and the effect of swirl on the integral nozzle performance are shown.

Finally, numerical algorithms developed for axisymmetric two dimensional flows are extended for three-dimensional viscous supersonic calculations. Both PNS and global Navier-Stokes procedures are demonstrated by flowfield predictions on a three-dimensional nozzle with a rectangular cross-section.

CHAPTER 2

THE APPLICATION OF TIME-ITERATIVE SCHEMES TO THE ONE-DIMENSIONAL EULER EQUATIONS

This research starts with quasi one-dimensional calculations of compressible flows for two reasons. First, the analytical solutions of these flows are easily obtained, and thus provide back-to-back checks of the accuracy of the numerical algorithms. Second, the simplicity in the formulation of the equations allows a series of numerical experiments to be done in order to explore the potential difficulties associated with multi-dimensional calculations. The central-differencing as well as the upwind-differenced Euler implicit schemes are applied to the calculation of quasi-one-dimensional flows through a convergent-divergent nozzle. Special emphases are placed on stability analyses of the numerical algorithms and the distinctive characteristics of supersonic flows.

2.1 Governing Equations

The unsteady quasi one-dimensional Euler equations are given by

$$\frac{\partial}{\partial t}(\rho a) + \frac{\partial}{\partial x}(\rho u a) = 0$$

$$\frac{\partial}{\partial t}(\rho u a) + \frac{\partial}{\partial x}(\rho u^2 a) + a\frac{\partial p}{\partial x} = 0 \qquad (2.1)$$

$$\frac{\partial}{\partial t}(e a) + \frac{\partial}{\partial x}[(e + p)u a] = 0$$

where, standard fluid dynamic notations have been used. These include the density ρ, velocity u, pressure p, and the cross-sectional area a. The total energy e per unit

volume is defined by

$$e = \rho\epsilon + \frac{1}{2}\rho u^2$$

in which, ϵ is the internal energy per unit mass. For compressible flow, the perfect gas relation is used to close the problem.

For easier implementation of numerical procedures, Eq. (2.1) is expressed in vector form as

$$\frac{\partial Q}{\partial t} + \frac{\partial E}{\partial x} = H \tag{2.2}$$

where Q, E, and H are flow variables, flux vector, and source vector, respectively. They are defined by

$$Q = [\rho a, \rho u a, e a]^T$$

$$E = [\rho u a, (\rho u^2 + p)a, (e + p)u a]^T$$

$$H = [0, p\frac{da}{dx}, 0]^T$$

where the superscript T refers to the transpose of the vector. Equation (2.2) is written in strong conservative form [35], which is preferred for numerical computations because it conserves mass, momentum, and energy identically in the discretized form. For flows with discontinuities, this conservative formulation allows the existence of weak solutions, thus allowing shock-capturing.

The unsteady Euler equations are hyperbolic in time and can be converted into uncoupled characteristic equations. If we define the Jacobian matrix A by

$$A \equiv \frac{\partial E}{\partial Q},$$

and use the chain rule, Eq. (2.2) becomes

$$\frac{\partial Q}{\partial t} + A\frac{\partial Q}{\partial x} = H. \tag{2.3}$$

For the present one-dimensional case, A is found to be

$$A = \begin{bmatrix} 0 & 1 & 0 \\ \frac{\gamma-3}{2}u^2 & (3-\gamma)u & \gamma - 1 \\ -\frac{\gamma e u}{\rho} + (\gamma - 1)u^3 & \frac{\gamma e}{\rho} - \frac{3}{2}(\gamma - 1)u^2 & \gamma u \end{bmatrix}.$$

The Jacobian matrix A can be transformed to a diagonal matrix via the similarity transformation defined by

$$A = M\Lambda M^{-1}. \tag{2.4}$$

The diagonal matrix Λ takes the form

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}.$$

where λ_1,λ_2,and λ_3 are eigenvalues of the matrix A. Matrices M and M^{-1} are composed of the left and right eigenvectors of the matrix A, respectively. For the matrix A given above, three eigenvalues are

$$\lambda_1 = u$$

$$\lambda_2 = u + c$$

$$\lambda_3 = u - c$$

in which, c is the speed of sound. The left and right eigenmatrices M and M^{-1} are given by

$$M = \begin{bmatrix} 1 & \frac{\rho}{\sqrt{2}c} & \frac{\rho}{\sqrt{2}c} \\ u & \frac{\rho}{\sqrt{2}}\left(\frac{u}{c}+1\right) & \frac{\rho}{\sqrt{2}}\left(\frac{u}{c}-1\right) \\ \frac{u^2}{2} & \frac{\rho}{\sqrt{2}}\left(\frac{u^2}{2c}+\frac{c}{\gamma-1}+u\right) & \frac{\rho}{\sqrt{2}}\left(\frac{u^2}{2c}+\frac{c}{\gamma-1}-u\right) \end{bmatrix}$$

and

$$M^{-1} = \begin{bmatrix} 1-\frac{(\gamma-1)u^2}{2c^2} & \frac{(\gamma-1)u}{c^2} & \frac{1-\gamma}{c^2} \\ \frac{1}{\sqrt{2}\rho}\left(\frac{\gamma-1}{2}\frac{u^2}{c}-u\right) & \frac{1}{\sqrt{2}\rho}\left(1+(1-\gamma)\frac{u}{c}\right) & \frac{\gamma-1}{\sqrt{2}\rho c} \\ \frac{1}{\sqrt{2}\rho}\left(\frac{\gamma-1}{2}\frac{u^2}{c}+u\right) & \frac{1}{\sqrt{2}\rho}\left(-1+(1-\gamma)\frac{u}{c}\right) & \frac{\gamma-1}{\sqrt{2}\rho c} \end{bmatrix}.$$

Equation (2.3) now becomes

$$\frac{\partial Q}{\partial t} + M\Lambda M^{-1}\frac{\partial Q}{\partial x} = H.$$

If we define the characteristic variable $\hat{Q}$ by

$$\frac{\partial \hat{Q}}{\partial t} \equiv M^{-1}\frac{\partial Q}{\partial t}$$

and multiply Eq. (2.2) by M^{-1}, we have

$$\frac{\partial \hat{Q}}{\partial t} + \Lambda \frac{\partial \hat{Q}}{\partial x} = \hat{H} \tag{2.5}$$

where

$$\hat{H} \equiv M^{-1} H.$$

Equation (2.5) is equivalent to the three decoupled characteristic equations

$$\frac{\partial \hat{q}_i}{\partial t} + \lambda_i \frac{\partial \hat{q}_i}{\partial x} = \hat{h}_i \qquad i = 1,2,3 \tag{2.6}$$

in which $\hat{q}_i$ and $\hat{h}_i$ are elements of $\hat{Q}$ and $\hat{H}$, respectively.

The procedure above demonstrates that the one-dimensional Euler equations can be transformed into three characteristic equations with each equation governing one-dimensional wave propagation with a specific direction. These characteristic equations can be obtained by multiplying the governing equations in vector form by the eigenmatrix M^{-1}. For subsonic flows, λ_1 and λ_2 are positive ,while λ_3 is negative. The equation set possesses both right and left running characteristics. For supersonic flows, all three eigenvalues are positive, thus the waves can only travel from upstream to downstream. As will be discussed later in this chapter, this allows a marching procedure to be used for supersonic flow calculations.

2.2 The Central-Differencing Algorithm

To solve Eq. (2.2) numerically, the central-differencing Euler implicit scheme is considered. Symbolically, the Euler implicit scheme can be expressed as

$$\frac{Q^{n+1} - Q^n}{\Delta t} + \left(\frac{\partial E}{\partial x} - H\right)^{n+1} = 0 \tag{2.7}$$

where superscripts $n+1$ imply these quantities are to be evaluated at the new time level. If we define $\Delta Q = Q^{n+1} - Q^n$, the flux vector E and the source vector H can be linearized according to the following local Taylor series expansions

$$E^{n+1} = E^n + A\Delta Q \tag{2.8}$$

$$H^{n+1} = H^n + D\Delta Q \tag{2.9}$$

in which D is the Jacobian matrix defined by $D = \partial H/\partial Q$. For the present quasi one-dimensional case, the matrix D is given by

$$D = \begin{bmatrix} 0 & 0 & 0 \\ \frac{\gamma-1}{2}u^2\frac{da}{dx} & (1-\gamma)u\frac{da}{dx} & (\gamma-1)\frac{da}{dx} \\ 0 & 0 & 0 \end{bmatrix}.$$

Upon substitution of Eq. (2.8)and Eq. (2.9) into Eq. (2.7), we have

$$(I - \Delta tD + \Delta t\frac{\partial}{\partial x}A)\Delta Q = -\Delta tR \tag{2.10}$$

where R is the residual vector evaluated at time level n,

$$R = (\frac{\partial E}{\partial x} - H)^n. \tag{2.11}$$

Note that all the derivatives $\partial/\partial x$ in Eq. (2.10) and Eq. (2.11) imply discretizations by central differencing. The left-hand side operator of Eq. (2.10) results in a block tri-diagonal matrix. Each block is a 3 × 3 matrix.

2.2.1 Boundary Conditions

For hyperbolic equations, the boundary conditions can be easily enforced by using the MOC boundary procedure suggested by Rai and Chaussee [36] and Chakravarthy [37]. As indicated earlier, the governing equations imply three waves travel with specific directions. Boundary conditions are imposed for those waves running into the computational domain, while for waves moving from inside the domain toward the boundary, the decoupled characteristic equations such as those given in Eq. (2.6) are used to allow the information to propagate from inside the domain.

For subsonic inflows, λ_1 and λ_2 are positive, which implies two conditions must be specified at the upstream end. A reasonable choice is to specify the stagnation

pressure P^0 and the stagnation temperature T^0. Let these specified values of P^0 and T^0 be given as

$$P^0 = K_1$$

$$T^0 = K_2.$$

If we define a vector Ω by

$$\Omega \equiv (P^0, T^0, 0)^T,$$

then, from Taylor series expansion of Ω, we have

$$\Delta\Omega \equiv \Omega^{n+1} - \Omega^n = \frac{\partial\Omega}{\partial Q}\Delta Q$$

where $\partial\Omega/\partial Q$ is the Jacobian matrix of Ω. To force Ω^{n+1} to be fixed at the value of Ω_t, where $\Omega_t = (K_1, K_2, 0)^T$, the following equation can be employed

$$\frac{\partial\Omega}{\partial Q}\Delta Q = \Omega_t - \Omega^n. \tag{2.12}$$

Since the third eigenvalue λ_3 is negative, we must select the decoupled characteristic equation corresponding to λ_3 from Eq. (2.6) to complete the upstream boundary conditions. If we define the selection matrix L^- by

$$L^- = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

and multiply L^-M^{-1} on both sides of Eq. (2.10), we obtain the characteristic equation corresponding to λ_3 as

$$L^-M^{-1}(I - \Delta tD + \Delta t\frac{\partial}{\partial x}A)\Delta Q = -\Delta tL^-M^{-1}R. \tag{2.13}$$

Combining Eq. (2.12) and Eq. (2.13), the discretized equation at the inlet boundary can be written as

$$[\frac{\partial\Omega}{\partial Q} + L^-M^{-1}(I - \Delta tD + \Delta t\frac{\partial}{\partial x}A)]\Delta Q = \Omega_t - \Omega^n - \Delta tL^-M^{-1}R. \tag{2.14}$$

For supersonic flows at the inlet, all three characteristics come from outside the domain; therefore all entries of the dependent variable Q have to be specified. For supersonic flows at the exit, all three characteristics are outgoing; therefore, the discretized equation itself, Eq. (2.10), can be applied directly without any special treatment. For subsonic flows at the exit, λ_3 is negative, hence one boundary condition has to be specified. Let the specified quantity be the back pressure $P_b = K_3$. The vector Ω now takes the form

$$\Omega = (0, 0, P_b)^T.$$

To select the characteristic equations corresponding to λ_1 and λ_2, one can choose the selection matrix L^+ as

$$L^+ = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Similarly, the discretized equation for this case is

$$[\frac{\partial \Omega}{\partial Q} + L^+ M^{-1}(I - \Delta t D + \Delta t \frac{\partial A}{\partial x})]\Delta Q = \Omega_e - \Omega^n - \Delta t L^+ M^{-1} R \qquad (2.15)$$

where the constant vector $\Omega_e = (0, 0, K_3)^T$.

In the discretized equations at the boundaries, Eq. (2.14) and Eq. (2.15), the centrally differenced spatial derivatives $\partial/\partial x$ are not applicable. To remedy this, we use two-point one-sided differences instead of central-differences. This approach retains the block tri-diagonal structure of the left hand side matrix but is only first order accurate.

In order to have better solution accuracy, three-point one-sided differences can be used. This results in extra elements at the first and the last row of the left hand side matrix, which can be eliminated easily by elementary matrix operations [38]. This approach retains second order spatial accuracy throughout the whole computational domain, and will be generally used for the discretized equations at the boundaries.

2.2.2 Stability Analysis

The application of Fourier or von Neumann stability analysis [24] has become a powerful tool for today's CFD. In developing a new numerical algorithm, the stability analysis provides abundant information about the convergence requirements of various parameters involved in the algorithm. Before attempting to solve Eq. (2.2) numerically, we consider the Fourier analysis of its discretized form, Eq. (2.10).

For any given function $f(x,t)$, the Fourier transform is defined by

$$\bar{f}(\omega,t) = \int_{-\infty}^{\infty} f(x,t)e^{-i\omega x}dx$$

where i is the square root of -1. This transform exists only if $f(x,t)$ is square summable, that is,

$$\int_{-\infty}^{\infty} f^2(x,t)dx < \infty.$$

The inverse transformation which transforms f from the frequency domain to the spatial main is defined by

$$f(x,t) = \int_{-\pi}^{\pi} \bar{f}(\omega,t)e^{i\omega x}\frac{d\omega}{2\pi}.$$

The analogous transform for a function $q(x,t)$ defined only at discretized points can be written as

$$\bar{q}(\omega,t) = \Delta x \sum_{i=-\infty}^{\infty} q_i(t)e^{-i\omega x_i}$$

or

$$\bar{q}^n(\omega) = \Delta x \sum_{i=-\infty}^{\infty} q_i^n e^{-i\omega x_i}$$

where the superscripts n denote the time step ($t = n\Delta t$) and the subscripts i represent the spatial step ($x_i = i\Delta x$). The inverse transformation for the discretized function q is given by

$$q_i^n = \int_{-\pi/\Delta x}^{\pi/\Delta x} \bar{q}^n(\omega)e^{i\omega x_i}\frac{d\omega}{2\pi}. \qquad (2.16)$$

Upon substitution of a specific Fourier mode with frequency ω into the discretized equation, we obtain the functional relationship of the amplification factor defined by

$$g = \frac{q^{n+1}(\omega)}{q^n(\omega)}. \tag{2.17}$$

The stability criteria for any specific algorithm are then determined by the magnitude of g. If $|g|$ is greater than unity, the amplitude corresponding to the wave mode ω is growing, and hence is unstable. If $|g|$ is less than unity for all wave modes, the algorithm is stable.

In solving the central-difference discretized equation numerically, second order as well as fourth order artificial dissipation terms are added to Eq. (2.2) to avoid odd-even decoupling and to damp out high frequency oscillations. This results in

$$\frac{\partial Q}{\partial t} + \frac{\partial E}{\partial x} - \frac{\epsilon_i}{4}\Delta x^2 \frac{\partial^3 Q}{\partial^2 x \partial t} = H - \frac{\epsilon_e \Delta x^4}{8\Delta t}\frac{\partial^4 Q}{\partial x^4}$$

where ϵ_i and ϵ_e are positive constants. The discretized equation, Eq. (2.10), now becomes

$$(I - \Delta t D + \Delta t \frac{\partial}{\partial x} A - \frac{\epsilon_i}{4}\Delta x^2 \frac{\partial^2}{\partial x^2} I)\Delta Q = -\Delta t R - \frac{\epsilon_e}{8}\Delta x^4 (\frac{\partial^4 Q}{\partial x^4})^n \tag{2.18}$$

For linear stability analysis A can be treated as a constant matrix. In the frequency domain Eq. (2.18) becomes

$$L_1 Q^{n+1} = L_2 Q^n \tag{2.19}$$

where L_1 and L_2 are given by

$$L_1 = I[1 + \frac{\epsilon_i}{2}(1 - \cos\omega_x)] + i\frac{\Delta t}{\Delta x} A \sin\omega_x - \Delta t D$$

and

$$L_2 = I[1 + \frac{\epsilon_i}{2}(1 - \cos\omega_x) - \frac{\epsilon_e}{2}(\cos\omega_x - 1)^2]$$

with ω_x representing the wave number defined by $\omega_x = \omega \Delta x$.

Analogous to the definition of the amplification factor for the scalar system, we can define the amplification matrix G by

$$Q^{n+1} = GQ^n.$$

The convergence criteria are then determined by the eigenvalues of the matrix G. A stable algorithm is ensured when the magnitudes of eigenvalues of G are all less than unity. From Eq. (2.19), G can be easily evaluated by $G = L_1^{-1} L_2$. At the high wave number limit ($\omega_x = \pi$), the eigenvalues of G are found to be

$$g_1 = \frac{1 + \epsilon_i - 2\epsilon_e}{1 + \epsilon_i}$$

$$g_2 = \frac{1 + \epsilon_i - 2\epsilon_e}{1 + \epsilon_i + (\gamma - 1)\frac{\sigma \Delta x}{a}\frac{da}{dx}}$$

$$g_3 = \frac{1 + \epsilon_i - 2\epsilon_e}{1 + \epsilon_i}$$

where σ is defined by

$$\sigma = \frac{u \Delta t}{\Delta x}$$

which is referred to as the Courant-Friedriches-Lewy (CFL) number. According to the absolute values of g_1 and g_3, it is required that

$$0 < \epsilon_e < 1 + \epsilon_i \tag{2.20}$$

to maintain numerical stability. The value of g_2 depends on the CFL number, ϵ_i, ϵ_e, and the geometry. The stability criteria associated with it are rather involved. However, several conclusions still can be drawn. First, in a divergent portion of the geometry ($\frac{da}{dx} > 0$), $|g_2|$ is always less than unity, thus the Euler implicit scheme is stable. Second, in a convergent section ($\frac{da}{dx} < 0$), there exists a certain range of CFL such that $|g_2| > 1$ for fixed ϵ_i and ϵ_e. Third, if no fourth order artificial dissipation is included ($\epsilon_e = 0$), $|g_2|$ is always greater than unity in the divergent

section, hence the algorithm is unstable. In practical situations, the geometry contains both convergent and divergent sections; therefore, the central-difference Euler implicit scheme for quasi one-dimensional flows is only conditionally stable.

The eigenvalues of G can be calculated numerically for various wave numbers. Figure 1 shows the plot of the magnitude of the maximum eigenvalue versus ω_x for $CFL = 1, 10, 100$ at a flow Mach number of 0.5. This figure clearly illustrates that increasing the CFL number tends to decrease the magnitude of the maximum eigenvalue, which is beneficial for the speed of convergence. Effects of the artificial dissipation are demonstrated in Fig. 2, where the maximum eigenvalue of G is plotted for $\epsilon_e = 0, 0.25, 0.5, 1.0$ for a fixed CFL of 10. It shows that the addition of fourth order dissipation damps out high frequency components of the wave. It is also observed that $\epsilon_e = 0.5$ is optimal as far as convergence is concerned.

The Fourier analysis discussed above is based on two major assumptions. First, the analysis is only valid for linear cases, in other words, the nonlinear effect of the Jacobian matrix A has been neglected. Second, the analysis assumes an infinite domain and excludes the effect of boundary conditions. Therefore, the results are qualitatively rather than quantitatively accurate.

From the results of Fourier stability analysis, it is apparent that the CFL number plays an important role on the speed of convergence. To obtain optimum convergence, the CFL number should be as large as possible provided that numerical stability is retained. Since the CFL number is directly related to the time step size, we can calculate Δt according to the desired CFL number. If one is interested in accurate solutions during a transient, Δt must be uniform throughout the flowfield. In this case, Δt is better determined according to the maximum value of u, that is,

$$\Delta t = \frac{\sigma \Delta x}{u_{max}}.$$

In general, the velocity u varies from point to point, the using of a uniform time step

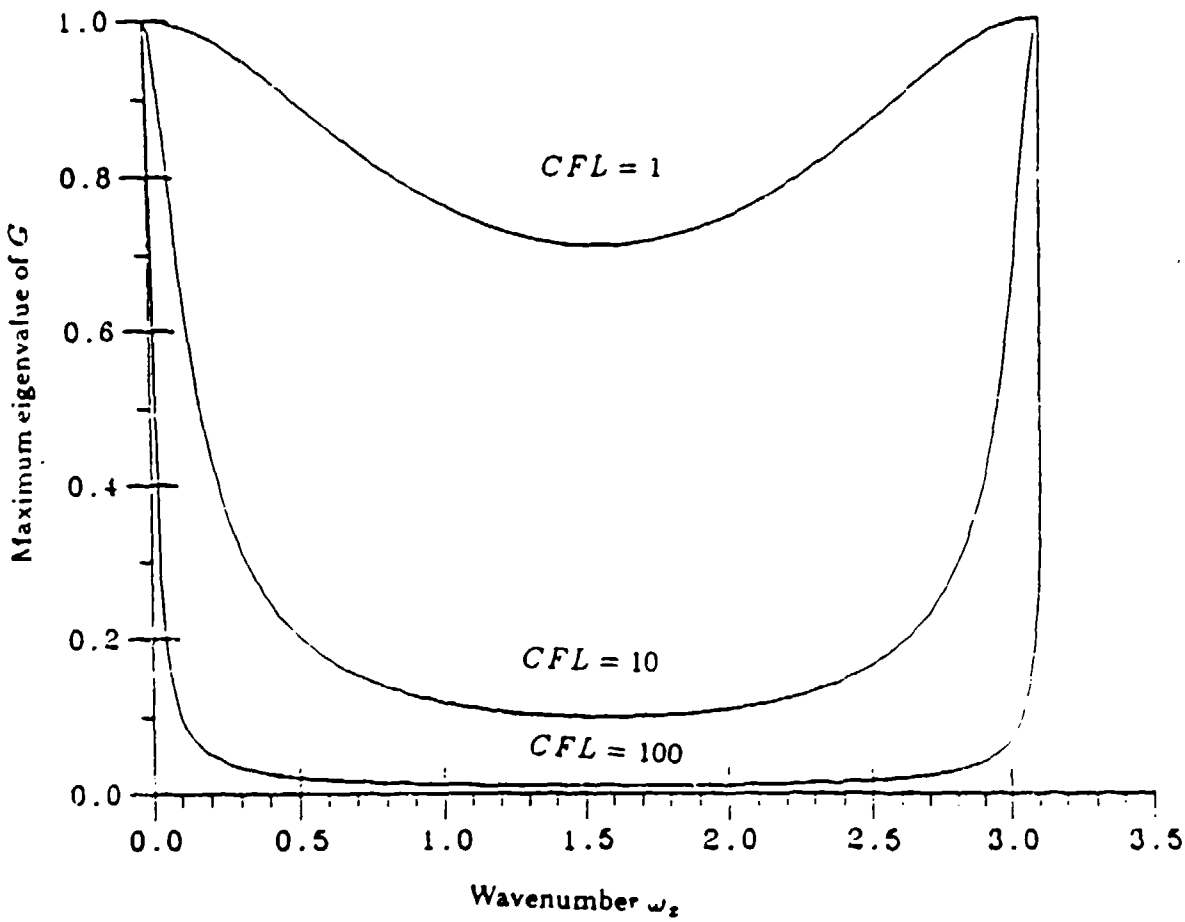


Figure 1. Stability results of 1-D implicit central-difference scheme without artificial viscosity

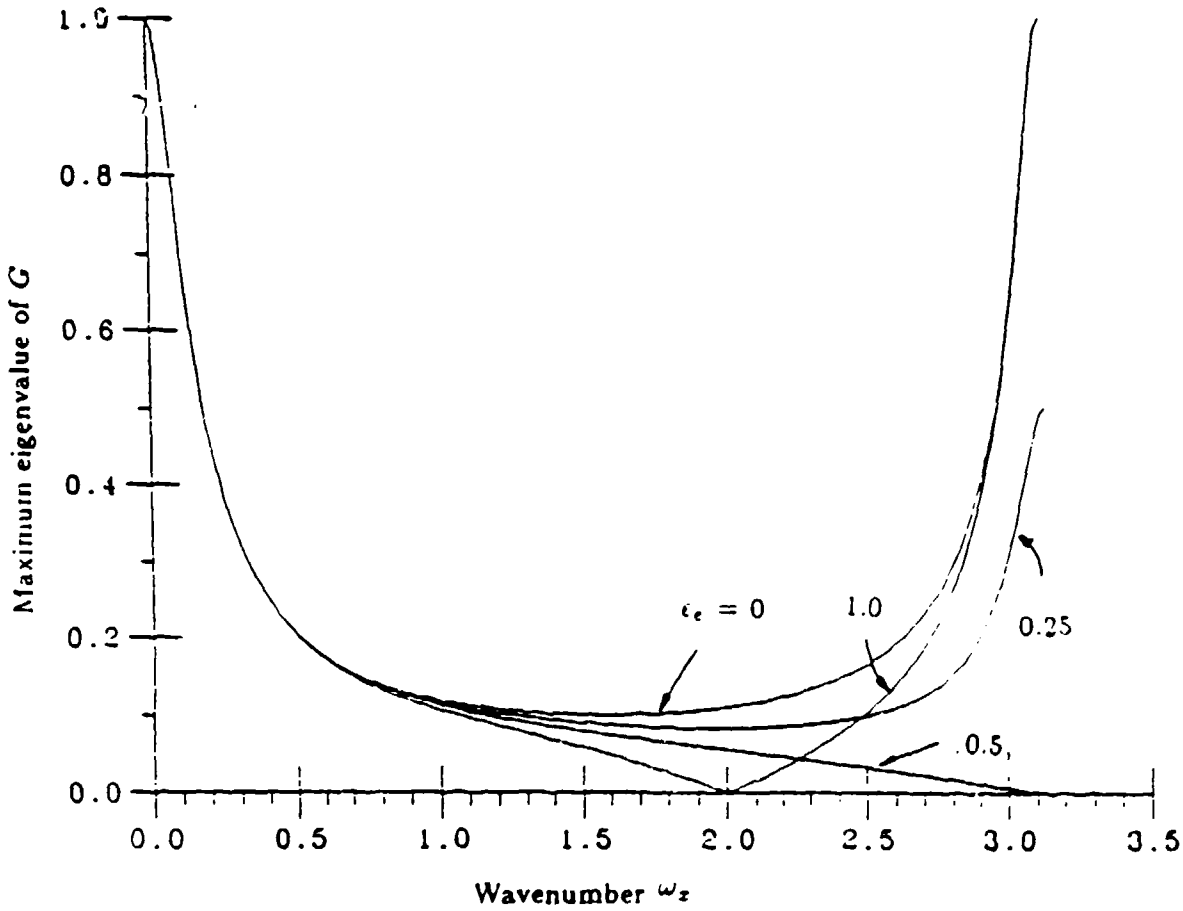


Figure 2. Stability results of 1-D implicit central-difference scheme with artificial viscosity

will result in non-uniform CFL numbers throughout the flowfield. Consequently, the overall convergence is deteriorated, especially when only steady state solutions are of interest. In order to have optimum convergence, Δt can be locally determined by the given CFL number, in other words, the time step size at each grid point is calculated according to

$$\Delta t = \frac{\sigma \Delta x}{u}.$$

This implies a constant CFL number has been enforced over the whole flowfield. The introduction of this spatially varying time step (or so-called constant CFL) greatly enhances the speed of convergence [39]. In this study, the constant CFL approach will in general be used for all calculations since only steady state solutions are concerned.

2.2.3 Computational Results

The one-dimensional flow through a convergent-divergent nozzle with the area variation given by

$$a(x) = \frac{A_{in} - A_{th}}{2} \cos(\frac{2\pi x}{L}) + \frac{A_{in} + A_{th}}{2}$$

is chosen as a test problem. The geometry associated with definitions of x, A_{in}, A_{th}, and L are shown in Fig. 3. A uniform grid with total of 40 points and an area ratio (A_{th}/A_{in}) of 0.8 are used for all calculations that follow.

Three typical cases are investigated, including pure subsonic, transonic, and pure supersonic flows. Figure 4 shows the convergence rates of these cases by plotting the L-2 norm of the non-dimensional change in Q ($\Delta Q/Q$) against the number of iterations. Each curve is obtained by using the optimum CFL number (the CFL number that gives the fastest convergence). The supersonic case converges the fastest among the three cases because of the predominantly hyperbolic nature of the flowfield. For pure subsonic flows, the system of equations is elliptic in

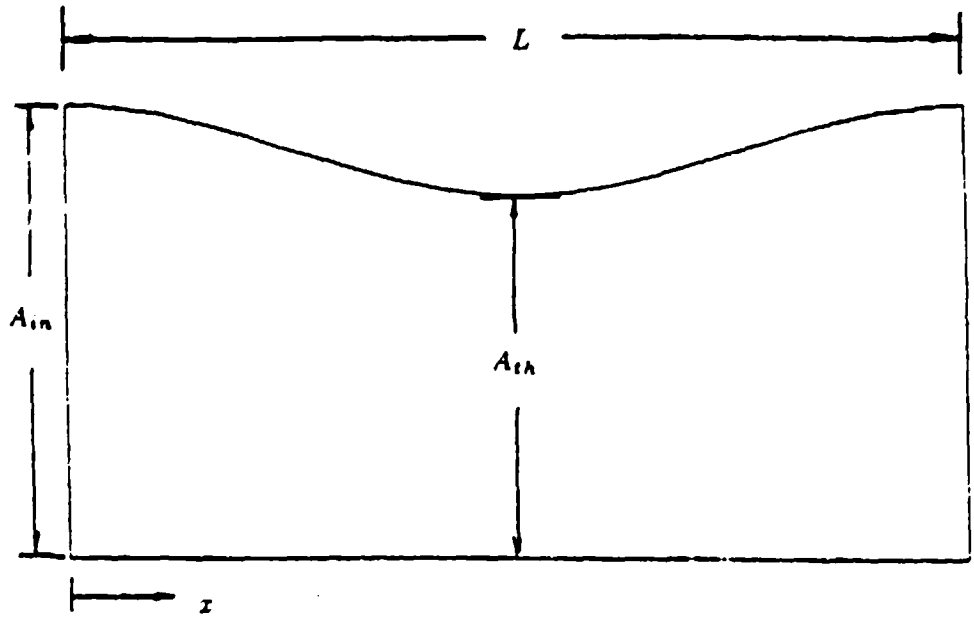


Figure 3. Nozzle geometry for 1-D calculations

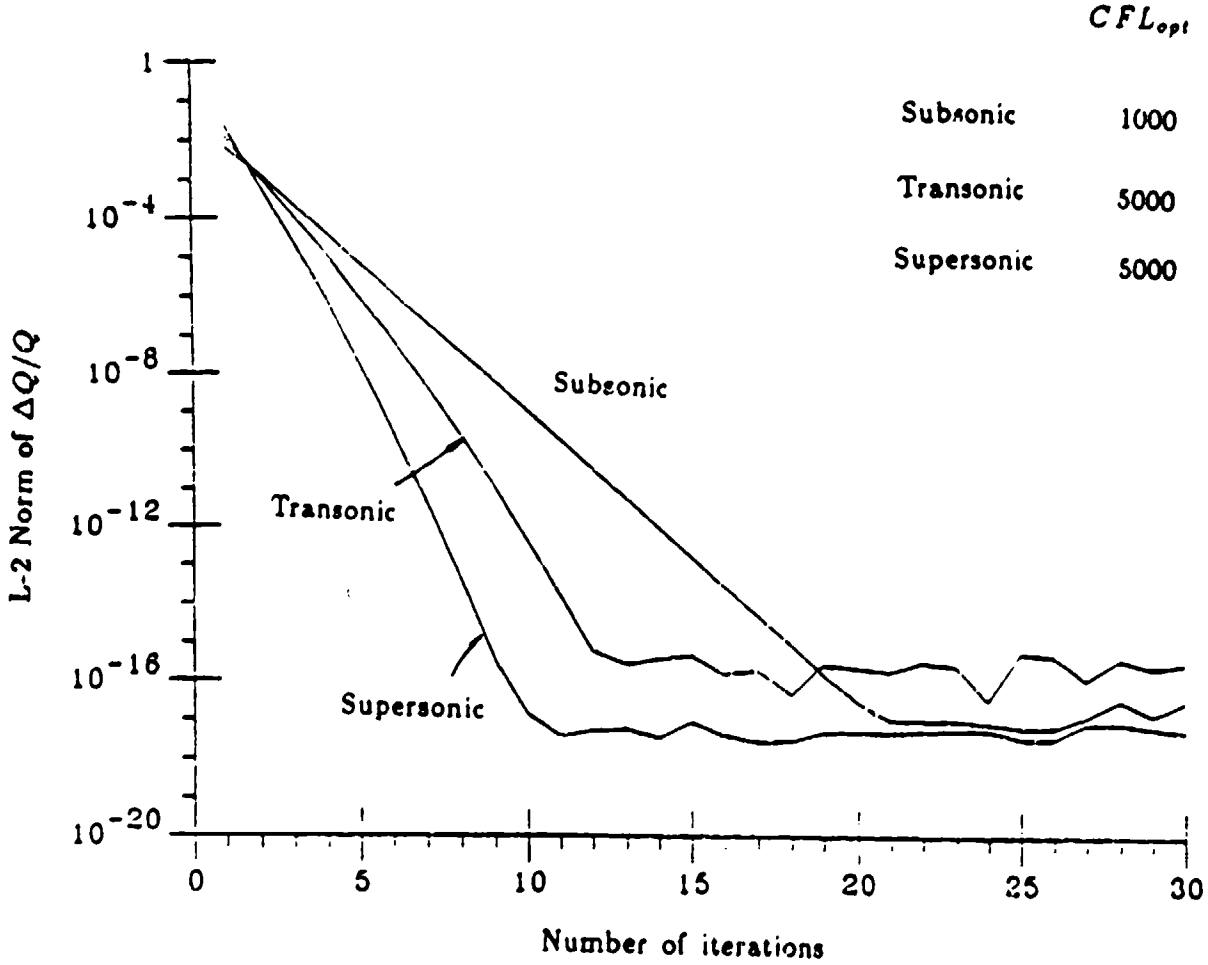


Figure 4. Convergence for 1-D implicit central-difference scheme

the spatial direction since both left-running and right-running characteristics are present in the flowfield. The subsonic case shows the slowest convergence due to this elliptic behavior. The accuracy of the central-difference formulation is demonstrated in Fig. 5 and Fig. 6, where the computed Mach number and pressure distributions along the streamwise direction for the transonic calculation are compared to those from exact solutions for the same nozzle. The comparison shows that computational results agree very well with exact solutions.

2.3 The Upwind-Differencing Algorithm

The main purpose of this study is to develop efficient numerical algorithms for supersonic calculations. As indicated earlier, the predominantly hyperbolic nature of supersonic flows distinguishes themselves from transonic and subsonic flows. To take advantage of this character, upwind schemes appear to be attractive. As we have seen in Section 2.1, the Jacobians of the governing equations generally contain both positive and negative eigenvalues. The sign of each eigenvalue implies the direction of wave propagation on the $x - t$ plane. The crux of flux-vector splitting upwind algorithms [16,22] is to separate the flux vector E into parts with definite (positive and negative) eigenvalues. The splitting can be formally indicated as

$$E = E^+ + E^- \tag{2.21}$$

where the eigenvalues of the Jacobian of E^+ are positive and those of E^- are negative. There are an infinite number of ways to accomplish this splitting. As an example, we have considered the Steger and Warming [16] splitting. From the similarity transformation of A defined by Eq. (2.4), we readily have

$$\Lambda = M^{-1} A M.$$

The Steger and Warming splitting takes the form

$$\Lambda^+ = (\Lambda + |\Lambda|)/2$$

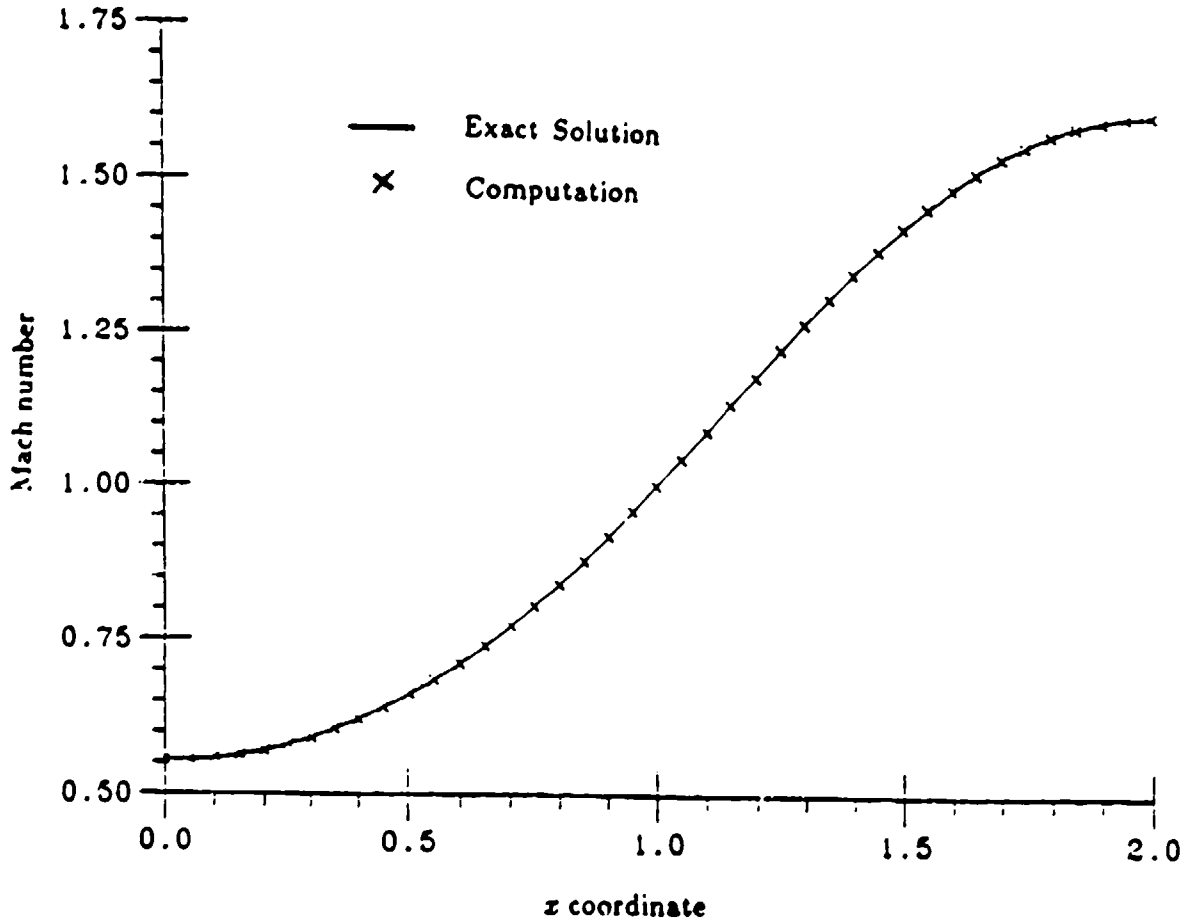


Figure 5. Mach number distribution of transonic 1-D calculation, central-difference solutions

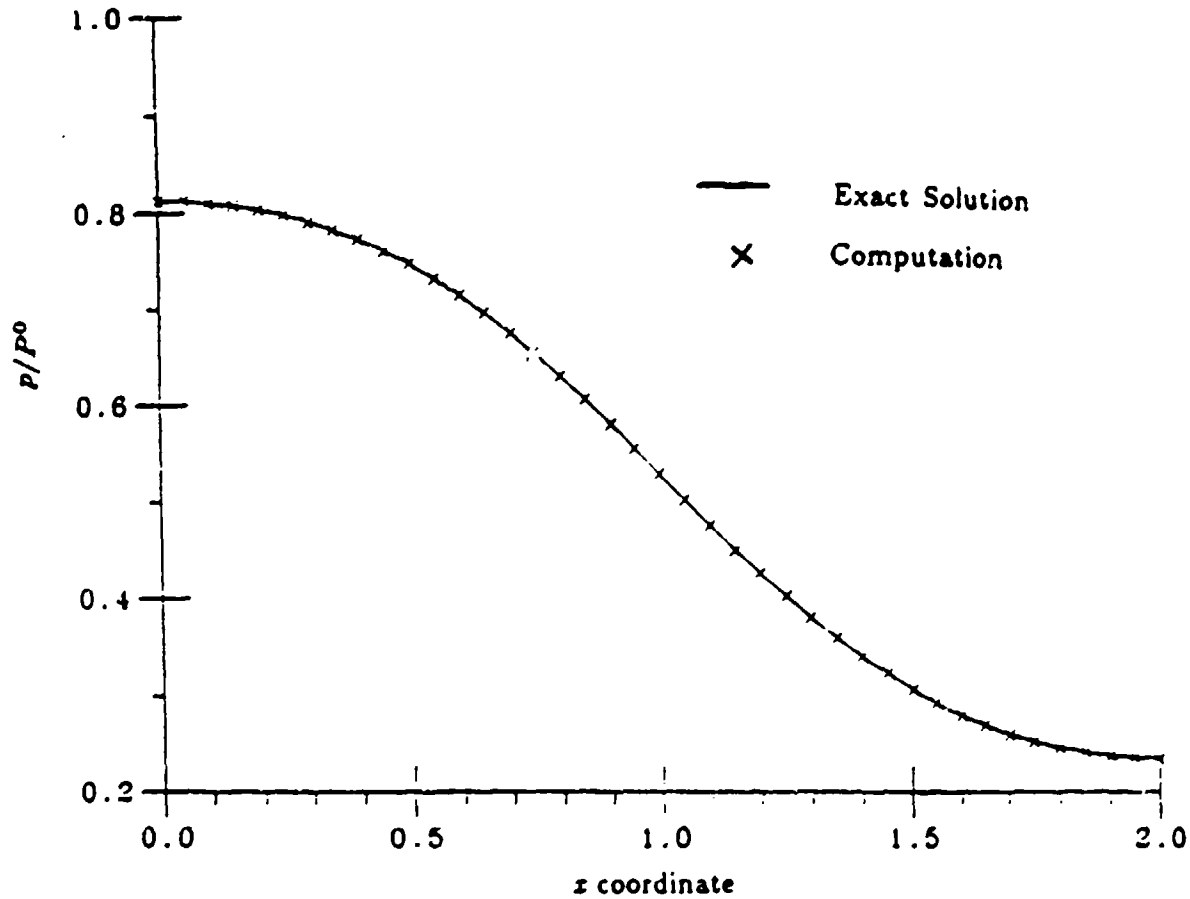


Figure 6. Pressure distribution of transonic 1-D calculation , central-difference solutions

$$\Lambda^- = (\Lambda - |\Lambda|)/2$$

where $|\Lambda|$ refers to the matrix composed of the absolute values of the elements of Λ. The matrix A can then be decomposed into

$$A^+ = M\Lambda^+ M^{-1}$$

$$A^- = M\Lambda^- M^{-1}$$

with $A = A^+ + A^-$. By using the homogeneous property of the matrix A, we have

$$E = (A^+ + A^-)Q = E^+ + E^-$$

in which, $E^+ = A^+Q$ and $E^- = A^-Q$.

Again, using Euler implicit differencing in time, the flux split system can be described by

$$\frac{Q^{n+1} - Q^n}{\Delta t} + \left(\frac{\partial E^+}{\partial x} + \frac{\partial E^-}{\partial x} - H\right)^{n+1} = 0. \tag{2.22}$$

A linearization similar to Eq. (2.8) can be applied to E^+ and E^-. The resulting delta form is

$$[I - \Delta t D + \Delta t(\frac{\partial}{\partial x}A_t^+ + \frac{\partial}{\partial x}A_t^-)]\Delta Q = -\Delta t R' \tag{2.23}$$

where the residual vector R' is

$$R' = \left(\frac{\partial E^+}{\partial x} + \frac{\partial E^-}{\partial x} - H\right)^n. \tag{2.24}$$

The Jacobian matrices A_t^+ and A_t^- are defined by $A_t^+ = \partial E^+/\partial Q$ and $A_t^- = \partial E^-/\partial Q$. Note that $A_t^+ \neq A^+$ and $A_t^- \neq A^-$ when the eigenvalues of A are of mixed sign. For the one-dimensional case, if the flow is supersonic ($u > c$), A_t^+ is exactly the same as A given in Section 2.1 and A_t^- is zero. As an approximation, A^+ and A^- can be used instead of A_t^+ and A_t^- in Eq. (2.23). The effect of using true or approximate Jacobian matrices is detrimental as will be discussed in the next section.

All spatial derivatives in Eq. (2.23) and Eq. (2.24) imply that they will be differentiated according to the signs of their eigenvalues. For example, $\partial E^+/\partial x$ and $\partial E^-/\partial x$ are differentiated according to

$$\frac{\partial E^+}{\partial x} = \frac{E_i^+ - E_{i-1}^+}{\Delta x} + \kappa \frac{E_i^+ - 2E_{i-1}^+ + E_{i-2}^+}{2\Delta x}$$

and

$$\frac{\partial E^-}{\partial x} = \frac{E_{i+1}^- - E_i^-}{\Delta x} - \kappa \frac{E_i^- - 2E_{i+1}^+ E_{i+2}^-}{2\Delta x},$$

where $\kappa = 0$ for the first-order scheme and $\kappa = 1$ for the second-order scheme.

The left-hand side matrix in Eq. (2.23) is block tri-diagonal if all spatial derivatives are discretized by first-order upwind differences. It is well known that first-order upwind differencing adds a large amount of artificial dissipation to the numerical algorithm, and is highly inaccurate. If second-order accurate differencing is employed, the left-hand side matrix becomes block penta-diagonal, which is more time-consuming to solve than the block tridiagonal matrix; however, if only steady-state solutions are of interest, one can use first order differencing on the left-hand side and second-order differencing on the right-hand side. This will retain the block tri-diagonal structure of the left-hand side matrix while maintaining the second-order accuracy of the steady state solution. Jespersen and Pulliam [40] have shown that this non-consistent first and second order differencing will reduce the stability bound of the CFL number and slow down the convergence. To make consistent second-order differencing possible, one can approximately factorize the left-hand side of Eq. (2.23) as,

$$(I - \Delta t D + \Delta t \frac{\partial}{\partial x} A_i^+)(I - \Delta t D)^{-1}(I - \Delta t D + \Delta t \frac{\partial}{\partial x} A_i^-)\Delta Q = -\Delta t R' \quad (2.25)$$

Equation (2.25) is equivalent to

$$(I - \Delta t D + \Delta t \frac{\partial}{\partial x} A_i^+)\Delta Q^* = -\Delta t R' \quad (2.26)$$

and

$$(I - \Delta t D + \Delta t \frac{\partial}{\partial x} A_i^-)\Delta Q = (I - \Delta t D)\Delta Q^* . \qquad (2.27)$$

Equation (2.26) can now be solved by space-marching from upstream to downstream since the left hand side matrix is lower bi-diagonal for first-order spatial differencing and is lower tri-diagonal for second-order spatial differencing . This forward marching together with the backward marching given in Eq. (2.27) will complete one iteration.

Two comments can be made at this point. First, as indicated by Steger and Warming [16], small oscillations occur by using this splitting when a sonic line is crossed because of the discontinuity in the first derivative of the split flux when the eigenvalues change sign. This oscillation can be partly removed by the introduction of a blending coefficient in the calculation of eigenvalues [41]. Second, if the flow is supersonic, E^- and A_i^- are identically zero. Equation (2.23) reduces to a form analogous to the forward-sweep step given in Eq. (2.26), namely

$$(I - \Delta t D + \Delta t \frac{\partial}{\partial x} A_i^+)\Delta Q = -\Delta t (\frac{\partial E^+}{\partial x} - H)^n . \qquad (2.28)$$

As in Eq. (2.26), the left hand side matrix of Eq. (2.28) is lower triangular and only the forward marching step of Eq. (2.26) is necessary to complete one iteration. This allows a pure marching solution to be obtained. In fact, we can rearrange Eq. (2.28) to

$$[I - \Delta t D + (1 + \frac{\kappa}{2})\frac{\Delta t}{\Delta x} A_i^+]\Delta Q_i = -\Delta t (\frac{E_i^+ - E_{i-1}^+}{\Delta x} + \kappa \frac{E_i^+ - 2E_{i-1}^+ + E_{i-2}^+}{2\Delta x} - H)^n \qquad (2.29)$$

where $\kappa = 0$ for first-order scheme and $\kappa = 1$ for second-order scheme. This equation is a marching equation, which can be solved by iterating in time at each grid point before advancing to the next streamwise location. Thus, the vector ΔQ_i on the left hand side of Eq. (2.29) can be driven to the desired tolerance by time-marching at the i-th grid point before the procedure marches to the i+1-th point.

This marching procedure will be extended for two-dimensional calculations in the following chapters and will not be discussed in detail here.

2.3.1 Boundary Conditions

The boundary procedures for upwind schemes are similar to those for central-difference schemes. At boundaries, the characteristics coming from outside the domain are not defined and are replaced by specified boundary conditions. By neglecting these incoming characteristics, the discretized equations reduce to

$$(I - \Delta t D + \Delta t \frac{\partial}{\partial x} A^-)\Delta Q = -\Delta t(\frac{\partial E^-}{\partial x} - H)^n \tag{2.30}$$

at the upstream inlet, and

$$(I - \Delta t D + \Delta t \frac{\partial}{\partial x} A^+)\Delta Q = -\Delta t(\frac{\partial E^+}{\partial x} - H)^n \tag{2.31}$$

at the downstream end. In these two equations, A^+ and A^- have been chosen instead of A_t^+ and A_t^- to enable the application of the MOC procedure. By using the identity matrix $I = MM^{-1}$ and the definition of A^+, Eq. (2.30) can be approximated by

$$M(I - \Delta t D + \Delta t \frac{\partial}{\partial x} \Lambda^-)M^{-1}\Delta Q = -\Delta t(\frac{\partial E^-}{\partial x} - H)^n .$$

Multiplying both sides by M^{-1}, we have

$$(I - \Delta t D + \Delta t \frac{\partial}{\partial x} \Lambda^-)\Delta \hat{Q} = -\Delta t M^{-1}(\frac{\partial E^-}{\partial x} - H)^n$$

which is equivalent to the previously defined decoupled characteristic equation. Again, multiplying this characteristic equation by a selection matrix L^+ and combining it with the specified boundary conditions discussed in Section 2.2.1 gives

$$[\frac{\partial \Omega}{\partial Q} + L^+ M^{-1}(I - \Delta t D + \Delta t \frac{\partial}{\partial x} A^-)]\Delta Q = \Omega_* - \Omega^n - \Delta t L^+ M^{-1}(\frac{\partial E^-}{\partial x} - H)^n \tag{2.32}$$

where the vectors Ω and Ω_i follow the same definitions given in Section 2.2.1. This boundary procedure is similar to Eq. (2.15) except different decoupled discretized equations have been used in Eq. (2.32). The discussion above illustrates that the boundary procedures formulated for the central-differencing algorithm given in Section 2.2.1 are also applicable to the upwind-differencing scheme.

2.3.2 Stability Analysis

The stability analysis is now studied for the upwind algorithm. For demonstration, only first-order differencing on both sides of Eq. (2.23) is considered. Following the definitions given in Section 2.2.2, the amplification matrix G can be expressed by $G = L_1^{-1} L_2$ with

$$L_1 = I - \Delta t D + \frac{\Delta t}{\Delta x}(1 - \cos\omega_x + i\sin\omega_x)A_t^+ + \frac{\Delta t}{\Delta x}(\cos\omega_x + i\sin\omega_x - 1)A_t^-$$

$$L_2 = I$$

where true Jacobian matrices A_t^+ and A_t^- have been used. Alternatively, if the approximate Jacobian matrices A^+ and A^- are used on the left hand side of Eq. (2.23), L_1 and L_2 become

$$\begin{aligned} L_1 &= I - \Delta t D + \frac{\Delta t}{\Delta x}(1 - \cos\omega_x + i\sin\omega_x)A^+ \\ &\quad + \frac{\Delta t}{\Delta x}(\cos\omega_x + i\sin\omega_x - 1)A^- \\ L_2 &= I + \frac{\Delta t}{\Delta x}(1 - \cos\omega_x + i\sin\omega_x)(A^+ - A_t^+) \\ &\quad + \frac{\Delta t}{\Delta x}(\cos\omega_x + i\sin\omega_x - 1)(A^- - A_t^-). \end{aligned}$$

As noted earlier, A_t^+ and A_t^- differ from A^+ and A^- when the flow is subsonic. Typical stability results for the approximate Jacobian case are shown in Fig. 7 for a flow Mach number of 0.5. An explicit-like CFL restriction ($\sigma < 1$) is observed. On the other hand, if true Jacobians are used on the left hand side of Eq. (2.23), the upwind algorithm is unconditionally stable, as shown in Fig. 8. For supersonic

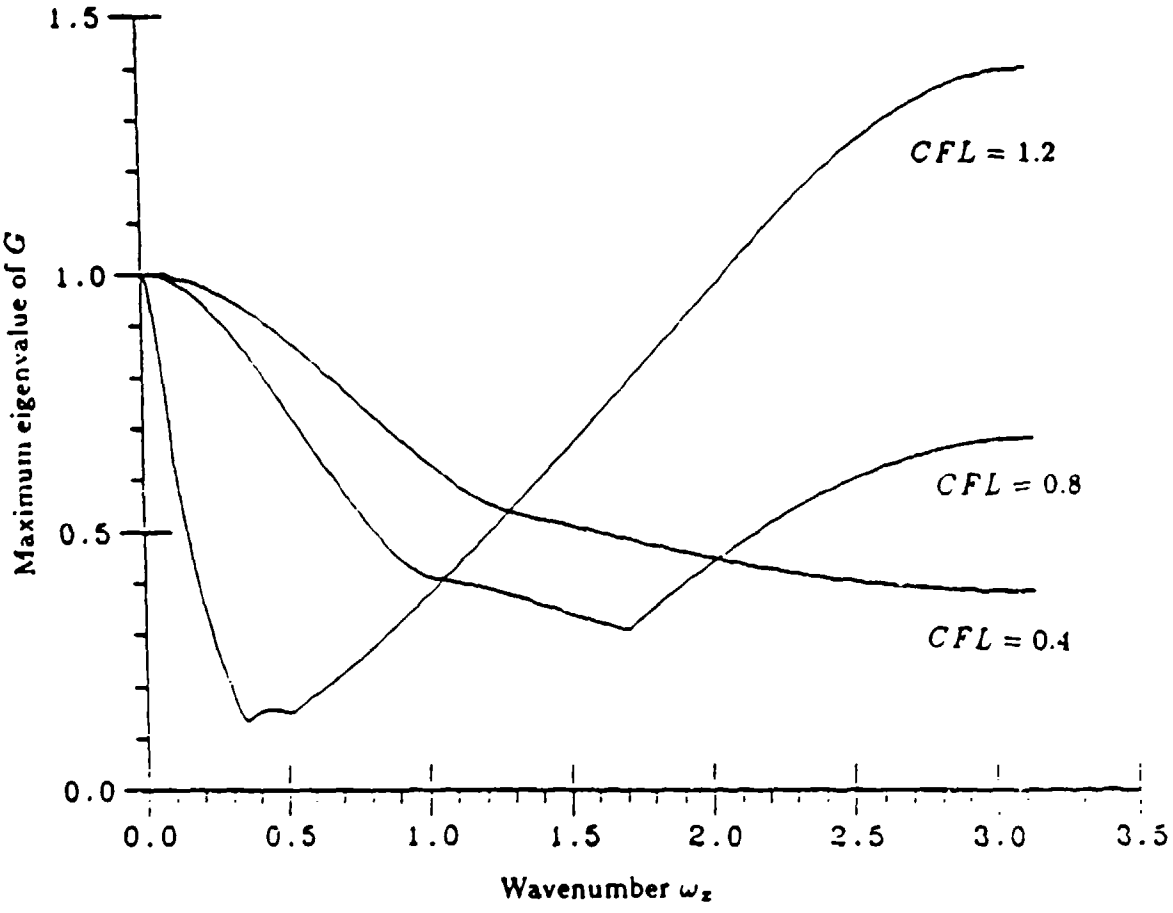


Figure 7. Stability results of 1-D implicit first-order upwind scheme, approximate Jacobian, $M = 0.5$

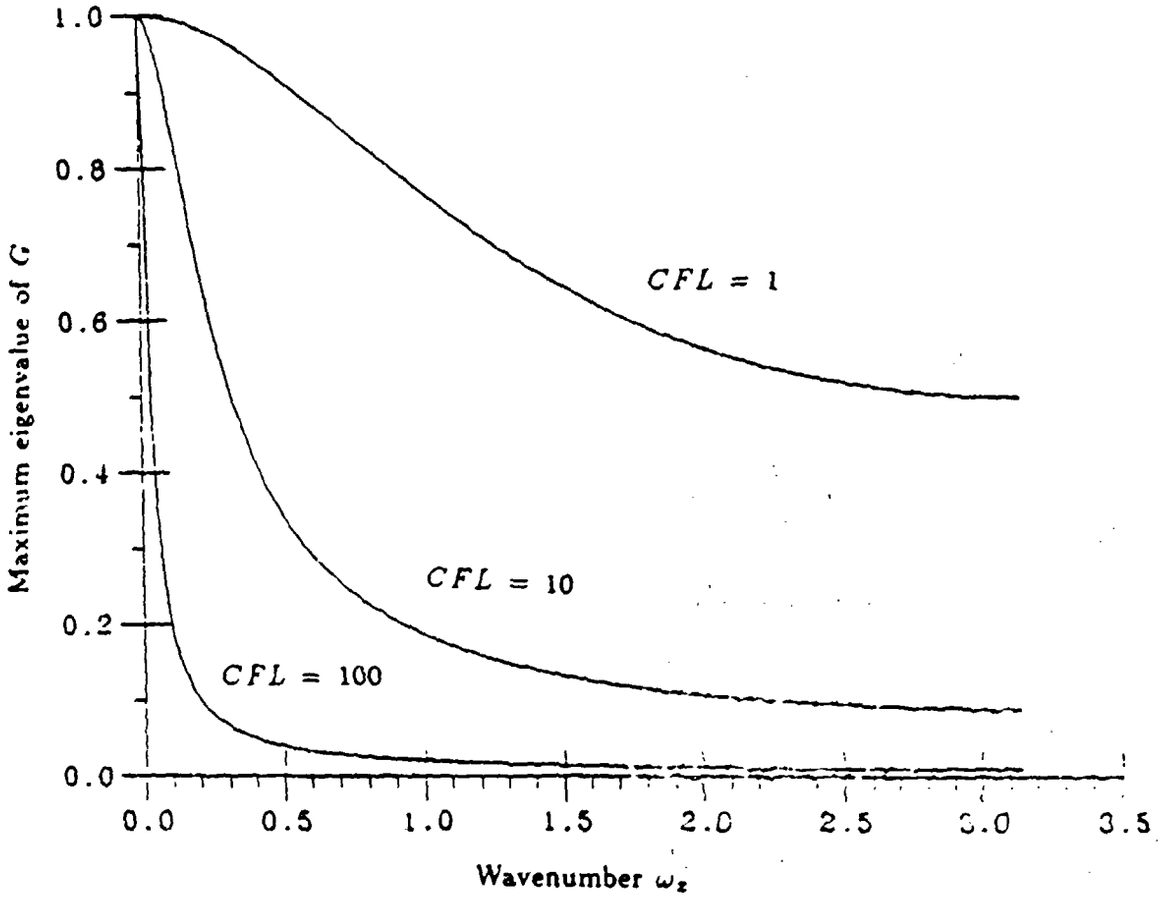


Figure 8. Stability results of 1-D implicit first-order upwind scheme, true Jacobian, $M = 0.5$

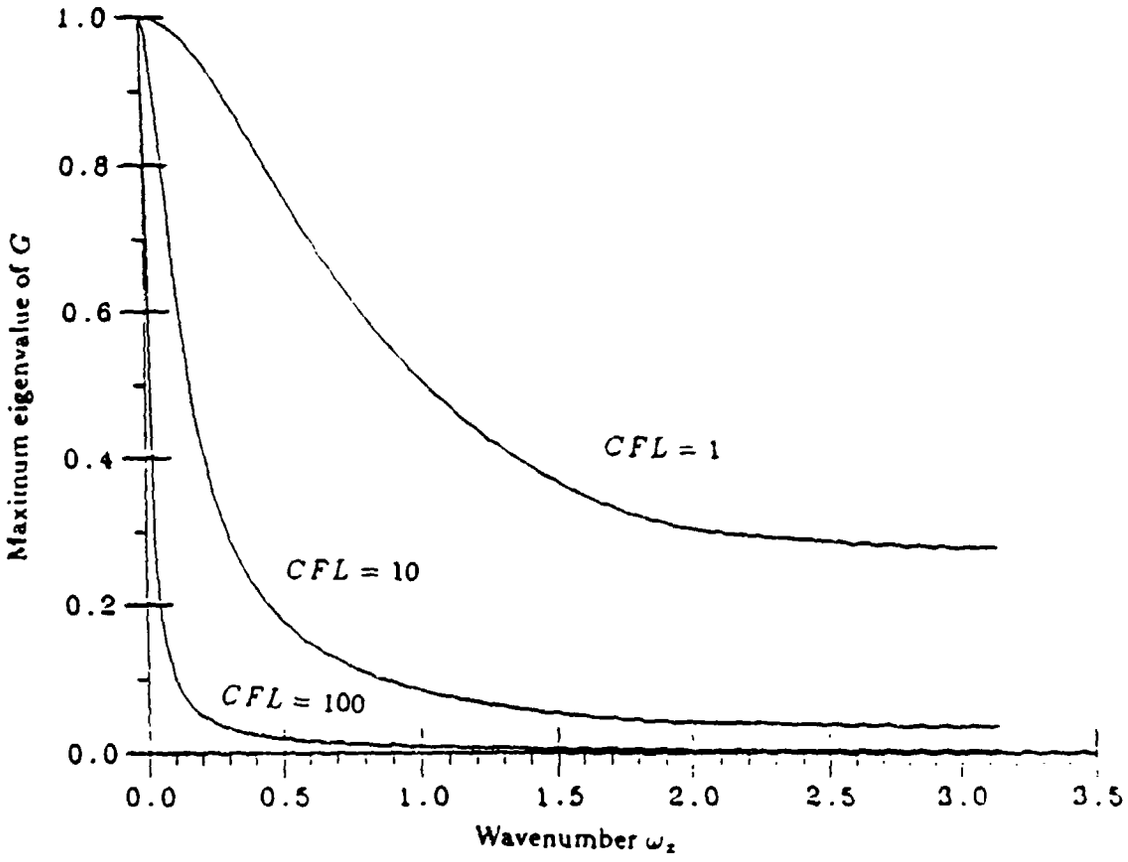


Figure 9. Stability results of 1-D implicit first-order upwind scheme, supersonic flow, $M = 2.0$

flows, $A_t^+ = A$ and A_t^- is identically zero. Corresponding stability results are shown in Fig. 9 for a flow Mach number of 2.0. The upwind scheme is again stable for all CFL numbers. As can be seen in Fig. 8 and Fig. 9, for a given CFL number, the maximum eigenvalue of G reaches its minimum value at the high wave number limit ($\omega_x = \pi$). This characteristic implies that the upwind scheme is naturally dissipative and no artificial viscosity is necessary to maintain numerical stability.

2.3.3 Computational Results

The same test problem given in Section 2.2.3 is calculated by using Eq. (2.23). Again, three typical cases are studied, they are pure subsonic, transonic, and pure supersonic flows. Figure 10 compares convergence rates obtained by using first-order differencing on both sides of Eq. (2.23) for all cases. It shows that the upwind algorithm is equally efficient as the central-difference algorithm (compare Fig. 4), except for the transonic case, for which the discontinuity across the sonic point substantially slows down the convergence. The very slow convergence of the subsonic case based on the approximate Jacobians where the optimum CFL number is found to be 0.9 is also shown on Fig. 10. In fact, with the use of the approximate Jacobian, the computer code diverges for both transonic and subsonic calculations if $\sigma \geq 1$, thus confirming the stability predictions given in the last section.

Computational results for the supersonic calculation are compared to exact solutions in Fig. 11 and Fig. 12 for both first order and second order accurate computations. The second-order scheme gives solutions that are much more accurate than the first-order scheme does. Therefore, second-order differencing should always be used to ensure accurate solutions.

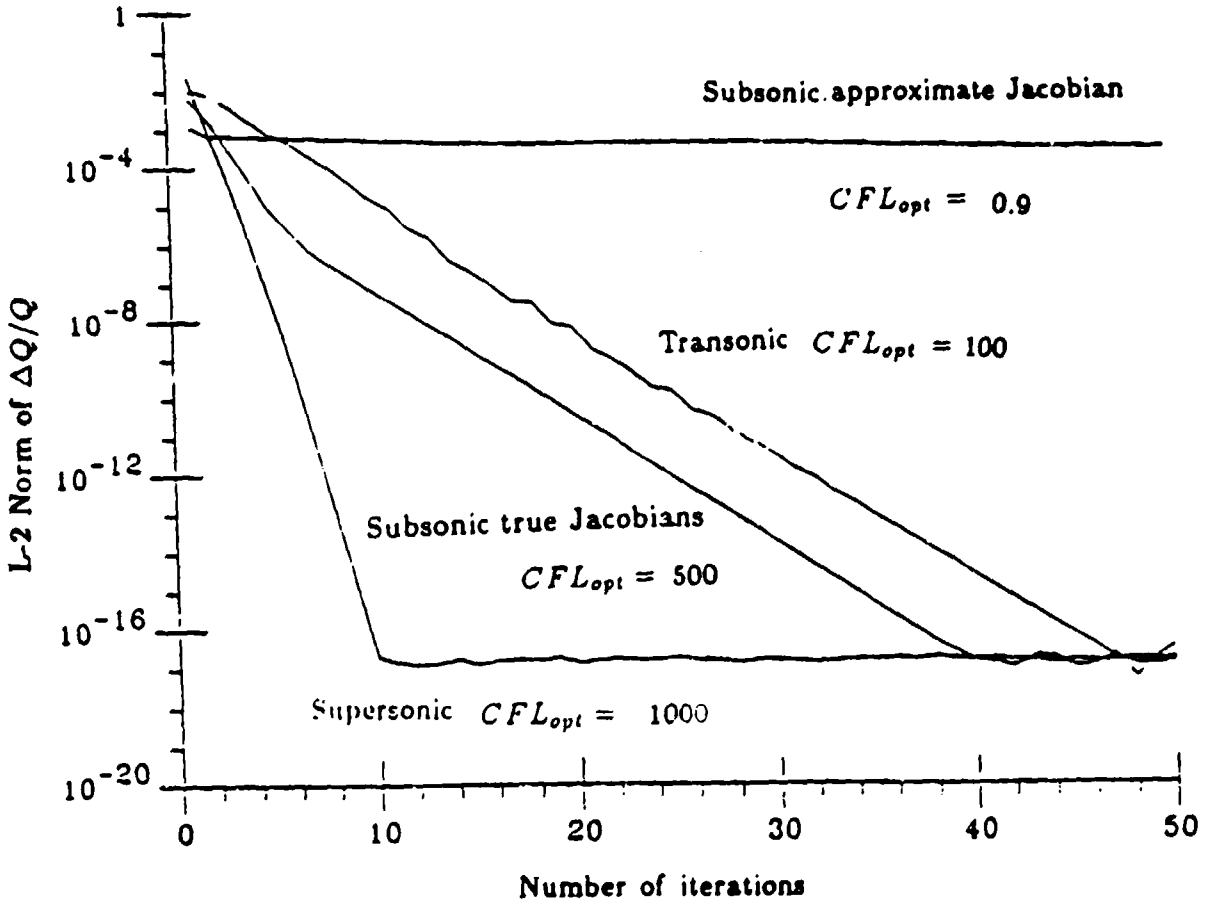


Figure 10. Convergence for 1-D implicit first-order upwind scheme

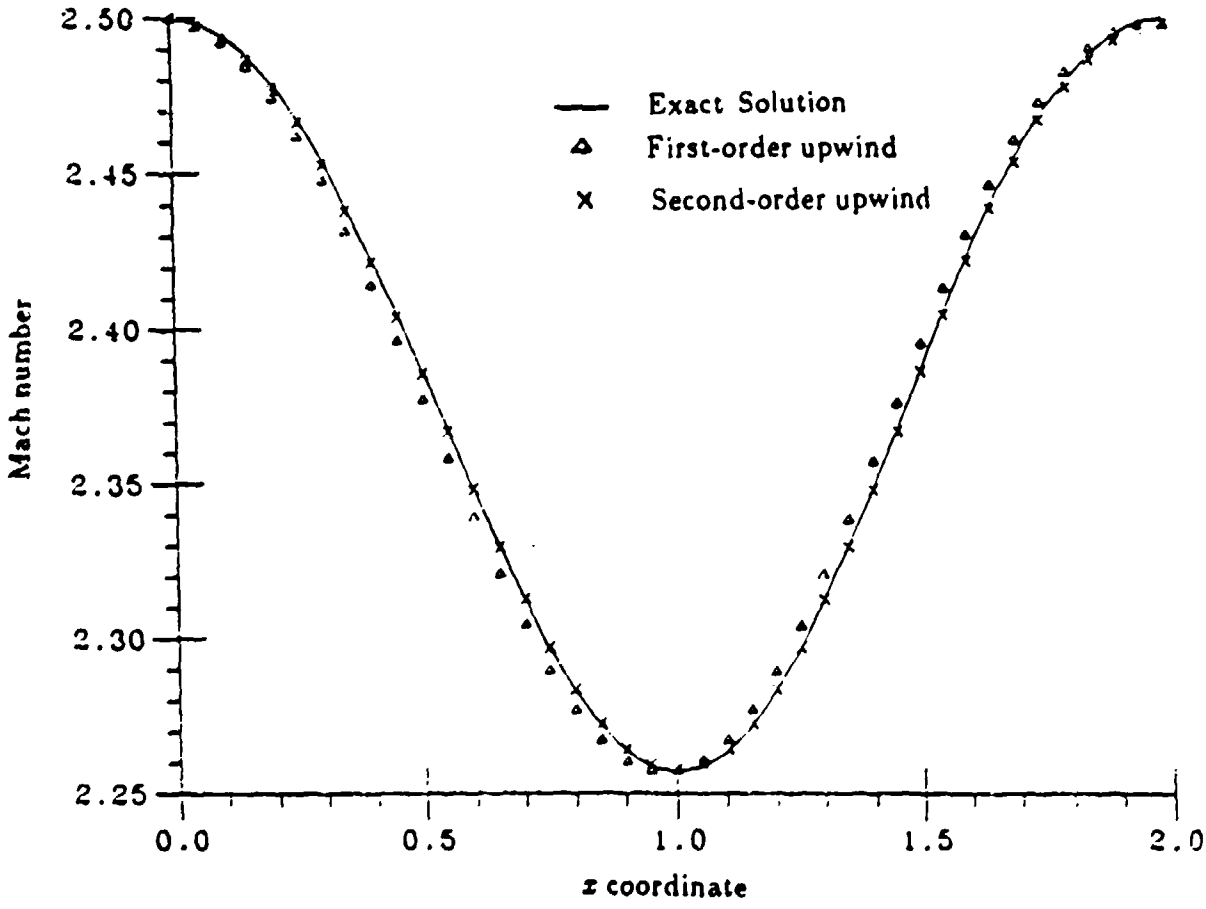


Figure 11. Mach number distribution of supersonic 1-D calculation, upwind solutions

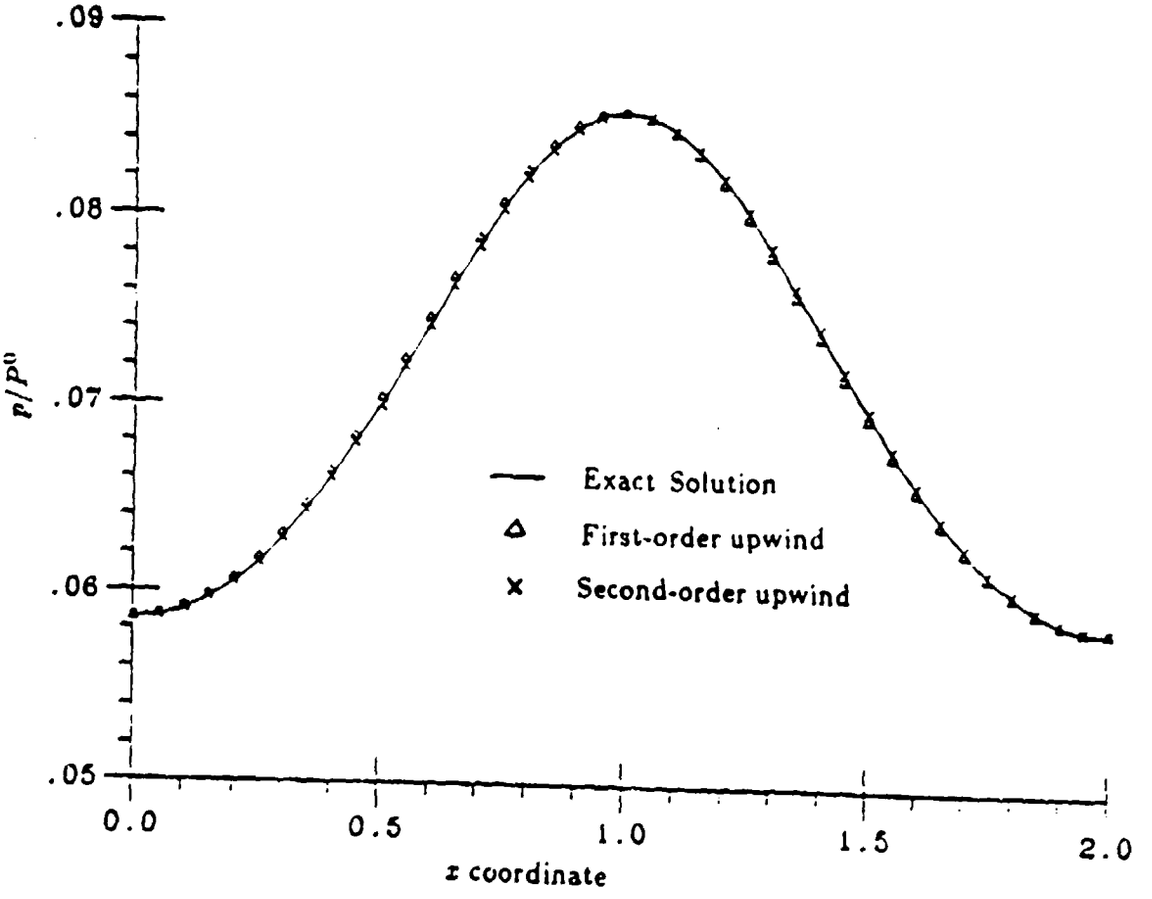


Figure 12. Pressure distribution of supersonic 1-D calculation , upwind solutions

CHAPTER 3

THE APPLICATION OF TIME-ITERATIVE SCHEMES TO THE TWO-DIMENSIONAL NAVIER-STOKES EQUATIONS

Based on the information gained from the application of time-marching schemes to the one-dimensional Euler equations, this chapter proceeds with numerical solutions of the axisymmetric two-dimensional Navier-Stokes (N-S) equations. The conventional implicit ADI procedure is first applied to transonic and supersonic viscous calculations. The appropriateness of this procedure when applied to predominantly supersonic flows will be identified. According to the physics of viscous supersonic flows, a discretized scheme using upwind flux-vector splitting in the streamwise direction and central-differencing in the cross-stream direction together with solution procedures are proposed. The Fourier stability analysis will be used to analyze the stability criteria of this new discretized scheme. Of the solution procedures, approximate algorithms as well as a direct solver are considered. These procedures will be used to calculate viscous supersonic flows through nozzles. In particular, attention will be paid to proper downstream boundary conditions for the subsonic portion of the outflow and global mass conservation.

3.1 Governing Equations

For practical applications, the two-dimensional Navier-Stokes equations are formulated in a cylindrical coordinate system. The equations for planar two-dimensional flows can be easily obtained by simplifying the cylindrical version.

Let x and y denote the axial and radial coo[illegible] The unsteady axisymmetric Navier-Stokes equations for laminar cor[illegible] are given as

$$\frac{\partial}{\partial t}(\rho y) + \frac{\partial}{\partial x}(\rho u y) + \frac{\partial}{\partial y}(\rho v y) = 0$$

$$\frac{\partial}{\partial t}(\rho u y) + \frac{\partial}{\partial x}|(\rho u^2 + p)y| + \frac{\partial}{\partial y}(\rho u v y) = \frac{\partial}{\partial x}|(2\mu\frac{\partial u}{\partial x} - \frac{2}{3}\mu\nabla\cdot\vec{V})y| + \frac{\partial}{\partial y}|\mu(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y})y|$$

$$\frac{\partial}{\partial t}(\rho v y) + \frac{\partial}{\partial x}(\rho u v y) + \frac{\partial}{\partial y}|(\rho v^2 + p)y| = p + \frac{2}{3}\mu\nabla\cdot\vec{V} - 2\mu\frac{v}{y} + \frac{\partial}{\partial x}|\mu(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y})y| + \frac{\partial}{\partial y}|\mu(2\frac{\partial v}{\partial y} - \frac{2}{3}\nabla\cdot\vec{V})y| \tag{3.1}$$

$$\frac{\partial}{\partial t}(ey) + \frac{\partial}{\partial x}|(e+p)uy| + \frac{\partial}{\partial y}|(e+p)vy| = \frac{\partial}{\partial x}\{|\mu u(2\frac{\partial u}{\partial x} - \frac{2}{3}\nabla\cdot\vec{V}) + \mu v(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}) + k\frac{\partial T}{\partial x}|y\} + \frac{\partial}{\partial y}\{|\mu u(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}) + \mu v(2\frac{\partial v}{\partial y} - \frac{2}{3}\nabla\cdot\vec{V}) + k\frac{\partial T}{\partial y}|y\}$$

Again, standard fluid dynamic notations are used, including the axial velocity u, the radial velocity v, the temperature T, the pressure p, the thermal conductivity k, the molecular viscosity μ, and the total velocity vector $\vec{V}$. The divergence in cylindrical coordinate is defined by

$$\nabla\cdot\vec{V} = \frac{1}{y}\frac{\partial}{\partial y}(yv) + \frac{\partial u}{\partial x}.$$

The total energy e in two dimensions is

$$e = \rho\varepsilon + \frac{1}{2}\rho(u^2 + v^2).$$

To close the problem, the perfect gas relation $p = \rho R_g T$ is also required.

When written in vector notation, Eq. (3.1) becomes

$$\frac{\partial Q}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} = H + \frac{\partial E_v}{\partial x} + \frac{\partial F_v}{\partial y}, \tag{3.2}$$

in which dependent variables are included in the vector Q defined by

$$Q = y(\rho, \rho u, \rho v, e)^T, \tag{3.3}$$

convection terms are expressed by the inviscid flux vectors E and F defined by,

$$E = y \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (e+p)u \end{bmatrix}, \qquad F = y \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (e+p)v \end{bmatrix}. \tag{3.4}$$

and the vectors E_v and F_v contain the second order viscous diffusion terms,

$$E_v = y \begin{bmatrix} 0 \\ \mu(\frac{4}{3}\frac{\partial u}{\partial x} - \frac{2}{3}\frac{\partial v}{\partial y}) \\ \mu(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}) \\ \mu v(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}) + \mu u(\frac{4}{3}\frac{\partial u}{\partial x} - \frac{2}{3}\frac{\partial v}{\partial y}) + k\frac{\partial T}{\partial x} \end{bmatrix} \tag{3.5}$$

$$F_v = y \begin{bmatrix} 0 \\ \mu(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}) \\ \mu(\frac{4}{3}\frac{\partial v}{\partial y} - \frac{2}{3}\frac{\partial u}{\partial x}) \\ \mu u(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}) + \mu v(\frac{4}{3}\frac{\partial v}{\partial y} - \frac{2}{3}\frac{\partial u}{\partial x}) + k\frac{\partial T}{\partial y} \end{bmatrix} \tag{3.6}$$

The source vector H includes all source terms associated with axisymmetric geometry and all remaining viscous terms,

$$H = \begin{bmatrix} 0 \\ -\frac{2}{3}\frac{\partial}{\partial x}(\mu v) \\ p - \frac{4}{3}\mu\frac{v}{y} + \frac{2}{3}\mu\frac{\partial u}{\partial x} - \frac{2}{3}v\frac{\partial \mu}{\partial y} \\ -\frac{2}{3}\frac{\partial}{\partial x}(\mu uv) - \frac{2}{3}\frac{\partial}{\partial y}(\mu v^2) \end{bmatrix}. \tag{3.7}$$

In this form, the corresponding equations in planar two dimensions can be easily obtained by dropping H and setting the y's in Eq. (3.3) through Eq. (3.6) to unity.

To facilitate numerical computation over arbitrary geometries, Eq. (3.2) is transformed to a general coordinate system by a transformation defined by

$$\xi = \xi(x, y)$$

$$\eta = \eta(x, y)$$

where, ξ and η are usually aligned with the streamwise and the cross-stream direction, respectively.

The transformed equation takes the form,

$$\frac{\partial \tilde{Q}}{\partial t} + \frac{\partial \tilde{E}}{\partial x} + \frac{\partial \tilde{F}}{\partial y} = \tilde{H} + \frac{\partial \tilde{E}_v}{\partial x} + \frac{\partial \tilde{F}_v}{\partial y} \tag{3.8}$$

in which the strong conservative form is achieved by placing all the metric terms (ξ_x,ξ_y,etc.) inside the derivatives and cancelling the arising terms by the metric identity. The new dependent variable $\tilde{Q}$ is

$$\tilde{Q} = \frac{y}{J}(\rho, \rho u, \rho v, e)^T$$

where, J is the Jacobian of the coordinate transformation, which is defined by,

$$J = \xi_x \eta_y - \xi_y \eta_x.$$

Inviscid flux vectors in the general coordinate system become

$$\tilde{E} = \frac{y}{J}\begin{bmatrix} \rho U \\ \rho u U + \xi_x p \\ \rho v U + \xi_y p \\ (e+p)U \end{bmatrix}, \qquad \tilde{F} = \frac{y}{J}\begin{bmatrix} \rho V \\ \rho u V + \eta_x p \\ \rho v V + \eta_y p \\ (e+p)V \end{bmatrix}$$

in which, U and V represent contravariant velocities in the general coordinate system,

$$U = \xi_x u + \xi_y v$$

$$V = \eta_x u + \eta_y v.$$

Viscous flux vectors can be expressed as

$$\tilde{E}_v = \frac{\xi_x E_v + \xi_y F_v}{J}$$

$$\tilde{F}_v = \frac{\eta_x E_v + \eta_y F_v}{J}$$

while the source vector is simply $\tilde{H} = H/J$.

For typical high Reynolds number flows, a highly stretched grid is required in the normal direction near the wall in order to resolve the large normal gradient inside

the boundary layer. As a consequence, the grid size in the streamwise direction generally cannot be refined enough to resolve corresponding streamwise gradients due to the limitation on computer storage even if streamwise diffusion terms are retained in the complete Navier-Stokes equations. Therefore, for high and even fairly low Reynolds number flows, streamwise diffusion terms can manytimes be neglected without losing accuracy. The resulting equation set is referred to as the thin-layer Navier-Stokes (TLNS) equations [11],

$$\frac{\partial \tilde{Q}}{\partial t} + \frac{\partial \tilde{E}}{\partial \xi} + \frac{\partial \tilde{F}}{\partial \eta} = \tilde{H} + \frac{\partial \tilde{F}_v}{\partial \eta}. \tag{3.9}$$

The TLNS equations will be used as the governing equations for viscous calculations in this study.

After the thin-layer approximation, the viscous flux vector $\tilde{F}_v$ and the source vector $\tilde{H}$ become

$$\tilde{F}_v = \frac{y}{J}\begin{bmatrix} 0 \\ \alpha_1 \frac{\partial u}{\partial \eta} + \alpha_2 \frac{\partial v}{\partial \eta} \\ \alpha_2 \frac{\partial u}{\partial \eta} + \alpha_3 \frac{\partial v}{\partial \eta} \\ \alpha_4 \frac{\partial}{\partial \eta}\frac{\varepsilon}{\rho} + \frac{\alpha_1 - \alpha_4}{2}\frac{\partial v^2}{\partial \eta} + \frac{\alpha_3 - \alpha_4}{2}\frac{\partial}{\partial \eta} + \alpha_2 \frac{\partial uv}{\partial \eta} \end{bmatrix}$$

and

$$\tilde{H} = \frac{1}{J}\begin{bmatrix} 0 \\ -\frac{2}{3}\eta_x \frac{\partial}{\partial \eta}(\mu v) \\ p - \frac{4}{3}\mu \frac{v}{y} + \frac{2}{3}\mu \eta_x \frac{\partial u}{\partial \eta} - \frac{2}{3} v \eta_y \frac{\partial \mu}{\partial \eta} \\ -\frac{2}{3}\eta_x \frac{\partial}{\partial \eta}(\mu u v) - \frac{2}{3}\eta_y \frac{\partial}{\partial \eta}(\mu v^2) \end{bmatrix}$$

where,

$$\alpha_1 = \mu\left(\frac{4}{3}\eta_x^2 + \eta_y^2\right)$$

$$\alpha_2 = \frac{\mu}{3}\eta_x \eta_y$$

$$\alpha_3 = \mu\left(\eta_x^2 + \frac{4}{3}\eta_y^2\right)$$

$$\alpha_4 = \frac{\gamma k}{C_p}\left(\eta_x^2 + \eta_y^2\right)$$

and C_p is the constant pressure specific heat. The viscous term $\partial \tilde{F}_v / \partial \eta$ can be further rearranged as

$$\frac{\partial F_v}{\partial \eta} = \frac{\partial}{\partial \eta}\left(R_1 \frac{\partial \dot{Q}_1}{\partial \eta} + R_2 \frac{\partial \dot{Q}_2}{\partial \eta}\right) \tag{3.10}$$

in which, R_1 and R_2 are 4 x 4 matrices defined by

$$R_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \alpha_1 & \alpha_2 & 0 \\ 0 & \alpha_2 & \alpha_3 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \qquad R_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \alpha_4 & \frac{\alpha_1 - \alpha_4}{2} & \frac{\alpha_3 - \alpha_4}{2} & \alpha_2 \end{bmatrix}.$$

The vectors $\tilde{Q}_1$ and $\tilde{Q}_2$ are defined by,

$$\tilde{Q}_1 = (\rho, u, v, e)^T, \qquad \tilde{Q}_2 = (e/\rho, u^2, v^2, uv)^T.$$

In this form, the viscous dissipation in the energy equation is separated from the remaining viscous terms so that formulations for which viscous dissipation can be neglected are easily obtained by setting R_2 to zero. A further advantage of this splitting is that matrices R_1 and R_2 contain only metric terms and properties of the gas (viscosity, thermal conductivity and specific heats). For cases where, μ, k and C_p are constants or nearly so, this division makes the linearization of the viscous terms particularly easy. For turbulent flows when these quantities vary rapidly, this form separates them from the dependent variables, making it possible to identify their effects on convergence more clearly.

With the substitution of Eq. (3.10) into Eq. (3.9), we readily have

$$\frac{\partial \tilde{Q}}{\partial t} + \frac{\partial \tilde{E}}{\partial \xi} + \frac{\partial \tilde{F}}{\partial \eta} = \tilde{H} + \frac{\partial}{\partial \eta}\left(R_1 \frac{\partial \tilde{Q}_1}{\partial \eta} + R_2 \frac{\partial \tilde{Q}_2}{\partial \eta}\right). \tag{3.11}$$

This form will be used for discussion in the remaining part of this chapter. Note that the Euler equations for axisymmetric two-dimensional inviscid flows can be obtained by dropping all the viscous diffusion terms in Eq. (3.11), this results in

$$\frac{\partial \tilde{Q}}{\partial t} + \frac{\partial \tilde{E}}{\partial \xi} + \frac{\partial \tilde{F}}{\partial \eta} = \tilde{H}_I \tag{3.12}$$

where, the vector $\tilde{H}_I$ represents the inviscid source vector,

$$\tilde{H}_I = \frac{1}{J} \begin{bmatrix} 0 \\ 0 \\ p \\ 0 \end{bmatrix}.$$

The TLNS approach described above differs from the classical boundary layer theory, in which both streamwise diffusion and normal pressure gradient are neglected as a result of the order of magnitude analysis. Three major advantages of the TLNS approach over the classical boundary layer theory are:

1. Flows containing large normal gradients, such as thick boundary layer or small reverse flow regions can be readily computed by using the TLNS equations, since the pressure gradient is retained inside the boundary layer.

2. The pressure inside the boundary layer couples with the pressure variation in the inviscid core region in the TLNS approach, hence the pressure distribution along the cross-stream direction can be solved in a coupled fashion without need for the inviscid-viscous patching, which is traditionally used in the classical approach for the numerical solution of high Reynolds number nozzle flows [4,5].

3. For internal flows, the TLNS equations automatically conserve mass, while the inviscid-viscous calculation based on the boundary-layer approach generally ignores the mass inside the boundary layer. Although, the effect of this mass deficit on the inviscid flow is compensated by offsetting the wall contour according to the displacement thickness, the patching procedure does not contain proper mass flow rate. This is especially critical for flows with thick boundary layers.

3.2 The Implicit ADI Scheme

As indicated in Chapter 1, the well-developed alternating direction implicit (ADI) scheme has been extensively used to solve compressible gas dynamics problems. In this section, this technique is formulated for the axisymmetric two-dimensional TLNS equations.

The Euler implicit algorithm for Eq. (3.11) can be formally expressed as

$$\frac{\tilde{Q}^{n+1}-\tilde{Q}^{n}}{\Delta t}+\left[\frac{\partial \tilde{E}}{\partial \xi}+\frac{\partial \tilde{F}}{\partial \eta}-\tilde{H}-\frac{\partial}{\partial \eta}\left(R_1\frac{\partial \tilde{Q}_1}{\partial \eta}+R_2\frac{\partial \tilde{Q}_2}{\partial \eta}\right)\right]^{n+1}=0. \qquad (3.13)$$

A local truncated Taylor series expansion can be used to linearize all the flux vectors and the source term. For example, $\tilde{E}$ is linearized by using

$$\tilde{E}^{n+1}=\tilde{E}^{n}+A\Delta\tilde{Q} \qquad (3.14)$$

where $\Delta\tilde{Q}=\tilde{Q}^{n+1}-\tilde{Q}^{n}$ and A is the Jacobian matrix $\partial\tilde{E}/\partial\tilde{Q}$ given by

$$A=\begin{bmatrix} 0 & \xi_x & \xi_y & 0 \\ \xi_x\Phi-uU & U-(\gamma-2)\xi_x u & \xi_y u-(\gamma-1)\xi_x v & (\gamma-1)\xi_x \\ \xi_y\Phi-vU & \xi_x v-(\gamma-1)\xi_y u & U-(\gamma-2)\xi_y v & (\gamma-1)\xi_y \\ U(2\Phi-\gamma\frac{\epsilon}{\rho}) & \xi_x(\gamma\frac{\epsilon}{\rho}-\Phi)-(\gamma-1)uU & \xi_y(\gamma\frac{\epsilon}{\rho}-\Phi)-(\gamma-1)vU & \gamma U \end{bmatrix},$$

in which, Φ is defined by $\Phi=\frac{\gamma-1}{2}(u^2+v^2)$.

Similarly, vectors $\tilde{F}$ and $\tilde{H}$ can be linearized as

$$\tilde{F}^{n+1}=\tilde{F}^{n}+B\Delta\tilde{Q} \qquad (3.15)$$

$$\tilde{H}^{n+1}=\tilde{H}^{n}+D\Delta\tilde{Q} \qquad (3.16)$$

where B and D are Jacobian matrices $\partial\tilde{F}/\partial\tilde{Q}$ and $\partial\tilde{H}/\partial\tilde{Q}$. The matrix B can be obtained by replacing U with V and ξ with η in the A matrix given above. The Jacobian matrix D is

$$D=\begin{bmatrix} 0 & 0 & 0 & 0 \\ d_{21} & 0 & d_{23} & 0 \\ d_{31} & d_{32} & d_{33} & d_{34} \\ d_{41} & d_{42} & d_{43} & 0 \end{bmatrix}$$

where,

$$d_{21} = \frac{2}{3}\frac{\eta_x}{J}\frac{\partial}{\partial\eta}(\frac{\mu v}{\rho}\cdot\frac{J}{y})$$

$$d_{23} = -\frac{2}{3}\frac{\eta_x}{J}\frac{\partial}{\partial\eta}(\frac{\mu}{\rho}\cdot\frac{J}{y})$$

$$d_{31} = \frac{\Phi}{y} + \frac{4}{3}\frac{\mu v}{\rho y^2} - \frac{2}{3}\mu\frac{\eta_x}{J}\frac{\partial}{\partial\eta}(\frac{u}{\rho}\cdot\frac{J}{y})$$

$$d_{32} = -(\gamma-1)\frac{u}{y} + \frac{2}{3}\mu\frac{\eta_x}{J}\frac{\partial}{\partial\eta}(\frac{1}{\rho}\cdot\frac{J}{y})$$

$$d_{33} = -(\gamma-1)\frac{v}{y} - \frac{4}{3}\frac{\mu}{\rho y^2}$$

$$d_{34} = \frac{\gamma-1}{y}$$

$$d_{41} = \frac{4}{3}\frac{\eta_x}{J}\frac{\partial}{\partial\eta}(\frac{\mu u v}{\rho}\cdot\frac{J}{y}) + \frac{4}{3}\frac{\eta_y}{J}\frac{\partial}{\partial\eta}(\frac{\mu v^2}{\rho}\cdot\frac{J}{y})$$

$$d_{42} = -\frac{2}{3}\frac{\eta_x}{J}\frac{\partial}{\partial\eta}(\frac{\mu v}{\rho}\cdot\frac{J}{y})$$

$$d_{43} = -\frac{2}{3}\frac{\eta_x}{J}\frac{\partial}{\partial\eta}(\frac{\mu u}{\rho}\cdot\frac{J}{y}) - \frac{4}{3}\frac{\eta_y}{J}\frac{\partial}{\partial\eta}(\frac{\mu v}{\rho}\cdot\frac{J}{y})$$

The viscous terms can be linearized according to

$$(R_1\frac{\partial\tilde{Q}_1}{\partial\eta} + R_2\frac{\partial\tilde{Q}_2}{\partial\eta})^{n+1} = (R_1\frac{\partial\tilde{Q}_1}{\partial\eta} + R_2\frac{\partial\tilde{Q}_2}{\partial\eta})^{n} + (R_1\frac{\partial}{\partial\eta}B_{v1} + R_2\frac{\partial}{\partial\eta}B_{v2})\Delta\tilde{Q} \tag{3.17}$$

where, only $\tilde{Q}_1$ and $\tilde{Q}_2$ are linearized. The viscous Jacobian matrices are defined by $B_{v1} = \partial\tilde{Q}_1/\partial\tilde{Q}$ and $B_{v2} = \partial\tilde{Q}_2/\partial\tilde{Q}$, they are found to be

$$B_{v1} = \frac{J}{y}\begin{bmatrix} 1 & 0 & 0 & 0 \\ -\frac{u}{\rho} & \frac{1}{\rho} & 0 & 0 \\ -\frac{v}{\rho} & 0 & \frac{1}{\rho} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and

$$B_{v2} = \frac{J}{y}\begin{bmatrix} -\frac{e}{\rho^2} & 0 & 0 & \frac{1}{\rho} \\ -2\frac{u^2}{\rho} & 2\frac{u}{\rho} & 0 & 0 \\ -2\frac{v^2}{\rho} & 0 & 2\frac{v}{\rho} & 0 \\ -2\frac{uv}{\rho} & \frac{v}{\rho} & \frac{u}{\rho} & 0 \end{bmatrix}.$$

Substituting Eq. (3.14) through Eq. (3.17) into Eq. (3.13), we have

$$\{I - \Delta t D + \Delta t[\frac{\partial}{\partial\xi}A + \frac{\partial}{\partial\eta}B - \frac{\partial}{\partial\eta}(R_1\frac{\partial}{\partial\eta}B_{v1} + R_2\frac{\partial}{\partial\eta}B_{v2})]\}\Delta\tilde{Q} = -\Delta t R \tag{3.18}$$

where, the residual vector R is,

$$R = [\frac{\partial \tilde{E}}{\partial \xi} + \frac{\partial \tilde{F}}{\partial \eta} - \tilde{H} - \frac{\partial}{\partial \eta}(R_1 \frac{\partial \tilde{Q}_1}{\partial \eta} + R_2 \frac{\partial \tilde{Q}_2}{\partial \eta})]^n . \tag{3.19}$$

All spatial derivatives in Eq. (3.18) imply that central-differencing will be used for the discretization.

The solution of the Euler implicit scheme, Eq. (3.18), requires the solution of a high band-width block matrix for each iteration, unlike the one-dimensional case, where the matrix is only block tri-diagonal. The computer storage and the CPU time required in solving this high band-width matrix are very large for typical two-dimensional problems. Hence, a procedure like Eq. (3.18) is impractical. The ADI or Approximate Factorization (AF) algorithm arises under this situation. The basic idea of a ADI scheme is to split the left hand side operator of Eq. (3.18) into two parts as,

$$\begin{aligned} &(I - \Delta t D + \Delta t \frac{\partial}{\partial \xi} A)(I - \Delta t D)^{-1} \\ &[I - \Delta t D + \Delta t \frac{\partial}{\partial \eta} B - \Delta t \frac{\partial}{\partial \eta}(R_1 \frac{\partial}{\partial \eta} B_{v1} + R_2 \frac{\partial}{\partial \eta} B_{v2})]\Delta \tilde{Q} = -\Delta t R \end{aligned} \tag{3.20}$$

The first operator on the left hand side of Eq. (3.20) contains only ξ derivatives, and the last one contains only η derivatives. The solution of Eq. (3.20) can be broken into two steps. For the ξ direction sweep, the equation

$$(I - \Delta t D + \Delta t \frac{\partial}{\partial \xi} A)\Delta \tilde{Q}^* = -\Delta t R$$

is solved. After obtaining $\Delta \tilde{Q}^*$ at each grid point, the η direction sweep equation,

$$[I - \Delta t D + \Delta t \frac{\partial}{\partial \eta} B - \Delta t \frac{\partial}{\partial \eta}(R_1 \frac{\partial}{\partial \eta} B_{v1} + R_2 \frac{\partial}{\partial \eta} B_{v2})]\Delta \tilde{Q} = (I - \Delta t D)\Delta \tilde{Q}^*$$

is solved for $\Delta \tilde{Q}$ over the entire domain. The dependent variable is then updated according to

$$\tilde{Q}^{n+1} = \tilde{Q}^n + \Delta \tilde{Q}.$$

Each sweep now requires only the solution of a block tri-diagonal matrix. This approach saves CPU time and computer storage.

3.2.1 Boundary Conditions for the ADI Scheme

The boundary procedures discussed in Section 2.2.1 can be easily extended to the present two dimensional calculation. First we notice that the similarity transformations

$$A = M_\xi \Lambda_\xi M_\xi^{-1}, \qquad B = M_\eta \Lambda_\eta M_\eta^{-1}$$

which convert Jacobian matrices into diagonal matrices Λ_ξ and Λ_η exist for both Jacobian matrices A and B. The left and right eigenmatrices for the Jacobian matrix A are

$$M_\xi = \begin{bmatrix} 1 & 0 & \frac{\rho}{\sqrt{2}c} & \frac{\rho}{\sqrt{2}c} \\ u & \rho k_2 & \frac{\rho}{\sqrt{2}}(\frac{u}{c}+k_1) & \frac{\rho}{\sqrt{2}}(\frac{u}{c}-k_1) \\ v & -\rho k_1 & \frac{\rho}{\sqrt{2}}(\frac{v}{c}+k_2) & \frac{\rho}{\sqrt{2}}(\frac{v}{c}-k_2) \\ \frac{\Phi}{\gamma-1} & \rho(uk_2-vk_1) & \Psi+\frac{\rho}{\sqrt{2}}\theta & \Psi-\frac{\rho}{\sqrt{2}}\theta \end{bmatrix}$$

and

$$M_\xi^{-1} = \begin{bmatrix} \frac{1-\Phi}{c^2} & (\gamma-1)\frac{u}{c^2} & (\gamma-1)\frac{v}{c^2} & -\frac{\gamma-1}{c^2} \\ \frac{-k_2u+k_1v}{\rho} & \frac{k_2}{\rho} & -\frac{k_1}{\rho} & 0 \\ -\frac{\theta}{\sqrt{2}\rho}+\frac{\Phi}{\sqrt{2}\rho c} & \frac{k_1}{\sqrt{2}\rho}+\frac{(1-\gamma)u}{\sqrt{2}\rho c} & \frac{k_2}{\sqrt{2}\rho}+\frac{(1-\gamma)v}{\sqrt{2}\rho c} & \frac{\gamma-1}{\sqrt{2}\rho c} \\ \frac{\theta}{\sqrt{2}\rho}+\frac{\Phi}{\sqrt{2}\rho c} & -\frac{k_1}{\sqrt{2}\rho}+\frac{(1-\gamma)u}{\sqrt{2}\rho c} & -\frac{k_2}{\sqrt{2}\rho}+\frac{(1-\gamma)v}{\sqrt{2}\rho c} & \frac{\gamma-1}{\sqrt{2}\rho c} \end{bmatrix}$$

where, $k_1 = \frac{\xi_x}{\sqrt{\xi_x^2+\xi_y^2}}$ and $k_2 = \frac{\xi_y}{\sqrt{\xi_x^2+\xi_y^2}}$. The quantities Ψ and θ are,

$$\Psi = \frac{\rho\Phi}{\sqrt{2}(\gamma-1)c} + \frac{\rho c}{\sqrt{2}(\gamma-1)}$$

and

$$\theta = k_1 u + k_2 v.$$

The eigenmatrices M_η and M_η^{-1} can be obtained by the substitution of ξ with η in M_ξ and M_ξ^{-1}, respectively. The transformed matrices Λ_ξ and Λ_η are diagonal matrices given by

$$\Lambda_\xi = diag(U, U, U+C_\xi, U-C_\xi), \qquad \Lambda_\eta = diag(V, V, V+C_\eta, V-C_\eta)$$

in which, $C_\xi = \sqrt{\xi_x^2 + \xi_y^2}c$ and $C_\eta = \sqrt{\eta_x^2 + \eta_y^2}c$. The non-vanishing elements of Λ_ξ and Λ_η are the eigenvalues of A and B, respectively.

For demonstration purposes, the boundary procedures for the flow through a nozzle are discussed. Extension to other types of flows is straightforward.

As indicated earlier, streamwise diffusion is negligible for a high Reynolds number flow. Hence, inviscid MOC boundary conditions are applicable at the inlet and the exit plane. If the inflow is supersonic, all eigenvalues are positive, thus four boundary conditions have to be specified in order to determine four unknowns at the inlet. In other words, the dependent variables must be completely specified. If the inflow is subsonic, only the fourth eigenvalue is negative since $U < C_\xi$. This implies three specified boundary conditions together with one decoupled characteristic equation must be imposed at the inlet. One reasonable choice is to the specify stagnation pressure P^0, the stagnation temperature T^0, and the flow angle v/u. If the selection matrix L^- is chosen as $L^- = diag(0,0,0,1)$ and the vector Ω is defined as $\Omega = (P^0, T^0, v/u, 0)$, a boundary procedure similar to Eq. (2.14) can be obtained by multiplying Eq. (3.20) by $L^- M_\xi$ and combining the resulting equation with $\partial\Omega/\partial\tilde{Q}$.

No special treatment is needed at the exit if the outflow is supersonic. For subsonic outflow, the back pressure is usually specified. Thus, a procedure similar to Eq. (2.15) can be applied at the exit.

At the centerline of the nozzle, the dependent variable $\tilde{Q}$ vanishes, since y is identically zero. Therefore $\Delta\tilde{Q}$ is always zero at the centerline. Flow variables can

be calculated after each iteration, by symmetrical conditions,

$$
\begin{aligned}
v &= 0 \\
\frac{\partial u}{\partial y} &= 0 \\
\frac{\partial p}{\partial y} &= 0 \\
\frac{\partial T}{\partial y} &= 0.
\end{aligned}
$$

At the wall, the MOC type procedure can be obtained for inviscid flows by specifying $V = 0$ to reflect the tangency condition and selecting the decoupled characteristic equations corresponding to outgoing characteristics. For viscous flows, the MOC procedure is not valid, the no-slip conditions,

$$
\begin{aligned}
u &= 0 \\
v &= 0
\end{aligned}
\tag{3.21}
$$

together with zero normal pressure gradient and the specified wall temperature,

$$
\begin{aligned}
\frac{\partial p}{\partial n} &= 0 \\
T &= T_w
\end{aligned}
\tag{3.22}
$$

can be used instead. Here, T_w is the specified wall temperature and n denotes the direction normal to the wall. The last equation can be replaced by adiabatic wall or specified heat flux conditions with $\kappa \partial T/\partial n = q''(\xi)$.

There are two different methods to apply these no-slip conditions at the wall. First, $\Delta\tilde{Q}$ can be set equal to zero at the wall [13] when solving the discretized equation, Eq. (3.20). Then flow variables are calculated according to Eqs. (3.21) and (3.22) by using the updated $\tilde{Q}$ from interior nodes. In this way, solutions at the wall lag those of the interior nodes by one iteration step, thus we can refer to this method as an explicit-type boundary procedure. Alternatively, the unknowns at the wall can be coupled to the unknowns at the interior points [14] by relating $\Delta\tilde{Q}$ at

wall to those at interior nodes according to the no-slip conditions. In this approach, the flow variables are forced to obey Eqs. (3.21) and (3.22) at the new time level. This approach solves the unknowns at wall and interior nodes simultaneously, and is referred to as an implicit-type boundary procedure. In the following, both boundary procedures are implemented at wall boundaries and their effects are identified.

3.2.2 Stability Analysis of the ADI Scheme

The stability behavior of the implicit ADI scheme has so far not been identified. By applying the double Fourier transform[24] to the fully implicit scheme, Eq. (3.18), we have

$$L_1\tilde{Q}^{n+1} = L_2\tilde{Q}^n$$

so that the amplification matrix G is

$$G = L_1^{-1}L_2$$

where, $L_1 = I + C_{FI}$ and $L_2 = I$ with C_{FI} given by

$$C_{FI} = -\Delta t D + i\frac{\Delta t}{\Delta\xi}\sin\omega_\xi A + i\frac{\Delta t}{\Delta\eta}\sin\omega_\eta B + 2\frac{\Delta t}{\Delta\eta^2}(1-\cos\omega_\eta)(R_1B_{v1} + R_2B_{v2}).$$

For the implicit ADI scheme, L_1 and L_2 are $L_1 = I + C_{FI} + C_{ADI}$ and $L_2 = I + C_{ADI}$, with

$$C_{ADI} = -\frac{\Delta t^2}{\Delta\xi\Delta\eta}\sin\omega_\xi\sin\omega_\eta A(I-\Delta t D)^{-1}B$$
$$+ 2\frac{\Delta t^2}{\Delta\xi\Delta\eta^2}\sin\omega_\xi(\cos\omega_\eta - 1)A(I-\Delta t D)^{-1}(R_1B_{v1} + R_2B_{v2}).$$

The matrix C_{ADI} is the contribution due to the approximate factorization. In these expressions, ω_ξ and ω_η represent the wave numbers in ξ and η directions, respectively.

As stated earlier, the numerical stability is controlled by the magnitude of the maximum eigenvalue of the matrix G. The variation of the maximum eigenvalue

as ω_ξ and ω_η change from 0 to π constitutes a three-dimensional surface. To make the results more readable, only the variation along the diagonal line ($\omega_\xi = \omega_\eta$) on the spectral plane is calculated. Figures 13 and 14 compare the results of the algorithm described above with and without approximate factorization. The flow Mach number is 0.5 and the Reynolds number is 10^4 for both cases. The time step size Δt is determined by the CFL number based on $U + C_\xi$. The maximum eigenvalue is shown to be always less than unity for the fully implicit scheme, and hence is unconditionally stable for the two-dimensional TLNS equations. As the CFL number increases, the maximum eigenvalue at moderate wave numbers (around $\pi/2$) decreases. This implies that the convergence rate will be speeded up by increasing the CFL value.

For the ADI scheme, the approximate factorization alters the shape of stability curves substantially, especially when CFL is large. In general, the maximum eigenvalue at the mid-wavenumber condition is much higher than that of the fully implicit scheme and approaches unity as the wavenumber increase. The results on the figure suggest that the ADI algorithm has an optimum convergence rate at a finite value of CFL.

Effects of the flow Mach number are indicated in Fig. 15, where consecutive cases are calculated for several Mach numbers with a CFL number of 10 for the ADI scheme. The eigenvalues approach unity for a very low Mach number (10^{-2}), thus slow convergence can be expected. As the Mach number increases, the eigenvalues near low and high wave numbers decrease, thus better convergence is implied. Although not shown here, effects of the flow Reynolds number on stability curves are less prominent as compared to those of the flow Mach number and the CFL number for the present local stability analysis.

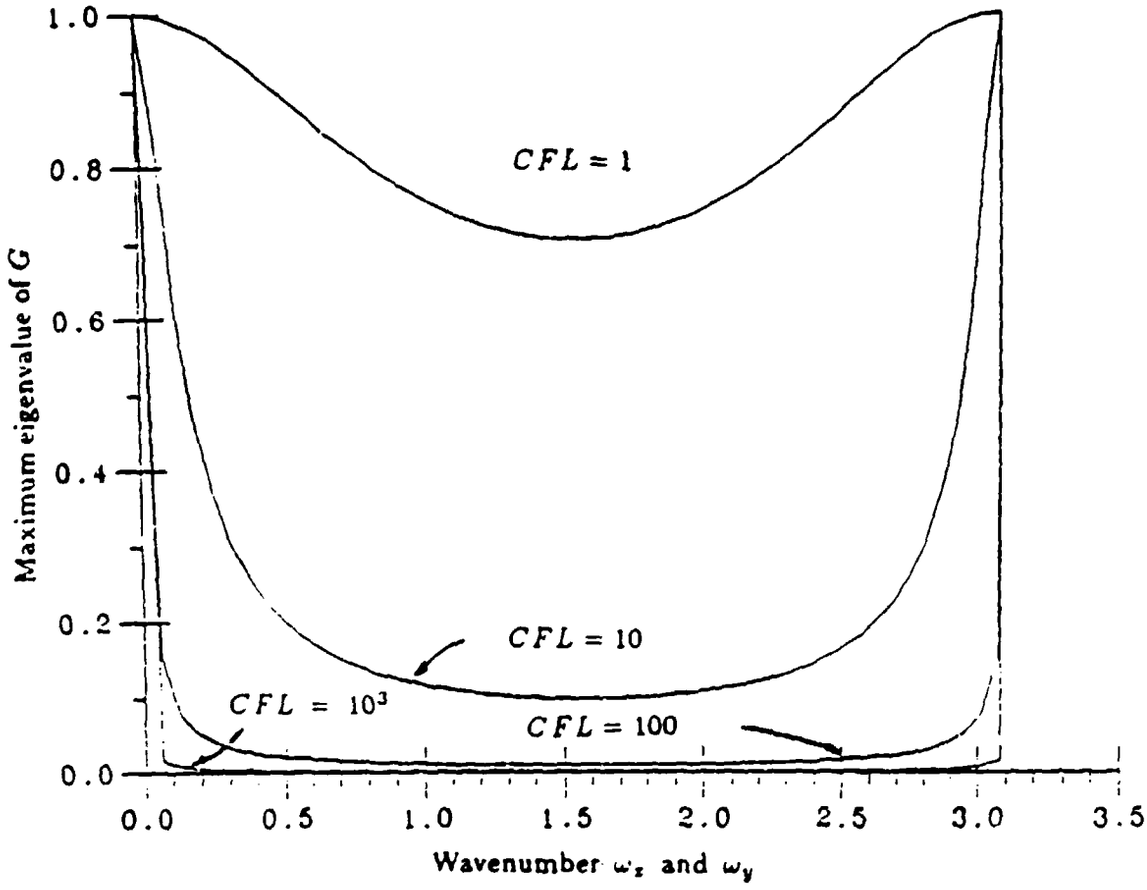


Figure 13. Stability results of 2-D fully implicit scheme without approximate factorization

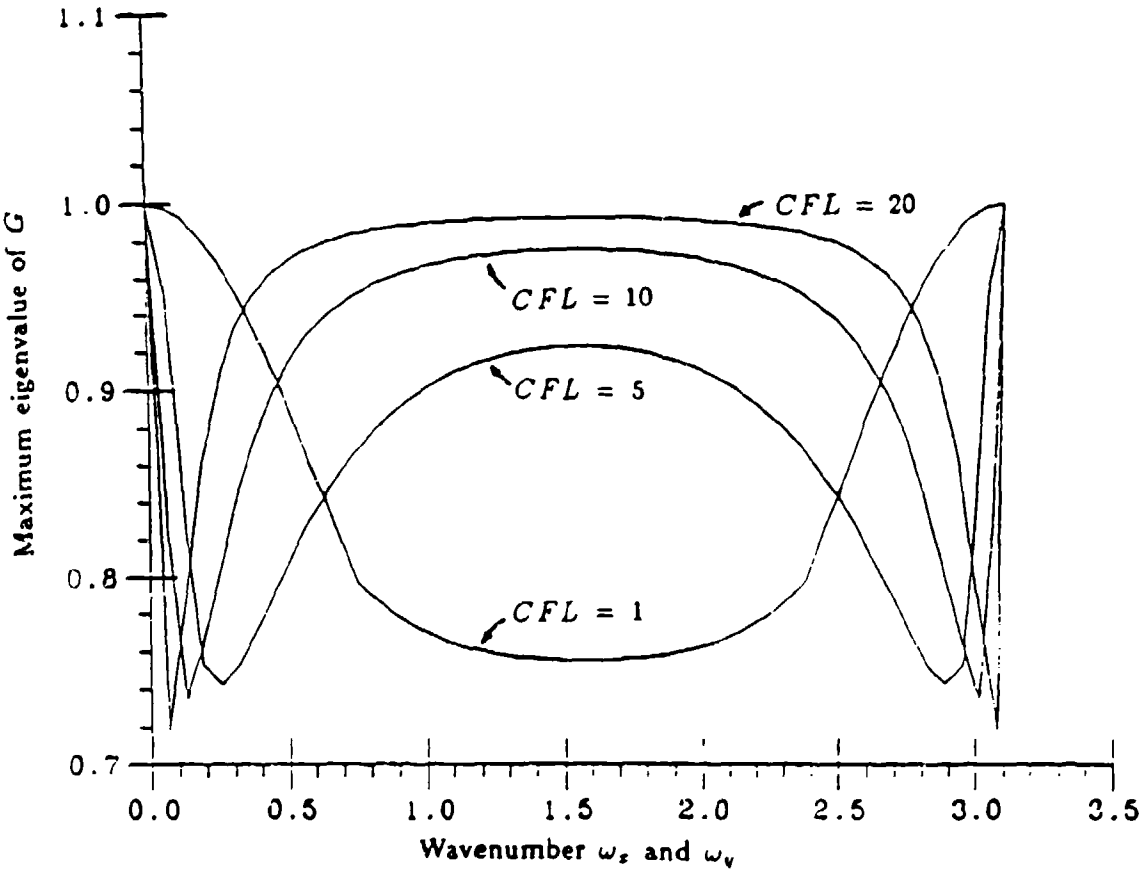


Figure 14. Stability results of 2-D implicit ADI scheme with approximate factorization

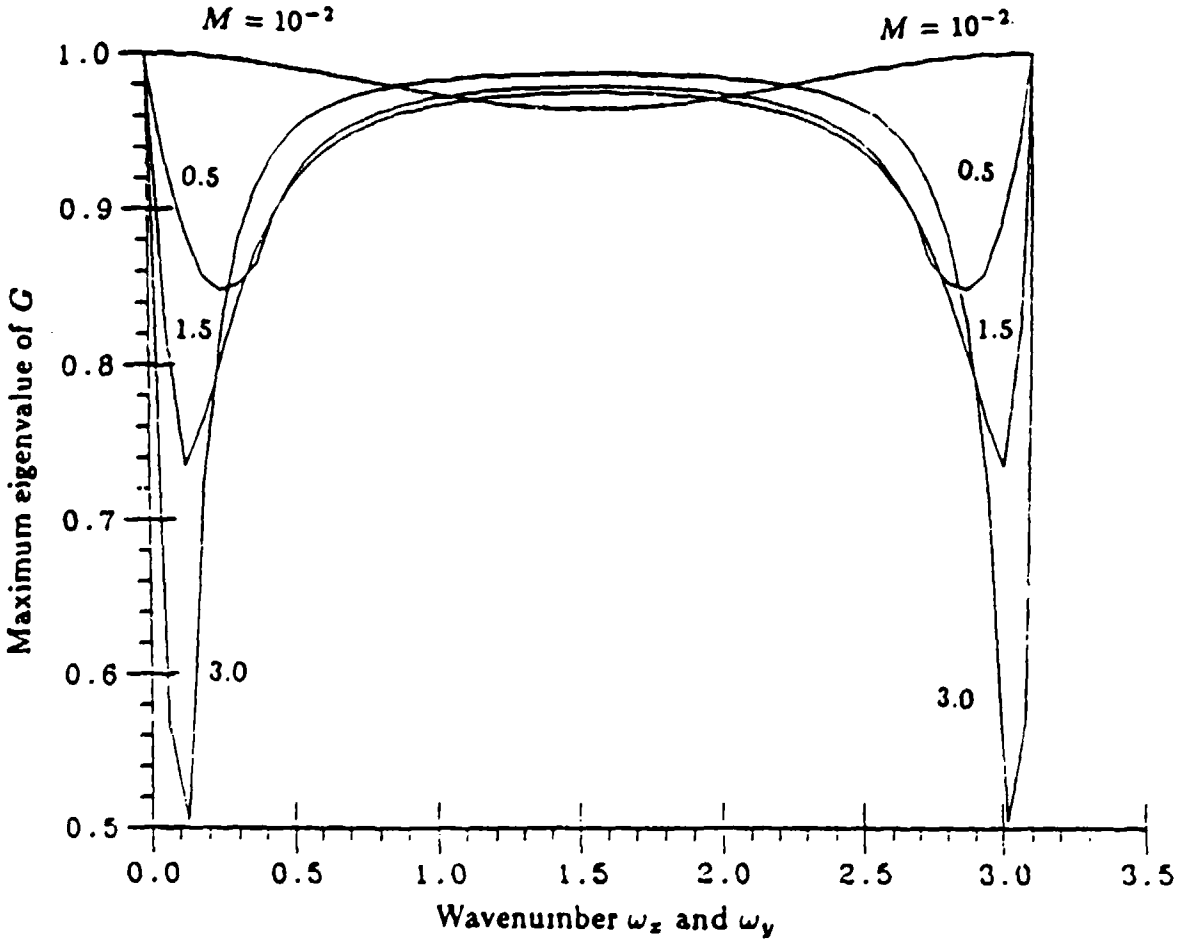


Figure 15. Stability results of 2-D implicit ADI scheme with approximate factorization for various Mach numbers

3.2.3 Results and Discussion

The flow through an axisymmetric converging-diverging nozzle with the wall shape defined by the radius variation,

$$r(x) = -2(AR - 1)x^3 + 3ARx^2 + 1$$

is calculated for illustration. Here AR is the ratio of the throat radius to the inlet radius. The geometry of the nozzle and a representative grid are shown in Fig. 16. Computational results are shown in Figs. 17-19. Two typical cases are calculated, including a pure subsonic flow, and a transonic flow through the c-d nozzle with $AR = 0.8$. The grid is 50×30 with 30 in the cross-stream direction for inviscid calculations and is 50×50 with a strong clustering near the wall for viscous calculations. The viscous grid is shown in Fig. 16.

The computational results have indicated that the resulting Mach numbers at the entrance are around 0.2 and 0.4 for the subsonic and the transonic cases, respectively. Figure 17 shows the L-2 norm of the change in the dependent variable, $\Delta\tilde{Q}/\tilde{Q}$, versus the number of iterations for the inviscid calculations by using the inviscid grid. Corresponding results for the viscous calculations based on the viscous grid are shown in Fig. 18. Both inviscid and viscous results show that the transonic case converges faster than the pure subsonic case. This is due to the low Mach number effects of the subsonic case.

The comparison of Fig. 17 and Fig. 18 indicates that inviscid calculations converge faster than corresponding viscous calculations. There are two possible reasons for this; one is the viscous diffusion inside the boundary layer, the other is the grid stretching near the wall. To understand which was controlling, the inviscid calculations are done on the viscous grid for both subsonic and transonic cases. The results are also shown on Fig. 18. As is seen, inviscid calculations based on the viscous

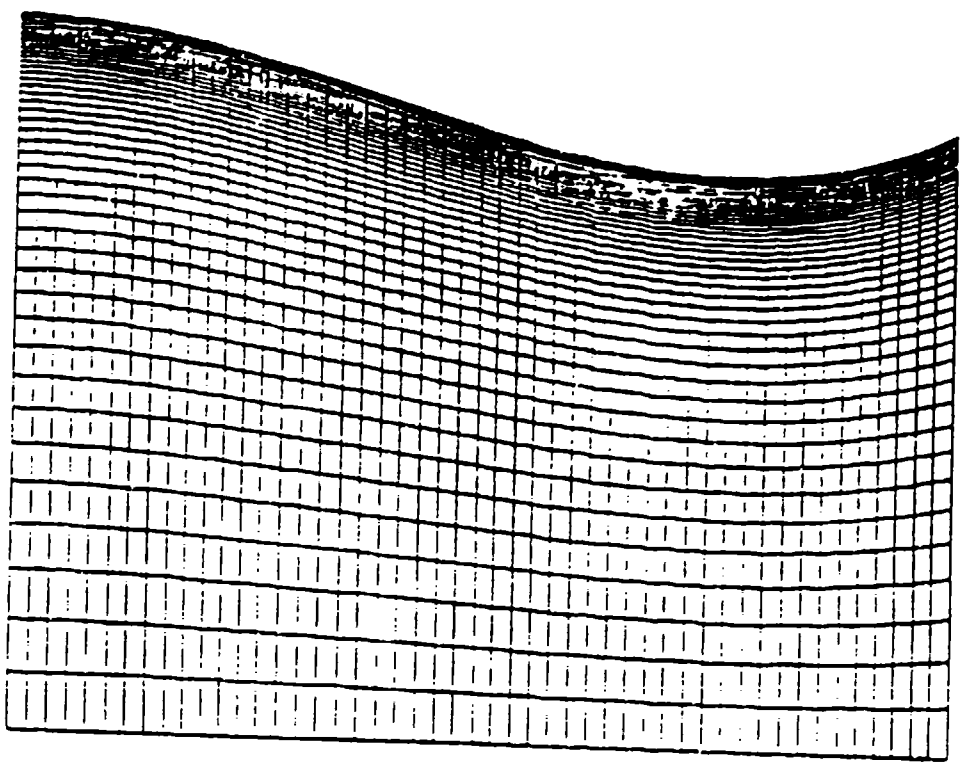


Figure 16. Nozzle geometry for 2-D transonic calculations

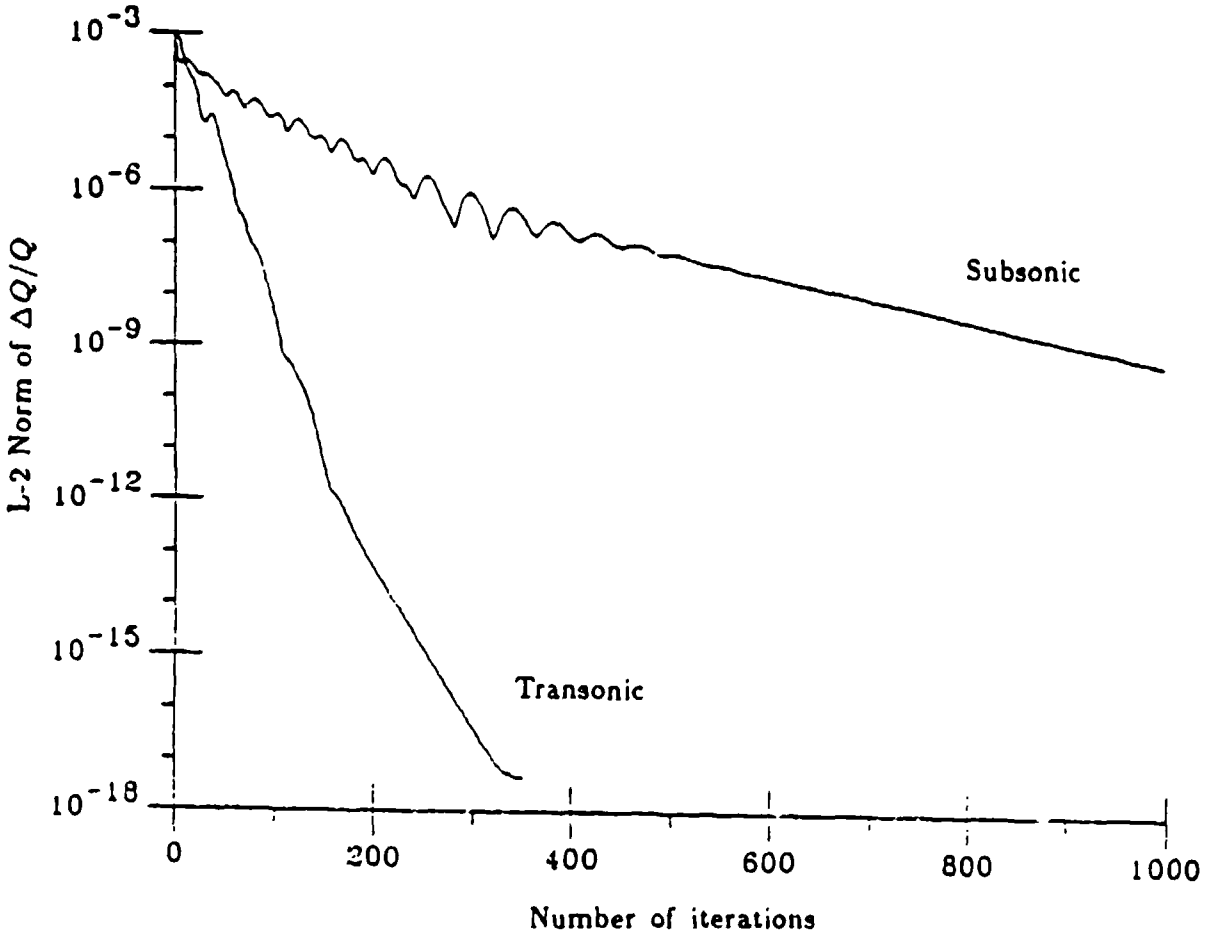


Figure 17. Convergence of 2-D ADI scheme on uniform grid

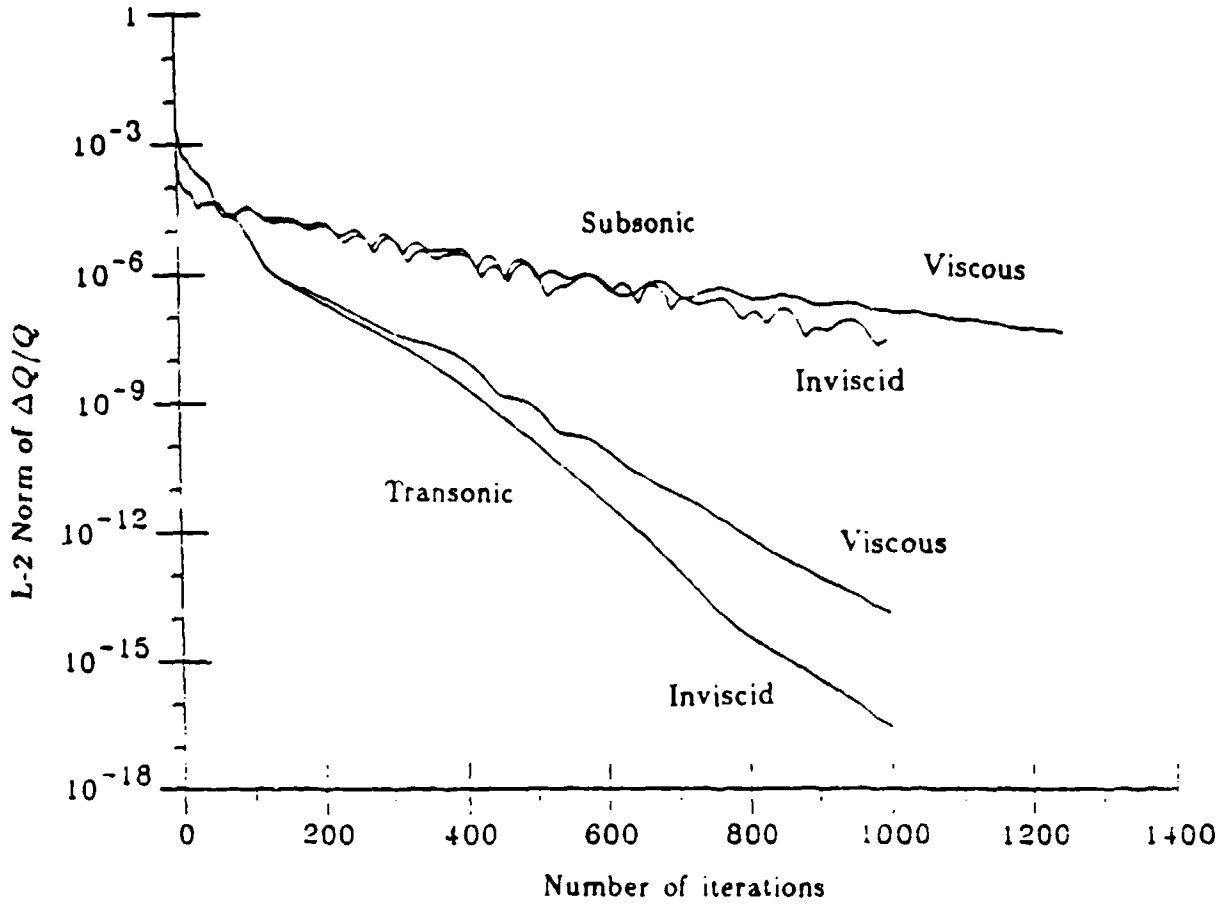


Figure 18. Convergence of 2-D ADI scheme on stretched grid

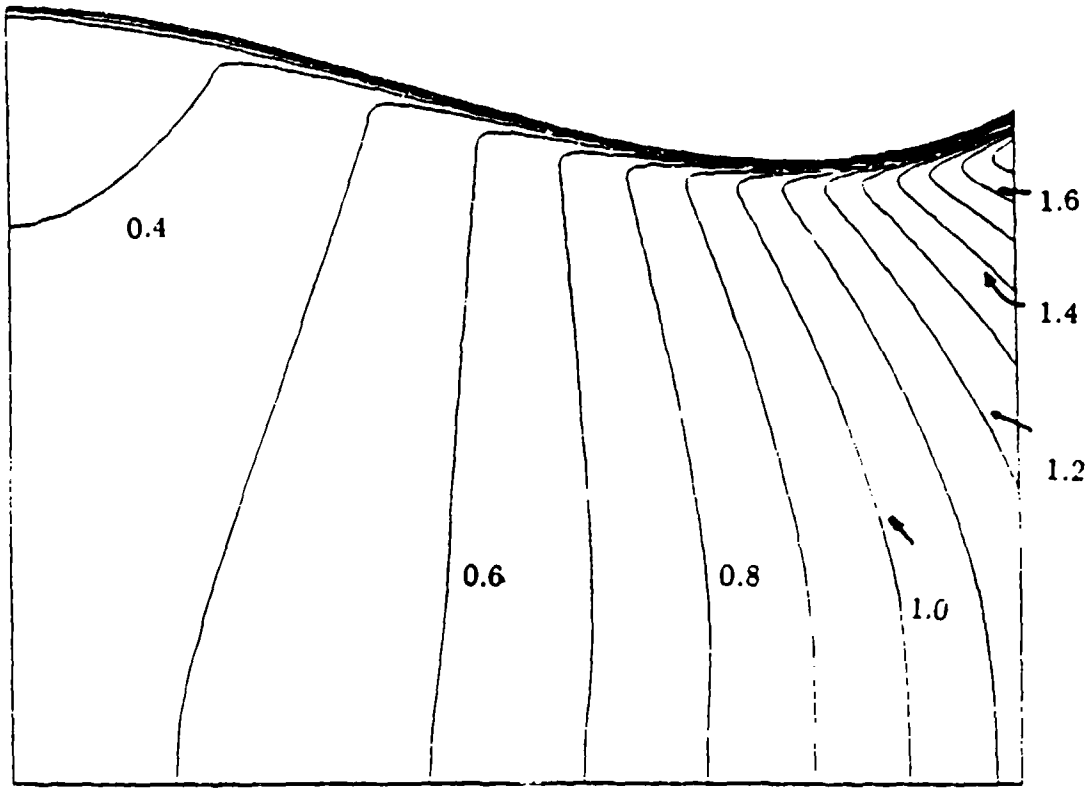


Figure 19. Mach number contours of transonic solutions for ADI scheme

grid give almost the same convergence rates as viscous calculations. Therefore, we can conclude that the grid stretching deteriorates the convergence.

The optimum CFL number for all calculations is around 5, which is typical for a factorization scheme. At a CFL number as low as the order of 10, further numerical experiments with the ADI scheme show that the explicit or the implicit wall boundary procedure has no effect on the convergence. Computed flowfield results are shown in Fig. 19 where Mach number contours for the viscous transonic case are plotted.

The ADI algorithm has also been applied to a pure supersonic calculation. To avoid the presence of shock waves, a 15 degree conical nozzle with an expansion ratio (the exit area to the inlet area) of 30 and an inlet Mach number of 1.02 is calculated. Figure 20 shows convergence curves for both viscous and inviscid results. Comparisons with Fig. 17 and Fig. 18 show that the inviscid supersonic case converges faster than the inviscid transonic calculation, while the viscous supersonic case is as slow as the viscous subsonic case. Although these preliminary supersonic calculations show fairly good results, further numerical experiments on a higher expansion ratio nozzle (100:1) have indicated that the analysis code based on the central-differenced ADI scheme is difficult to start with an arbitrary assigned initial guess. Usually a very low CFL number (around 1) has to be used at the initial stage. Also, the convergence rate deteriorates as the area ratio increases. Additional experiments of transonic flows through the convergent-divergent nozzle described above show that as the size of the divergent section increases by a certain amount, the central-differenced code becomes unstable and diverges and the upwind-central differencing method described later is recommended.

The computational results above illustrate that the implicit ADI procedure is efficient for flows at transonic speeds. For flows which are predominantly super-

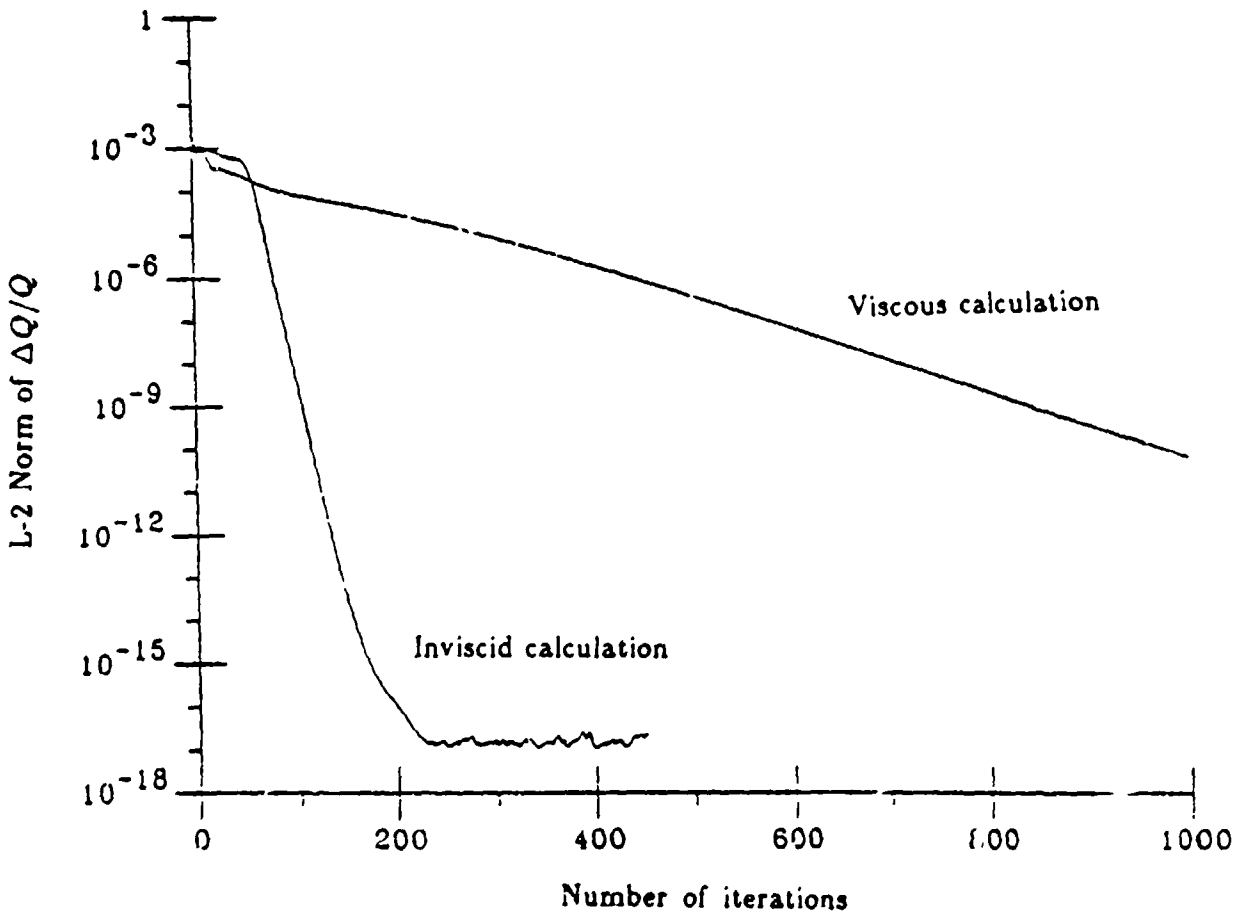


Figure 20. Convergence of 2-D ADI scheme, inviscid & viscous calculations, supersonic flows

sonic, such as those flows through high expansion ratio nozzles, the implicit ADI procedure is inefficient and sometimes even unstable. By contrast, our stability results have indicated that high Mach number flows should converge faster than low Mach number flows. The reason for this contradiction is as follows. Although high Mach number flows have stronger damping effects than lower Mach number flows, once the flow becomes supersonic, the equation set is hyperbolic in the streamwise direction since all eigenvalues of the Jacobian matrix A are positive. This hyperbolicity implies only upstream events can affect downstream events, however, the central-difference formulation in the ADI scheme allows upstream propagation. In the earlier stages of a computation, this upstream influence keeps on propagating wrong information (from the unconverged solution) from downstream back to upstream, and consequently slows down the convergence. This also explains why the code is difficult to start for supersonic cases.

3.3 Stability Consideration of Upwind Algorithms

As we have seen in the previous section, the implicit ADI algorithm becomes inefficient when the flow is predominantly supersonic due to the fact that the central-difference formulation allows the downstream influence of the unconverged solution to propagate upstream. To avoid this unwanted upstream propagation, upwind schemes appear to be desirable. The basic idea of a upwind difference is to model the physics correctly by using a difference stencil which involves only windward information, hence "wrong" propagation is prohibited. The upwind-difference formulation is chosen to develop suitable numerical algorithms for supersonic calculations. Based on the successful application in one-dimensional flows, the flux-vector splitting algorithm [16] will again be utilized to develop upwind algorithms. Before formulating the detail of the numerical procedures, the stability analysis of a modeled scalar equation is considered for several possible solution procedures.

To simulate present two-dimensional viscous flows, the Burger's equation,

$$\frac{\partial u}{\partial t} + a^+ \frac{\partial u}{\partial x} + a^- \frac{\partial u}{\partial x} + b\frac{\partial u}{\partial y} = \mu \frac{\partial^2 u}{\partial y^2} \tag{3.23}$$

is chosen as the modeled equation, in which a^+ and a^- are positive and negative constants, respectively. The second and the third terms on the left hand side are used to simulate the flux-splitting in the streamwise direction, where the flow is predominantly supersonic. Again, these terms must be differenced according to the windward directions. Although second order upwind differencing will be used to formulate the numerical procedures for Navier-Stokes calculations; for the modeled equation, only first order accurate differencing is considered. The last term on the left hand side implies that central-differencing will be retained since no preferable windward direction exists on the cross plane. The second order term on the right hand side is used to model the thin-layer viscous diffusion.

3.3.1 The Fully Implicit Scheme

Direct application of the Euler implicit scheme to Eq. (3.23) results in

$$[1 + \Delta t(a^+ \frac{\partial}{\partial x} + a^- \frac{\partial}{\partial x} + b\frac{\partial}{\partial y} - \mu \frac{\partial^2}{\partial y^2})]\Delta u = -\Delta t r \tag{3.24}$$

where, r is the residual given by

$$r = (a^+ \frac{\partial u}{\partial x} + a^- \frac{\partial u}{\partial x} + b\frac{\partial u}{\partial y} - \mu \frac{\partial^2 u}{\partial y^2})^n .$$

The amplification factor g_{FI} for Eq. (3.24) can be easily found to be

$$g_{FI} = \frac{1}{1 + C_{FI}}$$

where, C_{FI} is

$$C_{FI} = \sigma_x^+(1 - \cos\omega_x + i\sin\omega_x) + \sigma_x^-(\cos\omega_x + i\sin\omega_x - 1) + i\sigma_y \sin\omega_y + 2\nu(1 - \cos\omega_y).$$

The parameters, σ_x^+,σ_x^-,and σ_y are CFL numbers defined by

$$\sigma_x^+ = \frac{a^+\Delta t}{\Delta x}, \qquad \sigma_x^- = \frac{a^-\Delta t}{\Delta x}, \qquad \sigma_y = \frac{b\Delta t}{\Delta y}.$$

The parameter ν is the von Neumann number defined by

$$\nu = \frac{\mu\Delta t}{\Delta y^2}.$$

As stated earlier, the direct solution of a fully implicit scheme is impractical in multi-dimensions, hence approximate factorization is necessary. A number of approximate procedures can be identified for solving the left hand side operator in Eq. (3.24). Of them, three factorization procedures will be discussed.

3.3.2 The Standard ADI Scheme

The first approximate procedure is to split the operator in Eq. (3.24) as

$$[1 + \Delta t(a^+\frac{\partial}{\partial x} + a^-\frac{\partial}{\partial x})][1 + \Delta t(b\frac{\partial}{\partial y} - \mu\frac{\partial^2}{\partial y^2})]\Delta u = -\Delta t r. \qquad (3.25)$$

Equation (3.25) is analogous to the implicit ADI procedure, except upwinding has been used in the streamwise direction. This splitting generates error terms on the left hand side of order Δt^2. The resulting amplification factor g_{ADI} for Eq. (3.25) is

$$g_{ADI} = \frac{1 + C_{ADI}}{1 + C_{FI} + C_{ADI}}$$

where,

$$C_{ADI} = i\sigma_x^+\sigma_y(1 - \cos\omega_x + i\sin\omega_x)\sin\omega_y + i\sigma_x^-\sigma_y(\cos\omega_x + i\sin\omega_x - 1)\sin\omega_y$$
$$+ 2\sigma_x^+\nu(1 - \cos\omega_x + i\sin\omega_x)(1 - \cos\omega_y)$$
$$+ 2\sigma_x^-\nu(\cos\omega_x + i\sin\omega_x - 1)(1 - \cos\omega_y)$$

is the summation of factorization error terms.

3.3.3 The Diagonally Dominant ADI Scheme

The second splitting under consideration is the diagonally dominant alternating direction implicit (DDADI) method suggested by Lombard [14]. This splitting, when first order upwind differencing is used, can be expressed as

$$[d-\frac{\Delta t a^+}{\Delta x}u_{i-1,j}+\Delta t(b\frac{\partial}{\partial y}-\mu\frac{\partial^2}{\partial y^2})]d^{-1}[d+\frac{\Delta t a^-}{\Delta x}u_{i+1,j}+\Delta t(b\frac{\partial}{\partial y}-\mu\frac{\partial^2}{\partial y^2})]\Delta u = -\Delta t r \tag{3.26}$$

where d is the summation of diagonal elements given by $d = 1 + \frac{\Delta t}{\Delta x}(a^+ - a^-)$. Equation (3.26) can be solved directly by using alternating forward and backward sweeps. Alternatively, it can also be solved by using the line Gauss-Seidel relaxation by MacCormack [15] and Chakravarthy [17]. This method has also been referred to as the single-level scheme by Lombard [19], and the LU scheme by Yoon and Jameson [18].

The line relaxation method for the DDADI splitting is discussed as follows. First, Eq. (3.26) can be split into,

$$[d + \Delta t(b\frac{\partial}{\partial y} - \mu\frac{\partial^2}{\partial y^2})]\Delta u^* = -\Delta t r + \frac{\Delta t}{\Delta x}a^+\Delta u^*_{i-1,j} \tag{3.27}$$

and

$$[d + \Delta t(b\frac{\partial}{\partial y} - \mu\frac{\partial^2}{\partial y^2})]\Delta u = d\Delta u^*_{i,j} - \frac{\Delta t}{\Delta x}a^-\Delta u_{i+1,j} \tag{3.28}$$

Equation (3.27) can be rearranged as

$$[d + \Delta t(b\frac{\partial}{\partial y} - \mu\frac{\partial^2}{\partial y^2})]\Delta u^* = -\Delta t[a^+\frac{u^*_{i,j} - u^*_{i-1,j}}{\Delta x} + (a^-\frac{\partial u}{\partial x} + b\frac{\partial u}{\partial y} - \mu\frac{\partial^2 u}{\partial y^2})] \tag{3.29}$$

Equation (3.29) is lower bi-diagonal and can be solved by marching from the upstream toward the downstream at each cross-stream station after obtaining $u^*_{i-1,j}$ from the iteration at previous stations. Similar to this forward marching procedure, we can define a symmetrical backward marching procedure by substituting

Δu^* from Eq. (3.27) into Eq. (3.28) and neglecting unnecessary terms in order to make the resulting right hand side residual to be consistent with the steady state equation, $r = 0$, as Δu is driven to zero. The resulting backward marching equation can be written as,

$$[d + \Delta t(b\frac{\partial}{\partial y} - \mu\frac{\partial^2}{\partial y^2})]\Delta u = -\Delta t[a^- \frac{u^{n+1}_{i+1,j} - u^*_{i,j}}{\Delta x} + (a^+\frac{\partial u}{\partial x} + b\frac{\partial u}{\partial y} - \mu\frac{\partial^2 u}{\partial y^2})^*]. \tag{3.30}$$

Equation (3.30) now can be solved by marching from the downstream toward the upstream using the updated value of $u^{n+1}_{i+1,j}$ at the $i+1$ station. The combination of Eq. (3.29) and Eq. (3.30) completes one iteration step of the line-relaxation method for the DDADI splitting.

The amplification factor g^* for the intermediate forward marching, Eq. (3.29), is

$$g^* = \frac{u^*}{u^n} = \frac{1 - \sigma_x^-(\cos\omega_x + i\sin\omega_x)}{1 + \sigma_x^+(1 - \cos\omega_x + i\sin\omega_x) - \sigma_x^- + i\sigma_y\sin\omega_y + 2\nu(1 - \cos\omega_y)}.$$

and the amplification factor g^{**} for Eq. (3.30) is

$$g^{**} = \frac{u^{n+1}}{u^*} = \frac{1 + \sigma_x^+(\cos\omega_x - i\sin\omega_x)}{1 + \sigma_x^+ + \sigma_x^-(\cos\omega_x + i\sin\omega_x - 1) + i\sigma_y\sin\omega_y + 2\nu(1 - \cos\omega_y)}.$$

The overall amplification factor is then

$$g_{DDADI} = g^* g^{**}. \tag{3.31}$$

3.3.4 The Parabolized ADI Scheme

The third algorithm considered is the splitting,

$$[1 + \Delta t(a^+\frac{\partial}{\partial x} + b\frac{\partial}{\partial y} - \mu\frac{\partial^2}{\partial y^2})](1 + \Delta t a^-\frac{\partial}{\partial x})\Delta u = -\Delta t r. \tag{3.32}$$

As we shall see in the next section, the first operator on the LHS is similar to a parabolized Navier-Stoke (PNS) marching operator, hence, Eq. (3.32) is referred to

as a PNS-ADI splitting. The amplification factor for this splitting can be found to be,

$$g_{PNS-ADI} = \frac{1 + C_{PNS}}{1 + C_{FI} + C_{PNS}} \tag{3.33}$$

in which,

$$C_{PNS} = \sigma_x^+ \sigma_x^- (1 - \cos\omega_x + i\sin\omega_x)(\cos\omega_x + i\sin\omega_x - 1) +$$

$$i\sigma_x^- \sigma_y (\cos\omega_x + i\sin\omega_x - 1)\sin\omega_y + 2\sigma_x^- \nu (\cos\omega_x + i\sin\omega_x - 1)(1 - \cos\omega_y).$$

Note that if a^- vanishes, the error C_{PNS} is zero and the PNS-ADI splitting is equivalent to the fully implicit scheme.

3.3.5 Algorithm Comparisons

The amplification factors for all four algorithms noted above can be numerically computed over the entire spectral plane for ω_x and ω_y ranging from 0 to π. Again, to make results more concise and readable, only the variation along the diagonal line ($\omega_x = \omega_y$) on the spectral plane is calculated. Two cases are studied, they are,

1. Subsonic : $\sigma_x^+ = -\sigma_x^- = \sigma_y = \nu = CFL$
2. upersonic: $\sigma_x^+ = \sigma_y = \nu = CFL, \qquad \sigma_x^- = 0$

The first case is analogous to the subsonic flow, where both upstream and downstream propagations are significant. The second case simulates supersonic flow by setting $a^- = 0$. Both cases assume uniform CFL numbers in the x and y directions.

Results of these two cases are plotted on Fig. 21 and Fig. 22 for a CFL of 10. For the first case. both the fully implicit and the DDADI algorithms give monotonically decreasing eigenvalues, and these eigenvalues reach their minimums at the wavenumber π. The eigenvalue first decreases, then increases to a local maximum near the wave number π for the ADI and PNS-ADI schemes. In the second case, all algorithms except the ADI splitting give monotonically decreasing

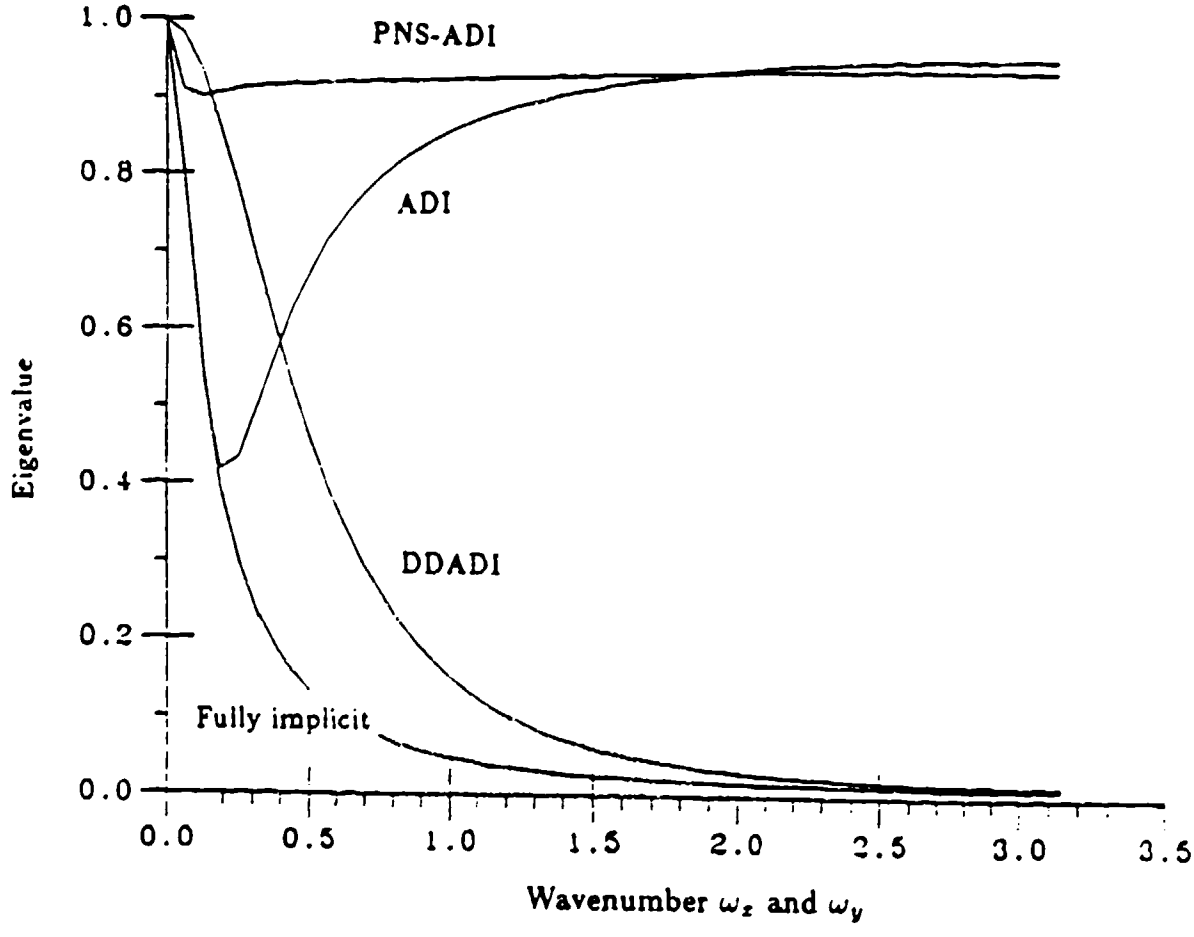


Figure 21. Stability results of first-order upwind-central scheme, $\sigma_x^+ = -\sigma_x^- = \sigma_y$: $\nu = 10$

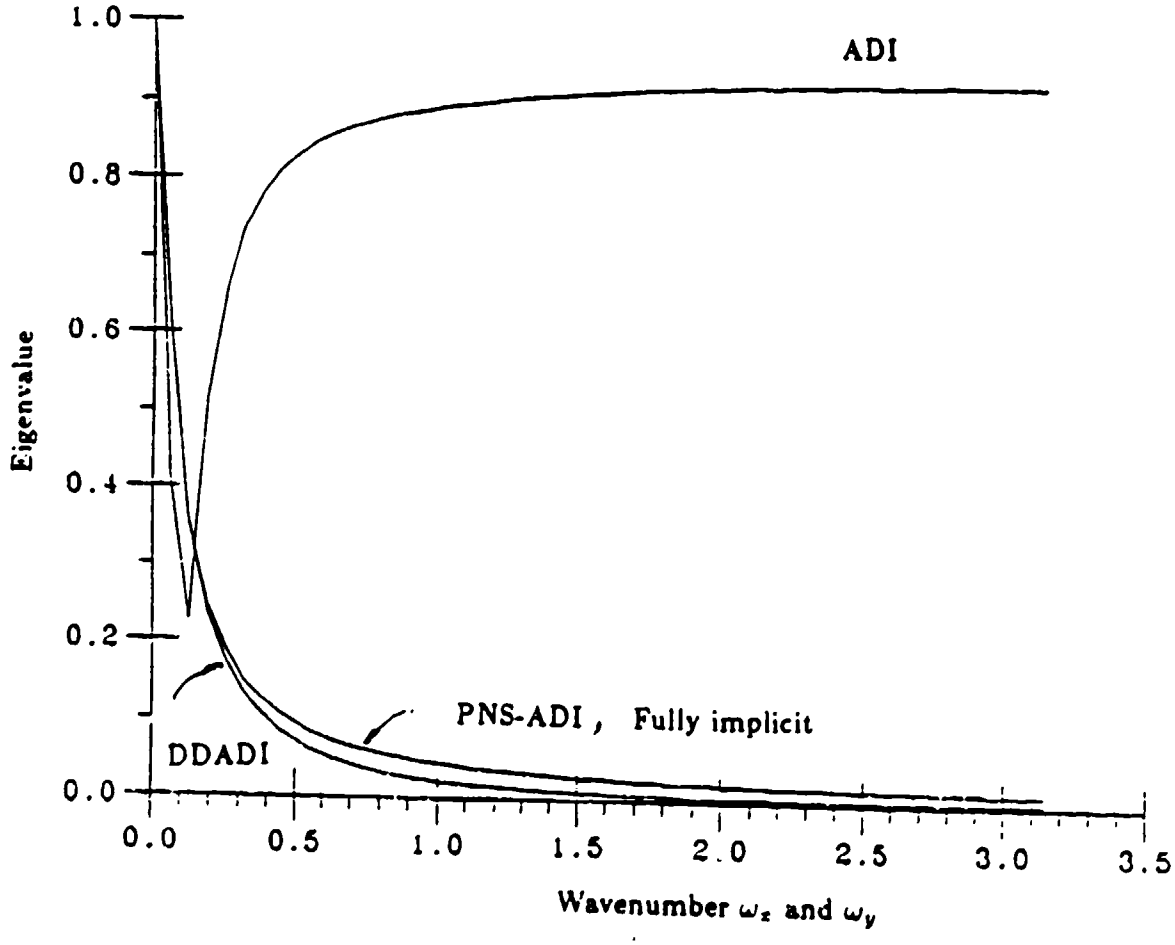


Figure 22. Stability results of first-order upwind-central scheme, $\sigma_x^+ = \sigma_y = \nu = 10$, $\sigma_x^- = 0$

eigenvalues. This is due to the factorization errors of the ADI splitting even when a^- is zero. The eigenvalue of the DDADI splitting is slightly smaller than that of the fully implicit scheme and the PNS-ADI splitting in the second case. Again, the naturally dissipative characteristic of upwind schemes is observed by noting that the eigenvalues are much less than unity near the wave number π.

As a final test case, the effects of CFL numbers on the magnitude of the eigenvalues are shown in Fig. 23 for the subsonic case ($a^- \neq 0$). As can be seen, increasing the CFL number tends to magnify the eigenvalues for the PNS-ADI splitting, thus a CFL number of 10^3 is inferior to a CFL number of 10 as far as convergence is concerned. On the other hand, the DDADI splitting is insensitive to the CFL number. Effects of the CFL number on ADI and fully implicit schemes are similar to those on PNS-ADI and DDADI schemes, respectively.

Among the four algorithms, the fully implicit scheme and the DDADI splitting can be expected to give better convergence rates than the other two splittings, since they have smaller eigenvalues over the entire spectrum. The DDADI splitting is superior to the other two approximate procedures and is as good as the fully implicit scheme. The PNS-ADI splitting will give better convergence than the standard ADI splitting since they have similar behavior in subsonic regions ($a^- \neq 0$) and the PNS-ADI scheme is more dissipative in supersonic regions ($a^- = 0$). All four algorithms can be shown to have eigenvalues always less than unity, hence they are unconditionally stable for the two-dimensional Burger's equation.

Based on the results of stability analysis, the proposed hybrid algorithm which uses upwind difference in the streamwise direction and central difference in the cross-stream direction proves to be unconditionally stable for the Burger's equation for fully implicit and all three approximate procedures investigated.

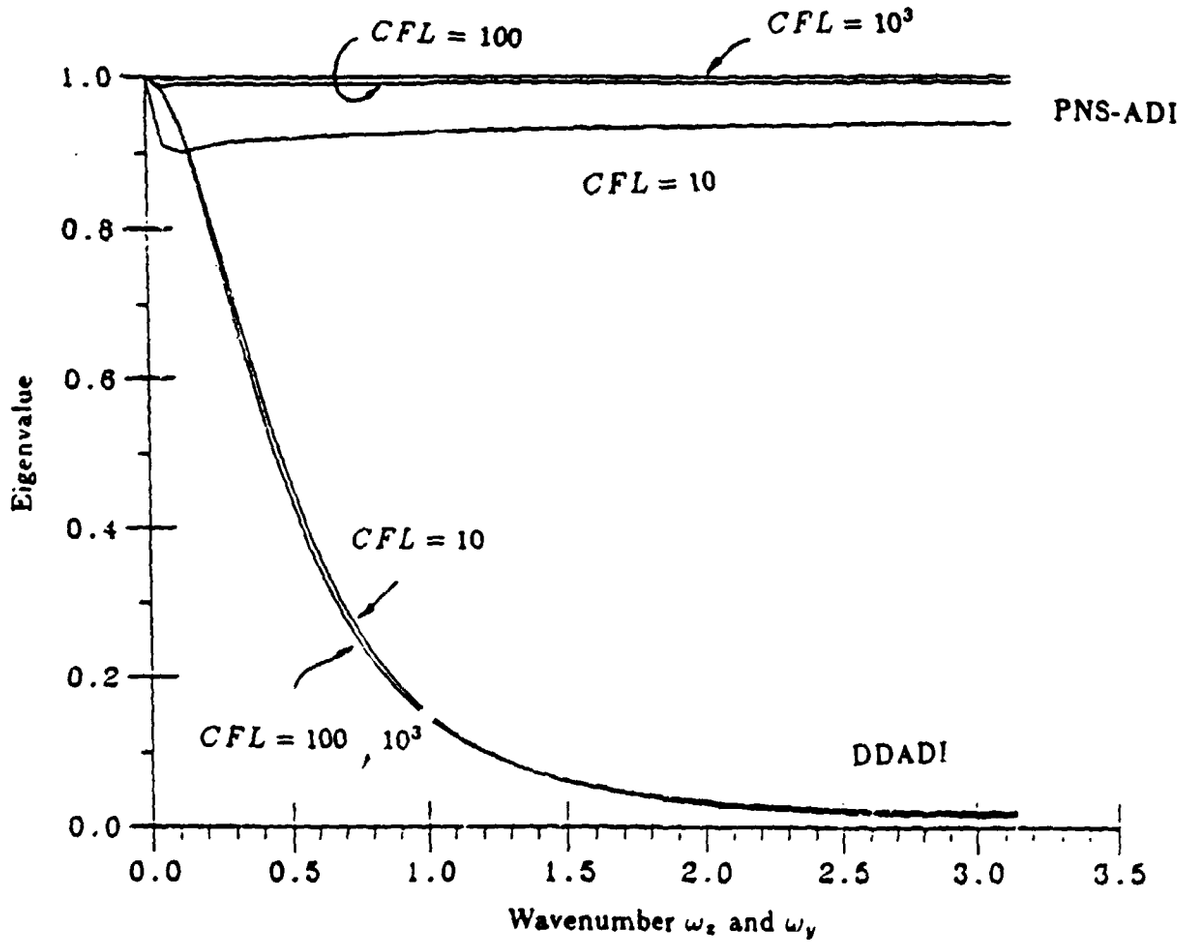


Figure 23. Stability results of first-order upwind-central scheme, varying CFL, $\sigma_z^- \neq 0$

The stability results discussed above are based on first-order upwind differencing and the scalar modeled equation. For algorithms based on more accurate second-order upwind differencing and the vector equation, the stability results may be quite different from the results given above. For instance, the approximate factorization error term associated with the product of $a^+\partial/\partial x$ and $a^-\partial/\partial x$ in the Parabolized ADI scheme (Eq. (3.32)) reduces to zero for the vector equation since A^+A^- is identically zero for both the subsonic and supersonic cases. This implies that the PNS-ADI procedure will give better convergence than that predicted by the stability analysis based on the Burger's equation. In the following discussion, this hybrid algorithm will be utilized to formulate numerical algorithms for viscous supersonic calculations.

3.4 Algorithms for Viscous Supersonic Flows

As is well known, the hyperbolic nature of supersonic flows allows inviscid problems to be computed in a single pass. This capability is lost in viscous flows for two reasons. The first is the streamwise diffusion, but this is weak for high and even moderate Reynolds number flows and may frequently be neglected, which results in the TLNS equations. The second and more significant reason is the subsonic region near the wall, where upstream influences exist. From a physical point of view, this subsonic region allows information to propagate upstream. This upstream propagation prohibits a single pass solution and renders the space-marching procedure unconditionally unstable. In the PNS approach, this upstream influence is removed by a parabolized procedure which neglects the contribution associated with this upstream influence so that a single sweep solution is allowed [26-29]. For problems with significant upstream influence, such as thick boundary layers inside the nozzles [5] and separated flows, global iterations are required. The main focus in this chapter falls into this category, thus upstream effects must be retained by considering

the global iteration from the Navier-Stokes equations. The simplification to PNS procedures will be discussed in the next chapter.

Based on the results of stability consideration, in discretizing the governing equations, we can take advantage of the predominantly supersonic character of the flowfield by using flux-vector splitting in the streamwise direction, while retaining central differencing in the cross-stream directions. Again, using Euler implicit differencing in time, the result can be expressed as

$$\frac{\tilde{Q}^{n+1}-\tilde{Q}^{n}}{\Delta t}+\left[\frac{\partial \tilde{E}^{+}}{\partial \xi}+\frac{\partial \tilde{E}^{-}}{\partial \xi}+\frac{\partial \tilde{F}}{\partial \eta}-\tilde{H}-\frac{\partial}{\partial \eta}\left(R_1\frac{\partial \tilde{Q}_1}{\partial \eta}+R_2\frac{\partial \tilde{Q}_2}{\partial \eta}\right)\right]^{n+1}=0 \quad (3.34).$$

If we assume the flux vector $\tilde{E}$ is homogeneous, it can be split according to

$$\tilde{E}=\tilde{E}^{+}+\tilde{E}^{-}=A^{+}\tilde{Q}+A^{-}\tilde{Q}.$$

Matrices A^{+} and A^{-} are the split Jacobian matrices given by

$$A^{+}=M_\xi \Lambda_\xi^{+} M_\xi^{-1}, \qquad A^{-}=M_\xi \Lambda_\xi^{-} M_\xi^{-1}$$

in which,

$$\Lambda_\xi^{+}=(\Lambda_\xi+|\Lambda_\xi|)/2, \qquad \Lambda_\xi^{-}=(\Lambda_\xi-|\Lambda_\xi|)/2.$$

The quantities $\tilde{E}^{+}$ and $\tilde{E}^{-}$ have to be upwind differenced according to the signs of their individual eigenvalues. In supersonic regions, $\tilde{E}^{-}$ vanishes and Eq. (3.34) provides an algorithm which can be solved by a marching procedure in the ξ direction. The difficulty is that the supersonic region is coupled to the subsonic region where upstream influence exists. Consequently, both regions must be solved multiple times. The following discussion formulates numerical procedures for Eq. (3.34) based on the algorithms considered in Section 3.3.

3.4.1 The Standard ADI Procedure

The first procedure considered is the standard approximate factorization [8,9] of the upwind differenced system. In this procedure, the ξ and η directions are split, and the discretized version of Eq. (3.34) in delta form becomes,

$$\begin{aligned}&[I-\Delta tD+\Delta t(\frac{\partial}{\partial\xi}A_t^+ + \frac{\partial}{\partial\xi}A_t^-)](I-\Delta tD)^{-1}\\&[I-\Delta tD+\Delta t\frac{\partial}{\partial\eta}B-\Delta t\frac{\partial}{\partial\eta}(R_1\frac{\partial}{\partial\eta}B_{v1}+R_2\frac{\partial}{\partial\eta}B_{v2})]\Delta\tilde{Q}=-\Delta tR'\end{aligned} \tag{3.35}$$

where, the residual vector R' is,

$$R'=[\frac{\partial\tilde{E}^+}{\partial\xi}+\frac{\partial\tilde{E}^-}{\partial\xi}+\frac{\partial\tilde{F}}{\partial\eta}-\tilde{H}-\frac{\partial}{\partial\eta}(R_1\frac{\partial\tilde{Q}_1}{\partial\eta}+R_2\frac{\partial\tilde{Q}_2}{\partial\eta})]^n. \tag{3.36}$$

The Jacobian matrices A_t^+ and A_t^- are,

$$A_t^+=\frac{\partial\tilde{E}^+}{\partial\tilde{Q}},\qquad A_t^-=\frac{\partial\tilde{E}^-}{\partial\tilde{Q}}$$

where the subscripts t are used to distinguish the true Jacobians $A_t^\pm$ from the matrices $A^\pm$. The first operator in Eq. (3.35) is block penta-diagonal for second order upwind differencing and is block tri-diagonal for the first order scheme. The solution procedure of Eq. (3.35) is similar to that of the implicit ADI scheme and will not be repeated here.

3.4.2 The Diagonally Dominant ADI Procedure

The second approximate factorization procedure is the DDADI method. To express this algorithm, we must discretize the equation in ξ and t before factoring. The explicit discretization in η is not necessary to specify the algorithm and the derivatives in η imply central differencing. The basic philosophy of the DDADI procedure is to place as many terms as possible on the diagonal element before

splitting. This procedure can be expressed as

$$\{I - \Delta t\dot{D} - \tilde{\kappa}\frac{\Delta t}{\Delta\xi}A_t^- + \Delta t[\frac{\partial}{\partial\xi}A_t^+ + \frac{\partial}{\partial\eta}B - \frac{\partial}{\partial\eta}(R_1\frac{\partial}{\partial\eta}B_{v1} + R_2\frac{\partial}{\partial\eta}B_{v2})]\}$$
$$[I - \Delta tD + \tilde{\kappa}\frac{\Delta t}{\Delta\xi}(A_t^+ - A_t^-)]^{-1}$$
$$\{I - \Delta tD + \tilde{\kappa}\frac{\Delta t}{\Delta\xi}A_t^+ + \Delta t[\frac{\partial}{\partial\xi}A_t^- + \frac{\partial}{\partial\eta}B - \frac{\partial}{\partial\eta}(R_1\frac{\partial}{\partial\eta}B_{v1} + R_2\frac{\partial}{\partial\eta}B_{v2})]\}\Delta\tilde{Q}$$
$$= -\Delta tR' \tag{3.37}$$

in which, $\tilde{\kappa} = 1 + \kappa/2$. The quantity κ is zero for first order upwind differencing and is one for second order upwind differencing.

Similar to the derivations discussed in Section 3.3.3, the DDADI splitting will be solved by the line Gauss-Seidel relaxation method, which includes the forward marching,

$$\{D' + \Delta t[\frac{\partial}{\partial\eta}B - \frac{\partial}{\partial\eta}(R_1\frac{\partial}{\partial\eta}B_{v1} + R_2\frac{\partial}{\partial\eta}B_{v2})]\}\Delta\tilde{Q}^* =$$
$$-\Delta t\{\frac{(\tilde{E}_{i,j}^+)^n - (\tilde{E}_{i-1,j}^+)^*}{\Delta\xi} + \kappa\frac{(\tilde{E}_{i,j}^+)^n - 2(\tilde{E}_{i-1,j}^+)^* + (\tilde{E}_{i-2,j}^+)^*}{2\Delta\xi} \tag{3.38}$$
$$+ [\frac{\partial\tilde{E}^-}{\partial\xi} + \frac{\partial\tilde{F}}{\partial\eta} - \tilde{H} - \frac{\partial}{\partial\eta}(R_1\frac{\partial\tilde{Q}_1}{\partial\eta} + R_2\frac{\partial\tilde{Q}_2}{\partial\eta})]^n\}$$

and the symmetrical backward marching,

$$\{D' + \Delta t[\frac{\partial}{\partial\eta}B - \frac{\partial}{\partial\eta}(R_1\frac{\partial}{\partial\eta}B_{v1} + R_2\frac{\partial}{\partial\eta}B_{v2})]\}\Delta\tilde{Q} =$$
$$-\Delta t\{\frac{(\tilde{E}_{i+1,j}^-)^{n+1} - (\tilde{E}_{i,j}^-)^*}{\Delta\xi} - \kappa\frac{(\tilde{E}_{i,j}^-)^* - 2(\tilde{E}_{i+1,j}^-)^{n+1} + (\tilde{E}_{i+2,j}^-)^{n+1}}{2\Delta\xi}$$
$$+ [\frac{\partial\tilde{E}^+}{\partial\xi} + \frac{\partial\tilde{F}}{\partial\eta} - \tilde{H} - \frac{\partial}{\partial\eta}(R_1\frac{\partial\tilde{Q}_1}{\partial\eta} + R_2\frac{\partial\tilde{Q}_2}{\partial\eta})]^*\} \tag{3.39}$$

where, $\Delta\tilde{Q}^* = \tilde{Q}^* - \tilde{Q}^n$ and $\Delta\tilde{Q}^{n+1} = \tilde{Q}^{n+1} - \tilde{Q}^*$ are used to update variables immediately after each forward or backward sweep, respectively. The diagonal term D' is,

$$D' = I - \Delta tD + \tilde{\kappa}\frac{\Delta t}{\Delta\xi}(A_t^+ - A_t^-).$$

Both forward and backward marching operators involve the solution of a block tri-diagonal matrix. The combination of Eq. (3.38) and Eq. (3.39) completes one iteration step.

3.4.3 The PNS-ADI Procedure

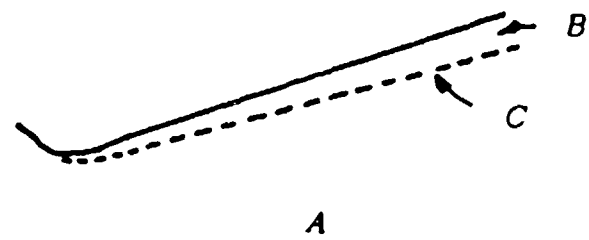
The third approximate factorization procedure considered is also a forward-backward scheme that is designed to give a PNS-like sweep in the forward direction and a partial backward sweep that is required only in those regions where the flow is subsonic. Because of its analogy with PNS algorithms, this procedure is referred to as a PNS-ADI scheme. This scheme can be described as,

$$\{I - \Delta t D + \Delta t[\frac{\partial}{\partial \xi} A_t^+ + \frac{\partial}{\partial \eta} B - \frac{\partial}{\partial \eta}(R_1 \frac{\partial}{\partial \eta} B_{v1} + R_2 \frac{\partial}{\partial \eta} B_{v2})]\}(I - \Delta t D)^{-1}$$
$$(I - \Delta t D + \Delta t \frac{\partial}{\partial \xi} A_t^-)\Delta \tilde{Q} = -\Delta t R' \tag{3.40}$$

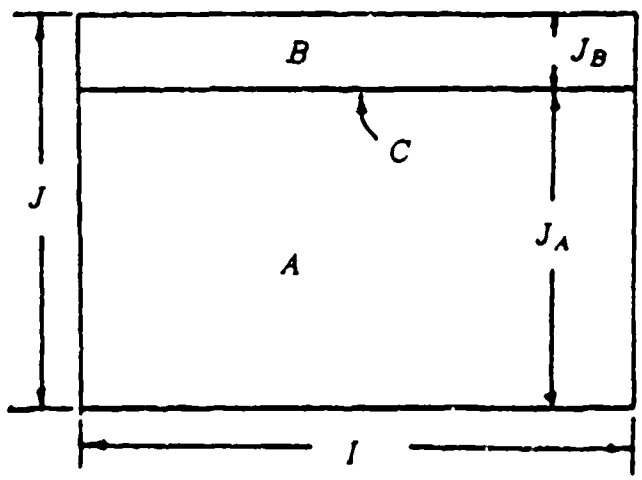
As will be shown later, the first operator is exactly the time-iterative version of a PNS procedure, as will be discussed in detail in chapter 4. Again, this operator is block tri-diagonal at each ξ location. The last operator reduces to the identity operator in supersonic regions where A_t^- vanishes, and is only block bi-diagonal in subsonic regions. Thus, in supersonic flows, the PNS-ADI procedure becomes a marching procedure, but in subsonic regions it retains the influence of the upstream acoustic waves. Because the backward operator is only necessary inside the subsonic layer, the computational time involved in one iteration for this PNS-ADI algorithm is less than that for the DDADI algorithm.

3.4.4 Direct Solution by Flowfield Partitioning

For a typical high Reynolds number supersonic flow, the flowfield can almost be solved by a pure marching procedure. Only the thin subsonic layer adjacent to the wall prevents this marching algorithm. This suggests a direct solver based on the



Physical Domain



Computational Domain

Grid Size $I \times J$

Figure 24. Partitioning of physical and computational domains

flowfield partitioning according to the distinct physical nature of this predominantly supersonic flow. As is given in Fig. 24, the flowfield is partitioned into three parts. Domain A is composed of all the points that are supersonic. Domain B includes all the points that are subsonic (although it may also include some supersonic points). Domain C is the interface line between domain A and domain B. It is chosen such that all points on the line (domain C) are supersonic. Let Q_A, Q_B, and Q_C denote all the dependent variables in these domains. These unknowns are related by

$$\begin{aligned} A_A Q_A \qquad + C_A Q_C &= R_A \\ B_B Q_B + C_B Q_C &= R_B \\ A_C Q_A + B_C Q_B + C_C Q_C &= R_C \end{aligned} \tag{3.41}$$

which indicates that the unknowns in domain A are coupled only to unknowns at domain C and are independent of unknowns at domain B. Similar statement holds for domain B, while domain C is coupled to both A and B.

Since all points are supersonic at domain A, the operator A_A can be efficiently solved by a marching procedure. The operator B_B is block tri-diagonal with block size equal to $4 \times J_B$, (J_B denotes the number of points in η direction inside domain B). This block tri-diagonal matrix is large, but the size decreases rapidly as the number of subsonic points decreases (as for thin subsonic layers). Both Q_A and Q_B can be solved easily if the unknowns Q_C are given. In fact, Q_C can be solved according to

$$[A_C A_A^{-1} C_A + B_C B_B^{-1} C_B - C_C] Q_C = A_C A_A^{-1} R_A + B_C B_B^{-1} R_B - R_C \tag{3.42}$$

which is obtained by the substitution of the first and the second equations in Eq. (3.41) into the last equation. The left hand side operator of Eq. (3.42) is a dense matrix with the size equal to $4 \times I$, with I representing the number of points in the ξ direction for the entire computational domain. This matrix can be efficiently solved by an iterative method in 4 to 6 iterations, because the influence of

downstream points on upstream points is weak if all points along domain C are supersonic.' This partitioning technique allows a direct solution procedure to be obtained. The direct solution without any approximate factorization implies the CFL number can be as large as possible, thus rapid convergence can be obtained. However, the computational time involved in one iteration for this direct method is much more than those for approximate procedures if too many subsonic points are involved.

3.5 Algorithm Comparisons

Based on the physics of supersonic flows, four different upwind algorithms associated with their stability analyses are discussed in the last two sections. However, the computational efficiency of these numerical procedures is yet to be identified. Before comparisons to be made, two concepts must be paid attention to. First, we note that all four methods described above are concerned with using different algorithms to solve the same equations. The residuals on the right hand side of Eq. (3.35), Eq. (3.37), Eq. (3.40), and Eq. (3.41) are identical in terms of both partial differential equations and finite difference representations. As $\Delta\tilde{Q}$ is driven to zero, all four methods provide the same steady state solution as given by Eq. (3.36). A check of the converged solutions from the computer codes verifies that these solutions are identical to six or seven digits. Second, since all methods give the same solutions, the only thing to compare is the path a specific algorithm takes to steady state. Therefore, it is of interest to compare both the number of iterations required for $\Delta\tilde{Q}$ to reach the machine accuracy and the total CPU time required. The number of iterations required shows the numerical efficiency of each algorithm, while the CPU time required indicates the cost per iteration step. The latter is affected by the arithmetic operations involved in an algorithm and the computer architecture. Present results are for the scalar machine, IBM3090-180. Slightly different results

may be obtained for a vector machine.

The test problem for the comparison follows is the laminar supersonic flow through a 15° conical nozzle with an expansion ratio of 30. The nozzle geometry and the 21 (axial) by 40 (radial) grid are shown on Fig. 25. At the starting plane, a constant Mach number flow ($M = 1.02$) with zero contravariant velocity V was chosen. The nozzle Reynolds number based on the throat radius is 10^5. The converged solution showed that with this highly stretched grid, the subsonic portion of the boundary layer had grown to nine points at the exit plane. All calculations are done without artificial viscosity for the upwind-central differencing formulations. Boundary conditions are implemented by the procedures given in Section 3.2.1.

Convergence rates based on the number of iterations for the four algorithms mentioned above are shown in Fig. 26. The linear convergence on these semi-logarithmic plots until machine accuracy is reached gives indication that the codes are error-free. All calculations started with initial conditions which were obtained from the single forward sweep through the flowfield with the PNS algorithm (The details of PNS algorithms will be given in chapter 4). The result shown for each algorithm corresponds to the optimum CFL for this scheme. These optimums are shown in the figure. Boundary conditions are implemented by the implicit wall boundary procedure. In terms of number of iterations required the ADI scheme is seen to be the slowest. It also has the lowest optimum CFL of 5. The DDADI scheme is the fastest of the three approximate methods and converges almost as rapidly as the direct method. The DDADI algorithm converged most rapidly with CFL at 5000, above this value, convergence becomes independent of CFL.

Figure 26 also shows that the PNS-ADI algorithm gives excellent convergence (it converges to machine accuracy in 70 iterations). This rapid convergence was not properly predicted by the stability analysis based on the scalar equation. The

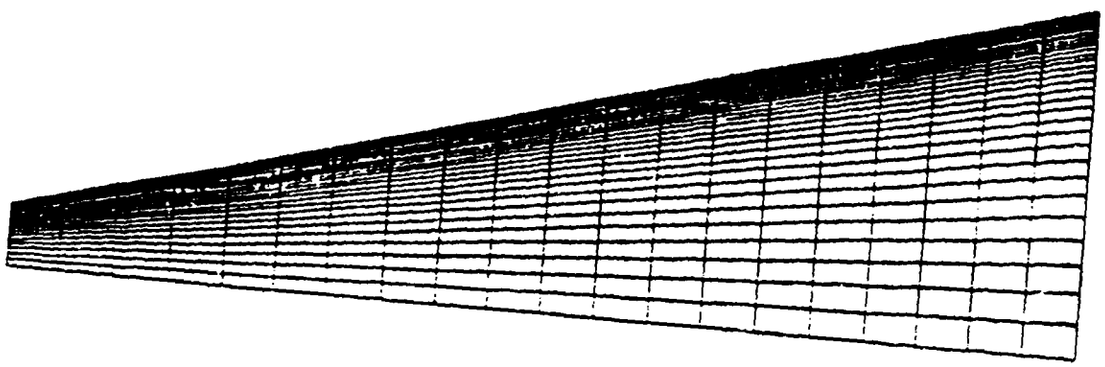


Figure 25. 15° conical nozzle for convergence comparison, grid size 21 × 40, Area ratio 30

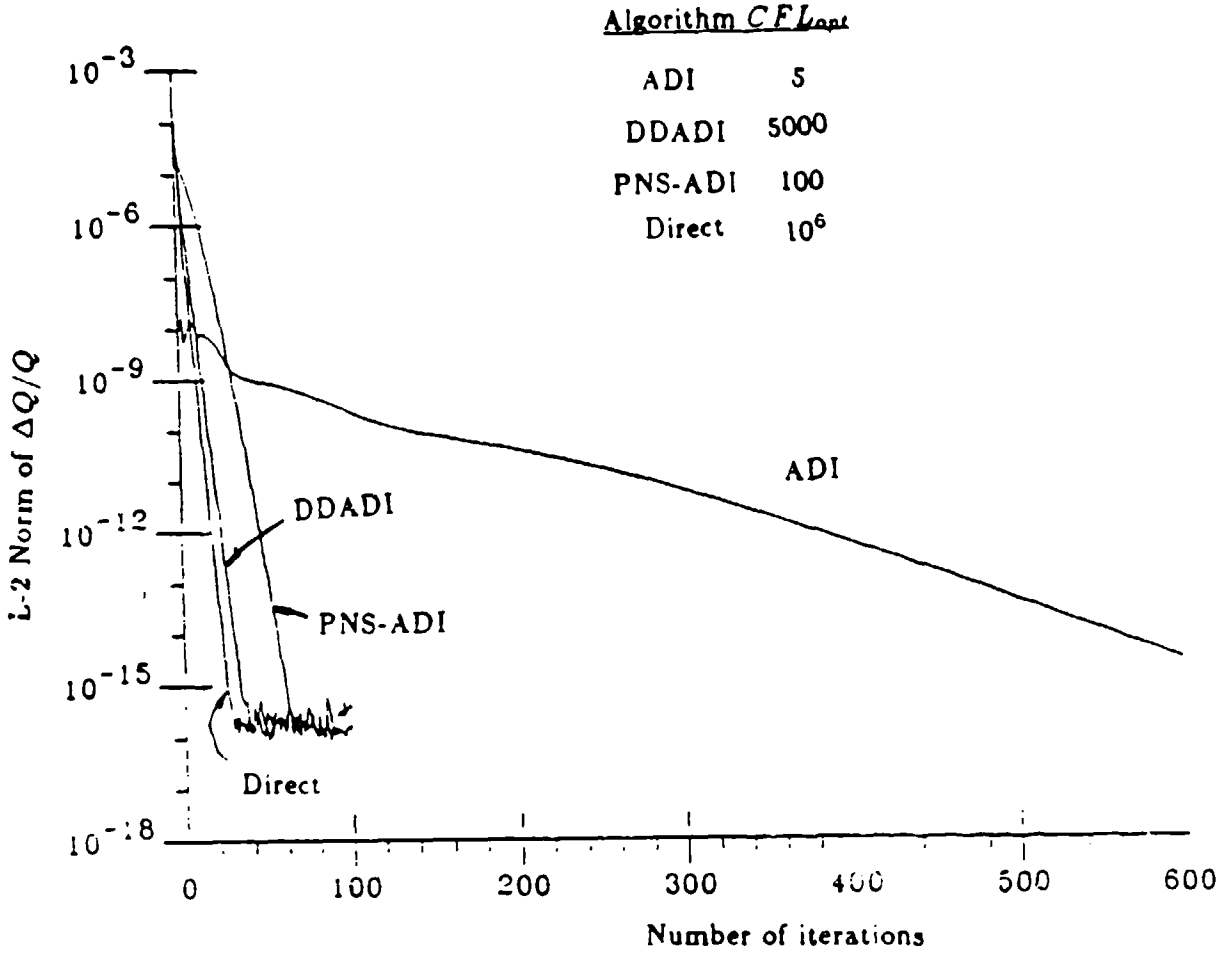


Figure 26. Convergence in terms of number of iteration of 2-D supersonic algorithms with true Jacobians

optimum CFL for the PNS-ADI procedure is around 100, beyond that, convergence rates tend to slow down. The optimum CFL for the direct solver was about 10^6 and the convergence remained the same for CFL numbers up to 10^{10}. The convergence rate for the direct procedure was somewhat disappointing in that it required 25 iterations to reach the machine accuracy. With the fully implicit scheme, the numerical procedure can normally converge to machine accuracy in $8-11$ iterations [42]. This normal convergence was not obtained with the current analysis code.

Comparisons of convergence rates in terms of CPU time are shown in Fig. 27. The ADI method is seen to require the most time to reach machine accuracy. The direct method is just slightly faster than the ADI method; however, if we could attain a factor of three improvement expected above, it would be competitive with the fastest procedure. The PNS-ADI and the DDADI schemes are quite competitive and are about a factor of ten faster than the standard ADI scheme. Although, in terms of number of iterations, the PNS-ADI method is slower than the DDADI method, the deficit is seen to be offset by less computational work involved in one iteration for the PNS-ADI procedure.

As demonstrated in Chapter 2, the use of the true Jacobian has significant effects on convergence. The same comparison was made for all four methods by using approximate Jacobians A^+ instead of A_i^+ on the left-hand side of corresponding discretized equations. The convergence rates in terms of both number of iterations and CPU time are shown in Fig. 28 and Fig. 29. All procedures except the ADI method slow down substantially due to the approximation in Jacobians. The convergence rate (both in terms of iterations and CPU time) for the ADI method actually improves slightly. This shows that the convergence rate for the ADI scheme is insensitive to the Jacobian matrices. Note also that corresponding optimum CFL numbers for three approximate methods change if approximate Jacobians are used.

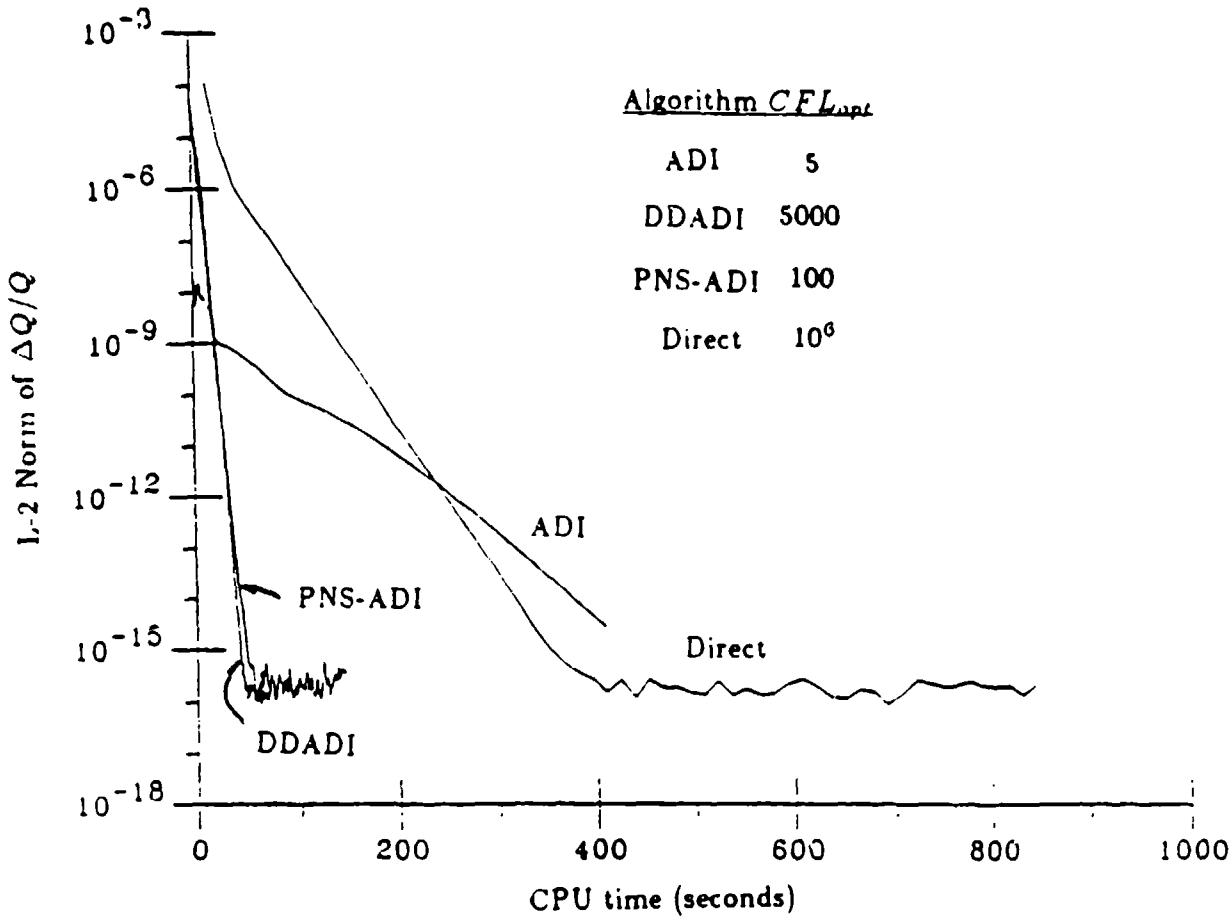


Figure 27. Convergence in terms of CPU time of 2-D supersonic algorithms with true Jacobians

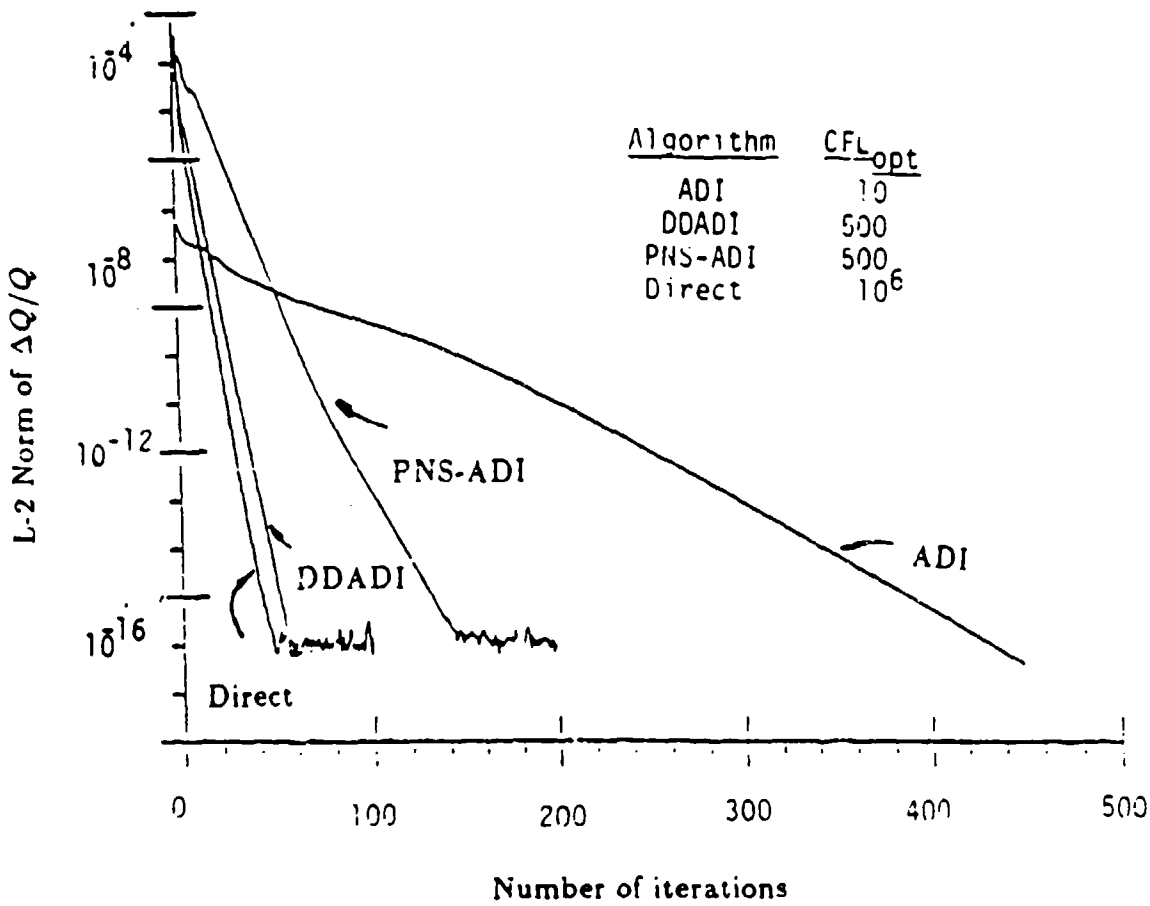


Figure 28. Convergence in terms of number of iteration of 2-D supersonic algorithms with approximate Jacobians

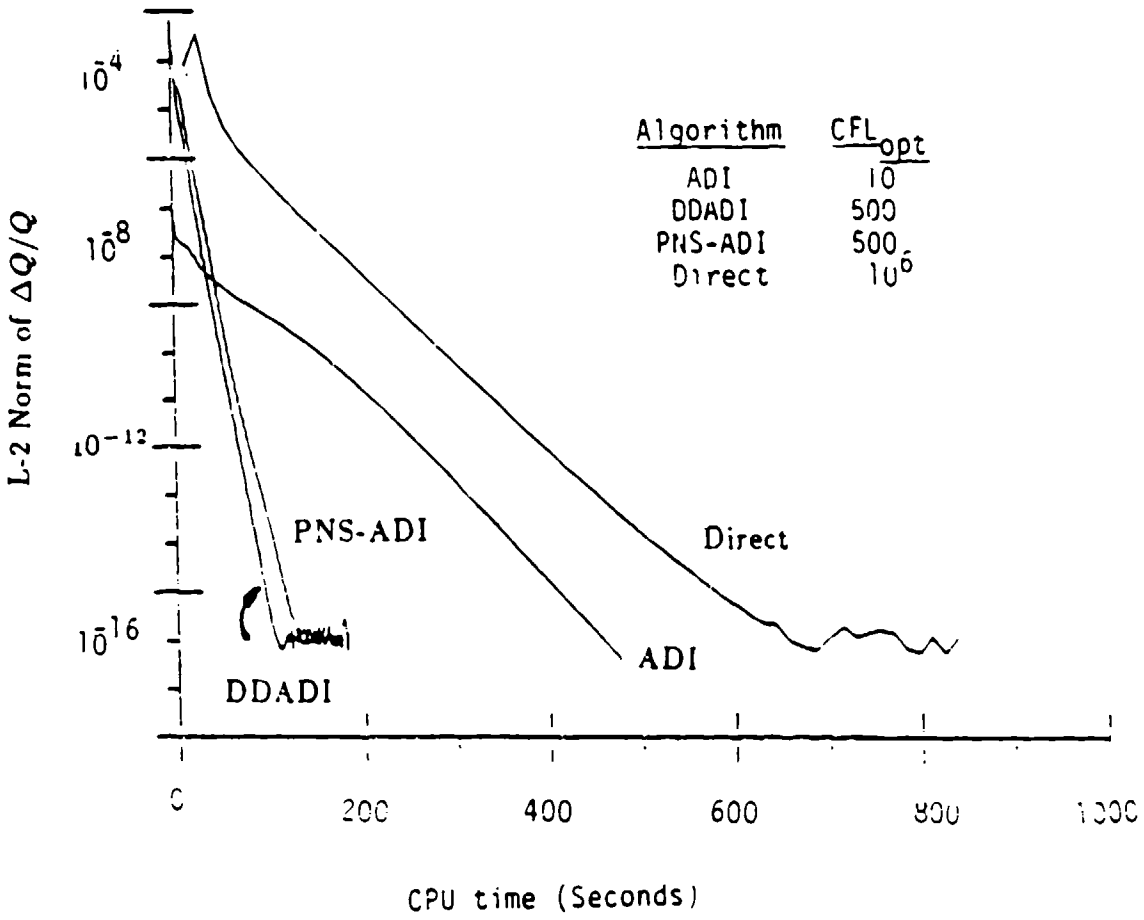


Figure 29. Convergence in terms of CPU time of 2-D supersonic algorithms with approximate Jacobians

3.6 Nozzle Flowfield Predictions

To demonstrate the capabilities of present upwind-central difference algorithms, the flowfields in two nozzle geometries were computed. The first geometry was the 15° conical nozzle shown on Fig. 25 except here a 21 × 70 grid was used. The second nozzle was the contoured geometry shown in Fig. 30, with a grid of 75 × 50. The area ratio of this nozzle was 272 : 1. In both nozzles, a throat radius of 10 mm. was chosen.

The properties of a typical rocket nozzle combustion gas were used for all calculations, including $\gamma = 1.24$ and $C_p = 3043 J/Kg°K$. Both nozzles were run at the stagnation pressures of 35 and 3.5 atm., corresponding to nozzle throat Reynolds numbers of 1.4×10^4 and 1.4×10^5. The stagnation temperature was chosen to be $3500°K$. The results for the conical nozzle were for laminar flows with the molecular viscosity varying according to the Sutherland law,

$$\frac{\mu}{\mu_r} = \left(\frac{T}{T_r}\right)^{3/2} \frac{T_r + S}{T + S}$$

where, μ_r is the reference viscosity at a reference temperature of T_r and S is the Sutherland constant. Those results for the contoured nozzle were calculated for both laminar and turbulent flows. For turbulent calculations, the algebraic model by Baldwin and Lomax [12] was used.

3.6.1 Verification of Solution Accuracy

To verify the accuracy of present upwind-central differencing algorithms, the results obtained from present analysis codes are compared to those from the MOC procedure [4]. Figure 31 plots Mach number contours for the inviscid supersonic flow through the high-expansion contoured nozzle. The upper half shows results from the MOC procedure and the lower half presents those from present algorithms

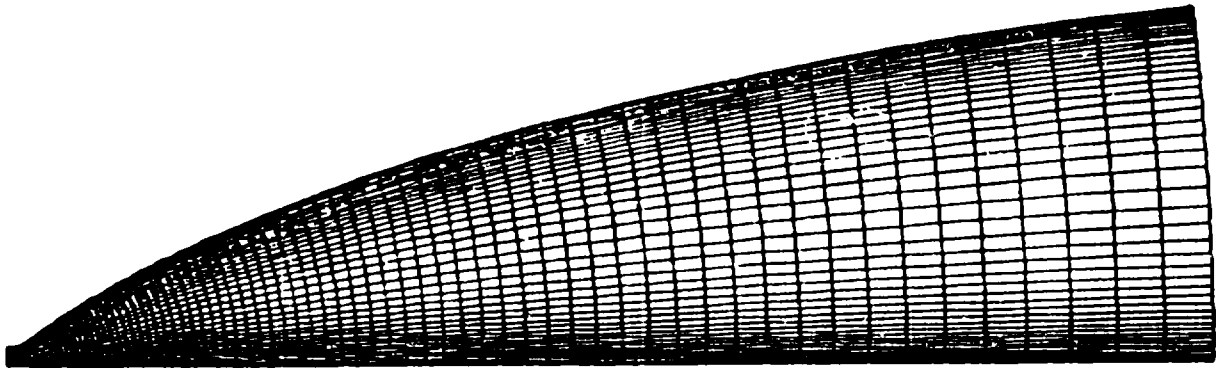


Figure 30. Contoured nozzle geometry with 75 × 50 grid

As shown here, the Mach number contours computed by using current upwind-central differencing schemes agree very well with the MOC predictions, except for the discrepancies near the centerline, where the MOC procedure fails to resolve the symmetry conditions due to the presence of a weak oblique shock.

Figure 32 shows the comparison between current algorithms and the MOC procedure by plotting the computed wall pressure distribution along the axial direction. Again, the inviscid results from current algorithms show excellent agreements with those from the MOC procedure. The corresponding wall pressure distribution for viscous calculations with a Reynolds number of 1.4×10^6 is also shown on the figure. The viscous wall pressure is slightly higher than that of inviscid results. This illustrates that viscous flows expand less due to the presence of the boundary layer.

3.6.2 Effects of Downstream Boundary Conditions

All numerical algorithms mentioned above require a downstream boundary condition at the subsonic part inside the boundary layer, since the eigenvalue $U - C_\xi$ is negative if $U < C_\xi$. Previous researchers have usually implemented this boundary condition by extrapolating from inside the computational domain. Although this does simplify the numerical procedures at the downstream boundary, it violates the physical conditions, especially when the boundary layer is thick. Further, an extrapolated boundary condition does not allow flowfields to respond to downstream pressure changes as they must in physical situations. In the case of nozzle flows, a high back pressure will cause the boundary layer to thicken and the flow to separate.

One case of interest is when an exhaust nozzle is operated in an altitude facility where the ambient pressure in the facility is only approximately matched to the nozzle expansion characteristics. This mismatch can provide significant differences between test stand performance and eventual performance in space because of the effect of the ambient pressure on the nozzle boundary layer at the exit. An analo-

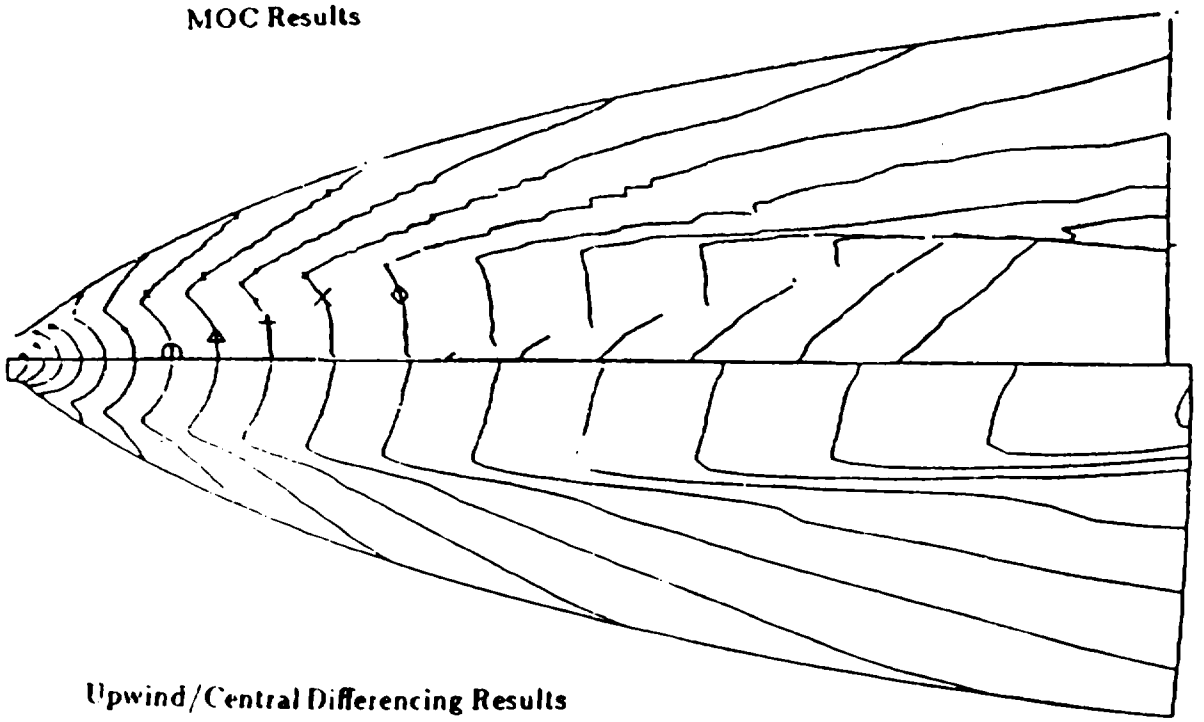


Figure 31. Comparison of Mach number contours for MOC and upwind/central difference algorithm

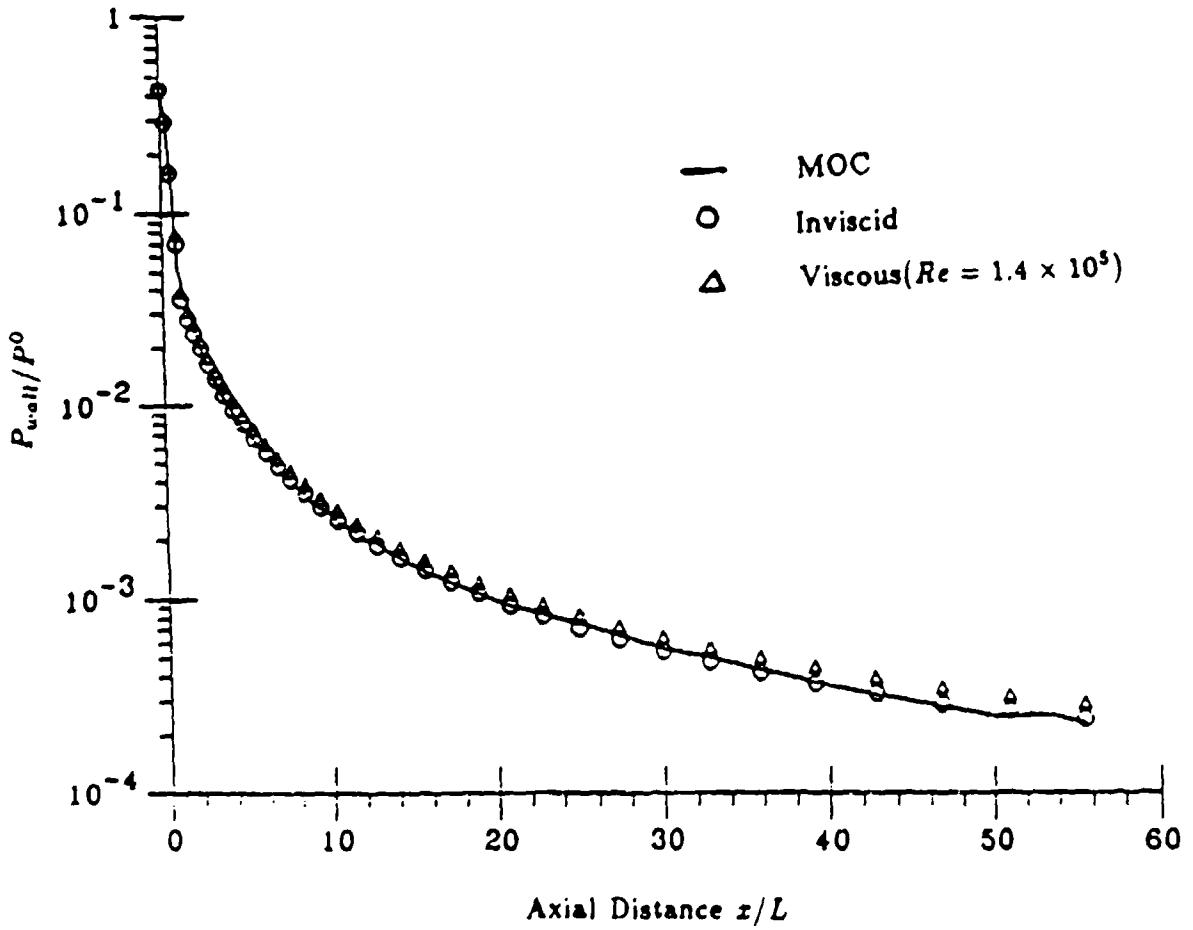


Figure 32. Comparison of wall pressure distribution for MOC and upwind/central difference algorithm

gous but much more severe result of this mismatch is that flow separation occurs inside the thick boundary layer of an altitude nozzle when it is operated at sea-level conditions. The examples shown later will refer only to the case of small back pressure mismatches and will not extend to the massive separations that are observed at sea level conditions. Calculations of these effects require that the downstream boundary conditions be correctly implemented.

The downstream boundary conditions for the present investigation are based on MOC procedures. At the early stages of iteration, a certain number of points at the exit plane will become subsonic due to viscous diffusion. The back pressure is then specified for these subsonic points on the exit plane. The procedure used is similar to that in Eq. (2.15). As the iteration proceeds, reverse flow will appear inside the subsonic layer if the back pressure is high enough. For these reentry flows, three eigenvalues (U, U, and $U - C_\xi$) become negative(assuming the reentry velocity is less than the sonic speed) and standard inflow boundary conditions are applied. This implies three conditions must be specified from outside the domain while one characteristic equation must be used for information coming from inside the domain. These three specified quantities are chosen as the stagnation pressure, the stagnation temperature, and the flow angle, corresponding to the ambient conditions. Since the external surroundings are interpreted as being at rest, the stagnation pressure is taken as the back pressure. For those subsonic points with positive (outflow) streamwise velocities, only the back pressure is specified and three equations are used in agreement with traditional outflow boundary procedures. The decision as to whether inflow or outflow boundary conditions are used depends on the signs of eigenvalues at each grid point as determined from the previous iteration. The above procedure for reentry flows is reliable for modestly sized recirculation regions but eventually breaks down for large recirculation regions because oscillations between

inflow and outflow occur.

Results of a series of calculations in which the pressure at the subsonic part of the outflow boundary was specified are shown on Figs. 33-38. Figure 33 shows Mach number contours for the conical nozzle at the lower Reynolds number. The top plot shows the results by using extrapolation boundary conditions. This boundary condition resulted in a back pressure to stagnation pressure ratio P_b/P^0 of about 2.5×10^{-3}. The high Mach number gradient near the wall gives an indication of the boundary layer thickness. The middle plot shows the effect of raising the back pressure to $P_b/P^0 = 5 \times 10^{-3}$. At this back pressure, the boundary layer is thicker and a small recirculation zone presents near the exit. As the back pressure is increased further, this recirculation region continues to grow and to propagate upstream as indicated by the bottom plot where $P_b/P^0 = 7 \times 10^{-3}$. Corresponding calculations with back pressures below 2.5×10^{-3} showed that the boundary layer at the exit accelerated and became thinner. The wall temperatures for these calculations are $3000°K$.

The dotted lines in Fig. 33 represent the Mach number contours from one-dimensional inviscid calculations corresponding to the same stagnation conditions. As is seen, the presence of the boundary layer in the two-dimensional calculations has resulted in less expansion and thus slower speeds near the exit.

As a further indication of the character of these flows, the axial velocity (u) profiles at the exit plane for various back pressure levels are shown on Fig. 34. This figure shows how the boundary layer grows as the back pressure is increased and the rate of thinning of the boundary layer as the back pressure is decreased. Furthermore, the width of the recirculation zone at the exit plane is clearly shown.

The results for turbulent flow in the contoured nozzle at the higher Reynolds number ($P^0 = 35atm$) are shown in Fig. 35. Again, the top plot shows Mach

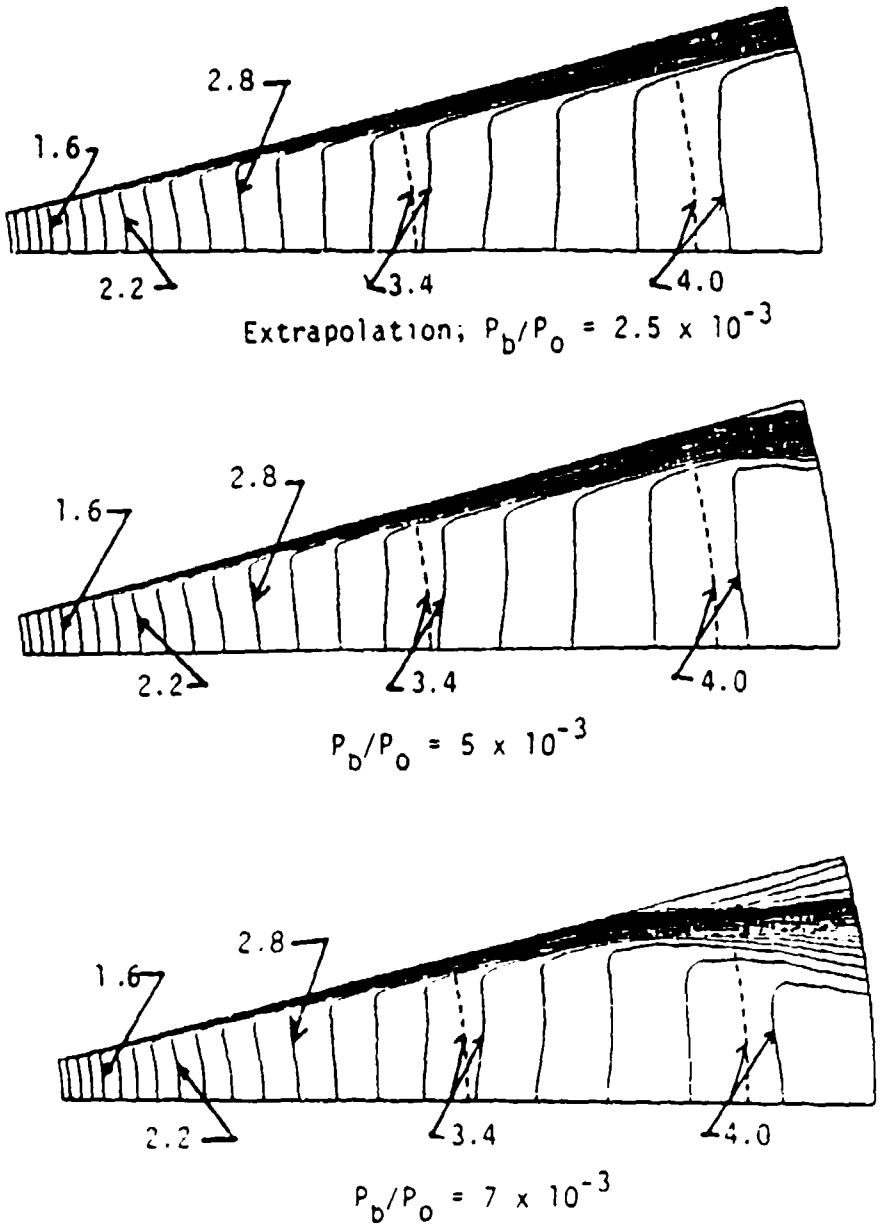


Figure 33. Converged Mach number contours for conical nozzle using various back pressures.

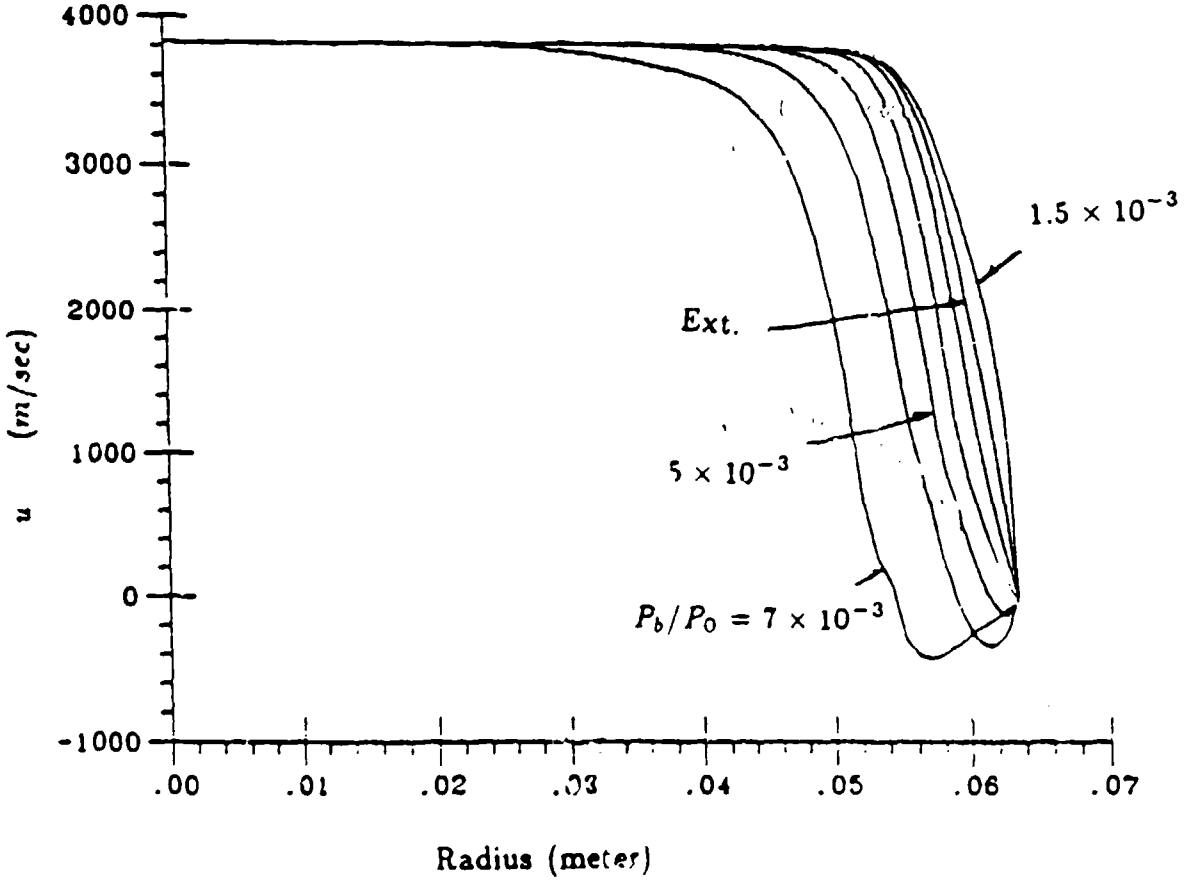
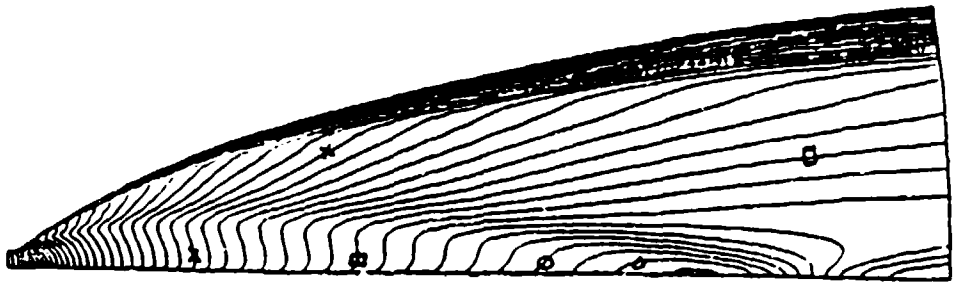
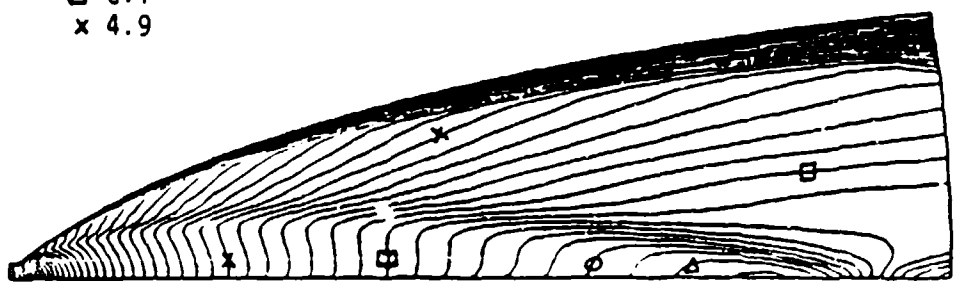


Figure 34. Velocity profiles at nozzle exit plane for conical nozzle using various back pressures

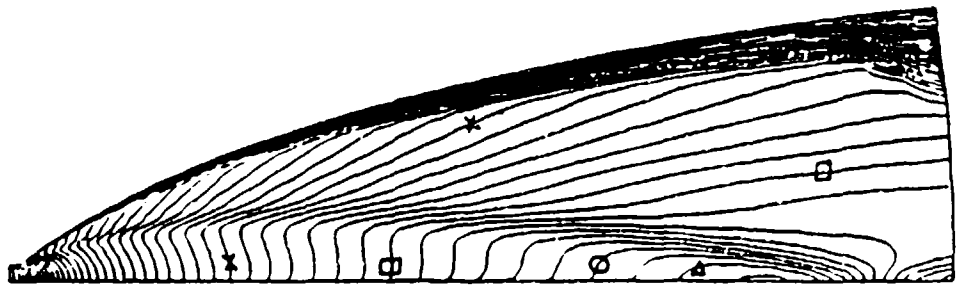


Extrapolation = $P_b/P_o = 0.33 \times 10^{-3}$

Δ 7.3
○ 7.0
□ 6.1
× 4.9

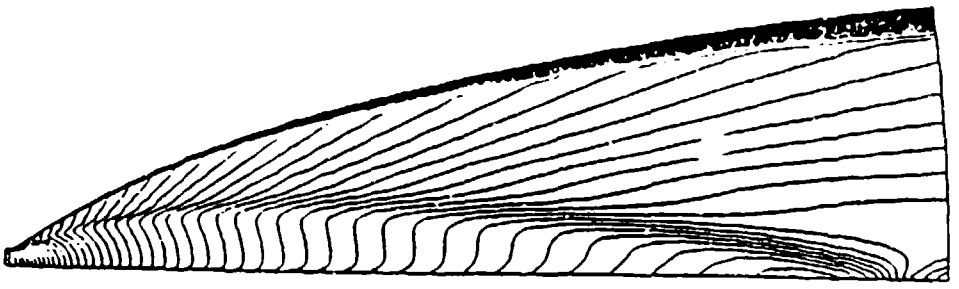


$P_b/P_o = 0.7 \times 10^{-3}$

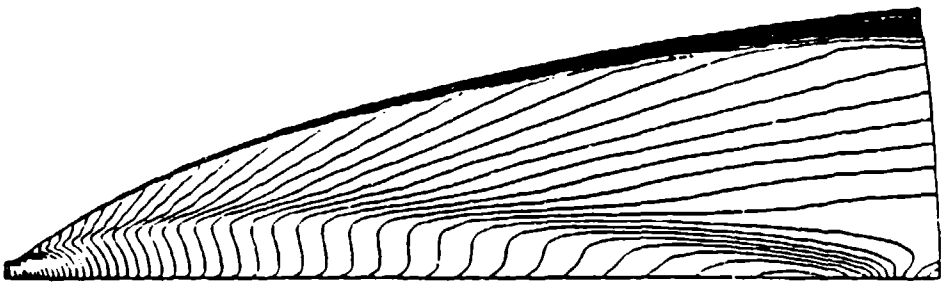


$P_b/P_o = 1.1 \times 10^{-3}$

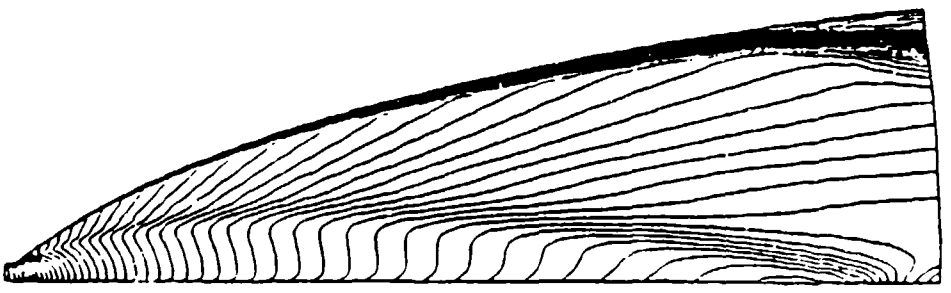
Figure 35. Turbulent Mach number contours in 272.1 contoured nozzle using various back pressure



Extrapolation, $P_b/P^0 = 0.29 \times 10^{-3}$



$P_b/P^0 = 0.48 \times 10^{-3}$



$P_b/P^0 = 0.7 \times 10^{-3}$

Figure 36. Laminar Mach number contours in 272:1 contoured nozzle using various back pressure

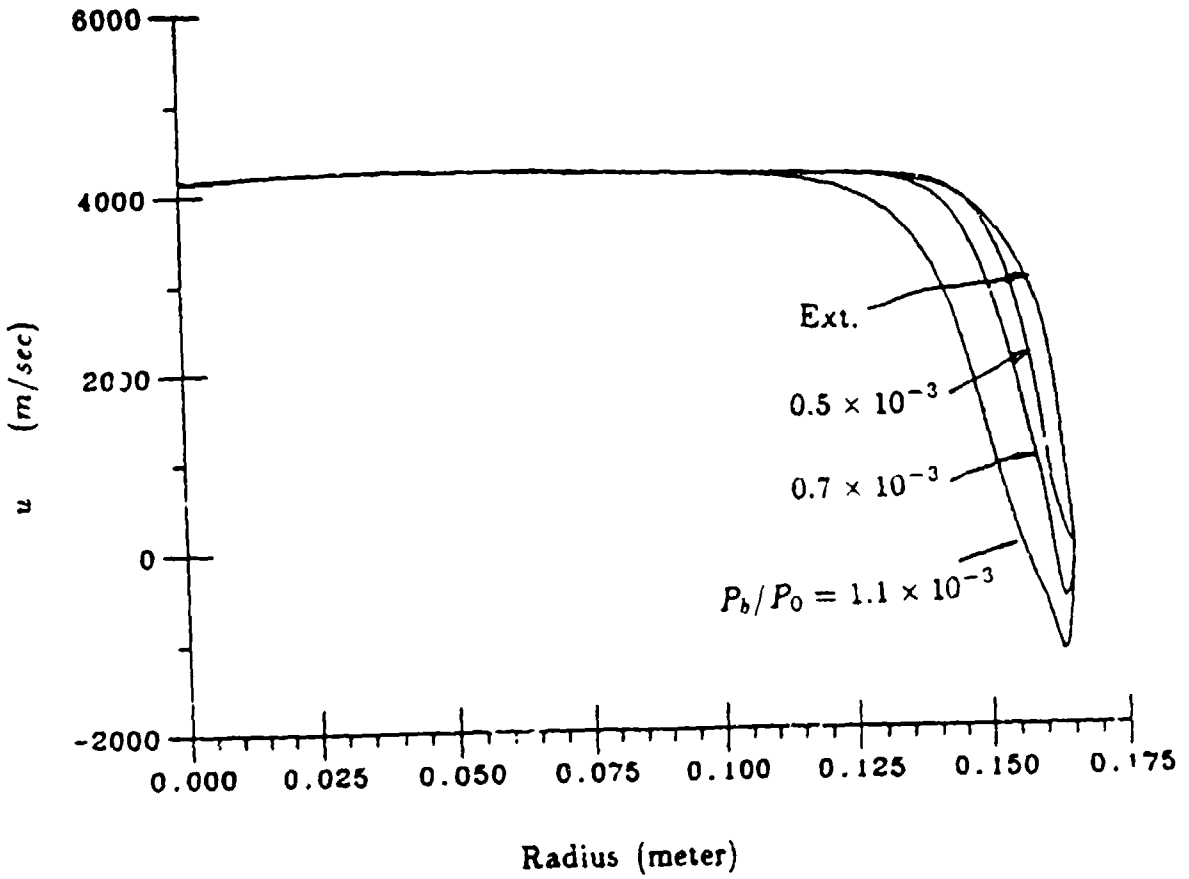


Figure 37. Velocity profiles at exit plane for turbulent flow in 272:1 contoured nozzle

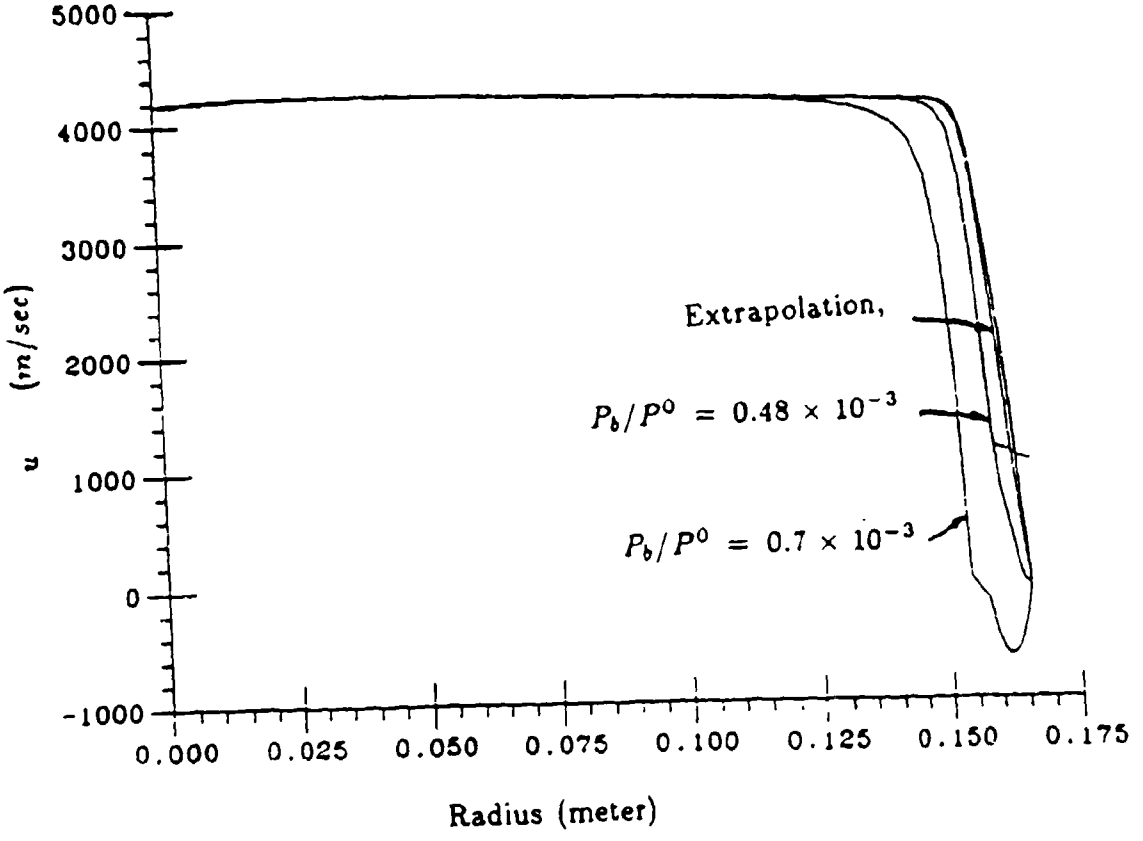


Figure 38. Velocity profiles at exit plane for laminar flow in 272:1 contoured nozzle

number contours for the extrapolated boundary condition case. The middle and the bottom plots show the results of higher back pressures. The extrapolation case here corresponds to the back pressure ratio P_b/P^0 of 0.33×10^{-3}. The middle plot is for a back pressure of 0.7×10^{-3} while the lower one is for 1.1×10^{-3}. Similar thickening and separation of the boundary layer is observed.

Because of the interchange of momentum inside the boundary layer for turbulent flows, a larger increase in P_b/P^0 is required to obtain the the same degree of separation, as is verified by comparing the laminar flow results shown on Fig. 36. For this laminar calculation, the extrapolated boundary condition corresponds to a lower back pressure ($P_b/P^0 = 0.29 \times 10^{-3}$) than to the turbulent case, because the much thinner boundary layer allows more expansion to be accomplished in the nozzle. Also, the laminar calculations show a larger separation region at $P_b/P^0 = 0.7 \times 10^{-3}$ than the turbulent boundary layer at $P_b/P^0 = 1.1 \times 10^{-3}$. The u velocity profile at the exit plane for laminar and turbulent boundary layers are shown on Fig. 37 and Fig. 38, respectively. The turbulent case has a thicker boundary layer and steeper gradients at the wall than the laminar calculation.

3.6.3 Effects of Reynolds Number and Wall Temperature

Although calculations with two different Reynolds numbers have been shown in the last section, some additional results of changing Reynolds numbers are demonstrated here along with comparisons of the effect of wall temperature. Figure 39 shows Mach number contours for the conical nozzle at a Reynolds number of 1.4×10^4 but with a lower wall temperature of $300°K$. The upper plot is for extrapolation boundary conditions, and the lower one is for a back pressure of 7×10^{-3}, analogous to the bottom plot of Fig. 33. These results show that colder wall temperatures give a much thinner boundary layer. As a consequence, a smaller separation region is observed as compared to the hot wall boundary layer.

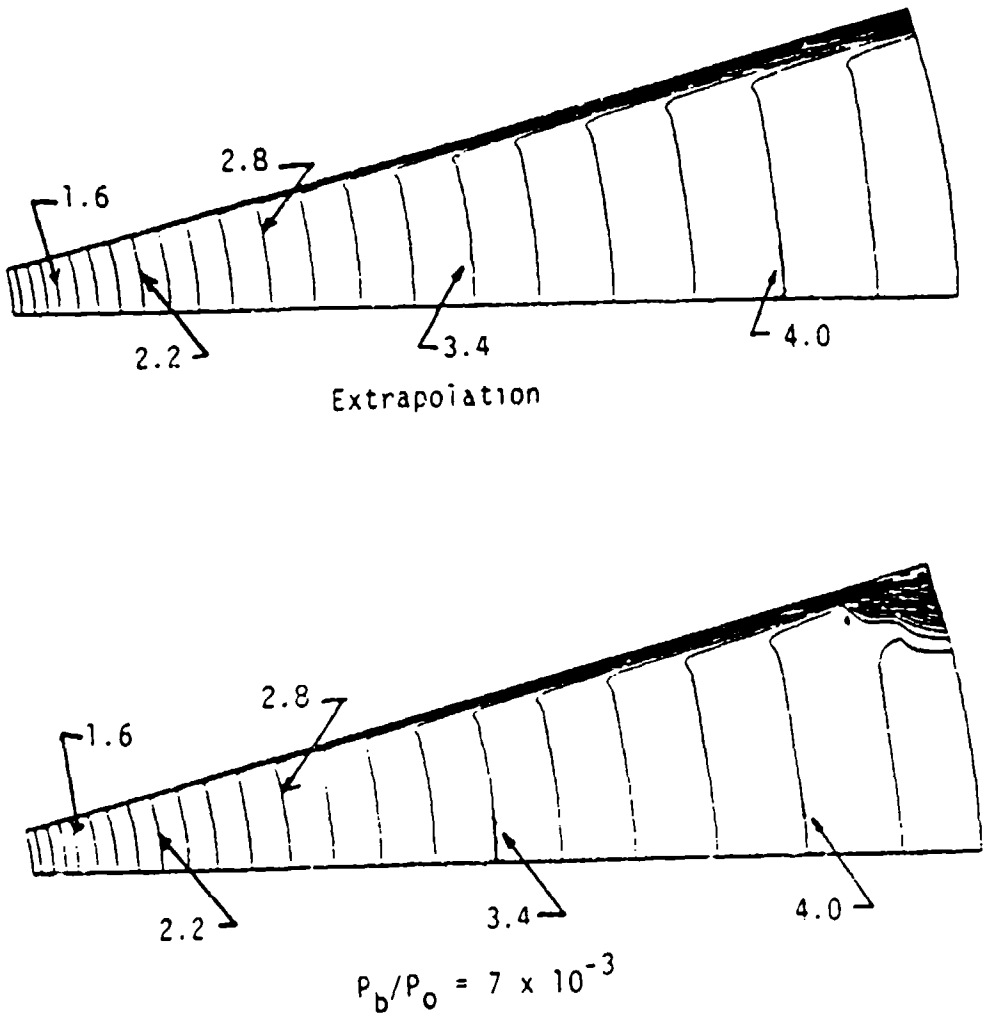


Figure 39. Converged Mach number contours for conical nozzle at lower Reynolds number and colder wall temperature

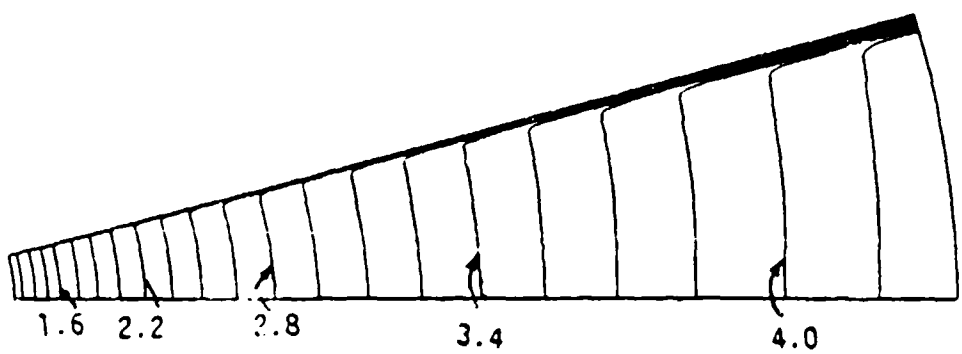


Figure 40. Converged Mach number contours for high Reynolds number Extrapolated boundary conditions

The effects of Reynolds numbers on the conical nozzle results can be seen by comparing the Mach number contours for a stagnation pressure of 35 atm in Fig. 40 with those for the lower Reynolds number case with $P^0 = 3.5$ atm on Fig. 33. Again, a thinner boundary layer is obtained for the higher Reynolds number case. Only the results based on the extrapolated boundary conditions are given in Fig. 40.

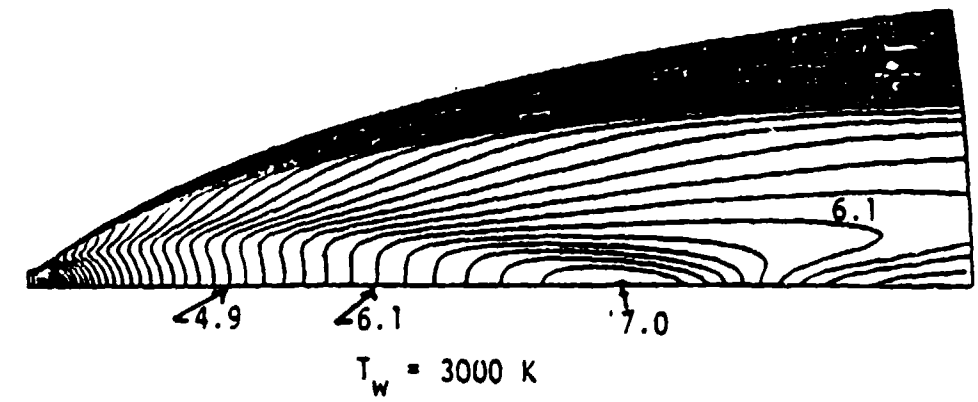
As a final comparison, results for the contoured nozzle are shown on Fig. 41 for the low Reynolds number case (3.5 atm) at two different wall temperatures, $3000°K$ and $300°K$, based upon turbulent boundary layer assumptions. As for the laminar case, the colder wall temperature results in a much thinner boundary layer.

The effects of Reynolds number can be seen by comparing the results in Fig. 41 with those for the turbulent case in Fig. 35. Again, a lower Reynolds number causes a dramatic increase in the boundary layer thickness. At this low Reynolds number, a check of the maximum eddy viscosity in the boundary layer profile reveals that the flow is no longer "turbulent" although the turbulence model is still included.

3.6.4 Effects of Coupled Wall Cooling and Nozzle Flows

As we have seen in previous sections, the wall temperature has a significant effect on the nozzle flowfield. While specified temperature or heat flux boundary conditions are frequently imposed at wall boundaries. In viscous problems, rocket nozzle walls are in general regeneratively cooled by propellant flowing inside the wall. This poses a problem when neither the heat flux nor the wall temperature are known a priori but both must be completed as part of the nozzle flowfield solution. In the present section, we develop a coupled method for solving the wall cooling flow along with the nozzle flow.

Figure 42 shows the schematic of a nozzle surrounded with cooling tubes. The coolant is assume to flow from the exit toward the inlet. To simplify the problem, the following assumptions are made:



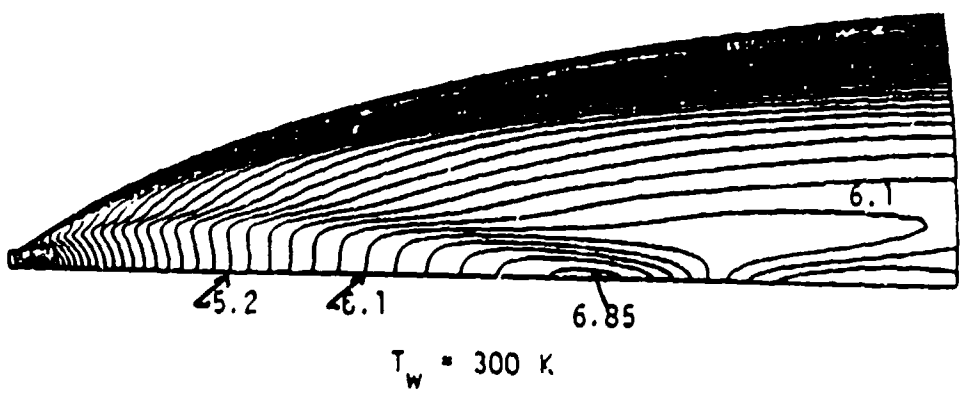
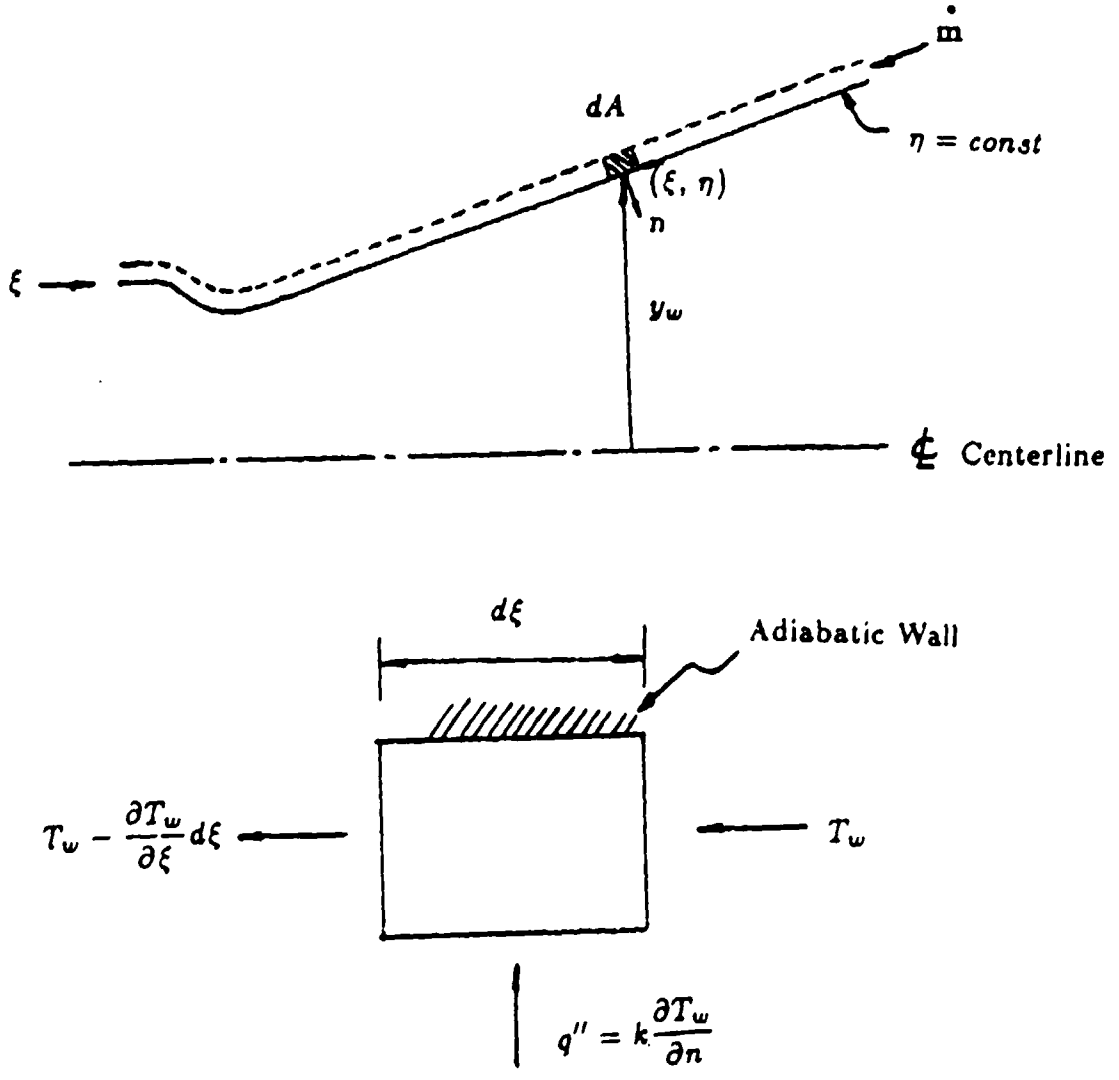


Figure 41. Converged Mach number contours for 272:1 nozzle at lower Reynolds number



control volume at (ξ, η)

Figure 42. Schematic of nozzle with wall cooling

1. The outer wall of the cooling tube is insulated.
2. The wall between the coolant the nozzle flow is thin and of high conductivity so that the temperature of the cooling liquid can be taken equal to that of the wall(T_w).
3. The cooling liquid has a constant specific heat (C_l).

Let y_w and n denote the radius and the inward normal direction at an arbitrary wall location (ξ, η). Referring to Fig. 42, the energy balance of the control volume at a wall location gives

$$\dot{m}C_l T_w + q'' dA = \dot{m}C_l(T_w - \frac{\partial T_w}{\partial \xi} d\xi)$$

where dA represent the surface area wetted by the cooling liquid. If the nozzle wall is completely surrounded by cooling tubes, the quantity dA can be expressed in the general coordinate system as

$$dA = \frac{2\pi y_w}{J}\sqrt{\eta_x^2 + \eta_y^2}\, d\xi.$$

By the Fourier's law and geometrical relations, the heat flux q'' is

$$q'' = k\frac{\partial T_w}{\partial n} = \frac{-k}{\sqrt{\eta_x^2 + \eta_y^2}}\left|(\xi_x\eta_x + \xi_y\eta_y)\frac{\partial T_w}{\partial \xi} + (\eta_x^2 + \eta_y^2)\frac{\partial T_w}{\partial \eta}\right|$$

in which, k is the thermal conductivity of the gas. Substituting expressions for q'' and dA into the energy balance equation, we readily obtain

$$\frac{1}{J}\left|(\xi_x\eta_x + \xi_y\eta_y)\frac{\partial T_w}{\partial \xi} + (\eta_x^2 + \eta_y^2)\frac{\partial T_w}{\partial \eta}\right| = \frac{Bi}{\tilde{y}}\frac{\partial T_w}{\partial \xi} \qquad (3.43)$$

where $\tilde{y}$ is the normalized wall radius ($\tilde{y} = y_w/y_t$) and Bi is the non-dimensional Biot number defined by

$$Bi = \frac{\dot{m}C_l}{2\pi k y_t}.$$

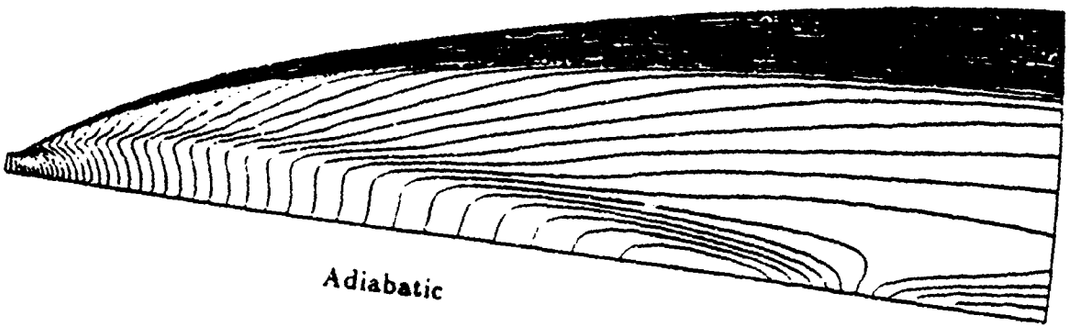
The non-dimensional Biot number represents the ratio of wall cooling to heat conduction in the gas.

The new thermal boundary condition, Eq. (3.43), can be coupled in an implicit manner to the discretized governing equations provided that the temperature at the downstream end of the wall (the inlet temperature of the cooling liquid) is given. The numerical procedure is based on the DDADI scheme. The wall temperature at the exit is fixed at the given coolant inlet temperature. In the discretized equation of Eq. (3.43), the derivative $\partial T_w/\partial \xi$ is backward differenced since the coolant flows from downstream to upstream. The derivative $\partial T_w/\partial \eta$ is one-sided differenced and is coupled to the unknowns of interior points. The implicit treatment of this discretized boundary equation is similar to the boundary procedure discussed in Section 3.2.1.

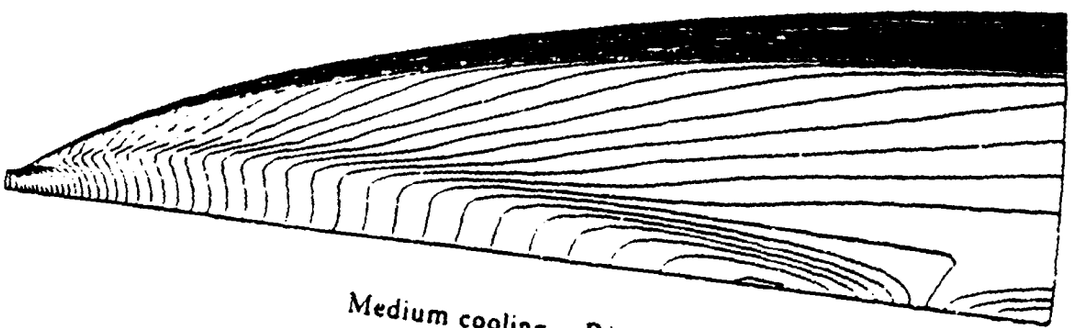
Typical Mach number contours for supersonic flows through the high expansion ratio nozzle given in Fig. 30 by using this wall cooling boundary condition are shown in Fig. 43. The top plot is for adiabatic wall conditions, the middle and the bottom plots are for $Bi = 10^3$ and $Bi = 10^5$, respectively. The inlet temperature of the coolant is $500^0 K$ for the last two cases. The Reynolds number is 1.4×10^4 and the flow is assumed laminar for all cases. Dramatic changes in the boundary layer thickness and the flowfield near the exit when this more appropriate cooling condition is incorporated can be observed. The results shown are for demonstration only, for practical applications, the parameter Bi should be calculated according to the true wetted area of cooling tubes and real properties of the coolant.

3.0.5 Nozzles with Subsonic Inflow

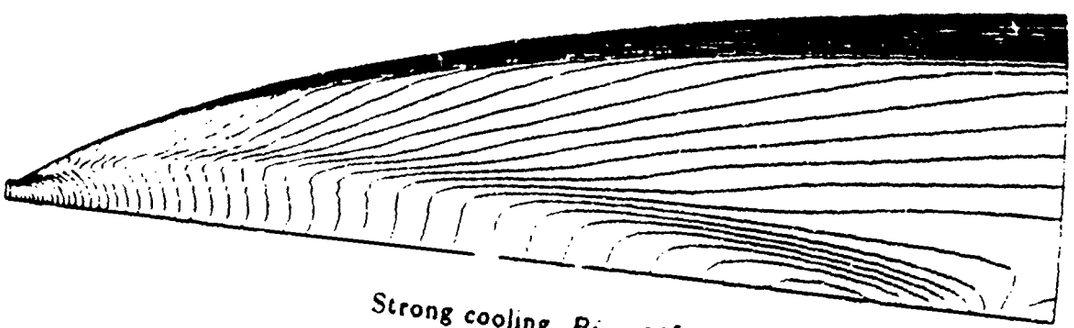
So far, the examples of supersonic nozzle flowfield predictions we have seen start from an arbitrarily given Mach number distribution at the inlet. For real nozzles, the flow enters the diverging section with a non-uniform Mach number distribution.



Adiabatic



Medium cooling $Bi = 10^3$



Strong cooling $Bi = 10^5$

Figure 43. Mach number contours for various wall cooling conditions

To take account of this effect, the computation must begin with the subsonic section of the nozzle. As we have shown, the transonic flow through the nozzle throat region can be efficiently calculated by the implicit ADI scheme. For flowfield computations over realistic nozzles which contain a short converging-diverging section and a very long diverging section, the following procedure is recommended. The nozzle can be segmented into two parts. The first part contains the entire converging section and a small portion of the diverging section. The divergent portion is chosen sufficiently large to ensure the flow at the flow at the last few rows of grid points is supersonic except for the boundary layer. The implicit ADI algorithm can be applied to this transonic portion and the resulting flowfield near the exit can be used as the input for subsequent supersonic calculations for the remaining part of the nozzle. The previously described supersonic algorithms can then be easily applied to flowfield computations in the diverging section.

Typical laminar results of the computation over the contoured nozzle starting from subsonic inflows by using the procedure above are shown in Fig. 44. The grid is 300×50 with 300 in the axial direction and the Reynolds number is 1.4×10^4. Comparisons with previous results using constant Mach number at the inlet (Fig. 41) show that the two-dimensionality near the throat has only minor effects on the flowfield results for this typical example.

3.6.6 Verification of Global Conservation

It is generally agreed that for flows with discontinuities, the strong conservative form of the equations plays an important role in global conservation. For flowfields that do not contain discontinuities, the fully conservative form is sometimes assumed to be less important. The primary application of present numerical algorithms are for rocket nozzle flowfield predictions. To accurately predict the flowfield and nozzle performance, global conservation is of great importance; however, "good" nozzle

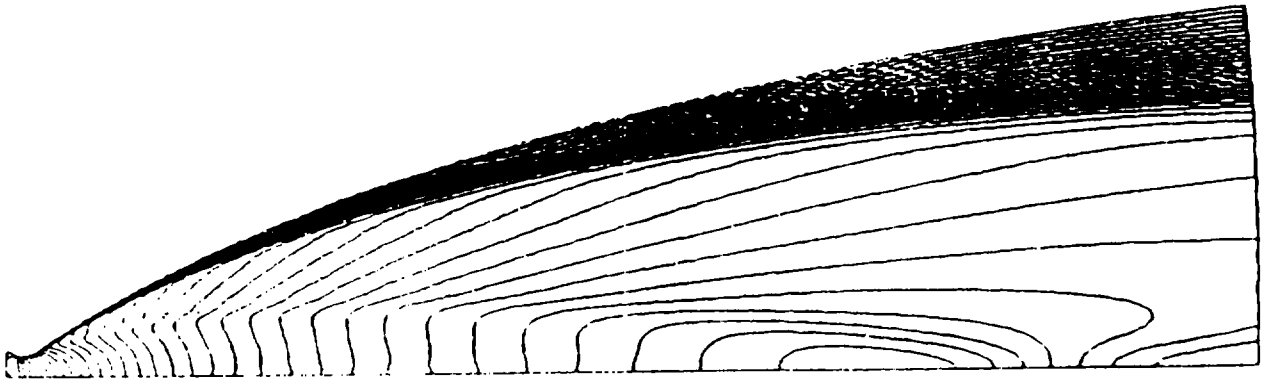


Figure 44. Mach number contours for nozzle with subsonic inflow

designs will generally be free from shocks and it might be assumed that the non-conservative equations are adequate. Because of the importance of accurate global mass conservation and to demonstrate the necessity of using the conservative form, mass conservation was monitored in all analysis codes throughout the course of this study. For better understanding of the necessity of the strong conservative formulation, a few computations with the weak conservative form of the governing equations were also done for typical high expansion nozzle flows without discontinuities. The weak conservative formulation is identical to Eq. (3.9) except that the metric coefficients are left outside the derivatives,

$$\frac{\partial Q}{\partial t} + \xi_x \frac{\partial E}{\partial \xi} + \eta_x \frac{\partial E}{\partial \eta} + \xi_y \frac{\partial F}{\partial \xi} + \eta_y \frac{\partial F}{\partial \eta} = H + \eta_x \frac{\partial E_v}{\partial \eta} + \eta_y \frac{\partial F_v}{\partial \eta}. \qquad (3.44)$$

The global mass conservation results for this weak conservative and the strong conservative formulations are compared in Fig. 45 for the high expansion ratio contoured nozzle. The 75 × 50 grid shown in Fig. 30 are used for both cases. As shown on this figure, the strong conservative form maintains the mass flow rate error within 1%, while the weak conservative form gives a maximum mass error of about 30%. In fact, for all the cases computed to date with the strong conservative form, including calculations for a nozzle with expansion ratio as high as 700, the maximum mass flow rate errors have been maintained below 1%.

The reason for the failure in weak conservative formulation is because it does not conserve the mass in its finite difference representation while the fully conservative form does. Therefore, the results shown in Fig. 45 is to be expected. The above results demonstrate that even for flows without discontinuities, the strong conservative formulation is necessary for maintaining global conservation.

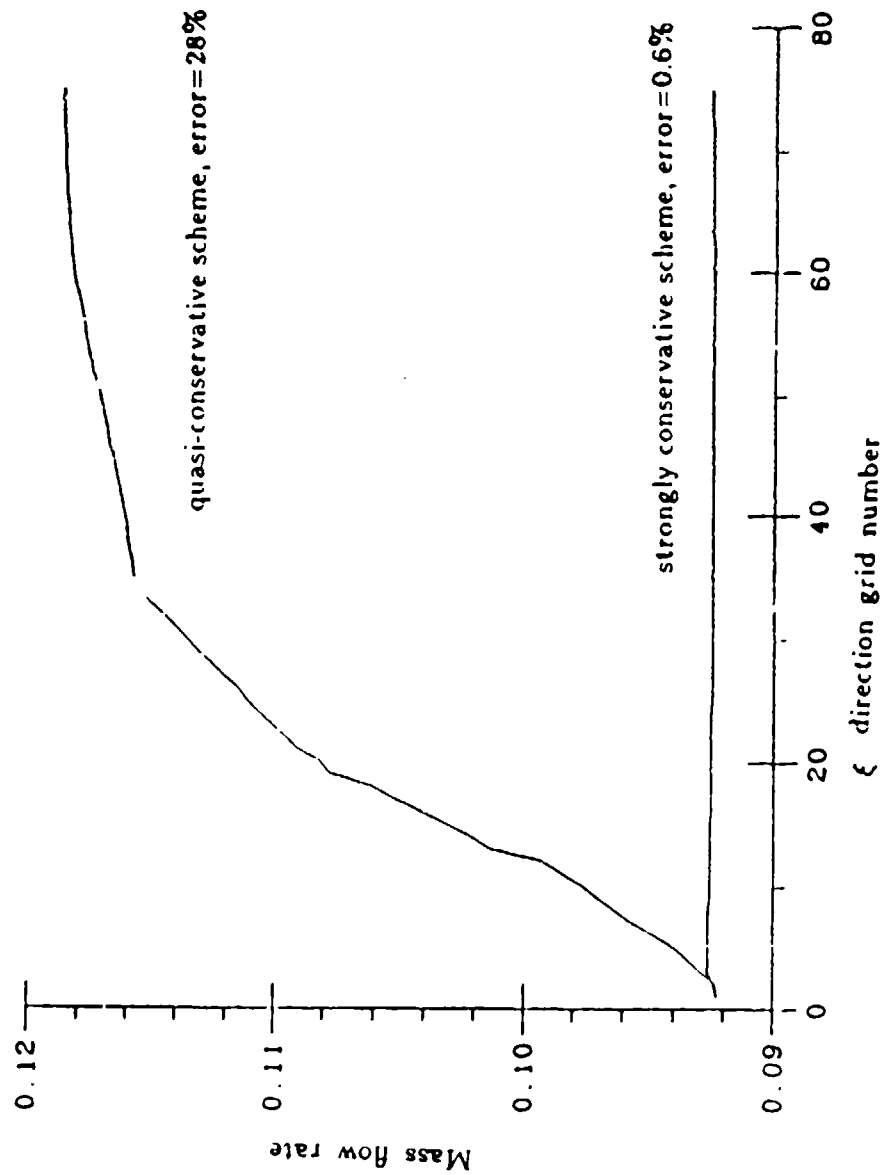


Figure 45. Comparison of global mass conservation for quasi-conservative and strongly conservative schemes

CHAPTER 4

THE APPLICATION OF TIME-ITERATIVE SCHEMES TO THE PARABOLIZED NAVIER-STOKES EQUATIONS

As indicated in Chapter 1, the parabolized Navier-Stokes equations have been extensively used as an alternative to the Navier-Stokes equations for the solution of compressible as well as incompressible viscous flows due to their computational efficiency. To assess this efficient numerical procedure and place it in a unified context with the present Navier-Stokes procedures, the parabolized technique is also addressed here. In Chapter 3, time-iterative numerical procedures based upon the predominant physics of the flow were formulated for the solution of thin-layer Navier-Stokes equations. Starting from these time-iterative Navier-Stokes equations, it is shown in this chapter that the parabolized equations can be obtained as a subset of the Navier-Stokes equations by means of flux splitting procedures. The traditional pressure gradient splitting devised by Vigneron is shown to be a specific type of flux vector splitting while the method of characteristics based splitting developed for the TLNS equations in Chapter 3 is also shown to split the streamwise flux vector into parts with positive and negative eigenvalues which can also be treated in a "parabolized" sense. This new PNS procedure is formulated by neglecting the flux vector with negative eigenvalues. The computational results obtained by using both the classical and the new PNS procedure are compared to those obtained with the TLNS algorithms to verify the accuracy. The flux splitting interpretation of the PNS procedure allows a well-behaved global iterative PNS procedure to be defined based on the PNS-ADI method presented in Chapter 3.

4.1 General Flux-Vector Splitting Navier-Stokes Equations

The thin-layer Navier-Stokes equations as given in Eq. (3.11) will be used as a starting point for the present approach. We first split the streamwise flux $\tilde{E}$ into two parts,

$$\tilde{E} = \tilde{E}^+ + \tilde{E}^- \tag{4.1}$$

where the eigenvalues of the Jacobians of $\tilde{E}^+$ and $\tilde{E}^-$ are positive and negative, respectively. With the substitution of Eq. (4.1) into Eq. (3.11), we have

$$\frac{\partial \tilde{Q}}{\partial t} + \frac{\partial \tilde{E}^+}{\partial \xi} + \frac{\partial \tilde{E}^-}{\partial \xi} + \frac{\partial \tilde{F}}{\partial \eta} = \tilde{H} + \frac{\partial}{\partial \eta}\left(R_1 \frac{\partial \tilde{Q}_1}{\partial \eta} + R_2 \frac{\partial \tilde{Q}_2}{\partial \eta}\right). \tag{4.2}$$

Note that the splitting given by Eq. (4.1) is used conceptually to indicate a general expression of flux-vector splitting, which might represent the Steger and Warming splitting defined in Chapter 3, but which could also represent any other splitting procedures. As examples, we will in addition to the Steger-Warming splitting also discuss a splitting based upon Vigneron's [26] parabolization method.

Although the approach is equally applicable to homogeneous or inhomogeneous flux vectors, for simplicity, we have restrict to the homogeneous case where $\tilde{E} = A\tilde{Q}$ with $A = \partial \tilde{E}/\partial \tilde{Q}$. Flux-splitting of the homogeneous vector is then reduced to splitting the matrix A as,

$$A = A^+ + A^-. \tag{4.3}$$

Here the eigenvalues of A^+ are positive and those of A^- are negative. From the homogeneity of the vector $\tilde{E}$, we have

$$\tilde{E}^+ = A^+\tilde{Q}, \qquad \tilde{E}^- = A^-\tilde{Q}$$

which obviously satisfies Eq. (4.1).

In principle, both the streamwise and the cross-stream fluxes could be split. Instead, we follow the traditional PNS approach, for which central differences are

generally used in the cross-stream direction while upwind differences are used in the streamwise direction. Thus, just as we did for the TLNS equations in Chapter 3, we have split only the streamwise flux $\hat{E}$.

Using Euler implicit differencing in time, the discretized version of Eq. (4.2) can be expressed as

$$\{I - \Delta t D + \Delta t[\frac{\partial}{\partial \xi} A_t^+ + \frac{\partial}{\partial \xi} A_t^- + \frac{\partial}{\partial \eta} B - \frac{\partial}{\partial \eta}(R_1 \frac{\partial}{\partial \eta} B_{v1} + R_2 \frac{\partial}{\partial \eta} B_{v2})]\} \Delta \tilde{Q} = -\Delta t R' \tag{4.4}$$

where R' has been previously defined by Eq. (3.36). Again, the true Jacobians of $\tilde{E}^+$ and $\tilde{E}^-$ are indicated by A_t^+ and A_t^-, respectively and the spatial derivatives in Eq. (4.4) must be treated consistently on both the left side and the right hand side. Note also that the derivatives containing A_t^+ and A_t^- must be upwind differenced in the manner defined previously. Efficient solution of Eq. (4.4) requires some sort of approximate factorization of the type discussed in Chapter 3.

4.1.1 Splitting Based on Characteristics

As the first of our two specific flux splitting procedures, we begin with one based on the method of characteristics. As we have seen in the previous discussion, the matrix A can be diagonalized according to the similarity transformation given by

$$\Lambda = M_\xi^{-1} A M_\xi$$

where the subscript of Λ has been dropped for simplicity. The matrix M_ξ is composed of the right eigenvectors of the matrix A and has been previously defined. The diagonal matrix Λ contains four entries, U, U, $U + C_\xi$, and $U - C_\xi$. A straightforward splitting suggested by Steger and Warming [16] is,

$$\Lambda^+ = \frac{(\Lambda + |\Lambda|)}{2}$$
$$\Lambda^- = \frac{(\Lambda - |\Lambda|)}{2} \tag{4.5}$$

in which, $|A|$ refers to the matrix composed of the absolute values of the elements of A. From Eq. (4.5) we readily obtain

$$A^+ = M_\xi \Lambda^+ M_\xi^{-1}, \qquad A^- = M_\xi \Lambda^- M_\xi^{-1}. \tag{4.6}$$

For the homogeneous case, the split flux vectors thus become

$$\tilde{E}^+ = A^+ \tilde{Q}, \qquad \tilde{E}^- = A^- \tilde{Q}.$$

According to Eq. (4.5), for supersonic flows, the matrix Λ^- is identically zero, and Λ^+ is equal to Λ. For subsonic flows, these matrices are $\Lambda^+ = diag(U, U, U + C_\xi, 0)$ and $\Lambda^- = diag(0, 0, 0, U - C_\xi)$.

4.1.2 Splitting Based on Pressure Gradient

The second splitting under consideration is based on the suggestion by Vigneron [26], who split the streamwise pressure gradient into two parts. His discussion of $\partial p/\partial x$ can also be interpreted as a flux splitting procedure given by,

$$\tilde{E}^+ = \frac{y}{J}\begin{bmatrix} \rho U \\ \rho u U + \omega \xi_x p \\ \rho v U + \omega \xi_y p \\ (e+p)U \end{bmatrix} \qquad \tilde{E}^- = \frac{y}{J}\begin{bmatrix} 0 \\ (1-\omega)\xi_x p \\ (1-\omega)\xi_y p \\ 0 \end{bmatrix}. \tag{4.7}$$

This splitting recognizes that the streamwise ellipticity arises from the pressure gradient term inside the subsonic portion of the boundary layer. Due to this pressure gradient, downstream information can propagate upstream. Thus, Vigneron's notion was to separate the streamwise pressure gradient into parts, then, by choosing the coefficient ω properly, one can place those parts responsible for upstream influence into the flux vector $\tilde{E}^-$. The vector $\tilde{E}^+$ therefore contains only those parts governing the propagation from upstream to downstream.

The eigenvalues of the Jacobian matrix of $\tilde{E}^+$ in Eq. (4.7) are the four roots of the following polynomial,

$$(\lambda - U)^2\{\lambda^2 - [\gamma + 1 - \omega(\gamma - 1)]U\lambda + (\gamma + \omega - \omega\gamma)U^2 - \omega C_\xi^2\} = 0. \tag{4.8}$$

The eigenvalues of A_ξ^- are the roots of

$$\lambda^3[\lambda - (1-\gamma)(1-\omega)U] = 0.$$

These roots are found to be

$$\Lambda^+ = diag(U, U, 1/2\{|(\gamma+1) - \omega(\gamma-1)|U \pm \sqrt{(\gamma-1)^2(\omega-1)^2U^2 + 4\omega C_\xi^2}\})$$
$$\Lambda^- = diag(0, 0, 0, -(\gamma-1)(1-\omega)U) \tag{4.9}$$

In keeping with our purpose, all four eigenvalues of $A_\xi^+ = \partial\hat{E}^+/\partial\hat{Q}$ must be positive, and those eigenvalues of $A_\xi^- = \partial\hat{E}^-/\partial\hat{Q}$ must be negative. For supersonic flows ($U > C_\xi$), if we set ω equal to unity, then Λ^+ becomes $\Lambda^+ = diag(U, U, U + C_\xi, U - C_\xi)$ and Λ^- is identically zero. All entries of Λ^+ are positive, thus the splitting is completed by setting $\omega = 1$ for supersonic flows.

For subsonic flows ($U < C_\xi$), the eigenvalues of A_ξ^- are negative as long as $\omega < 1$. On the other hand, the eigenvalues of A_ξ^+ are positive if the three inequalities

$$\begin{aligned} U &> 0 \\ |\gamma + 1 - \omega(\gamma-1)|U &> 0 \\ (\gamma + \omega - \omega\gamma)U^2 - \omega C_\xi^2 &> 0 \end{aligned} \tag{4.10}$$

are satisfied. The first inequality is straightforward while the second is equivalent to

$$\omega < \frac{\gamma+1}{\gamma-1}$$

which is valid if $\omega < 1$. The third inequality in Eq. (4.10) results in

$$\omega < \frac{\gamma M_\xi^2}{1 + (\gamma-1)M_\xi^2} \tag{4.11}$$

where M_ξ is the streamwise Mach number defined by $M_\xi = U/C_\xi$.

In summary, pressure gradient splitting based on Vigeneron's approach gives the proper signs for eigenvalues of both A_ξ^+ and A_ξ^- when the following conditions are met:

1. The streamwise contravariant velocity is positive.
2. ω is unity if the flow is supersonic.
3. ω satisfies Eq. (4.11) if the flow is subsonic.

Three observations are noted here. First, the flux vector splitting in Eq. (4.7) is analogous to traditional Vigneron-based PNS procedures. Second, the derivations of the conditions for proper pressure gradient splitting given above differ from the derivations given by previous investigators [3,26], in which the same splitting criteria are obtained by considering the steady state TLNS equations in both the inviscid and viscous limits of the corresponding simplified equations. By requiring these simplified equations to be hyperbolic along the streamwise direction in the inviscid limit, and to be parabolic in the viscous limit, they arrived at the same conclusions given above. In the present approach, the unsteady TLNS equations are considered, the splitting criteria are then obtained by taking into account only the signs of eigenvalues of the Jacobian matrix in the streamwise direction. Since the unsteady version of the TLNS equations is hyperbolic in time, by forcing these eigenvalues to be positive, we can easily complete the splitting without considering the TLNS equations in the inviscid or the viscous limits separately. Third, in the formulation of the pressure gradient splitting, the flux vectors $\hat{E}^+$ and $\hat{E}^-$ are defined without a prior specification of the matrices A^+ and A^-. In fact, the flux vectors $\hat{E}^+$ and $\hat{E}^-$ defined by Eq. (4.7) are not homogeneous; thus no explicit representations of A^+ and A^- for the pressure gradient splitting exist (as that given in Eq. (4.6)). However, if we neglect the variation of ω with respect to $\hat{Q}$, the relations

$$\hat{E}^+ = A_t^+ \hat{Q}, \qquad \hat{E}^- = A_t^- \hat{Q}$$

similar to Eq. (4.6) can be obtained, in which $A_t^+ = \partial \hat{E}^+ / \partial \hat{Q}$ and $A_t^- = \partial \hat{E}^- / \partial \hat{Q}$.

Just as in the discussion of characteristics splitting procedure in Section 4.1.1, we have split the flux vector into positive and negative parts.

4.2 Obtaining the PNS Procedure from the Navier-Stokes Algorithm

In the last section, we have formulated general flux vector splitting for the TLNS equations. In the special case if the splitting is based on characteristics, we have shown in Chapter 3 that the resulting TLNS equations can be efficiently solved by approximate factorization procedures. The traditional Vigeneron's parabolization procedure has also been interpreted as a special case of the generalized flux-vector splitting TLNS equations. Based on this interpretation, the Vigeneron's Parabolized procedure is equivalent to ignoring the reverse sweep in a specific flux-vector splitting TLNS procedure. This suggests that a general parabolization method can be devised based on an arbitrary flux splitting.

This generalized parabolization procedure can be achieved by simply neglecting the parts of the flux vector $\hat{E}$ contributing to upstream propagation. If the vector $\hat{E}^-$ is identically zero (as it is in supersonic flows) the algorithm given in Eq. (4.4) describes an alternating procedure in one direction. For those cases where $\hat{E}^-$ is not zero, we can likewise obtain a "marching" procedure by ignoring the contribution of $\hat{E}^-$. In other words, the streamwise ellipticity is suppressed by ignoring the elliptic parts of governing equations, thus the new approximate equations become parabolic in the streamwise direction. Again, we note that we must maintain consistency on both sides of Eq. (4.4), so we also drop the operator $\partial A_i^-/\partial\xi$ on the left-hand side. With this approximation, the left hand side of Eq. (4.4) becomes a parabolic operator,

$$\{I - \Delta t D + \Delta t[\frac{\partial}{\partial\xi}A_i^+ + \frac{\partial}{\partial\eta}B - \frac{\partial}{\partial\eta}(R_1\frac{\partial}{\partial\eta}B_{v1} + R_2\frac{\partial}{\partial\eta}B_{v2})]\}\Delta\hat{Q} = -\Delta t R'' \quad (4.12)$$

where the modified residual R'' is also parabolized,

$$R'' = [\frac{\partial\hat{E}^+}{\partial\xi} + \frac{\partial\hat{F}}{\partial\eta} - \hat{H} - \frac{\partial}{\partial\eta}(R_1\frac{\partial\hat{Q}_1}{\partial\eta} + R_2\frac{\partial\hat{Q}_2}{\partial\eta})]^n . \quad (4.13)$$

Equation (4.12) defines a general parabolized procedure based upon any flux splitting for which the eigenvalues of A_t^+ are positive. This implies that there are an infinite number of ways to accomplish the Parabolized Navier-Stokes procedures and Vigneron's pressure gradient method is only a special case of these parabolized procedures. With this general form of parabolized procedures, the splitting based upon characteristics seems to be more "natural" than the pressure gradient splitting in the physical sense. These two special cases of general parabolized procedures are considered in the next sections.

4.2.1 Pressure Gradient Splitting

In the special case when $\tilde{E}^+$ is given by Eq. (4.7), Eq. (4.12) becomes the traditional PNS operator as given by numerous authors (for example, Refs.26-29) except that the time derivative is included. These time-iterative PNS equations are to be solved by iterations at each streamwise location. In other words, since Eq. (4.12) is now a marching equation (this implies no upstream influences exist), it is clearly better to iterate to convergence in time at each line before advancing to the next streamwise station. To define this time-iterative procedure more precisely, Eq. (4.12) is rearranged as,

$$\{I-\Delta t D+(1+\frac{\kappa}{2})\frac{\Delta t}{\Delta\xi}A_t^+ + \Delta t[\frac{\partial}{\partial\eta}B-\frac{\partial}{\partial\eta}(R_1\frac{\partial}{\partial\eta}B_{v1}+R_2\frac{\partial}{\partial\eta}B_{v2})]\}\Delta\tilde{Q} =$$
$$-\Delta t\{\frac{(\hat{E}^+_{i,j})^n-(\hat{E}^+_{i-1,j})^*}{\Delta\xi}+\kappa\frac{(\hat{E}^+_{i,j})^n-2(\hat{E}^+_{i-1,j})^*+(\hat{E}^+_{i-2,j})^*}{2\Delta\xi}$$
$$+[\frac{\partial\tilde{F}}{\partial\eta}-\tilde{H}-\frac{\partial}{\partial\eta}(R_1\frac{\partial\tilde{Q}_1}{\partial\eta}+R_2\frac{\partial\tilde{Q}_2}{\partial\eta})]^n\} \tag{4.14}$$

where κ is 0 for first-order upwind differencing in the streamwise derivative and is 1 for second order differencing. The superscripts * denote that these quantities are to be evaluated based on the converged solutions. The value of $\Delta\tilde{Q}$ on the left-hand side of Eq. (4.14) is driven to machine accuracy by time marching at one station,

and then the procedure marches to the next ξ-station and so forth. As will be demonstrated later, this iteration can be driven to machine accuracy in less than 10 iterations for two-dimensional problems.

4.2.2 Characteristics Splitting

If the flux vector $\tilde{E}^+$ in Eq. (4.14) is chosen as that given in Eq. (4.6), a similar time-dependent PNS procedure can also be obtained. This marching procedure differs from the more traditional pressure gradient-split PNS procedure in the parabolized approximation. As we have seen, the pressure gradient splitting algorithm omits parts of the pressure gradient, while the characteristic splitting algorithm neglects those parts with upstream-propagating acoustic wave. The latter is more appropriately described by an appeal to the physics of the flow. The differences between these two algorithms are also indicated by the different eigenvalues of the Jacobians of $\tilde{E}^+$. As will be shown later, the calculations based on this PNS procedure give results that are almost identical to or even slightly better than those based on the pressure gradient splitting that is traditionally used.

4.2.3 Non-Iterative PNS Procedure

The PNS algorithms discussed above include the temporal derivative, while in the traditional PNS procedure, the solutions are obtained by a simple space marching without iterations. To obtain this marching procedure, we first re-write Eq. (4.12) without using the delta form. By cancelling terms on the left-hand side with those on the right-hand side (given in Eq. (4.13)), the time-dependent algorithm becomes

$$\{\tilde{Q} + \Delta t(\frac{\partial \tilde{E}^+}{\partial \xi} + \frac{\partial \tilde{F}}{\partial \eta} - \tilde{H}) - \Delta t \frac{\partial}{\partial \eta}(R_1 \frac{\partial}{\partial \eta} B_{v1} + R_2 \frac{\partial}{\partial \eta} B_{v2})\tilde{Q}\}^{n+1} = \tilde{Q}^n. \quad (4.15)$$

Note that for characteristic splitting, $\tilde{E}^+$ does not cancel with $A_t^+ \tilde{Q}$ since, $A_t^+ \neq A^+$ if the flow is subsonic. Therefore, Eq. (4.15) is only approximately valid for the

characteristic splitting algorithm, but this is not severe since in general the subsonic layer is very thin. If we allow Δt to go to infinity, and use the chain rule,

$$\frac{\partial \tilde{E}^+}{\partial \xi} = \frac{\partial \tilde{E}^+}{\partial \tilde{Q}} \frac{\partial \tilde{Q}}{\partial \xi} = A_i^+ \frac{\partial \tilde{Q}}{\partial \xi}. \tag{4.16}$$

Equation (4.15) then becomes

$$\frac{\partial \tilde{E}^+}{\partial \xi} = A_i^+ \frac{\partial \tilde{Q}}{\partial \xi} = -\frac{\partial \tilde{F}}{\partial \eta} + \tilde{H} + \frac{\partial}{\partial \eta}(R_1 \frac{\partial}{\partial \eta} B_{v1} + R_2 \frac{\partial}{\partial \eta} B_{v2})\tilde{Q}. \tag{4.17}$$

We can now linearize each term in Eq. (4.17) according to the Taylor series expansions,

$$\begin{aligned} \tilde{F}_{i+1} &= \tilde{F}_i + B\Delta\tilde{Q} \\ \tilde{H}_{i+1} &= \tilde{H}_i + D\Delta\tilde{Q} \end{aligned} \tag{4.18}$$

where subscripts represent the ξ direction grid number, B and D are Jacobians of $\tilde{F}$ and $\tilde{H}$, and $\Delta\tilde{Q}$ is now interpreted in a spatial rather than a temporal sense, $\Delta\tilde{Q} = \tilde{Q}_{i+1} - \tilde{Q}_i$.

With the substitution of Eq. (4.18) into Eq. (4.17), we have,

$$\begin{aligned} \{A_i^+ - \Delta\xi D + \Delta\xi |\frac{\partial}{\partial \eta} B - \frac{\partial}{\partial \eta}(R_1 \frac{\partial}{\partial \eta} B_{v1} + R_2 \frac{\partial}{\partial \eta} B_{v2})|\}_i \Delta\tilde{Q} = \\ - \Delta\xi |\frac{\partial \tilde{F}}{\partial \eta} - \tilde{H} - \frac{\partial}{\partial \eta}(R_1 \frac{\partial \tilde{Q}_1}{\partial \eta} + R_2 \frac{\partial \tilde{Q}_2}{\partial \eta})|_i . \end{aligned} \tag{4.19}$$

This equation now can be used to solve $\tilde{Q}_{i+1}$ without iterations at $i+1$ station provided that $\tilde{Q}_i$ is given. Equation (4.19) is referred to in the literature as a space-marching PNS algorithm. The formulation above shows that any flux-vector splitting defined by Eq. (4.1) can be used to obtain a distinct non-iterative PNS algorithm. In particular, the pressure gradient splitting noted above gives the traditional PNS procedure. Equation (4.17) also shows that the characteristic splitting suggested by Steger and Warming can be used to formulate a parabolized algorithm as well.

4.2.4 Comparisons of Time-Iterative and Space-Marching Algorithms

In the discussion above, both the time-iterative and space-marching PNS procedures are formulated. By intuition, one might expect that the space-marching algorithm is more efficient than the time-iterative algorithm because it does not require local iterations at each ξ station. There are, however, other issues involved. To demonstrate this, we compare the differences between two approaches. First, we consider the difference in the final converged solutions of the two methods. From Eq. (4.12), if $\Delta\tilde{Q}$ is driven to zero, the steady state solutions of the time-iterative procedure can be obtained as,

$$\frac{\partial}{\partial\xi}(A^+\tilde{Q}) + \frac{\partial\tilde{F}}{\partial\eta} - V.T. = 0 \tag{4.20}$$

where $A^+\tilde{Q} = \tilde{E}^+$ and $V.T.$ is used to represent the viscous terms. On the other hand, the solutions of the space-marching algorithm are,

$$A^+\frac{\partial\tilde{Q}}{\partial\xi} + \frac{\partial\tilde{F}}{\partial\eta} - V.T. = 0. \tag{4.21}$$

The η derivatives and the viscous terms are exactly the same for both methods. The difference lies in the ξ derivative, in which the time-iterative method utilizes the conservative form, while the space-marching method employs a non-conservative form.

To explore this difference further, we compare the finite-difference representations of the ξ derivative in Eqs. (4.20) and (4.21). For clarity, we restrict to first order in ξ. Upon discretization, the ξ derivative in Eq. (4.20) becomes

$$\frac{(A^+\tilde{Q})_{i+1} - (A^+\tilde{Q})_i}{\Delta\xi}$$

while the one in Eq. (4.21) is

$$A_i^+\frac{\tilde{Q}_{i+1} - \tilde{Q}_i}{\Delta\xi}.$$

Because the first row of the matrix A^+ is related to the continuity equation, the non-conservative form in Eq. (4.21) can be expected to give mass conservation errors. For a grid which has no stretching in the ξ direction, these mass errors are not severe since the metric coefficients are constants in the ξ direction. However, when grid stretching is used in ξ, the mass error can be expected to accumulate with ξ because of the variation in the metrics. Numerical experiments using the non-iterative (space-marching) scheme show that a global mass error of order one is observed for even a moderately stretched grid. For highly stretched grids, this accumulation leads to numerical instability. On the other hand, the time-iterative algorithm worked well with either uniform or highly non-uniform grids. Consequently, the penalty paid for local iterations in the time-iterative algorithm can be at least partly offset by using a stretched grid.

An alternative procedure for ensuring mass conservation with non-iterative scheme on stretched grids has been proposed by Schiff and Steger [43], although it does not appear to have been widely used. In their approach, instead of directly using the chain rule given by Eq. (4.16) to represent $\partial \tilde{E}^+ / \partial \xi$ in Eq. (4.17), the flux vector $\tilde{E}$ is linearized before discretizing. This procedure is obtained by noting that the flux vectors at two consecutive locations can be linearized according to

$$\tilde{E}^+_{i+1} = \tilde{E}^+_i + A^+_i (\tilde{Q}_{i+1} - \tilde{Q}_i) \tag{4.22}$$

$$\tilde{E}^+_i = \tilde{E}^+_{i-1} + A^+_{i-1} (\tilde{Q}_i - \tilde{Q}_{i-1}). \tag{4.23}$$

By using Eq. (4.22) and Eq. (4.23), we have

$$\frac{\partial \tilde{E}^+}{\partial \xi} = \frac{\tilde{E}^+_{i+1} - \tilde{E}^+_i}{\Delta \xi} = \frac{1}{\Delta \xi} A^+_i \Delta \tilde{Q} + \frac{1}{\Delta \xi} (A^+_i - A^+_{i-1}) \tilde{Q}_i. \tag{4.24}$$

With the use of Eq. (2.24) in Eq. (4.17), the space-marching PNS algorithm gives better mass conservation. Numerical experiments with this approach prove to be

able to conserve mass within an error of 1% for a non-uniform grid in a moderate expansion ratio (around 30) nozzle. However, for more realistic problems such as flows through the 272 : 1 nozzle investigated in Chapter 3, the 75 axial grid lines had to be increased to 300 to enable the modified space-marching algorithm to match the conservative time-iterative method with 75 axial grid lines in accuracy. The space-marching procedure without the Schiff-Steger modification led to global mass errors of more than 50% even with 300 axial grid lines.

The second difference to be addressed is the requirement of a safety factor, $\bar{\sigma}$, in defining the parabolized operator. Parabolized Navier-Stokes calculations reported in the literature (for example, Ref. [29]) traditionally use a safety factor in Eq. (4.11). This results in

$$\omega < \frac{\bar{\sigma}\gamma M_\xi^2}{1 + (\gamma - 1)M_\xi^2}$$

where the safety factor $\bar{\sigma}$ is generally chosen as 0.85 or smaller. Numerical experiments with the space-marching algorithm indicate that $\bar{\sigma}$ can not be greater than 0.85 without numerical instability. On the other hand, with the use of the time-iterative algorithm, $\bar{\sigma}$ can always be set equal to unity. The results presented in the next section also show that the solutions with $\bar{\sigma} = 1$ are more accurate than those with $\bar{\sigma} = 0.85$ as compared to the Navier-Stokes solutions. From the derivation of ω given in Section 4.1.2, it is clear that there is no theoretical reason for requiring a safety factor. The necessity of a safety factor in the space-marching method is only to make the algorithm stable, and is not inherent with the parabolized equations.

As a final comparison, we note that the left hand side operator of the time-iterative procedure is more diagonally dominant than that of the space-marching algorithm. In fact, vanishing elements appear on the diagonal of the left hand side matrix in the space-marching algorithm due to the absence of the identity matrix I in Eq. (4.19). As a consequence, pivoting strategies are required to solve

Eq. (4.19). Contrarily, the diagonal elements are always non-zero for the time-iterative algorithm, and pivoting is not required. Our experience shows that a 30% saving in computational time per iteration is gained by the solution of a block tri-diagonal matrix without pivoting. Again, this difference would help to make the time-iterative PNS procedure more economically competitive with the non-iterative procedure.

4.2.5 Stability Analysis of Time-Iterative PNS Algorithms

To validate the time-iterative algorithm developed above, the Fourier stability analysis of Eq. (4.14) is given as follows. The amplification matrix of the variable $\tilde{Q}$ is defined

$$\tilde{Q}^{n+1} = G\tilde{Q}^n.$$

From a von Neumann analysis, G can be found to be

$$L_1 G = I$$

where the matrix L_1 is,

$$L_1 = I - \Delta t D + \frac{\Delta t}{\Delta \xi} A_t^+ + i\frac{\Delta t}{\Delta \eta} B \sin\omega_\eta + 2\frac{\Delta t}{\Delta \eta^2}(R_1 B_{v1} + R_2 B_{v2})(1 - \cos\omega_\eta)$$

and ω_η is the η direction wavenumber. Figure 46 shows maximum eigenvalues of the G matrix versus wavenumber for typical supersonic and subsonic conditions. The results show that Eq. (4.14) is unconditionally stable, and that rapid convergence can be expected for high values of CFL. The stability results given above are for the pressure gradient splitting method. Those of characteristic splitting are qualitatively the same.

4.2.6 Results and Discussion

We have discussed two parabolized procedures so far. Now, the question to ask is which method is better in terms of both computational efficiency and solution

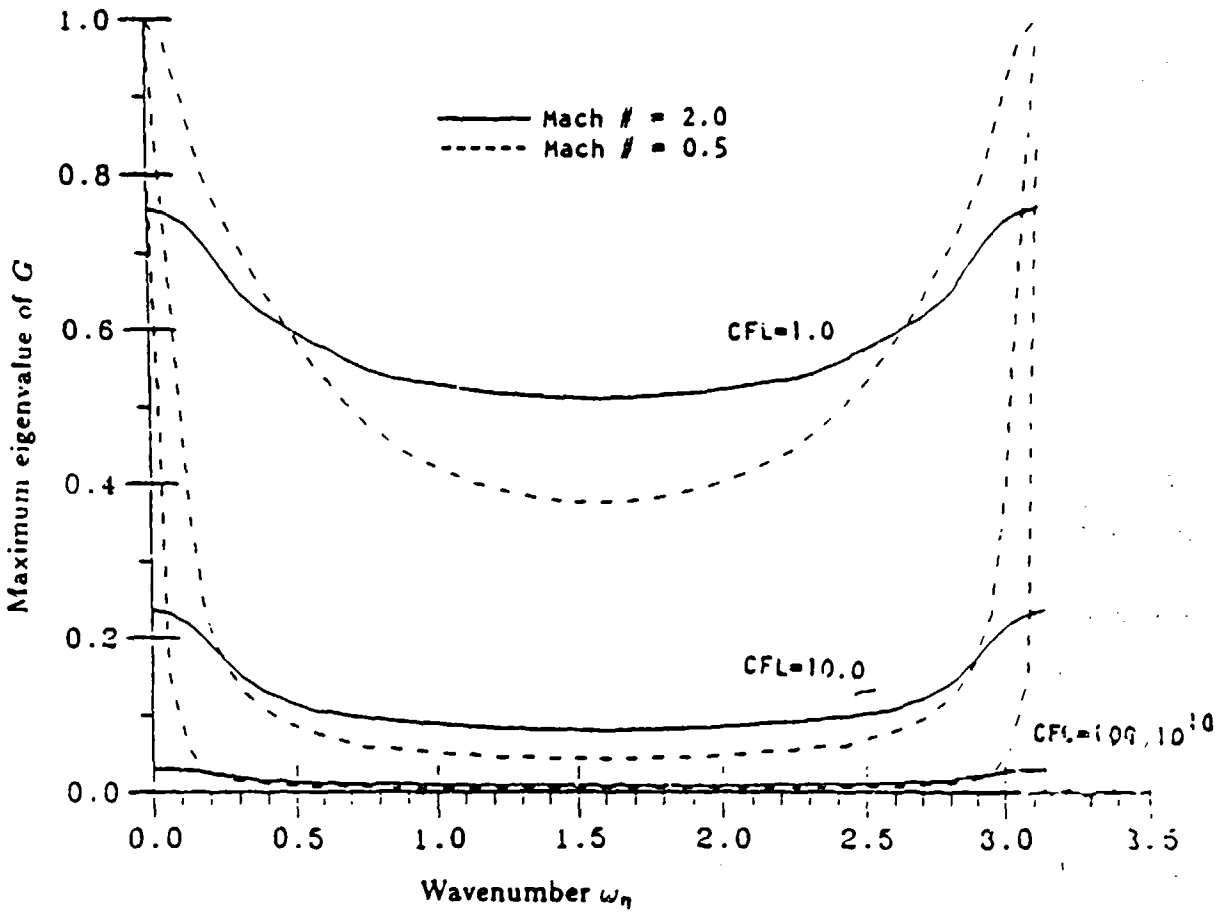


Figure 46. Stability results for the time-iterative PNS scheme with pressure gradient splitting

accuracy? To answer this question, a series of numerical computations were done and their solutions were compared to those from the TLNS algorithms provided in Chapter 3. Before showing the results of these comparisons, we note here that all PNS algorithms given above require boundary conditions in the cross-stream direction. The procedures at the wall and the centerline discussed in Section 3.2.1 are equally applicable to PNS algorithms. At the starting plane, the flow variable $\tilde{Q}$ must be given.

The test problem for the comparisons following is again the flow through the high expansion ratio nozzle with an area ratio of 272 : 1. The same flow conditions and properties described in Section 3.6 were used, including the inlet conditions, the ratio of specific heats, and the 75 × 50 grid system (for the non-iterative algorithm, a more refined 300 × 50 grid was used). The Reynolds number was taken to be 1.4×10^4 based on the throat radius for all calculations presented. All flowfield results presented are for laminar calculations.

The numerical efficiency of the time-iterative PNS procedure is shown in Fig. 47 for representative conditions. This figure shows the convergence at a specific ξ station by plotting the L-2 norm of $\Delta\tilde{Q}/\tilde{Q}$ associated with the four equations (continuity, momentum, and energy equations) as a function of the iteration number. Both inviscid and viscous results are shown on Fig. 47. A CFL number of 10^6 was chosen for both cases. The convergence clearly indicates that machine accuracy was reached in less than 10 iterations and the inviscid case converges slightly faster than the viscous case. Acceptable convergence (corresponding to a reduction of six order of magnitude in the L-2 norm) is obtained in 4 iterations. As we can see from Eq. (4.14), when the time step Δt goes to infinity, the time-iterative algorithm approaches Newton's method. Hence, the quadratic convergence shown in Fig. 47 is to be expected. This rapid convergence has been generally observed for all PNS

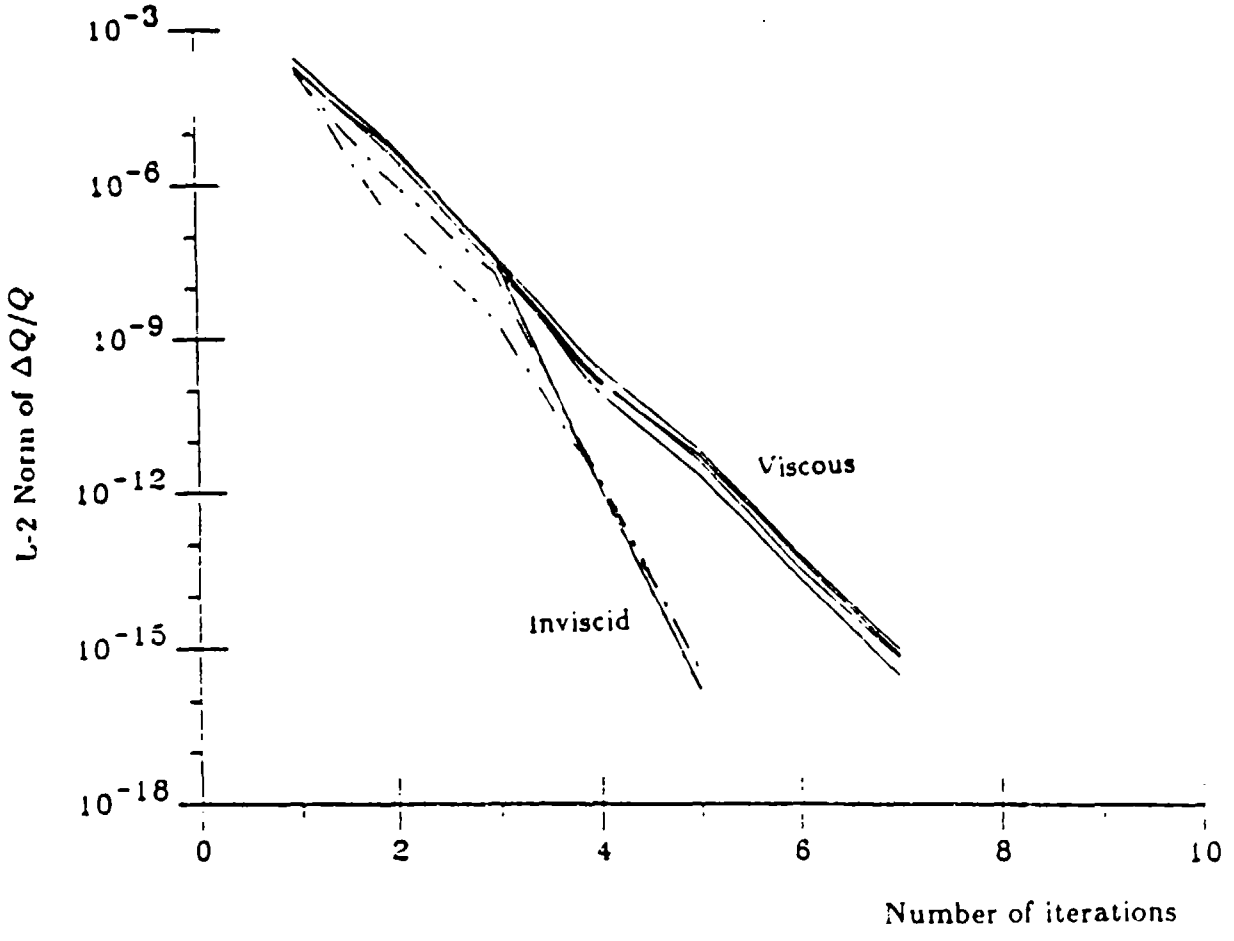


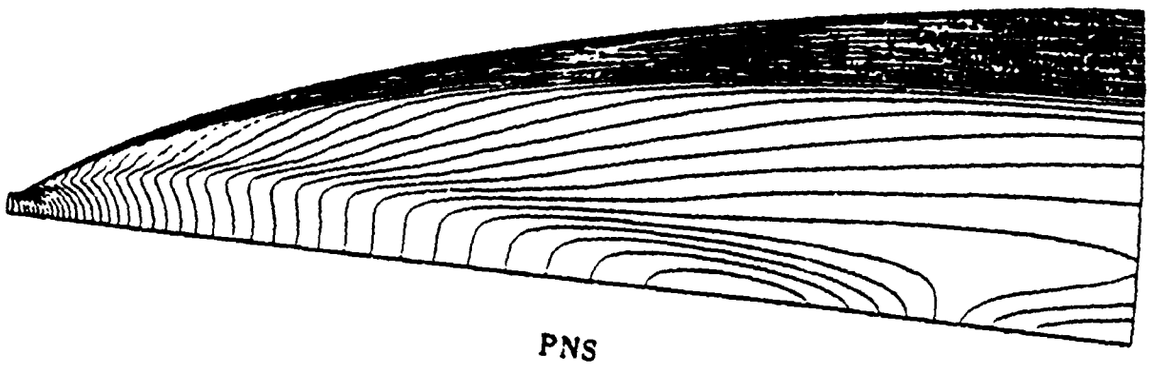
Figure 47. Convergence rate of time-iterative PNS procedure at typical axial location

calculations to date. The convergence shown in Fig. 47 is based upon pressure gradient splitting, but is also representative for characteristic splitting algorithm.

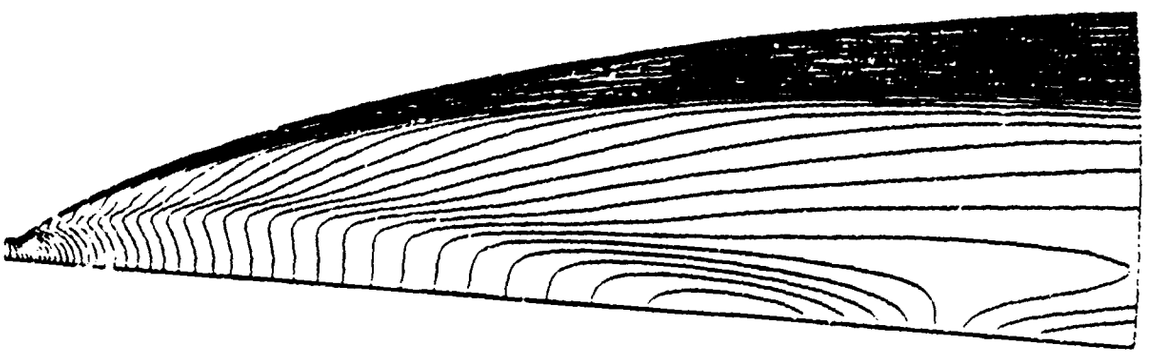
The solutions of PNS algorithms are compared to those from thin-layer Navier-Stokes calculations in Fig. 48. The upper plot shows the Mach number contours by using the pressure gradient splitting PNS method, the lower plot shows similar results for the TLNS solutions. This comparison indicates that the PNS procedure gives solutions that are almost identical to those of the TLNS equations. As will be shown later, the characteristic splitting also gives results that are even closer to the TLNS solutions.

To further compare the flowfield details, the pressure distribution and streamwise velocity profiles at the exit plane are plotted in Fig. 49 and Fig. 50. Each figure compares four different procedures. They are the TLNS method, the pressure gradient splitting PNS with $\delta = 1.0$ and $\delta = 0.85$, and the characteristic splitting PNS. The pressure profiles in Fig. 49 show that the computed pressure distribution by pressure gradient splitting without safety factor is almost identical to that by characteristic splitting, except the former slightly overshoots the pressure at the centerline. Both methods agree very well with the TLNS results and the characteristics splitting method is slightly better than the pressure gradient PNS. Figure 49 also shows that the use of a safety factor of 0.85 in pressure gradient splitting deteriorates the solution accuracy. As can be seen, the use of the safety factor causes about a 25% undershoot in the pressure at the centerline and about a 5% overshoot at the wall.

Similar comparisons associated with the velocity profiles shown in Fig. 50 show that all four procedures predict fairly close velocity distributions. Correct values of velocity together with incorrect values of static pressure indicate that the entropy (stagnation pressure) is not well conserved, a phenomenon frequently encountered



PNS



TLNS

Figure 48. Comparison of Mach number contours computed by PNS algorithm based on pressure gradient splitting and the TLNS algorithm

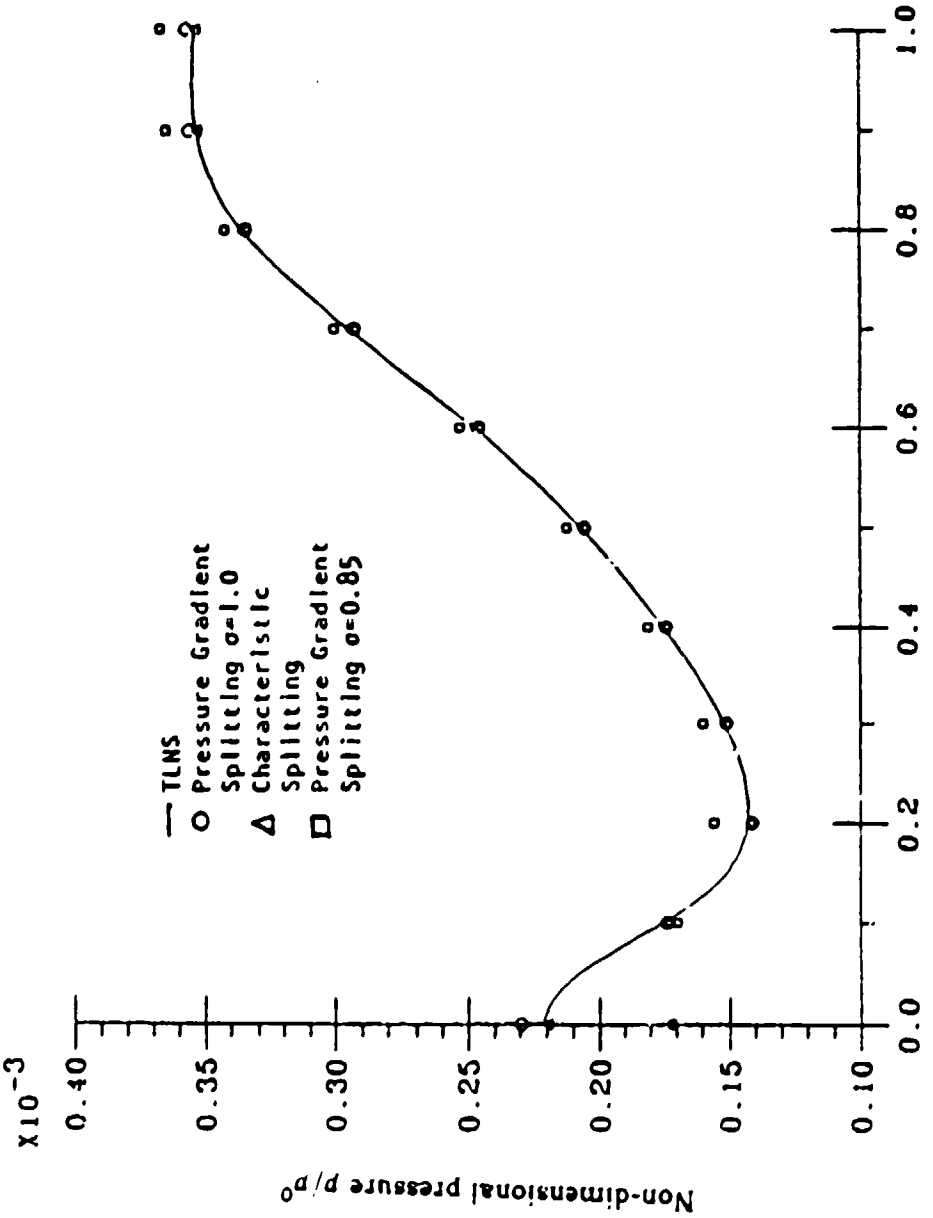


Figure 49. Comparison of cross-stream pressure profile at exit plane for various PNS results with TLNS calculations

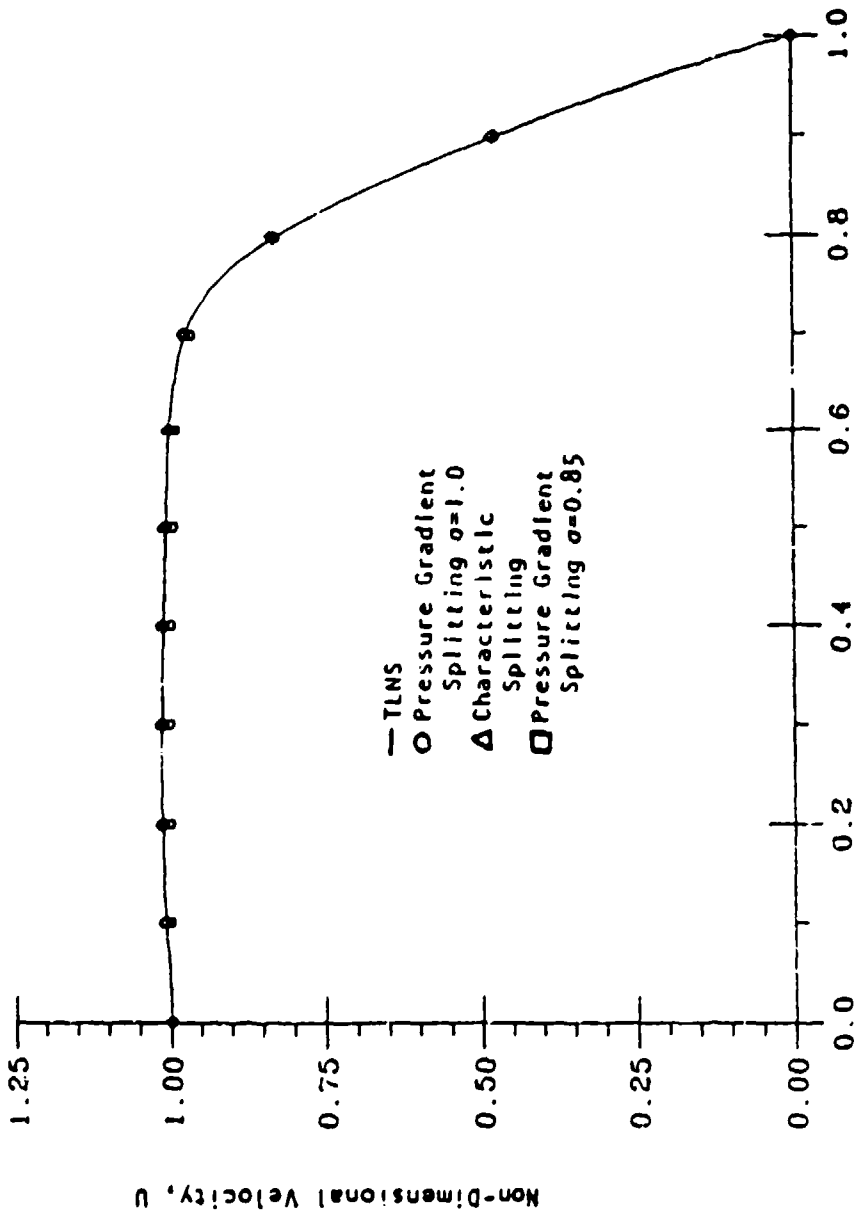


Figure 50. Comparison of cross-stream velocity profile at exit plane for various PNS results with TLNS calculations

in numerical schemes.

The effect of setting $\sigma = 0.85$ is also indicated in Fig. 51, which [illegible] number contours for this case. Comparison of this figure with the [illegible] solutions given in Fig. 48 shows that the over-suppression of the streamwise pressure gradient by using safety factor other than one alters the flowfield structure dramatically.

Further comparisons are shown in Fig. 52 and Fig. 53, where the wall pressure distributions and the locations of the sonic line along the axial direction are plotted. Figure 52 shows that all four methods give almost identical wall pressure distributions but again the case with the safety factor included is not quite as accurate. The sonic line locations shown in Fig. 53 are obtained by interpolation between grid points. This figure shows that the three PNS algorithms give basically the same subsonic layer thickness (the distance from the wall to the sonic point). Although, the $\sigma = 0.85$ case underpredicts the thickness of subsonic layer by about 1%, which is the worst among the three algorithms. These results indicate that the PNS approximation gives solutions that are acceptable in accuracy for the high Reynolds number flow without separation, as in current test problem.

The discussion above demonstrates that for better solution accuracy, the safety factor should not be less than one (which is easily done by using the time-iterative algorithm). Furthermore, the characteristic splitting PNS procedure gives solutions that are as accurate as, or even more accurate than (as in current test problem) the traditional pressure gradient splitting PNS method.

So far, all results shown for supersonic viscous calculations including both from TLNS and PNS algorithms are obtained by using second order differencing in the ξ direction. To demonstrate the difference in accuracy between first order and second order accurate upwind differencing in two dimensions, the first-order PNS results of the same test problem (flows through the high expansion nozzle) are shown in

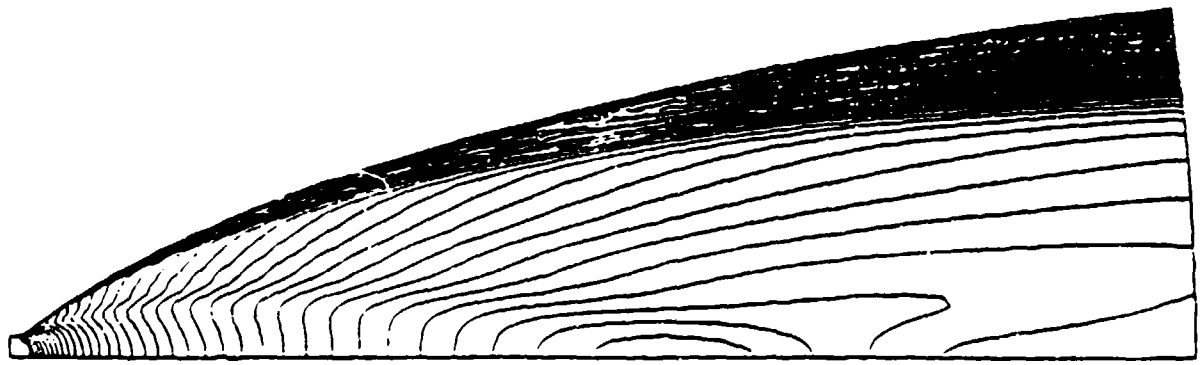


Figure 51. Mach number contours computed by PNS algorithm based on pressure gradient splitting and using a safety factor of 0.85

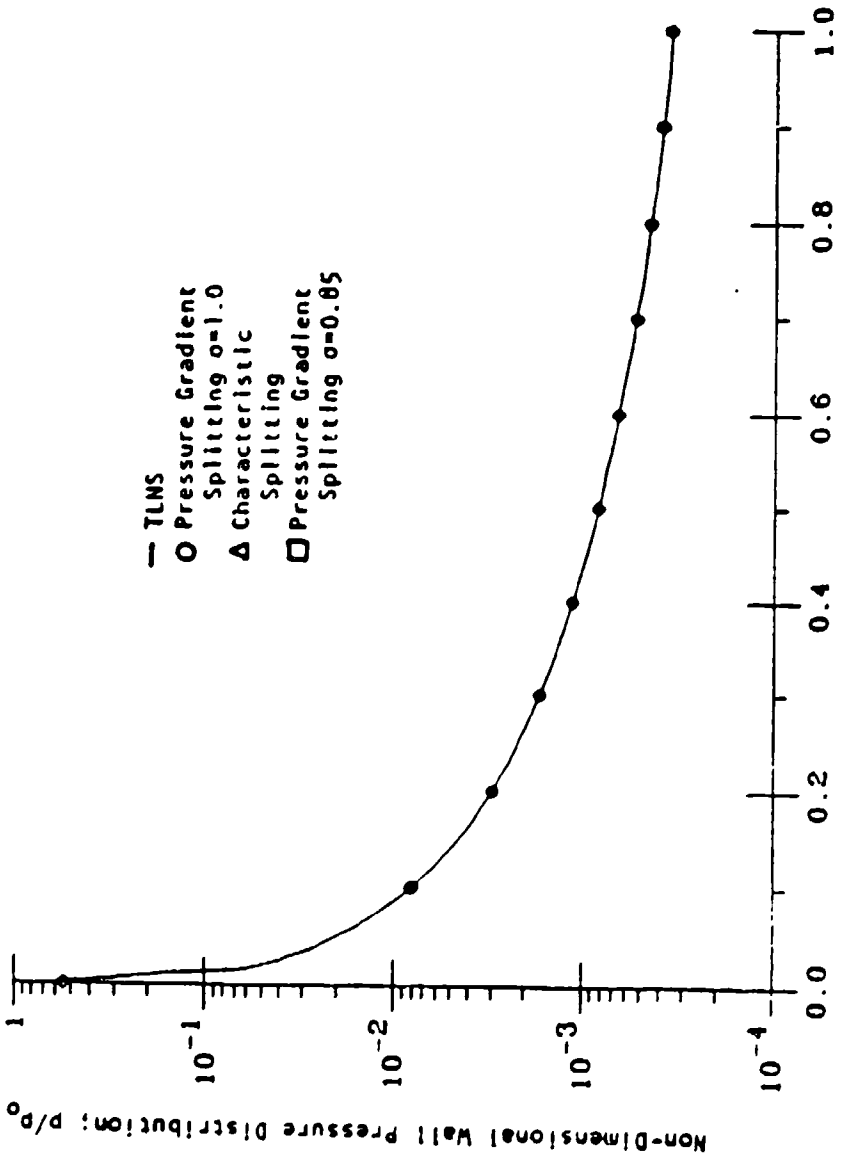


Figure 52. Pressure distribution on the wall for TLNS and PNS procedures

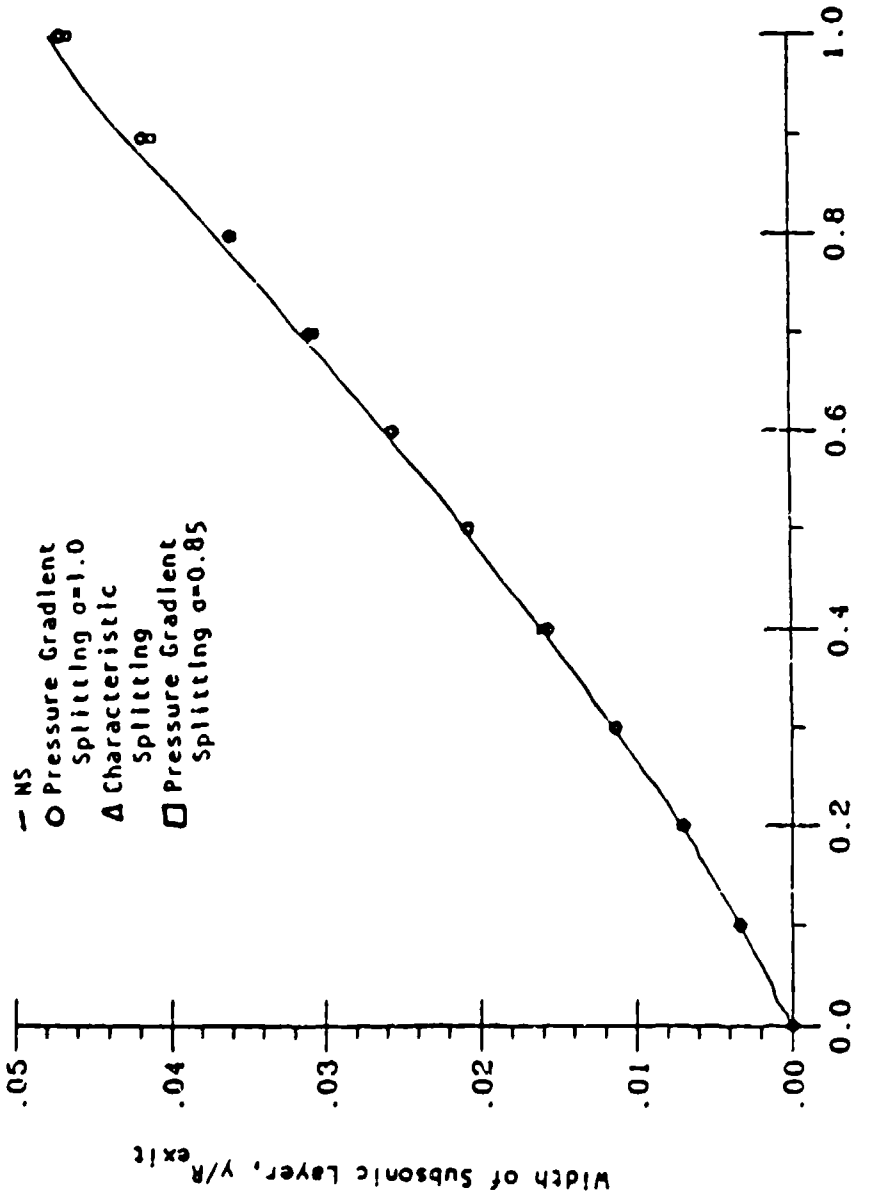


Figure 53. Width of subsonic layer as a function of axial distance along the wall, showing results of TLNS and various PNS calculations

Fig. 54. Comparisons of Fig. 48 and Fig. 54 show that the oblique shock wave from the first-order solutions is not as sharp as that from second-order solutions due to the smearing effect resulting from the inherent second order artificial dissipation of the first-order upwind differencing. Therefore, for better solution accuracy, second order upwind differencing should always be used.

4.3 Global PNS Procedures

For flows with strong upstream influences such as separated flows, the marching type PNS procedure as discussed in Section 4.2 can lead to serious errors in the numerical solution due to the suppression of the streamwise ellipticity. To allow the upstream propagation of acoustic wave inside the subsonic layer, thus preserving the streamwise elliptic behavior, the $\partial \tilde{E}^- / \partial \xi$ term in Eq. (4.2) must be included; thus the Navier-Stokes procedures discussed in Chapter 3 must be used instead of the parabolized algorithms provided in this chapter.

In the traditional PNS approach, numerous attempts have been made to take into account the upstream influences by identifying global pressure iterations. The basic idea of global pressure iterations is to update the pressure field by providing some sort of stable differencing scheme for the the omitted $(1-\omega)\partial p/\partial \xi$ term. This is usually done by evaluating $(1-\omega)\partial p/\partial \xi$ from a forward difference and using the updated value of pressures at downstream locations, as suggested in the works of Rakich [44], and Lin and Rubin [45]. Davis and co-workers [46] and Barnett and Davis [47] also developed a global pressure iteration procedure by appending a fictitious unsteady term, $\partial p/\partial \tau$, on the steady state equation, then updating the pressure field by a two-step alternating direction explicit (ADE) procedure. These global pressure iterations are summarized by Thompson and Anderson [48].

In the present study, by obtaining the PNS procedure from the TLNS equations, the procedure for incorporating a global pressure iteration procedure becomes

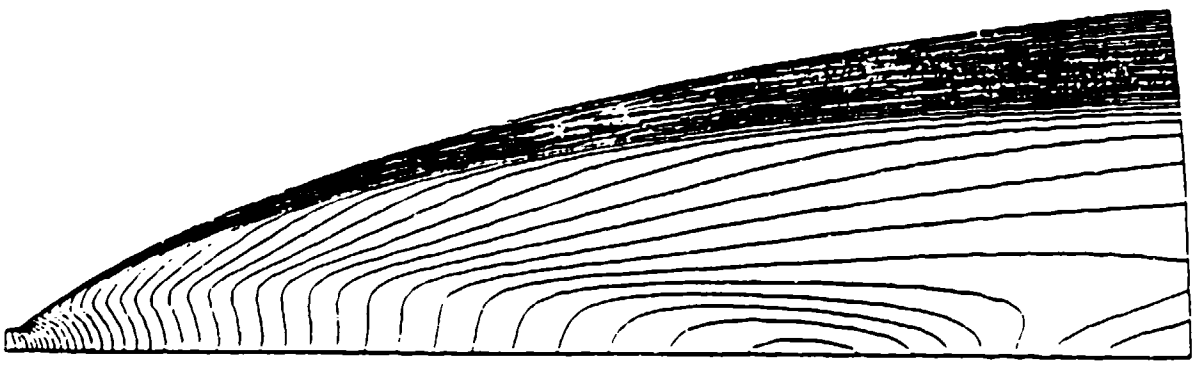


Figure 54. Mach number contours computed by PNS algorithm based on pressure gradient splitting, first-order upwind results.

obvious. We need only return to the complete TLNS equations (Eq. (4.2)). Therefore, all three approximate factorization algorithms provided in Chapter 3 can be interpreted as global pressure iteration procedures.

The TLNS algorithms (or so-called global pressure iteration procedures in the traditional PNS approach) developed in this study are based upon the approximate factorization of the TLNS equations, and therefore have both physical and mathematical connections to the equations of motions, while the global pressure iteration algorithms are concerned with arbitrary iterative processes for the pressure gradient $(1-\omega)\partial p/\partial \xi$, which are unrelated to the physical equations, as a consequence, some sort of relaxation scheme is required. This suggests that the mathematically and physically well-behaved TLNS algorithms based on approximate factorization can be used instead of global pressure iteration procedures.

As an example of the interpretation of global pressure iterations based on TLNS algorithms, the following procedure is suggested:

1. Obtain an initial PNS solution by marching from upstream to downstream using Eq. (4.4).
2. Solve the discretized equation of the PNS-ADI algorithm, Eq. (3.40), by the following two equations,

$$\{I - \Delta t D + \Delta t[\frac{\partial}{\partial \xi}A_t^+ + \frac{\partial}{\partial \eta}B - \frac{\partial}{\partial \eta}(R_1\frac{\partial}{\partial \eta}B_{v1} + R_2\frac{\partial}{\partial \eta}B_{v2})]\}\Delta\tilde{Q}^* = -\Delta t R' \tag{4.25}$$

$$(I - \Delta t D + \Delta t\frac{\partial}{\partial \xi}A_t^-)\Delta\tilde{Q} = (I - \Delta t D)\Delta\tilde{Q}^* \tag{4.26}$$

3. Update the dependent variable $\tilde{Q}$, according to,

$$\tilde{Q}^{n+1} = \tilde{Q}^n + \Delta\tilde{Q}$$

until the converged steady state is reached.

The first step is used to obtain a good initial condition for the TLNS calculations. The first equation in the second step, Eq. (4.25), is equivalent to the time-iterative PNS algorithm and the second equation, Eq. (4.26), is augmented in order to provide a mechanism to allow upstream propagation to take place inside the subsonic layer. In the supersonic region, A_ξ^- is identically zero and the left hand side operator in Eq. (4.26) reduces to an identity matrix; hence, only Eq. (4.25) has to be solved.

Figure 55 shows typical convergence of the TLNS procedure mentioned above when applied to the high-expansion ratio nozzle calculation given in Section 4.2.6. It requires only 110 iterations to reach machine accuracy; acceptable convergence is achieved in 25 iterations.

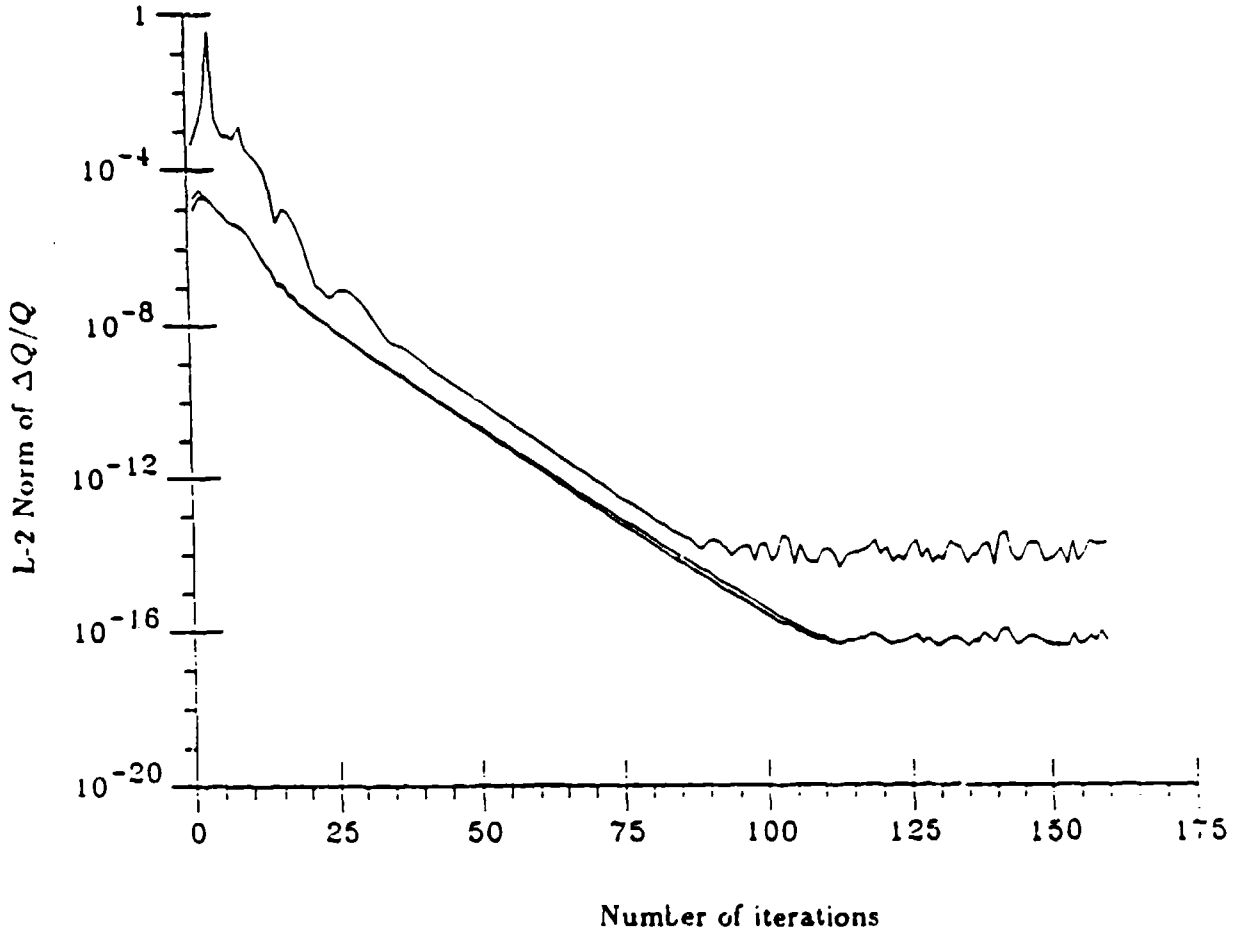


Figure 55. Convergence of global pressure iteration for the TLNS equations for 272:1 contoured nozzle

CHAPTER 5

THE APPLICATION OF TIME-ITERATIVE SCHEMES TO VISCOUS SWIRLING NOZZLE FLOWS

As examples of the application of the Navier-Stokes algorithms, swirling viscous flows in transonic and supersonic propulsion nozzles are investigated in this chapter. The central-differenced ADI and the flux-vector splitting algorithms discussed in Chapter 3 are utilized to solve the thin-layer Navier-Stokes equations for axisymmetric two-dimensional flows with swirl. The effects of swirl on viscous flows are identified for nozzles with mild to high expansion ratios. Both flowfield details and the integral nozzle performance are compared to previously published inviscid calculations.

5.1 Governing Equations and Numerical Algorithms

The swirling nozzle flow inside an axisymmetric nozzle can be described by the three-dimensional Navier-Stokes equations. If we assume the flow is axisymmetric, all circumferential derivatives vanish and the system of equations reduces to two dimensions. The resulting Navier-Stokes equations in vector form can be written as

$$\frac{\partial Q}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} = H + \frac{\partial E_v}{\partial x} + \frac{\partial F_v}{\partial y}. \tag{5.1}$$

where x and y are the axial and radial coordinates, respectively. The flow variable Q in Eq. (5.1) is defined as

$$Q = y(\rho, \rho u, \rho v, \rho w, e)^T$$

in which u, v, and w represent axial, radial, and circumferential velocity components. The inviscid flux vectors E and F are given by

$$E = y \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ (e+p)u \end{bmatrix}, \qquad F = y \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ (e+p)v \end{bmatrix}.$$

Viscous terms are included in flux vectors E_v and F_v,

$$E_v = y \begin{bmatrix} 0 \\ \mu(\frac{4}{3}\frac{\partial u}{\partial x} - \frac{2}{3}\frac{\partial v}{\partial y}) \\ \mu(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}) \\ \mu\frac{\partial w}{\partial x} \\ \mu v(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}) + \mu u(\frac{4}{3}\frac{\partial u}{\partial x} - \frac{2}{3}\frac{\partial v}{\partial y}) + \mu w\frac{\partial w}{\partial x} + k\frac{\partial T}{\partial x} \end{bmatrix}$$

$$F_v = y \begin{bmatrix} 0 \\ \mu(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}) \\ \mu(\frac{4}{3}\frac{\partial v}{\partial y} - \frac{2}{3}\frac{\partial u}{\partial x}) \\ \mu\frac{\partial w}{\partial y} \\ \mu u(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}) + \mu v(\frac{4}{3}\frac{\partial v}{\partial y} - \frac{2}{3}\frac{\partial u}{\partial x}) + \mu w\frac{\partial w}{\partial y} + k\frac{\partial T}{\partial y} \end{bmatrix}.$$

The source vector H is defined by

$$H = \begin{bmatrix} 0 \\ -\frac{2}{3}\frac{\partial}{\partial x}(\mu v) \\ p + \rho w^2 - \frac{4}{3}\mu\frac{v}{y} + \frac{2}{3}\mu\frac{\partial u}{\partial x} - \frac{2}{3}v\frac{\partial \mu}{\partial y} \\ -\rho vw - \mu\frac{w}{y} - w\frac{\partial \mu}{\partial y} \\ -\frac{2}{3}\frac{\partial}{\partial x}(\mu uv) - \frac{2}{3}\frac{\partial}{\partial y}(\mu v^2) - 2\mu w\frac{\partial w}{\partial y} - w^2\frac{\partial \mu}{\partial y} \end{bmatrix}.$$

The system of equations, Eq. (5.1), is similar to the Navier-Stokes equations in axisymmetric two dimensions (Eq. (3.2)) except the tangential momentum equation is included in Eq. (5.1) to take into account the variation of the circumferential velocity in axial and radial directions.

Following similar procedures given in Section 3.1, the thin-layer version of Eq. (5.1) in general coordinate system for two-dimensional swirling flows can be written as

$$\frac{\partial \hat{Q}}{\partial t} + \frac{\partial \hat{E}}{\partial \xi} + \frac{\partial \hat{F}}{\partial \eta} = \hat{H} + \frac{\partial \hat{F}_v}{\partial \eta} \tag{5.2}$$

where the flow variable and the flux vectors are

$$\tilde{Q} = \frac{y}{J}(\rho, \rho u, \rho v, \rho w, e)^T,$$

$$\tilde{E} = \frac{y}{J}\begin{bmatrix} \rho U \\ \rho u U + \xi_x p \\ \rho v U + \xi_y p \\ \rho w U \\ (e+p)U \end{bmatrix}, \qquad \tilde{F} = \frac{y}{J}\begin{bmatrix} \rho V \\ \rho u V + \eta_x p \\ \rho v V + \eta_y p \\ \rho w V \\ (e+p)V \end{bmatrix},$$

and

$$\tilde{F}_v = \frac{y}{J}\begin{bmatrix} 0 \\ \alpha_1 \frac{\partial u}{\partial \eta} + \alpha_2 \frac{\partial v}{\partial \eta} \\ \alpha_2 \frac{\partial u}{\partial \eta} + \alpha_3 \frac{\partial v}{\partial \eta} \\ \alpha_5 \frac{\partial w}{\partial \eta} \\ \alpha_4 \frac{\partial}{\partial \eta}\frac{e}{\rho} + \frac{\alpha_1 - \alpha_4}{2}\frac{\partial u^2}{\partial \eta} + \frac{\alpha_3 - \alpha_4}{2}\frac{\partial v^2}{\partial \eta} + \frac{\alpha_4 - \alpha_4}{2}\frac{\partial w^2}{\partial \eta} + \alpha_2 \frac{\partial uv}{\partial \eta} \end{bmatrix}.$$

In which α_1 through α_4 follow the same definitions described in Chapter 3 and α_5 is

$$\alpha_5 = \mu(\eta_x^2 + \eta_y^2).$$

The source vector $\tilde{H}$ for swirling flows is simply $\tilde{H} = H/J$. Again, if μ is set equal to zero and $\partial \tilde{F}_v / \partial \eta$ is omitted, Eq. (5.2) reduces to the Euler equations which describe inviscid swirling flows.

Equation (5.2) takes the same form as Eq. (3.9) except for the additional entries arising from the tangential momentum equation. Therefore, all numerical algorithms discussed in Chapter 3 and Chapter 4 are presumably applicable for the present governing equations. According to the nature of the flow, different numerical algorithms will be employed to solve transonic and supersonic flows. For transonic flows, we choose the implicit ADI procedure instead of MacCormack's explicit algorithm which was used by previous workers [32-34]. The details of this implicit ADI scheme have been discussed in Section 3.2 and will not be repeated here.

As mentioned earlier, the implicit ADI procedure becomes inefficient and sometimes even leads to numerical instability if a large portion of the flowfield is supersonic For this viscous supersonic swirling flow, the hybrid upwind-central differencing algorithms described in Section 3.4 can be chosen. For swirling flows where upstream influence is not significant, the parabolized procedures discussed in Chapter 4 are also applicable. The major difference in numerical procedures between the present swirling flow solvers and the non-swirling axisymmetric solvers discussed in Chapter 3 is that the block size of the left hand side matrix for the present case is 5×5 while that for the non-swirling case is 4×4. Therefore, numerical procedures for the swirling flows are more time-consuming than those for the non-swirling cases.

5.2 Boundary Conditions

Previously defined boundary procedures can be easily extended to swirling flow calculations. For supersonic flows in the meridian plane at the inlet, the flow variable $\bar{Q}$ is completely specified. For subsonic inflows at the inlet, the stagnation temperature, the stagnation pressure, the meridian plane streamline angle $\psi = tan^{-1}(v/u)$, and the swirl angle $\phi = tan^{-1}(w/u)$ are specified, the remaining one unknown comes from the characteristic equation corresponding to the single negative eigenvalue. The swirl angle profile at the inlet is assumed to be one of constant angle, free vortex, or forced vortex, which are the same as in Ref. [34] except that the swirl angle asymptotically approaches zero at the wall for all viscous calculations.

At the wall, four characteristic equations and the tangency condition are imposed for inviscid calculations, while no-slip conditions together with zero normal pressure gradient and isothermal or adiabatic wall conditions are used. Symmetrical conditions are applied at the centerline. At the exit, either extrapolation or fixed back pressure conditions can be chosen as described in Chapter 3.

To give assessments of the nozzle performance, several integral parameters are defined as the following. These include the discharge coefficient C_D, the vacuum stream thrust efficiency η_{vs}, the specific impulse efficiency η_{SI}, and the nozzle flowfield as a function of the inlet swirl number S_i,

$$C_D \equiv \frac{\dot{m}}{\dot{m}_{id}} = 2\int_{y_c}^{y_w} \rho u y dy / (y_{wt}^2 - y_{ct}^2)(\rho^* u^*)_{id}$$

$$\eta_{vs} \equiv 2\int_{y_{ce}}^{y_{we}} (p + \rho u^2) y dy / (y_{we}^2 - y_{ce}^2)(p_e + \rho_e u_e^2)_{id}$$

$$\eta_{SI} \equiv \frac{\eta_{vs}}{C_D}$$

$$S_i \equiv \int_{y_{ci}}^{y_{wi}} \rho u w y^2 dy / y_{wi} \int_{y_{ci}}^{y_{wi}} \rho u^2 y dy.$$

The subscripts i, t, e, c, w, and id denote inlet, throat, exit, centerline or centerbody, wall, and ideal conditions, respectively. The quantity $\dot{m}$ is the mass flow rate. The ideal conditions are obtained from one-dimensional isentropic values at the same stagnation conditions as the actual flow. The discharge coefficient can be interpreted as a measure of the loss in mass flow rate due to two-dimensionality and the swirl. The swirl number is defined as the ratio of the axial flux of flow angular momentum divided by the axial flux of axial momentum and is a direct measure of the level of swirl at the nozzle inlet.

5.3 Nozzle Flowfield Predictions

To place present viscous swirling calculations in perspective with previously published inviscid results, the implicit ADI scheme is applied to calculate transonic flows through a convergent-divergent nozzle, an annular plug nozzle and a converging nozzle, which all have been investigated by Dutton [34]. As an example of predominantly supersonic flowfield, the viscous swirling flow through the 272 : 1 contoured nozzle previously given is calculated by using the upwind-central differencing algorithms. Only laminar results are shown for all three transonic cases and both laminar and turbulent results are presented for the contoured nozzle.

5.3.1 Convergent-Divergent Nozzle

The $35^0 - 18.5^0$ convergent-diverging nozzle calculated by Dutton is analyzed in the first series of computations. The geometry of this nozzle is shown in Fig. 56 and Fig. 57 for inviscid and viscous computations, respectively. The 63×30 equally spaced grid in both the axial and radial directions as shown in Fig. 56 is for inviscid calculations, while the 63×50 grid with strong clustering near the wall as shown in Fig. 57 is for viscous calculations.

The convergence rates of inviscid and viscous cases are shown in Fig. 58 and Fig. 59, respectively. For inviscid computations, as shown in Fig. 58, the L-2 norm of $\Delta\tilde{Q}/\tilde{Q}$ reduces 9 orders of magnitude in 250 iterations for the zero-swirl case, which is typical for a ADI scheme. Also, the presence of swirl is seen to slow down the convergence rate substantially. The convergence rate for the viscous calculation is dominated by the boundary layer near the wall, hence it is in general slower than that of inviscid calculations as is seen in Fig. 59 (for 300 iterations, the L-2 norm drops only four orders of magnitude). These results show that the convergence for viscous calculations is insensitive to swirl.

Calculations of the flow in the converging-divergent nozzle have been completed for a number of nozzle Reynolds number conditions including the inviscid case. Figures 60 and 61 compare Mach number contours for the no-swirl and the swirl cases for the inviscid and one of the low Reynolds number viscous calculations, respectively. The inviscid results are in good agreement with Dutton's calculations. The viscous case is for a Reynolds number of 7000 based on the inlet radius and inflow properties. These viscous Mach number contours indicate that the introduction of swirl primarily affects the axial velocity near the centerline as in the inviscid case, although there are some changes beginning to occur near the wall in the diverging section.

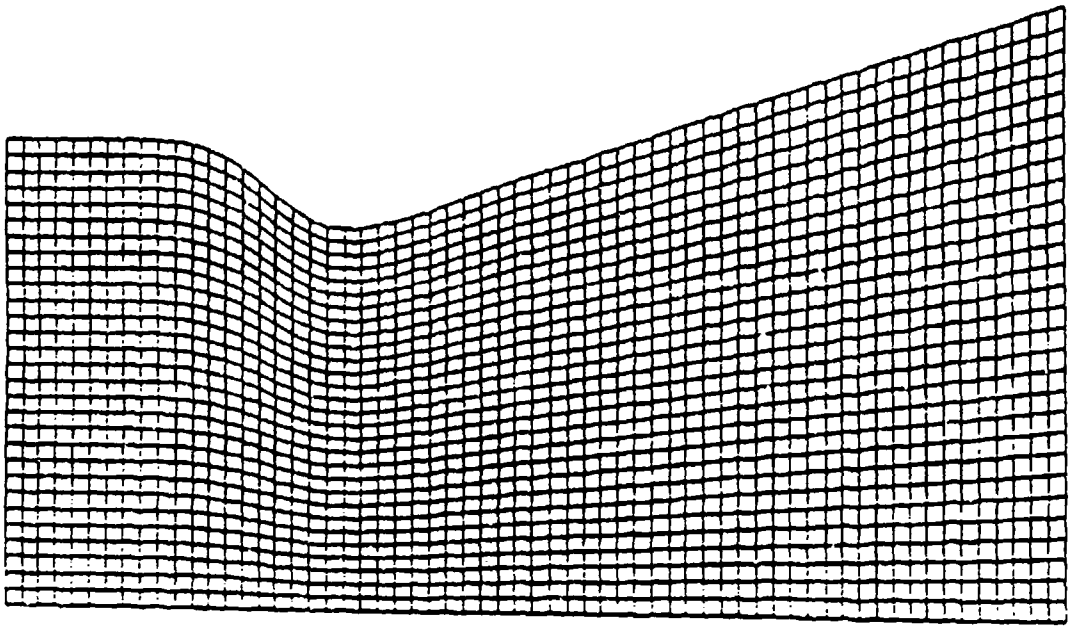


Figure 56. Inviscid grid for 35^0-18.5^0 convergent-divergent nozzle

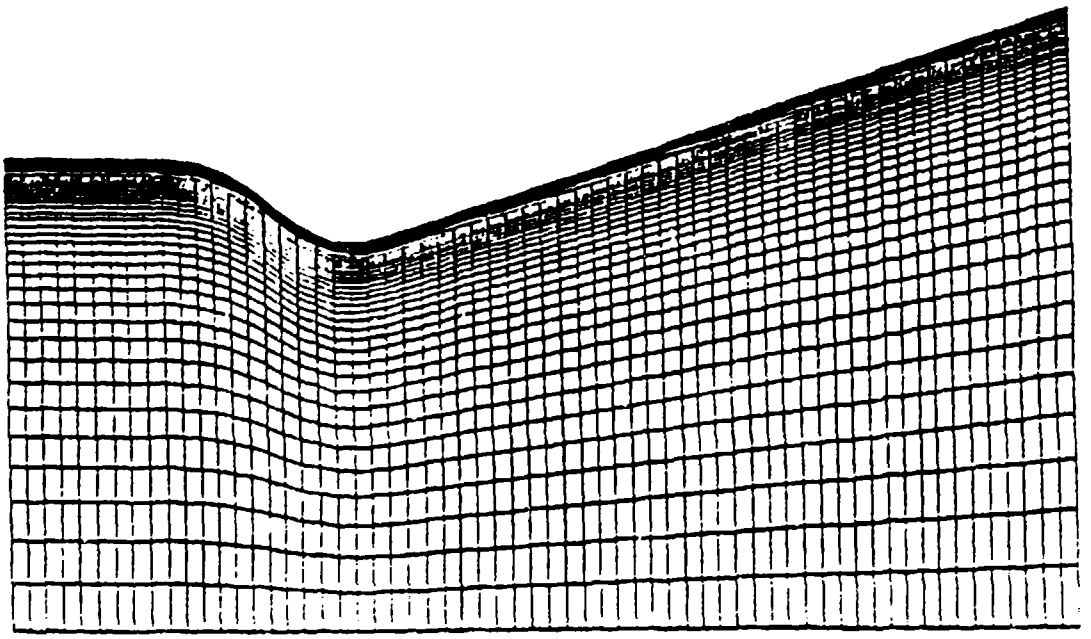


Figure 57.. Viscous grid for 35^{0}-18.5^{0} convergent-divergent nozzle

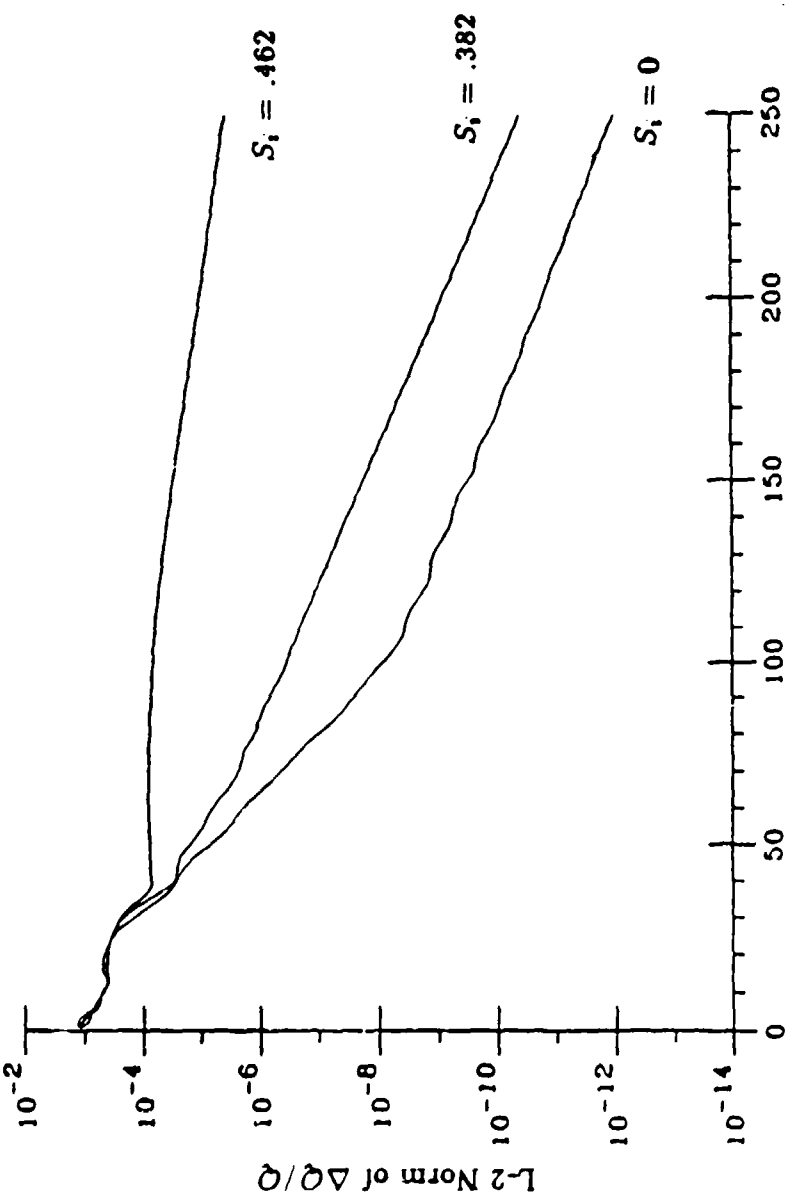


Figure 58. Convergence rates of C-D nozzle with swirl, inviscid calculation

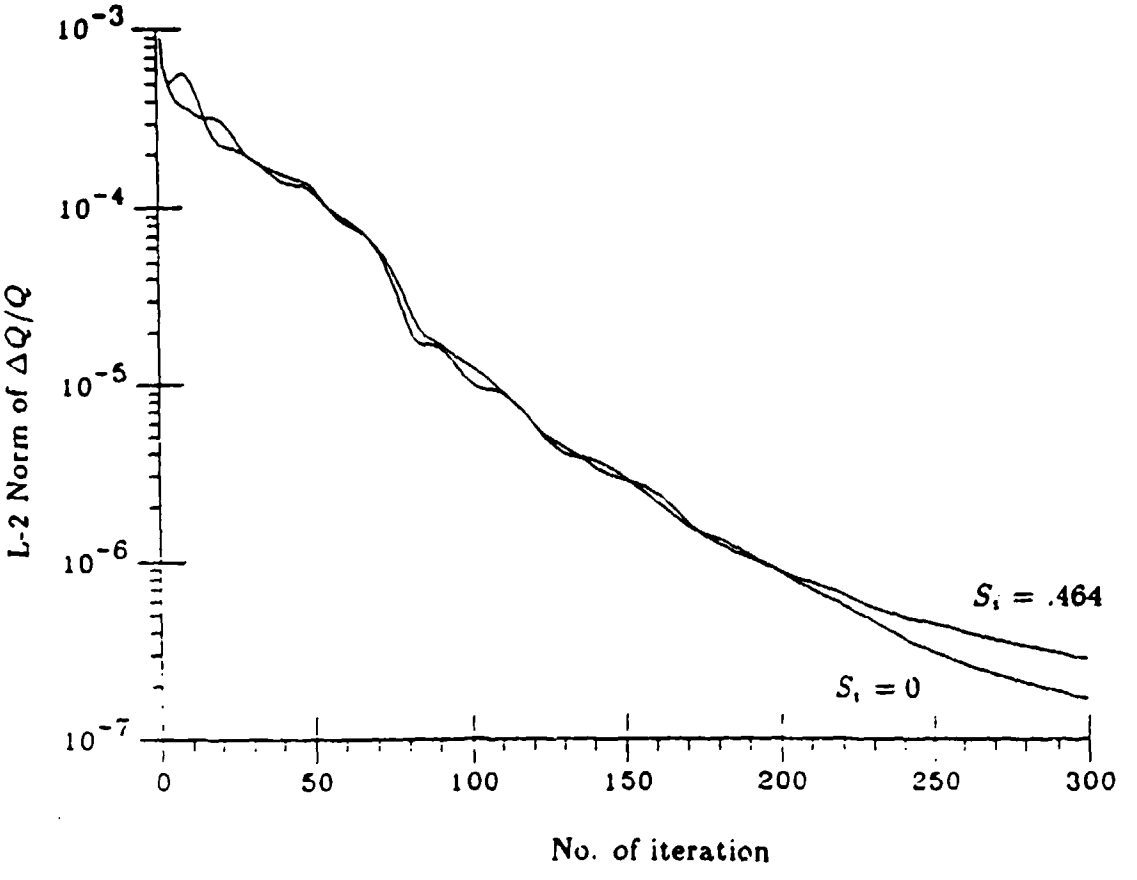
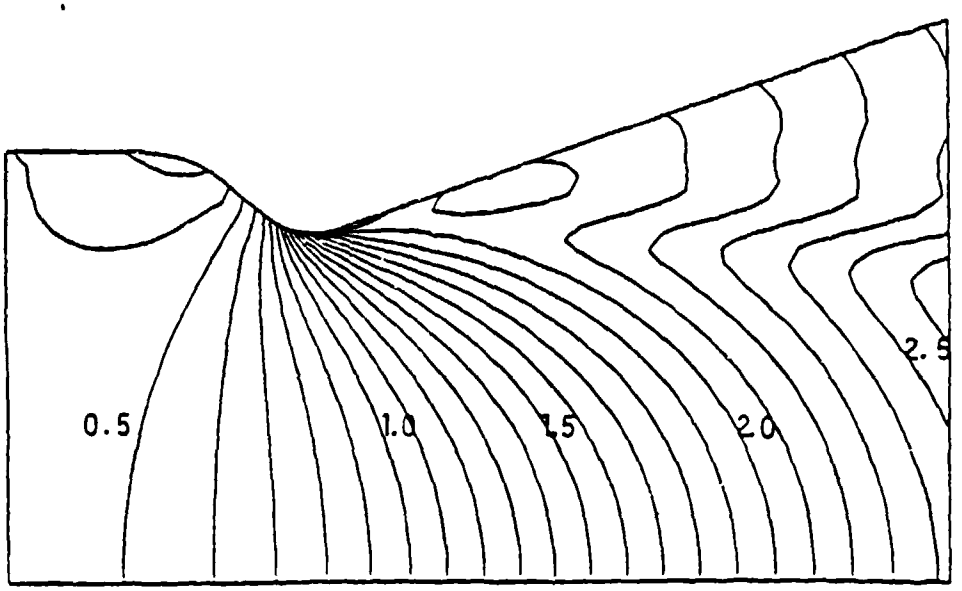
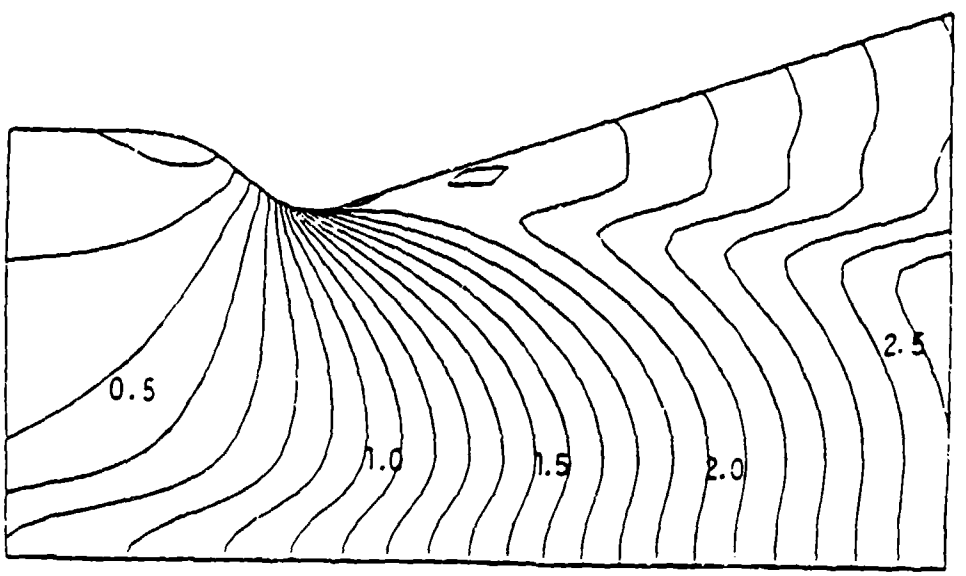


Figure 59. Convergence rates of C-D nozzle with swirl, viscous calculation

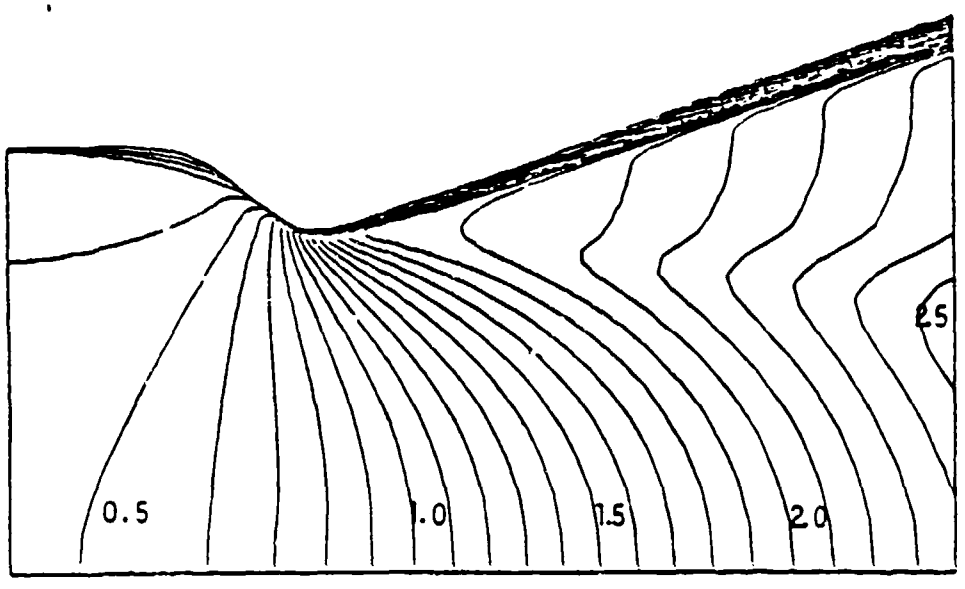


$S_t = 0$

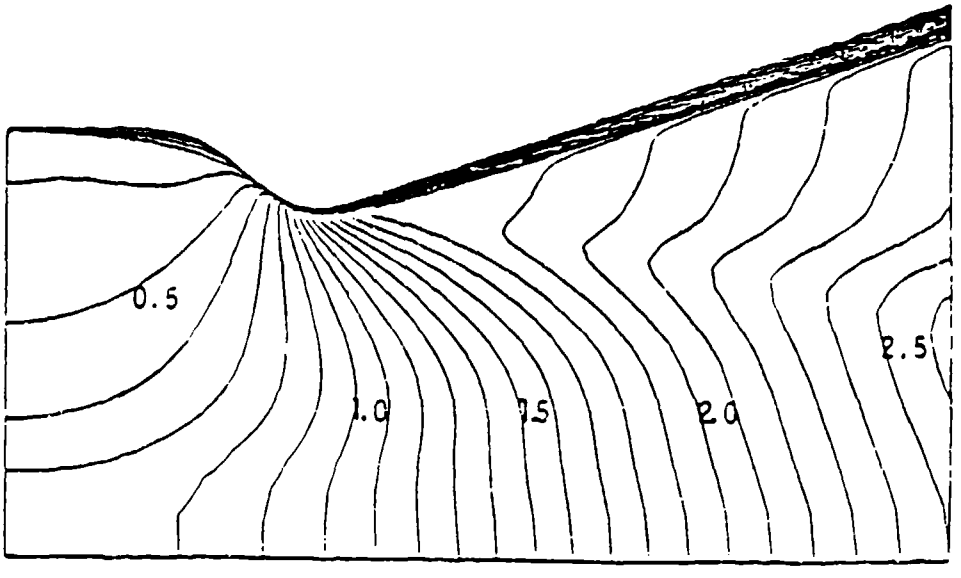


$S_t = .382$

Figure 60. Inviscid Mach number contours for C-D nozzle, with $S_t = 0$ and $S_t = .382$



$S_t = 0$



$S_t = .464$

Figure 61. Viscous Mach number contours for C-D nozzle with $Re = 7,000$, $S_t = 0$ and $S_t = .464$

The integral nozzle performance in the presence of viscosity is plotted in Fig. 62 against the swirl number. This figure shows the discharge coefficient (C_D) and the vacuum stream thrust efficiency 4 (η_{vs}) as a function of swirl numbers for free vortex, forced vortex, and constant angle inlet swirl profiles. The predicted C_D and η_{vs} values are about 2% less than those of inviscid calculations of Dutton at the Reynolds number of 7000. At a given swirl number, the reduction in C_D and η_{vs} is most prominent for the free vortex case because a relatively larger swirl angle must be specified near the centerline in order to achieve the same swirl number. This larger swirl angle sresults in a larger reduction in the mass flow rate. A similar phenomenon was also observed in Dutton's inviscid calculations [34]. The specific impulse efficiency (η_{SI}) for the viscous case is essentially constant and is similar to Dutton's inviscid results, except the value is 0.965 instead of 0.971.

The effects of Reynolds numbers are shown in Fig. 63. The computed C_D and η_{vs} values as functions of the Reynolds number are plotted for $S_i = 0$ and $S_i = 0.361$. A constant angle swirl profile was used for these computations. Asymptotic values obtained from present inviscid calculations and from Dutton's calculations are given on the right. As the Reynolds number increases, C_D and η_{vs} approach the values of inviscid calculations. These results show the degree of error incurred by making the inviscid assumption for high Reynolds number flows.

5.3.2 Convergent Nozzle and Plug Nozzle

Viscous calculations are proceeded with the elliptically contoured converging nozzle and the annular plug nozzle. A small portion of the wall has been appended to the converging nozzle after the throat, such that the flow at the downstream boundary is predominantly supersonic in the streamwise direction. This allows easier implementation of the downstream boundary conditions and does not alter the flowfield before the throat. The resulting Mach number contours are shown in

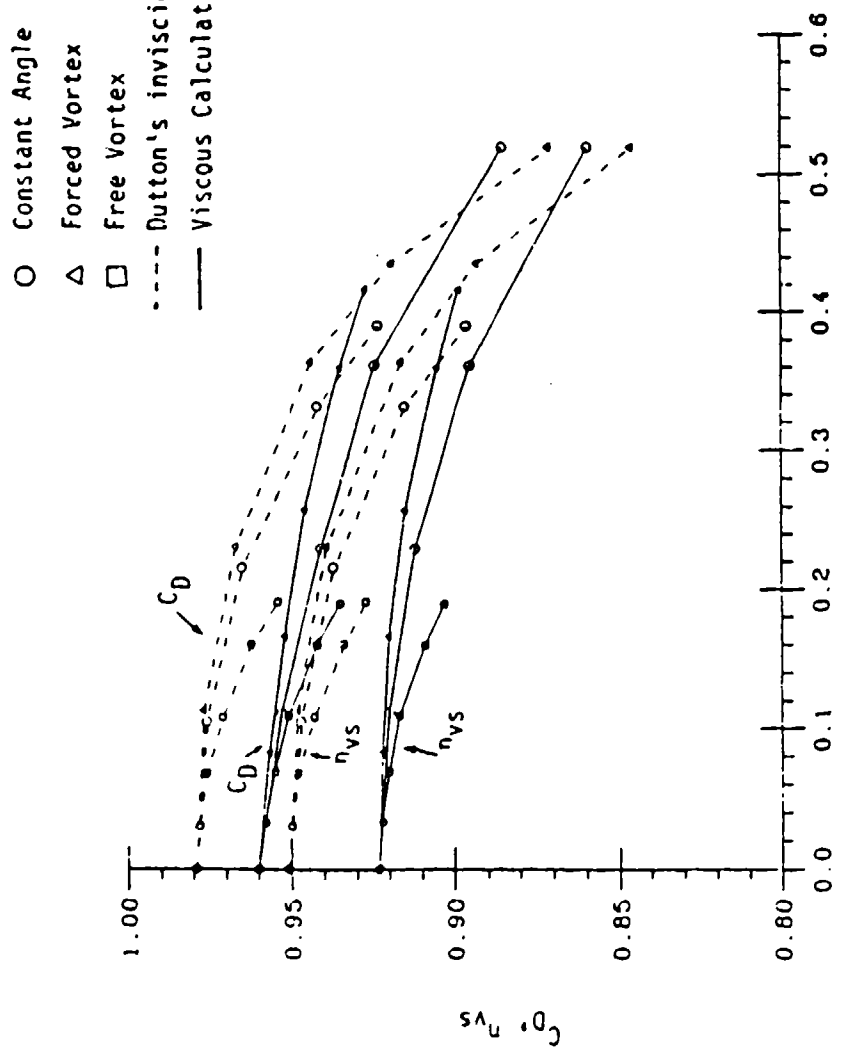


Figure 62. Dependence of integral performance on inlet swirl number for C-D nozzle

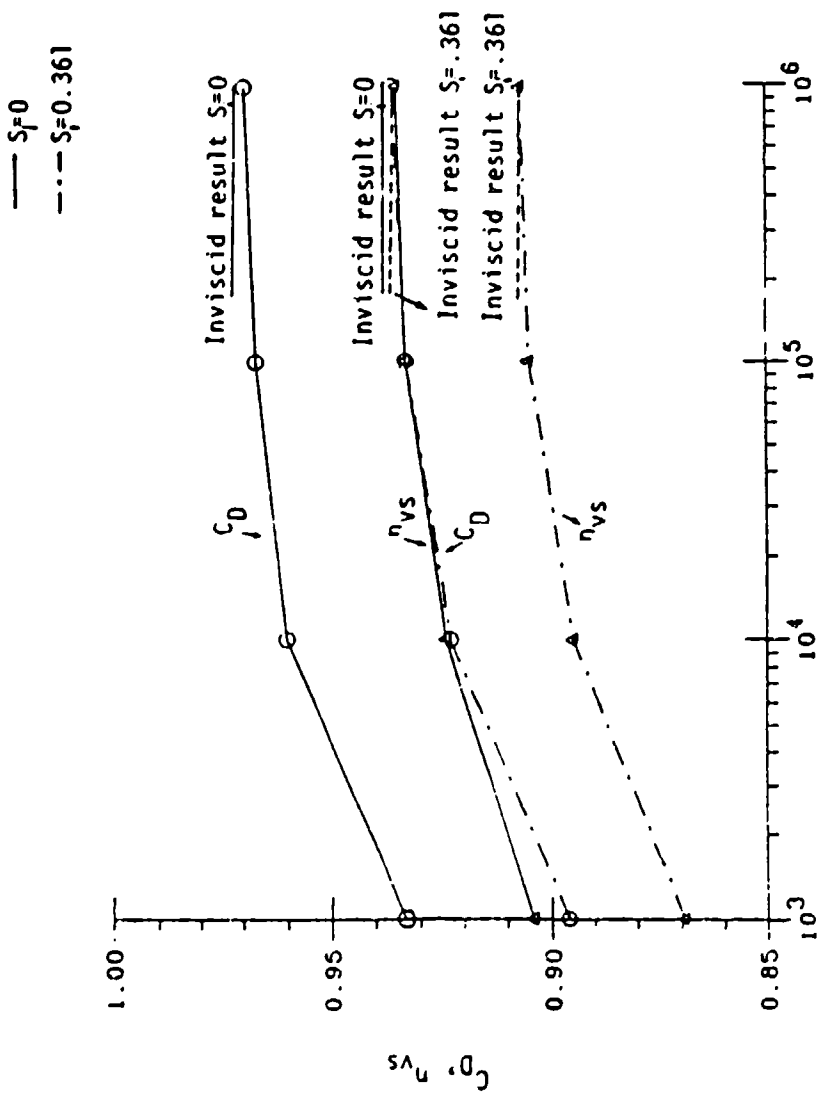


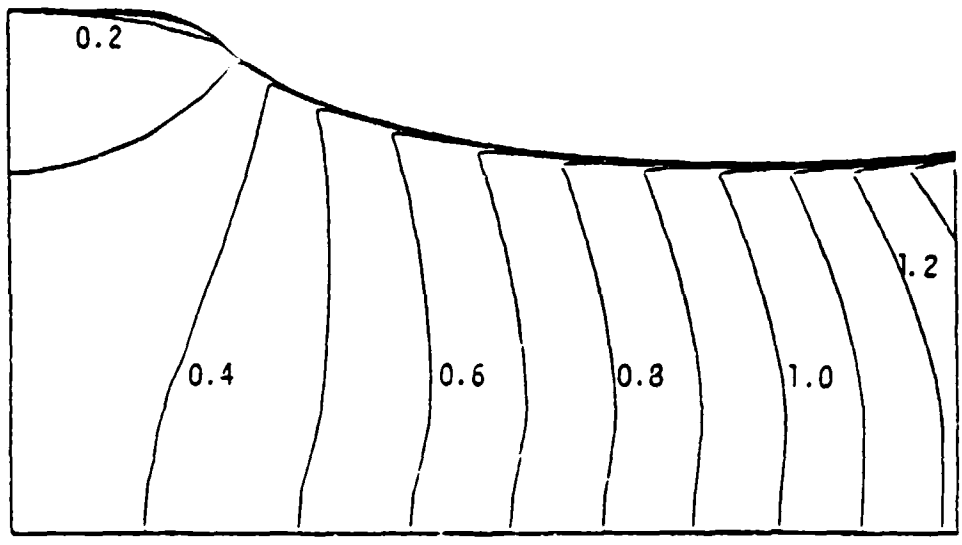
Figure 63. Dependence of nozzle performance on Reynolds number for $S_i = 0$ and $S_i = .361$

Fig. 64 for $S_i = 0$ and $S_i = 0.43$. Corresponding performance curves are plotted in Fig. 65. The predicted C_D and η_{vs} values are about 1% less than those of Dutton's inviscid calculations over the entire range of inlet swirl numbers for a Reynolds number of 1.1×10^4. The values of η_{SI} are about 0.7% less than those of inviscid results.

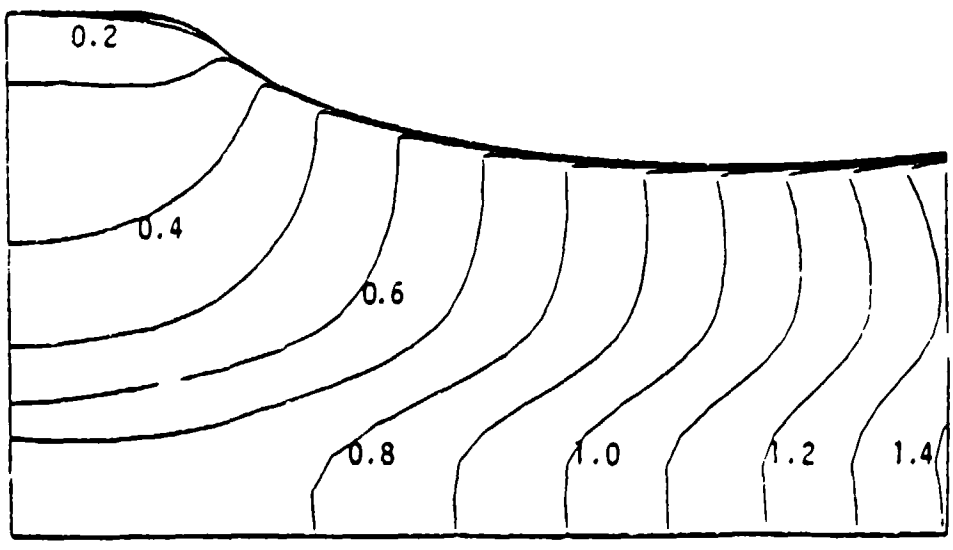
The results of the annular nozzle are shown in Fig. 66 for viscous calculations for a Reynolds number of 10^4. The flowfield of the high swirl case $S_i = 1.706$ is very different from that of the zero swirl case near the inlet region due to the combined effect of boundary layer and the inlet swirling. The total Mach number contours for this highly swirled viscous flow differ from Dutton's inviscid results due to the viscous effect on the circumferential velocity. This discrepancy demonstrates the importance of viscous analysis for low Reynolds number flows. About 4% of the reduction in C_D and η_{vs} compared to the inviscid case can be observed in Fig. 67. Again, the reduction in specific impulse efficiency for viscous calculations is less than that for the inviscid results.

5.3.3 High Expansion Nozzle

As indicated earlier, the effect of viscosity on high expansion ratio nozzles with swirl are considerably greater than that on C-D nozzles. Supersonic flows through a contoured nozzle with an expansion ratio of 272 : 1 as that given in Chapter 3 were computed by using the PNS-ADI algorithm (Eq. (3.40)). A 75×50 grid as shown in Fig. 30 was used and the Reynolds number based on inlet (throat) radius and inflow conditions was 1.4×10^5 for both laminar and turbulent calculations. The convergence rates for zero-swirl and typical swirling case are shown in Fig. 68. The results show that swirling has minor effects on the convergence rate for the PNS-ADI algorithm.



$S_i = 0$



$S_i = .43$

Figure 64. Viscous Mach number contours for convergent nozzle

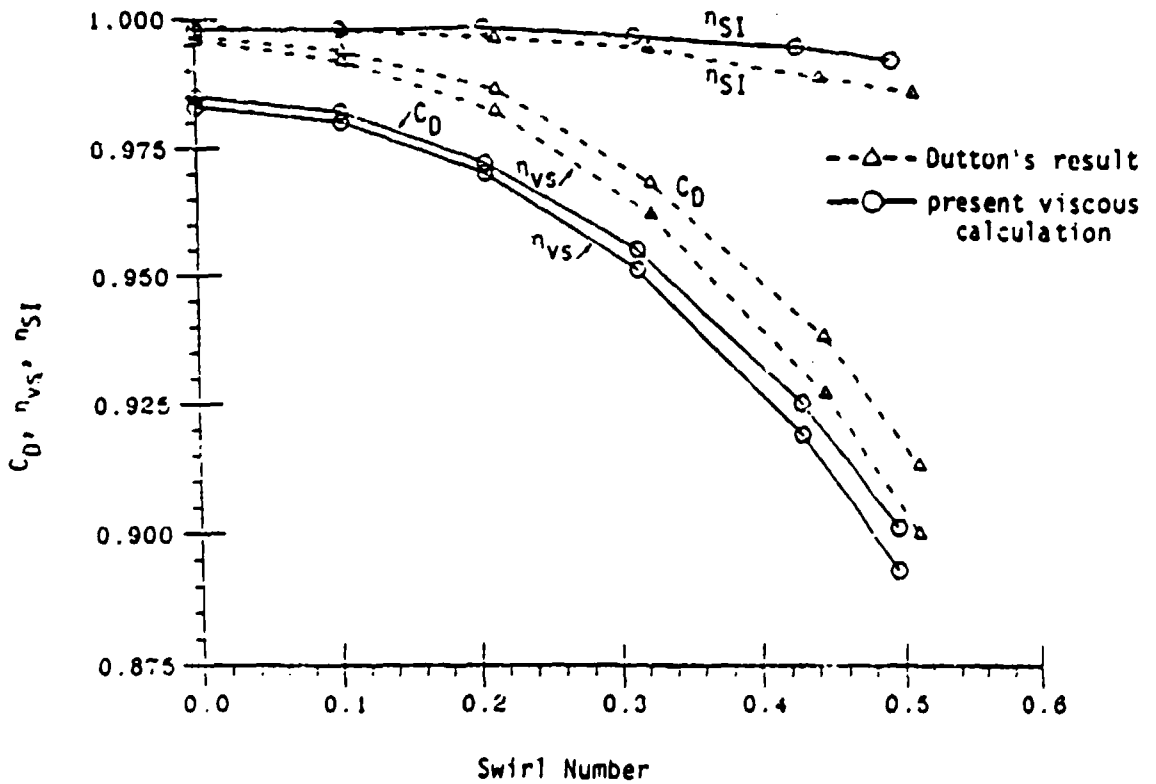
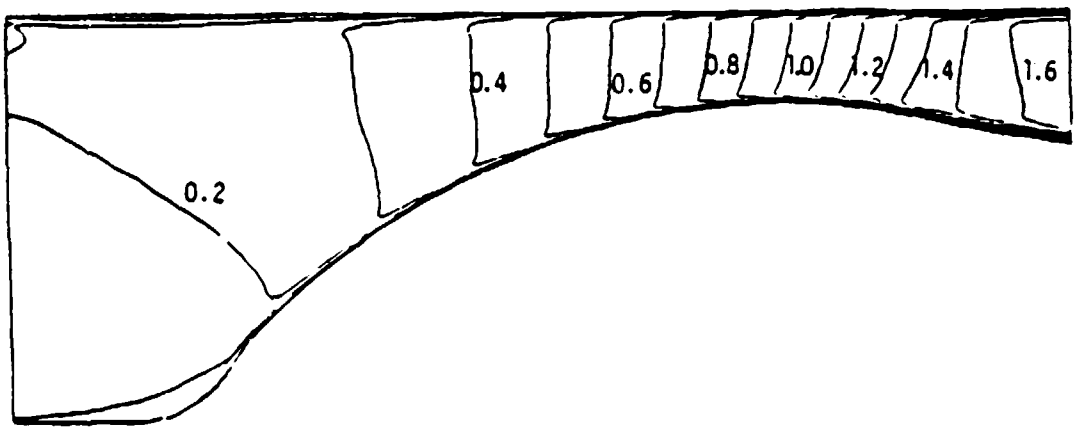
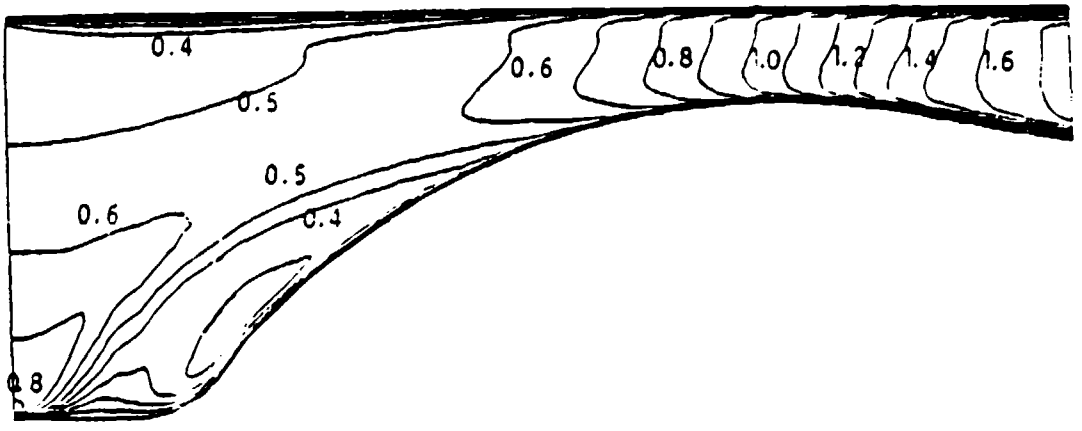


Figure 65. Integral performance of transonic flow through convergent nozzle



$S_i = 0$



$S_i = 1.706$

Figure 66. Viscous Mach number contour for annular nozzle

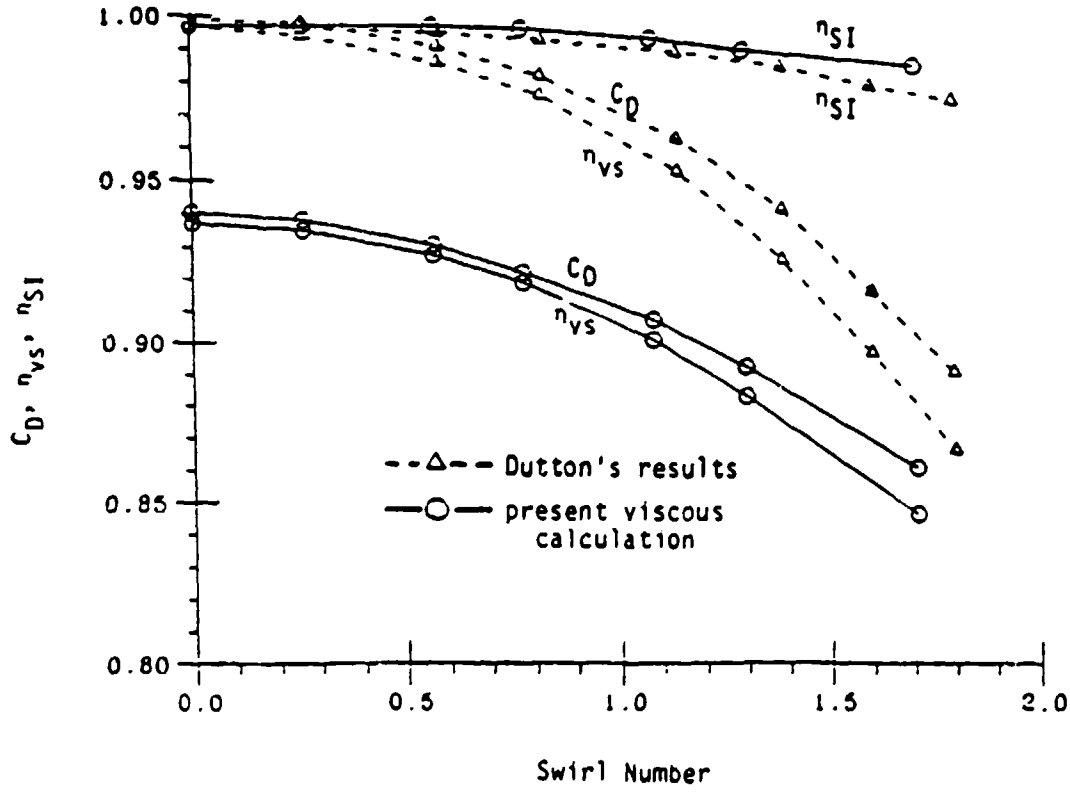


Figure 67. Integral performance of transonic flow through annular plug nozzle

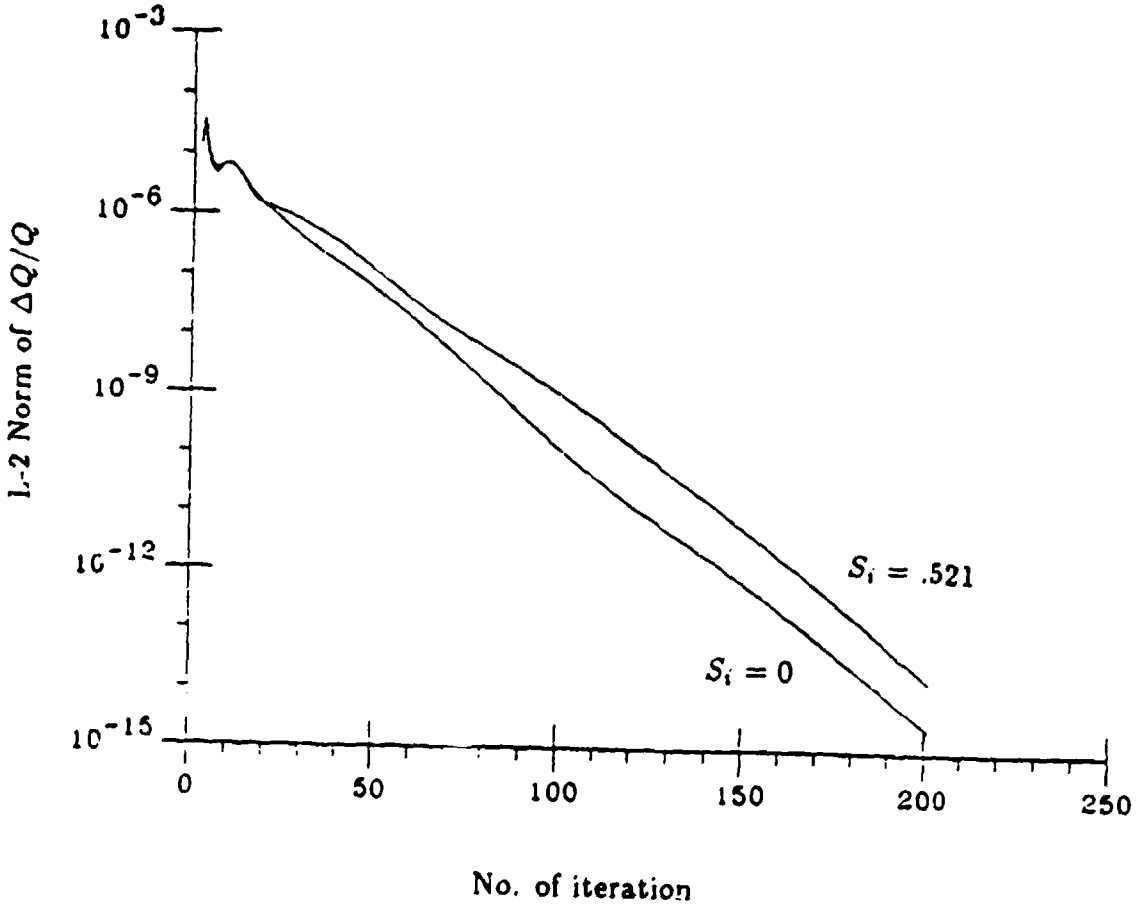


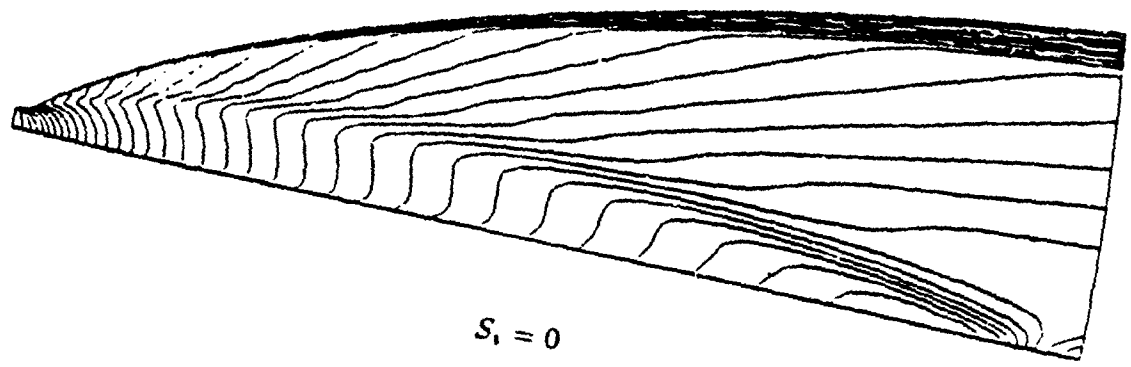
Figure 68. Convergence rates of contoured nozzle with swirl for viscous calculations

The Mach number contours from laminar results for $S_i = 0$ and $S_i = 0.521$ are shown in Fig. 69. The presence of swirl increases the axial velocity near the centerline and thus results in a shifting of iso-Mach lines and the weak oblique shock. The resulting boundary layers are thicker in these calculations than those in C-D nozzles even for a Reynolds number as high as 1.4×10^5. A much thicker boundary layer can be expected for lower Reynolds number flows where inviscid assumptions appear to be inadequate.

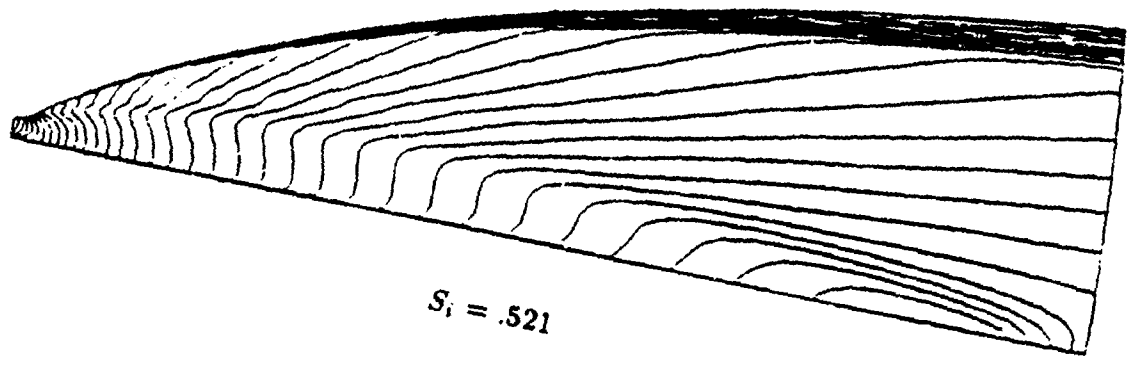
Turbulent results by using Baldwin and Lomax model [12] are shown in Fig. 70 based on the same Reynolds number and two inlet swirl numbers of 0 and 0.55. Comparing with Fig. 69, relatively thicker boundary layer is seen. Figure 71 plots C_D and η_{v_θ} as functions of S_i for both laminar and turbulent results. Large reduction in C_D and η_{v_θ} for both laminar and turbulent results can be observed. For a highly swirled flow ($S_i = 2.5$), C_D and η_{v_θ} are about 20% less than those of the zero-swirl flow, even for a moderate swirl, a 10% reduction in C_D and η_{v_θ} may occur. Slightly less reduction in C_D and η_{v_θ} for turbulent results are noted. These results demonstrate the effects of swirl on high expansion ratio nozzles are much more prominent than those on mild C-D nozzles.

5.3.4 Verification of Global Conservation

To validate the numerical algorithms, the mass flow rate at each axial location is calculated in the analysis codes. This provides a back-to-back check for global mass conservation. For the transonic results presented above, the maximum mass error has been maintained below 1%. For the more difficult high expansion nozzle case, which has the largest mass error to date, the maximum deviation is about 0.8%. This again verifies the necessity of fully conservative form for the internal flow calculations.

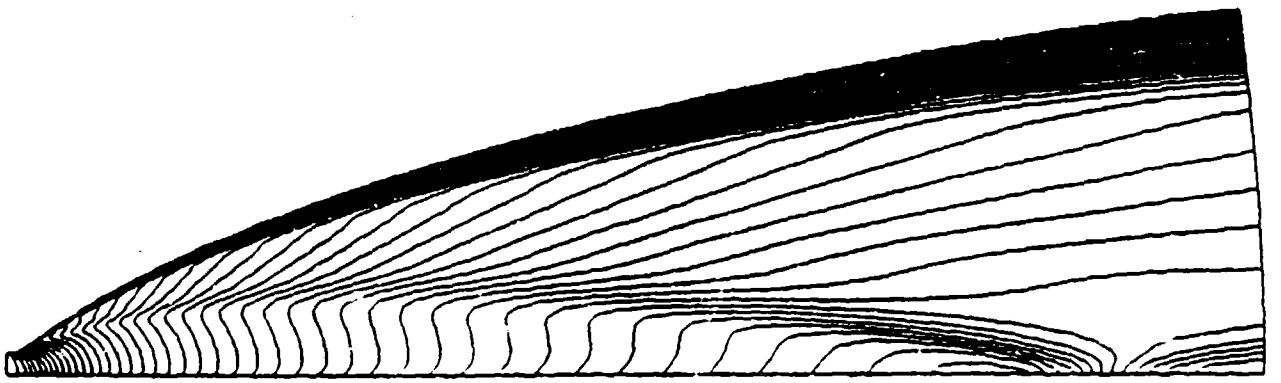


$S_i = 0$

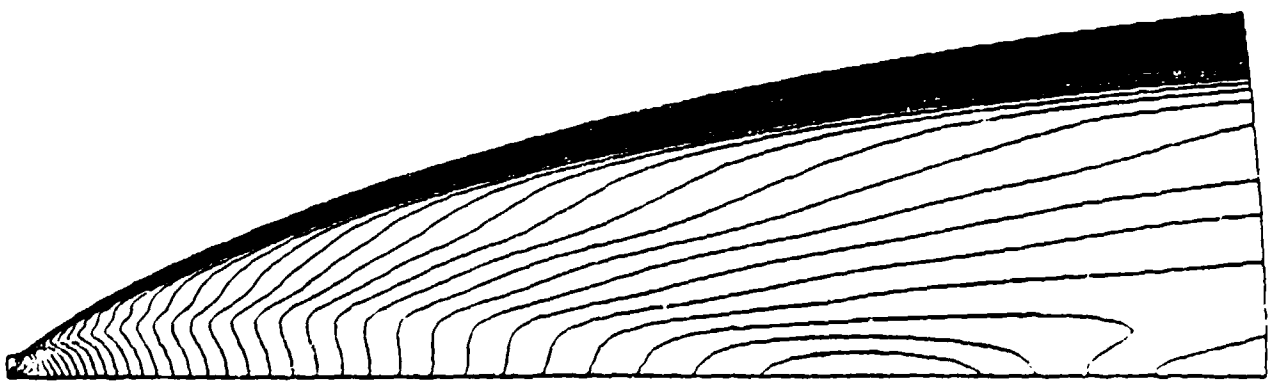


$S_i = .521$

Figure 69. Laminar Mach number contour for high expansion contoured nozzle for $S_i = 0$ and $S_i = .521$



$S_1 = 0$



$S_1 = .551$

Figure 70. Turbulent Mach number contour for high expansion contoured nozzle for $S_1 = 0$ and $S_1 = .551$

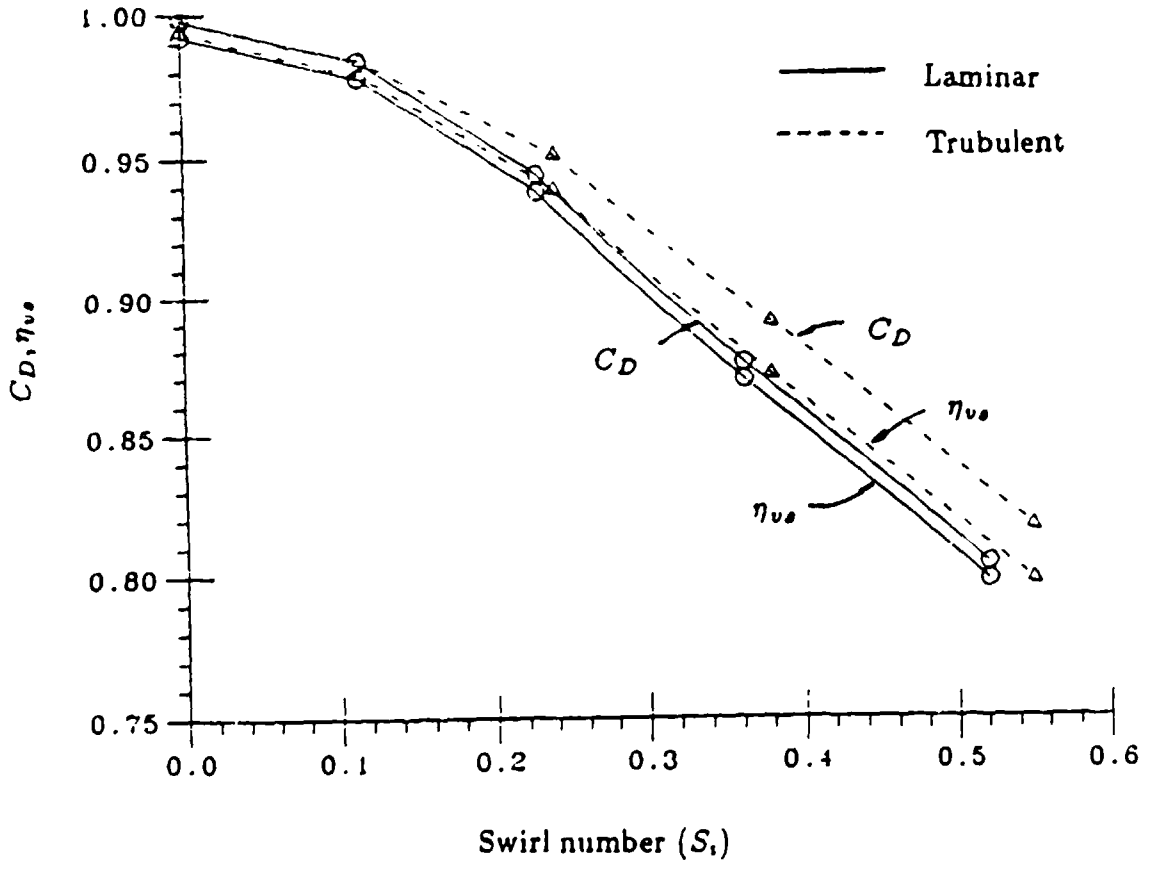


Figure 71. Integral performance for supersonic flow through contoured nozzle

CHAPTER 6

THREE-DIMENSIONAL NOZZLE FLOWS

The hybrid upwind-central algorithms proposed in Chapter 2 are extended to three-dimensional viscous supersonic calculations in this chapter. The three-dimensional thin-layer Navier-Stokes equations are simplified by neglecting the streamwise diffusion while retaining all viscous terms on the cross-stream plane. Both the Parabolized Navier-Stokes procedure and the time-iterative TLNS algorithm are studied for three-dimensional nozzle flowfield predictions. These algorithms are formulated based on the DDADI splitting for the streamwise derivative and central differencing in cross-plane derivatives. Supersonic flows through a three-dimensional nozzle with a rectangular cross section are computed for demonstration.

6.1 Governing Equations

The three-dimensional Navier-Stokes equations in a Cartesian coordinate system can be written in vector notation as

$$\frac{\partial Q}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} + \frac{\partial G}{\partial z} = \frac{\partial E_v}{\partial x} + \frac{\partial F_v}{\partial y} + \frac{\partial G_v}{\partial z} \tag{6.1}$$

where the dependent variable Q is

$$Q = (\rho, \rho u, \rho v, \rho w, e)^T.$$

The inviscid flux vectors E, F, and G are

$$E = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ (e+p)u \end{bmatrix}, \qquad F = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ (e+p)v \end{bmatrix},$$

$$G = \begin{bmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ (e+p)w \end{bmatrix}.$$

Viscous terms are included in the flux vectors E_v, F_v, and G_v as

$$E_v = \begin{bmatrix} 0 \\ \frac{4}{3}\mu\frac{\partial u}{\partial x} - \frac{2}{3}\mu(\frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}) \\ \mu(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}) \\ \mu(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z}) \\ \mu u[\frac{4}{3}\frac{\partial u}{\partial x} - \frac{2}{3}(\frac{\partial v}{\partial y} + \frac{\partial w}{\partial z})] + \mu v(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}) + \mu w(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z}) + k\frac{\partial T}{\partial x} \end{bmatrix}$$

$$F_v = \begin{bmatrix} 0 \\ \mu(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}) \\ \frac{4}{3}\mu\frac{\partial v}{\partial y} - \frac{2}{3}\mu(\frac{\partial u}{\partial x} + \frac{\partial w}{\partial z}) \\ \mu(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}) \\ \mu u(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}) + \mu v[\frac{4}{3}\frac{\partial v}{\partial y} - \frac{2}{3}(\frac{\partial u}{\partial x} + \frac{\partial w}{\partial z})] + \mu w(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}) + k\frac{\partial T}{\partial y} \end{bmatrix}$$

$$G_v = \begin{bmatrix} 0 \\ \mu(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z}) \\ \mu(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}) \\ \frac{4}{3}\mu\frac{\partial w}{\partial z} - \frac{2}{3}\mu(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}) \\ \mu u(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z}) + \mu v(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}) + \mu w[\frac{4}{3}\frac{\partial w}{\partial z} - \frac{2}{3}(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y})] + k\frac{\partial T}{\partial z} \end{bmatrix}$$

To facilitate computations on arbitrary grids, the Cartesian coordinates x, y, and z are transformed to general coordinates ξ, η, and ς according to

$$\begin{aligned} \xi &= \xi(x,y,z) \\ \eta &= \eta(x,y,z) \\ \varsigma &= \varsigma(x,y,z). \end{aligned} \tag{6.2}$$

By using the transformation defined by Eq. (6.2), Eq. (6.1) can be transformed to

$$\frac{\partial \hat{Q}}{\partial t} + \frac{\partial \hat{E}}{\partial \xi} + \frac{\partial \hat{F}}{\partial \eta} + \frac{\partial \hat{G}}{\partial \varsigma} = \frac{\partial \hat{E}_v}{\partial \xi} + \frac{\partial \hat{F}_v}{\partial \eta} + \frac{\partial \hat{G}_v}{\partial \varsigma}. \tag{6.3}$$

The dependent variable now becomes $\tilde{Q}$, which is defined by

$$\tilde{Q} = \frac{1}{J}(\rho, \rho u, \rho v, \rho w, e)^T$$

where J is the Jacobian of the coordinate transformation and can be expressed as

$$J^{-1} = x_\xi y_\eta z_\zeta + x_\zeta y_\xi z_\eta + x_\eta y_\zeta z_\xi - x_\xi y_\zeta z_\eta - x_\eta y_\xi z_\zeta - x_\zeta y_\eta z_\xi.$$

The inviscid flux vectors now become

$$\tilde{E} = \frac{1}{J}\begin{bmatrix} \rho U \\ \rho u U + \xi_x p \\ \rho v U + \xi_y p \\ \rho w U + \xi_z p \\ (e+p)U \end{bmatrix}, \qquad \tilde{F} = \frac{1}{J}\begin{bmatrix} \rho V \\ \rho u V + \eta_x p \\ \rho v V + \eta_y p \\ \rho w V + \eta_z p \\ (e+p)V \end{bmatrix},$$

$$\tilde{G} = \frac{1}{J}\begin{bmatrix} \rho W \\ \rho u W + \zeta_x p \\ \rho v W + \zeta_y p \\ \rho w W + \zeta_z p \\ (e+p)W \end{bmatrix}$$

in which the contravariant velocities U, V, and W in three dimensions are

$$U = \xi_x u + \xi_y v + \xi_z w$$

$$V = \eta_x u + \eta_y v + \eta_z w$$

$$W = \zeta_x u + \zeta_y v + \zeta_z w.$$

The transformed viscous flux vectors are defined by

$$\tilde{E}_v = (\xi_x E_v + \xi_y F_v + \xi_z G_v)/J$$

$$\tilde{F}_v = (\eta_x E_v + \eta_y F_v + \eta_z G_v)/J$$

$$\tilde{G}_v = (\zeta_x E_v + \zeta_y F_v + \zeta_z G_v)/J.$$

As mentioned in Chapter 3, the streamwise diffusion can be neglected without losing accuracy even for a fairly low Reynolds number. For three-dimensional flows inside a nozzle, the viscous terms in η and ζ directions cannot be neglected. This results in the TLNS equations in three dimensions as

$$\frac{\partial \tilde{Q}}{\partial t} + \frac{\partial \tilde{E}}{\partial \xi} + \frac{\partial \tilde{F}}{\partial \eta} + \frac{\partial \tilde{G}}{\partial \zeta} = \frac{\partial \tilde{F}_v}{\partial \eta} + \frac{\partial \tilde{G}_v}{\partial \zeta}. \tag{6.4}$$

The viscous terms on the right hand side can be further rearranged to

$$\frac{\partial \tilde{F}_v}{\partial \eta} + \frac{\partial \tilde{G}_v}{\partial \zeta} = \frac{\partial \tilde{F}_\eta}{\partial \eta} + \frac{\partial \tilde{F}_\zeta}{\partial \eta} + \frac{\partial \tilde{G}_\eta}{\partial \zeta} + \frac{\partial \tilde{G}_\zeta}{\partial \zeta}$$

where $\tilde{F}_\eta$ and $\tilde{G}_\zeta$ contain pure second order derivatives with respect to η and ζ, respectively. All cross derivatives are included in $\tilde{F}_\zeta$ and $\tilde{G}_\eta$. The viscous flux vector $\tilde{F}_\eta$ can be expressed as

$$\tilde{F}_\eta = \begin{bmatrix} 0 \\ \alpha_1 \frac{\partial u}{\partial \eta} + \alpha_2 \frac{\partial v}{\partial \eta} + \alpha_3 \frac{\partial w}{\partial \eta} \\ \alpha_2 \frac{\partial u}{\partial \eta} + \alpha_4 \frac{\partial v}{\partial \eta} + \alpha_5 \frac{\partial w}{\partial \eta} \\ \alpha_3 \frac{\partial u}{\partial \eta} + \alpha_5 \frac{\partial v}{\partial \eta} + \alpha_6 \frac{\partial w}{\partial \eta} \\ \alpha_0 \frac{\partial}{\partial \eta} \frac{e}{\rho} + \frac{\alpha_1 - \alpha_0}{2} \frac{\partial u^2}{\partial \eta} + \frac{\alpha_4 - \alpha_0}{2} \frac{\partial v^2}{\partial \eta} + \frac{\alpha_6 - \alpha_0}{2} \frac{\partial w^2}{\partial \eta} \\ + \alpha_2 \frac{\partial uv}{\partial \eta} + \alpha_3 \frac{\partial uw}{\partial \eta} + \alpha_5 \frac{\partial vw}{\partial \eta} \end{bmatrix}$$

with

$$\alpha_0 = \frac{\gamma k (\eta_x^2 + \eta_y^2 + \eta_z^2)}{J C_v}, \qquad \alpha_1 = \frac{\mu}{J}\left(\tfrac{4}{3}\eta_x^2 + \eta_y^2 + \eta_z^2\right), \qquad \alpha_2 = \frac{\mu}{J}\frac{\eta_x \eta_y}{3},$$

$$\alpha_3 = \frac{\mu}{J}\frac{\eta_x \eta_z}{3}, \qquad \alpha_4 = \frac{\mu}{J}\left(\eta_x^2 + \tfrac{4}{3}\eta_y^2 + \eta_z^2\right), \qquad \alpha_5 = \frac{\mu}{J}\frac{\eta_y \eta_z}{3},$$

$$\alpha_6 = \frac{\mu}{J}\left(\eta_x^2 + \eta_y^2 + \tfrac{4}{3}\eta_z^2\right).$$

The vector $\tilde{G}_\zeta$ can be obtained by replacing all η's in the expression of $\tilde{F}_\eta$ with ζ's.

The vectors $\tilde{F}_\zeta$ and $\tilde{G}_\zeta$ are

$$\tilde{F}_\zeta = \begin{bmatrix} 0 \\ \gamma_1 \frac{\partial u}{\partial \zeta} + \gamma_2 \frac{\partial v}{\partial \zeta} + \gamma_3 \frac{\partial w}{\partial \zeta} \\ \gamma_4 \frac{\partial u}{\partial \zeta} + \gamma_5 \frac{\partial v}{\partial \zeta} + \gamma_6 \frac{\partial w}{\partial \zeta} \\ \gamma_7 \frac{\partial u}{\partial \zeta} + \gamma_8 \frac{\partial v}{\partial \zeta} + \gamma_9 \frac{\partial w}{\partial \zeta} \\ \gamma_0 \frac{\partial}{\partial \zeta} \frac{e}{\rho} + \frac{\gamma_1 - \gamma_0}{2} \frac{\partial u^2}{\partial \zeta} + \frac{\gamma_4 - \gamma_0}{2} \frac{\partial v^2}{\partial \zeta} + \frac{\gamma_9 - \gamma_0}{2} \frac{\partial w^2}{\partial \zeta} \\ + \gamma_2 u \frac{\partial v}{\partial \zeta} + \gamma_4 v \frac{\partial u}{\partial \zeta} + \gamma_3 u \frac{\partial w}{\partial \zeta} + \gamma_7 w \frac{\partial u}{\partial \zeta} \\ + \gamma_6 v \frac{\partial w}{\partial \zeta} + \gamma_8 w \frac{\partial v}{\partial \zeta} \end{bmatrix}$$

and

$$\tilde{G}_\eta = \begin{bmatrix} 0 \\ \gamma_1 \frac{\partial u}{\partial \eta} + \gamma_4 \frac{\partial v}{\partial \eta} + \gamma_7 \frac{\partial w}{\partial \eta} \\ \gamma_2 \frac{\partial u}{\partial \eta} + \gamma_5 \frac{\partial v}{\partial \eta} + \gamma_8 \frac{\partial w}{\partial \eta} \\ \gamma_3 \frac{\partial u}{\partial \eta} + \gamma_6 \frac{\partial v}{\partial \eta} + \gamma_9 \frac{\partial w}{\partial \eta} \\ \gamma_0 \frac{\partial}{\partial \eta} \frac{e}{\rho} + \frac{\gamma_1 - \gamma_0}{2} \frac{\partial u^2}{\partial \eta} + \frac{\gamma_4 - \gamma_0}{2} \frac{\partial v^2}{\partial \eta} + \frac{\gamma_9 - \gamma_0}{2} \frac{\partial w^2}{\partial \eta} \\ + \gamma_4 u \frac{\partial v}{\partial \eta} + \gamma_2 v \frac{\partial u}{\partial \eta} + \gamma_7 u \frac{\partial w}{\partial \eta} + \gamma_3 w \frac{\partial u}{\partial \eta} \\ + \gamma_8 v \frac{\partial w}{\partial \eta} + \gamma_6 w \frac{\partial v}{\partial \eta} \end{bmatrix}$$

with

$$\gamma_0 = \frac{\gamma k(\eta_x\varsigma_x+\eta_y\varsigma_y+\eta_z\varsigma_z)}{JC_\mu}, \quad \gamma_1 = \frac{\mu}{J}\left(\frac{4}{3}\eta_x\varsigma_x + \eta_y\varsigma_y + \eta_z\varsigma_z\right),$$

$$\gamma_2 = \frac{\mu}{J}\left(\eta_y\varsigma_x - \frac{2}{3}\eta_x\varsigma_y\right), \quad \gamma_3 = \frac{\mu}{J}\left(\eta_z\varsigma_x - \frac{2}{3}\eta_x\varsigma_z\right),$$

$$\gamma_4 = \frac{\mu}{J}\left(\eta_x\varsigma_y - \frac{2}{3}\eta_y\varsigma_x\right), \quad \gamma_5 = \frac{\mu}{J}\left(\eta_x\varsigma_x + \frac{4}{3}\eta_y\varsigma_y + \eta_z\varsigma_z\right),$$

$$\gamma_6 = \frac{\mu}{J}\left(\eta_z\varsigma_y - \frac{2}{3}\eta_y\varsigma_z\right), \quad \gamma_7 = \frac{\mu}{J}\left(\eta_x\varsigma_z - \frac{2}{3}\eta_z\varsigma_x\right),$$

$$\gamma_8 = \frac{\mu}{J}\left(\eta_y\varsigma_z - \frac{2}{3}\eta_z\varsigma_y\right), \quad \gamma_9 = \frac{\mu}{J}\left(\eta_x\varsigma_x + \eta_y\varsigma_y + \frac{4}{3}\eta_z\varsigma_z\right).$$

With the substitution of the new expressions of Viscous terms, Eq. (6.4) becomes

$$\frac{\partial\tilde{Q}}{\partial t} + \frac{\partial\tilde{E}}{\partial\xi} + \frac{\partial\tilde{F}}{\partial\eta} + \frac{\partial\tilde{G}}{\partial\varsigma} = \frac{\partial\tilde{F}_\eta}{\partial\eta} + \frac{\partial\tilde{F}_\varsigma}{\partial\eta} + \frac{\partial\tilde{G}_\eta}{\partial\varsigma} + \frac{\partial\tilde{G}_\varsigma}{\partial\varsigma}. \tag{6.5}$$

6.2 Three-Dimensional Supersonic Algorithms

Numerical algorithms for the solution of Eq. (6.5) can be formulated in a number of ways. Based upon the results from Chapter 3, the algorithm for three-dimensional flows will be formulated according to flux-vector splitting in the streamwise direction and central-differencing in cross-stream directions. Before discussing the details of numerical algorithms for the vector governing equations, the Fourier stability analysis for a scalar modeled equation is studied.

6.2.1 Stability Analysis of the Scalar Equation

The three-dimensional Burger's equation,

$$\frac{\partial u}{\partial t} + a^+\frac{\partial u}{\partial x} + a^-\frac{\partial u}{\partial x} + b\frac{\partial u}{\partial y} + c\frac{\partial u}{\partial z} = \mu\left(\frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right) \tag{6.6}$$

is chosen as the modeled equation. This modeled equation implies only the streamwise (x) direction is flux-vector split, while the remaining derivatives on the cross plane (y and z directions) are to be evaluated according to central differences.

For simplicity, we restrict only to first-order upwind differencing for the discussion of stability analysis although the numerical computations shown later are

based on second-order upwind differencing. By using the line Gauss-Seidel version for the DDADI method, the discretized equation for Eq. (6.6) can be expressed as a forward marching,

$$[d + \Delta t(b\frac{\partial}{\partial y} - \mu\frac{\partial^2}{\partial y^2}) + \Delta t(c\frac{\partial}{\partial z} - \mu\frac{\partial^2}{\partial z^2})]\Delta u^* = -\Delta t r^n \qquad (6.7)$$

and a backward marching,

$$[d + \Delta t(b\frac{\partial}{\partial y} - \mu\frac{\partial^2}{\partial y^2}) + \Delta t(c\frac{\partial}{\partial z} - \mu\frac{\partial^2}{\partial z^2})]\Delta u = -\Delta t r^*. \qquad (6.8)$$

In Eqs. (6.7) and (6.8), the right hand side residuals r^n and r^* are

$$r^n = a^+\frac{u^n_{i,j,k} - u^*_{i-1,j,k}}{\Delta x} + [a^-\frac{\partial u}{\partial x} + b\frac{\partial u}{\partial y} + c\frac{\partial u}{\partial z} - \mu(\frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2})]^n$$

and

$$r^* = a^-\frac{u^{n+1}_{i+1,j,k} - u^*_{i,j,k}}{\Delta x} + [a^+\frac{\partial u}{\partial x} + b\frac{\partial u}{\partial y} + c\frac{\partial u}{\partial z} - \mu(\frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2})]^*.$$

and the quantity d is the diagonal element defined by

$$d = 1 + \frac{\Delta t}{\Delta x}(a^+ - a^-).$$

Equations (6.7) and (6.8) are based on a straightforward extension of the two-dimensional algorithm. This implies that the y and z derivatives (from both inviscid and viscous terms) are treated implicitly. Consequently, the resulting left hand side matrices of Eqs. (6.7) and (6.8) are very expansive to solve due to their high bandwidth structure.

A more practical way to solve these two equations is to factorize the left hand side operators of Eqs. (6.7) and (6.8). This results in the factored forward marching,

$$[d + \Delta t(b\frac{\partial}{\partial y} - \mu\frac{\partial^2}{\partial y^2})]d^{-1}[d + \Delta t(c\frac{\partial}{\partial z} - \mu\frac{\partial^2}{\partial z^2})]\Delta u^* = -\Delta t r^n \qquad (6.9)$$

and the factored backward marching,

$$[d + \Delta t(b\frac{\partial}{\partial y} - \mu\frac{\partial^2}{\partial y^2})][d^{-1}][d + \Delta t(c\frac{\partial}{\partial z} - \mu\frac{\partial^2}{\partial z^2})]\Delta u = -\Delta t r^*. \qquad (6.10)$$

The solutions of these factored equations require alternating sweeps in y and z directions, each sweep involves only consecutive solutions of a scalar tri-diagonal matrix for the present scalar modeled equation. In the vector governing equations of interest, the factored algorithm requires the solutions of a block tri-diagonal matrix with a block size of 5×5.

The amplification factor for the unfactored forward marching Eq. (6.7) is then

$$g_u^* \equiv \frac{u^*}{u^n} = \frac{D_u^*}{C_u^*}$$

where, D_u^* and C_u^* are

$$D_u^* = 1 - \sigma_x^-(\cos\omega_x + i\sin\omega_x)$$

and

$$C_u^* = 1 + \sigma_x^+(1 - \cos\omega_x + i\sin\omega_x) - \sigma_x^- + i\sigma_y\sin\omega_y + i\sigma_z\sin\omega_z$$
$$+ 2\nu_y(1 - \cos\omega_y) + 2\nu_z(1 - \cos\omega_z).$$

and that for the unfactored backward marching Eq. (6.8) is

$$g_u^{**} \equiv \frac{u^{n+1}}{u^*} = \frac{D_u^{**}}{C_u^{**}}$$

where

$$D_u^{**} = 1 + \sigma_x^+(\cos\omega_x + i\sin\omega_x)$$

and

$$C_u^{**} = 1 + \sigma_x^+ + \sigma_x^-(\cos\omega_x + i\sin\omega_x - 1) + i\sigma_y\sin\omega_y + i\sigma_z\sin\omega_z$$
$$+ 2\nu_y(1 - \cos\omega_y) + 2\nu_z(1 - \cos\omega_z).$$

The parameters σ_x^+, σ_x^-, σ_y, and σ_z are CFL numbers defined by

$$\sigma_x^+ = \frac{a^+\Delta t}{\Delta x}, \qquad \sigma_x^- = \frac{a^-\Delta t}{\Delta x}$$
$$\sigma_y = \frac{b\Delta t}{\Delta y}, \qquad \sigma_z = \frac{c\Delta t}{\Delta z}$$

and ν_y and ν_z are von Neumann numbers defined by

$$\nu_y = \frac{\mu \Delta t}{\Delta y^2}, \qquad \nu_z = \frac{\mu \Delta t}{\Delta z^2}.$$

The wavenumbers in x, y, and z directions are represented by ω_x, ω_y, and ω_z, respectively. The overall amplification factor for the unfactored method is then $g_u = g_u^{\bullet} g_u^{\bullet\bullet}$.

The amplification factors for the factored forward and backward marching are

$$g_f^{\bullet} = \frac{D_u^{\bullet} + C_f}{C_u^{\bullet} + C_f}$$

and

$$g_f^{\bullet\bullet} = \frac{D_u^{\bullet\bullet} + C_f}{C_u^{\bullet\bullet} + C_f},$$

respectively. The quantity C_f is due to the approximate factorization and is

$$C_f = (1 + \sigma_x^+ - \sigma_x^-)[i\sigma_y \sin\omega_y + 2\nu_y(1 - \cos\omega_y)][i\sigma_z \sin\omega_z + 2\nu_z(1 - \cos\omega_z)].$$

The overall amplification factor for the factored method is $g_f = g_f^{\bullet} g_f^{\bullet\bullet}$.

Similar to the discussion in Section 3.3.5, the stability results are presented for two special cases, they are

1. subsonic :$\sigma_x^+ = -\sigma_x^- = \sigma_y = \sigma_z = \nu_y = \nu_z = CFL$
2. supersonic:$\sigma_x^+ = \sigma_y = \sigma_z = \nu_y = \nu_z = CFL, \qquad \sigma_x^- = 0.$

Here, the first case simulates the subsonic flow, while the second case is analogous to the supersonic flow.

The results for the first case with a CFL number of 10 are shown in Fig. 72 and Fig. 73 for the unfactored and factored schemes, respectively. These two figures plot the amplification factors versus the wavenumber ω_y and ω_z for three typical x-direction wavenumbers, $\omega_x = 0$, $\omega_x = \pi/2$, and $\omega_x = \pi$. As can be seen, the approximate factorization of the factored scheme results in a much higher amplification factor near the high wavenumber region as compared to the unfactored

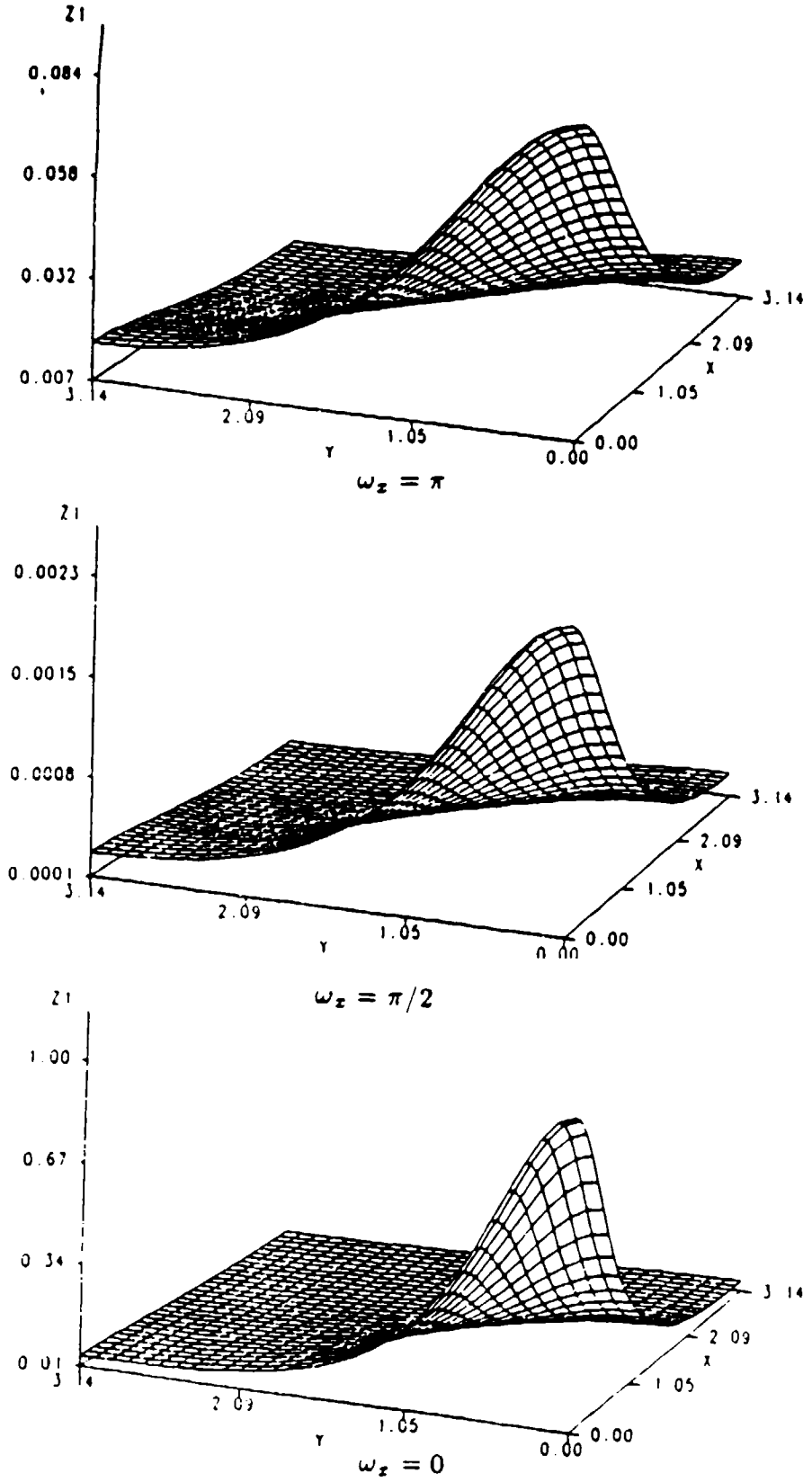


Figure 72. Stability results of 3-D Burger's equation based on unfactored scheme for subsonic case

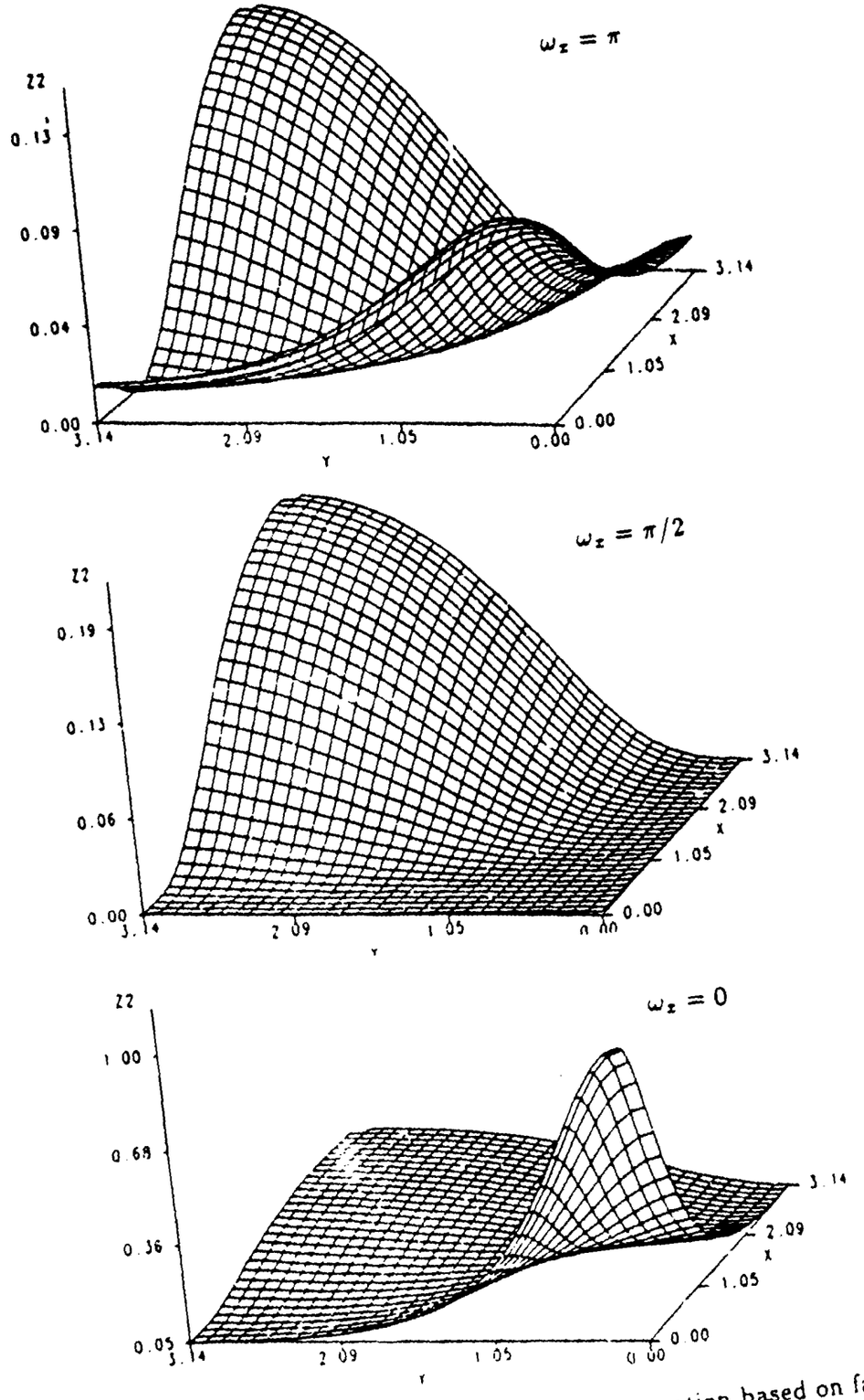


Figure 73. Stability results of 3-D Burger's equation based on factored scheme for subsonic case

case. Therefore, the factored scheme is expected to give much slower convergence than the unfactored scheme. The stability results for the second case are shown in Fig. 74 and Fig. 75. Similar effects of approximate factorization as in the first case are clearly shown.

The results above indicate that both the factor and unfactored three-dimensional DDADI schemes are unconditionally stable for the three-dimensional Burger's equation. Furthermore, the necessary approximate factorization for a practical three-dimensional DDADI algorithm results in increasing the eigenvalues near the high wavenumber region. This implies that convergence of the three-dimensional DDADI algorithm is inferior to that of the two-dimensional DDADI algorithm given in Section 3.4. In fact, due to similar approximate factorization required, we expect this three-dimensional algorithm will give similar convergence as that of the two-dimensional central-differenced ADI scheme.

6.2.2 Numerical Algorithms of the TLNS Equation

Similar to the formulations of the DDADI algorithm in axisymmetric two dimensional flows, we first split the streamwise flux vector $\hat{E}$ according to

$$\hat{E} = \hat{E}^+ + \hat{E}^-.$$

This splitting can be done by using either the characteristic splitting,

$$\hat{E}^+ = (M_\xi \Lambda_\xi^+ M_\xi^{-1})\hat{Q} \qquad \hat{E}^- = (M_\xi \Lambda_\xi^- M_\xi^{-1})\hat{Q},$$

or the pressure gradient splitting,

$$\hat{E}^+ = \frac{1}{J}\begin{bmatrix} \rho U \\ \rho u U + \omega \xi_x p \\ \rho v U + \omega \xi_y p \\ \rho w U + \omega \xi_z p \\ (e+p)U \end{bmatrix}, \qquad \hat{E}^- = \frac{1}{J}\begin{bmatrix} 0 \\ (1-\omega)\xi_x p \\ (1-\omega)\xi_y p \\ (1-\omega)\xi_z p \\ 0 \end{bmatrix}.$$

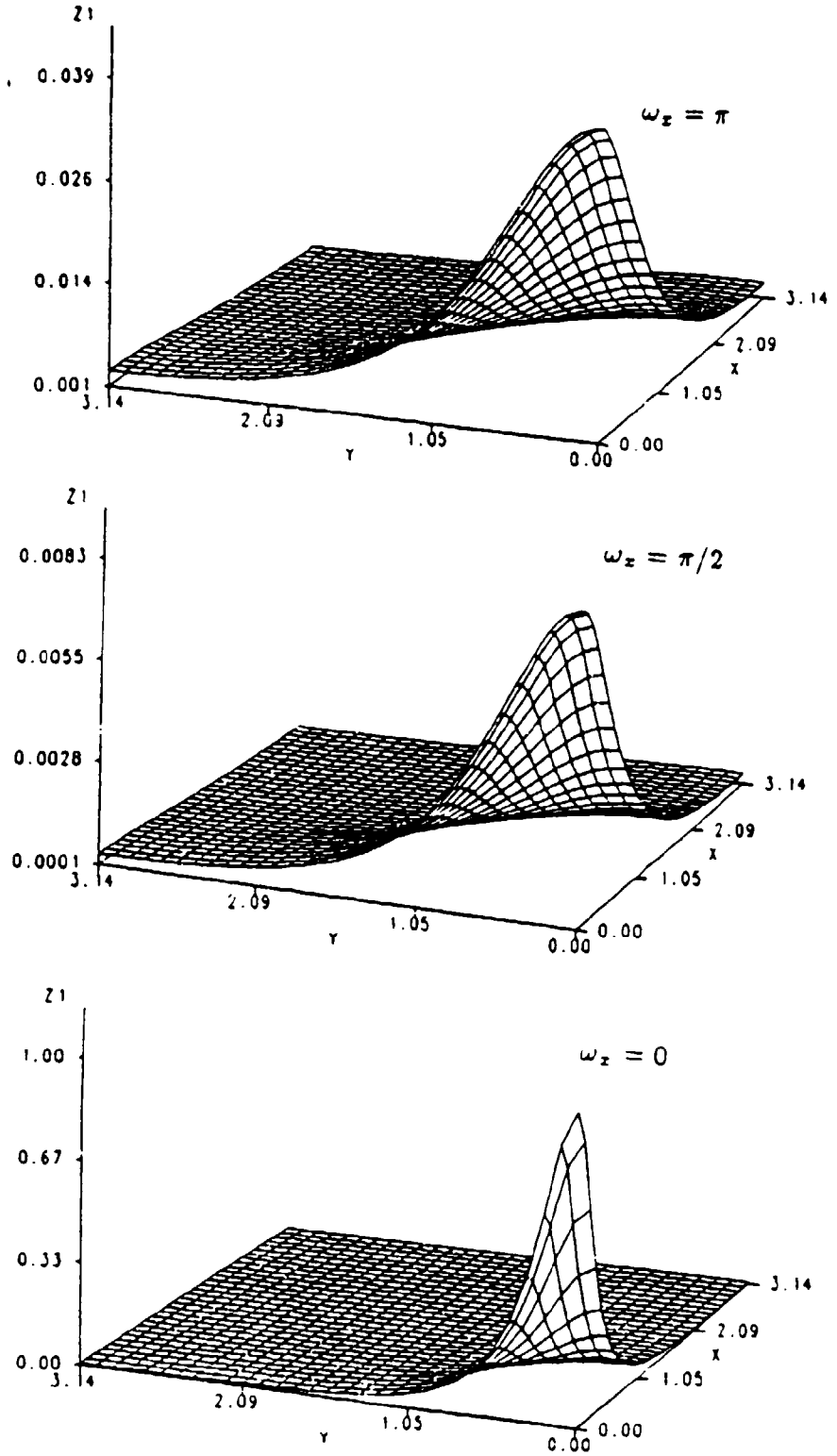


Figure 74. Stability results of 3-D Burger's equation based on unfactored scheme for supersonic case

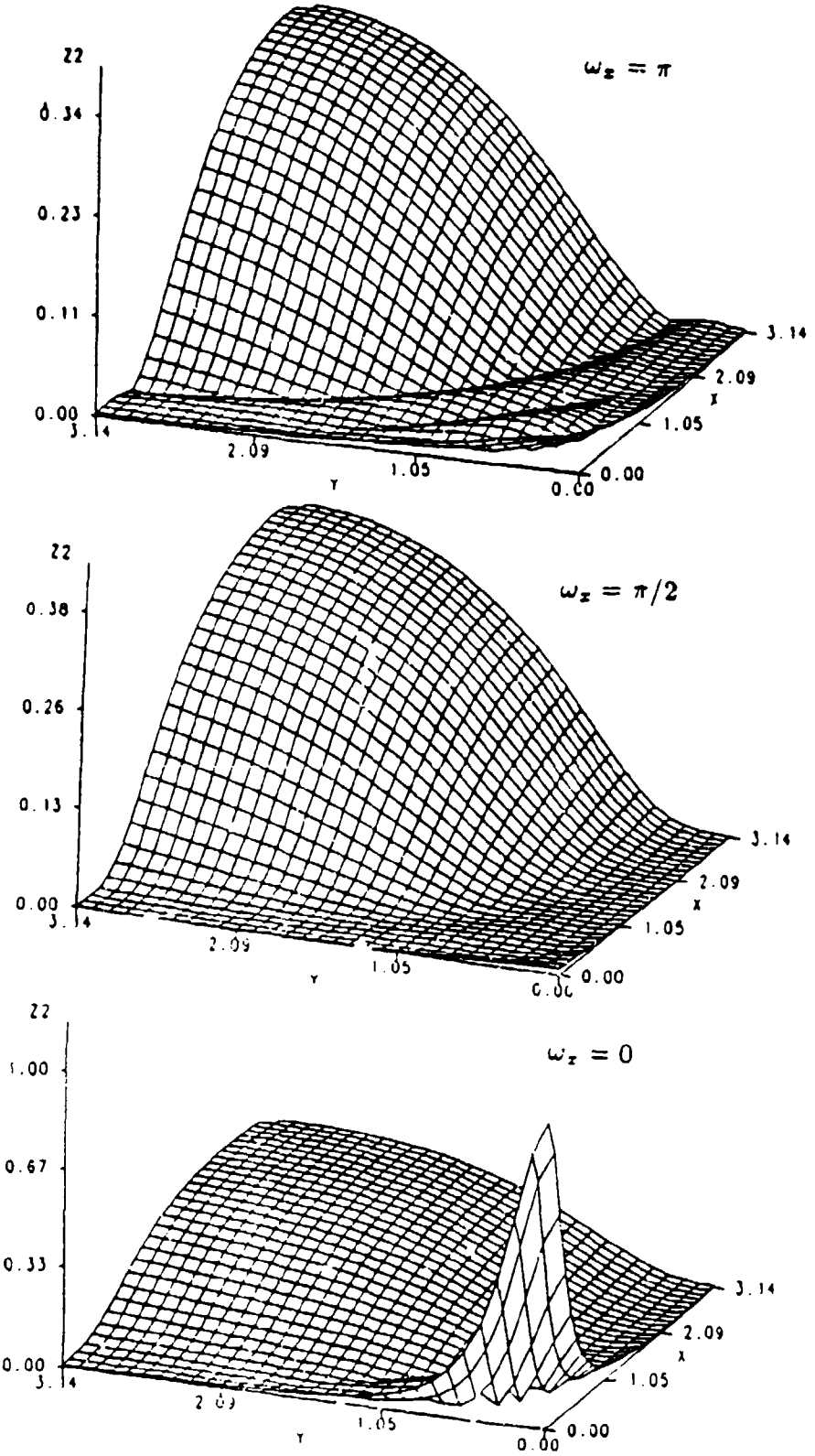


Figure 75. Stability results of 3-D Burger's equation based on factored scheme for supersonic case

The matrices M_ξ and M_ξ^{-1} are the right and the left eigenmatrices of the Jacobian matrix $A = \partial\tilde{E}/\partial\tilde{Q}$, and the diagonal matrices Λ_ξ^+ and Λ_ξ^- are

$$\Lambda_\xi^+ = \frac{\Lambda_\xi + |\Lambda_\xi|}{2}, \qquad \Lambda_\xi^- = \frac{\Lambda_\xi - |\Lambda_\xi|}{2}$$

where $\Lambda_\xi = diag(U, U, U, U + C_\xi, U - C_\xi)$ with $C_\xi = \sqrt{\xi_x^2 + \xi_y^2 + \xi_z^2}c$. The parameter ω of the pressure gradient splitting in three dimensions is equal to unity if $U \geq C_\xi$ and must satisfy

$$\omega < \frac{\gamma M_\xi^2}{1 + (\gamma - 1)M_\xi^2}$$

if $U < C_\xi$, where M_ξ is the streamwise Mach number ($M_\xi = U/C_\xi$).

The flux vectors $\tilde{E}^\pm$, $\tilde{F}$, and $\tilde{G}$ in Eq. (6.5) can be linearized by the truncated local Taylor series,

$$(\tilde{E}^\pm)^{n+1} = (\tilde{E}^\pm)^n + A^\pm \Delta\tilde{Q}$$

$$\tilde{F}^{n+1} = \tilde{F}^n + B\Delta\tilde{Q}$$

$$\tilde{G}^{n+1} = \tilde{G}^n + C\Delta\tilde{Q}$$

in which, $A^\pm$, B, and C are Jacobian matrices of $\tilde{E}^\pm$, $\tilde{F}$, and $\tilde{G}$, respectively. Viscous terms containing pure second order derivatives can be linearized according to

$$(\frac{\partial\tilde{F}_\eta}{\partial\eta})^{n+1} = (\frac{\partial\tilde{F}_\eta}{\partial\eta})^n + \frac{\partial}{\partial\eta}(\tilde{B}_v\Delta\tilde{Q})$$

$$(\frac{\partial\tilde{G}_\zeta}{\partial\zeta})^{n+1} = (\frac{\partial\tilde{G}_\zeta}{\partial\zeta})^n + \frac{\partial}{\partial\zeta}(\tilde{C}_v\Delta\tilde{Q})$$

where the viscous Jacobians $\tilde{B}_v$ and $\tilde{C}_v$ can be expressed in a similar way as in Eq. (3.10) For example, $\tilde{B}_v$ is

$$\tilde{B}_v = R_1\frac{\partial}{\partial\eta}(\frac{\partial\tilde{Q}_1}{\partial\tilde{Q}}) + R_2\frac{\partial}{\partial\eta}(\frac{\partial\tilde{Q}_2}{\partial\tilde{Q}})$$

where the matrices R_1 and R_2 are

$$R_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \alpha_1 & \alpha_2 & \alpha_3 & 0 \\ 0 & \alpha_2 & \alpha_4 & \alpha_5 & 0 \\ 0 & \alpha_3 & \alpha_5 & \alpha_6 & 0 \\ \alpha_3 & 0 & 0 & 0 & \alpha_5 \end{bmatrix}$$

$$R_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \alpha_0 & \frac{\alpha_1 - \alpha_0}{2} & \frac{\alpha_4 - \alpha_0}{2} & \frac{\alpha_6 - \alpha_0}{2} & \alpha_2 \end{bmatrix}$$

and the vectors $\tilde{Q}_1$ and $\tilde{Q}_2$ are

$$\tilde{Q}_1 = (uv, u, v, , w, vw)^T, \qquad \tilde{Q}_2 = (e/\rho, u^2, v^2, w^2, uv)^T.$$

The matrix C_v can be obtained by replacing all α's with β's in the expressions of R_1 and R_2 above. The cross-derivative viscous flux vectors in Eq. (6.5) will not be linearized because the linearization of these vectors will results in a high band-width left hand side matrix.

Direct application of the DDADI splitting to the ξ derivative of Eq. (6.5) results in the unfactored forward marching

$$[D' + \Delta t(\frac{\partial}{\partial \eta}B + \frac{\partial}{\partial \zeta}C - \frac{\partial}{\partial \eta}B_v - \frac{\partial}{\partial \zeta}C_v)]\Delta\tilde{Q}^* = -\Delta t R^n \qquad (6.11)$$

and the unfactored backward marching

$$[D' + \Delta t(\frac{\partial}{\partial \eta}B + \frac{\partial}{\partial \zeta}C - \frac{\partial}{\partial \eta}B_v - \frac{\partial}{\partial \zeta}C_v)]\Delta\tilde{Q} = -\Delta t R^* \qquad (6.12)$$

where the residuals R^n and R^* are

$$R^n = \frac{(\tilde{E}^+_{i,j,k})^n - (\tilde{E}^+_{i-1,j,k})^*}{\Delta\xi} + \kappa\frac{(\tilde{E}^+_{i,j,k})^n - 2(\tilde{E}^+_{i-1,j,k})^* + (\tilde{E}^+_{i-2,j,k})^*}{2\Delta\xi}$$
$$+ [\frac{\partial \tilde{E}^-}{\partial \xi} + \frac{\partial \tilde{F}}{\partial \eta} + \frac{\partial \tilde{G}}{\partial \zeta} - \frac{\partial \tilde{F}_v}{\partial \eta} - \frac{\partial \tilde{F}_\zeta}{\partial \eta} - \frac{\partial \tilde{G}_\eta}{\partial \zeta} - \frac{\partial \tilde{G}_\zeta}{\partial \zeta}]^n$$

and

$$R^* = \frac{(\tilde{E}^-_{i+1,j,k})^{n+1} - (\tilde{E}^-_{i,j,k})^*}{\Delta\xi} - \kappa\frac{(\tilde{E}^-_{i,j,k})^* - 2(\tilde{E}^-_{i+1,j,k})^{n+1} + (\tilde{E}^-_{i+2,j,k})^{n+1}}{2\Delta\xi}$$
$$+ [\frac{\partial\tilde{E}^+}{\partial\xi} + \frac{\partial\tilde{F}}{\partial\eta} + \frac{\partial\tilde{G}}{\partial\zeta} - \frac{\partial\tilde{F}_\eta}{\partial\eta} - \frac{\partial\tilde{F}_\zeta}{\partial\eta} - \frac{\partial\tilde{G}_\eta}{\partial\zeta} - \frac{\partial\tilde{G}_\zeta}{\partial\zeta}]^*.$$

The diagonal matrix D' is

$$D' = I + (1 + \kappa/2)\frac{\Delta t}{\Delta\xi}(A^+ - A^-)$$

where the quantity κ is 0 for first order and is 1 for second order upwind differencing in ξ.

To avoid the solution of a high band-width matrix, the left hand side operators of Eq. (6.11) and Eq. (6.12) must be further factorized into two one-dimensional operators. This results in the factored forward marching,

$$[D' + \Delta t(\frac{\partial}{\partial\eta}B - \frac{\partial}{\partial\eta}B_v)](D')^{-1}[D' + \Delta t(\frac{\partial}{\partial\zeta}C - \frac{\partial}{\partial\zeta}C_v)]\Delta\hat{Q}^* = -\Delta t R^n \quad (6.13)$$

and the factored backward marching,

$$[D' + \Delta t(\frac{\partial}{\partial\eta}B - \frac{\partial}{\partial\eta}B_v)](D')^{-1}[D' + \Delta t(\frac{\partial}{\partial\zeta}C - \frac{\partial}{\partial\zeta}C_v)]\Delta\hat{Q} = -\Delta t R^*. \quad (6.14)$$

Note that the corresponding Parabolized Navier-Stokes procedure can be obtained by neglecting the flux vector $\tilde{E}^-$ and its Jacobian A^- in Eq. (6.13). Either the traditional pressure gradient splitting or the characteristic-based splitting can be chosen to form this three-dimensional PNS algorithm. Also, the combination of Eq. (6.13) and Eq. (6.14) provides a three-dimensional TLNS solver.

To assess the numerical efficiency of these three-dimensional algorithms, the supersonic flow through a 15^0 expanding three-dimensional nozzle with rectangular cross-sections was chosen for numerical experiments. The nozzle geometry is shown in Fig. 76 and the grids on the inlet plane and the side wall surface are shown in

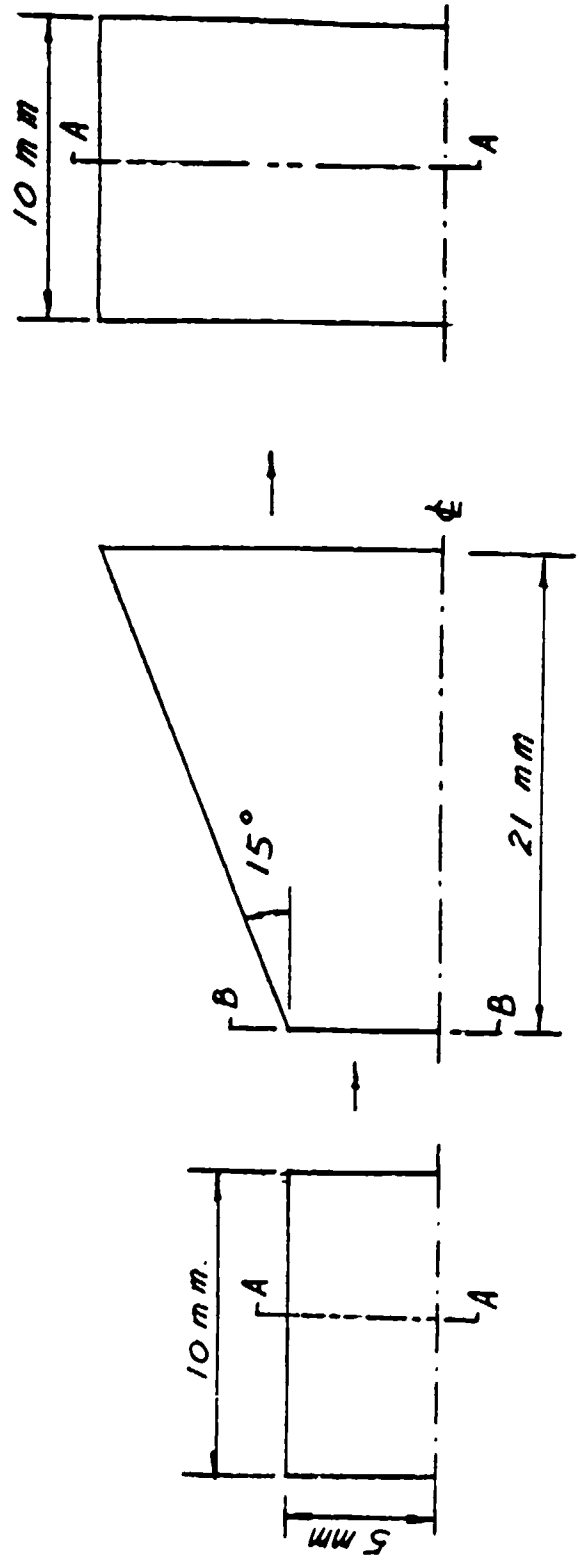


Figure 76. Geometry of 15° expanding 3-D nozzle for convergence test

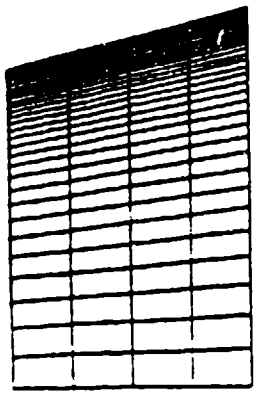
Fig. 77. A constant Mach number of 1.2 and zero contravariant velocities of V and W were imposed at the inlet. The flow was assumed laminar and the Reynolds number was taken to be 10^5 based on the inlet hydraulic radius and inflow properties.

The numerical experiments were done for both the three-dimensional PNS and TLNS algorithms with approximate factorization. Typical convergence curves for the PNS algorithm are shown in Fig. 78. As we can see, due to the additional approximate factorization of the left hand side operator, the quadratic convergence in two-dimensional PNS procedures cannot be obtained for the three-dimensional PNS solver. The optimum CFL number for this case is 20 and this results in 300 local iterations to reach 7 orders of magnitude reduction in the L-2 norm. But acceptable convergence (5 orders reduction in the L-2 norm) can be achieved in 40 iterations. However, this time-iterative three-dimensional PNS algorithm has been found to be very robust and is insensitive to grid-stretching in the ζ direction.

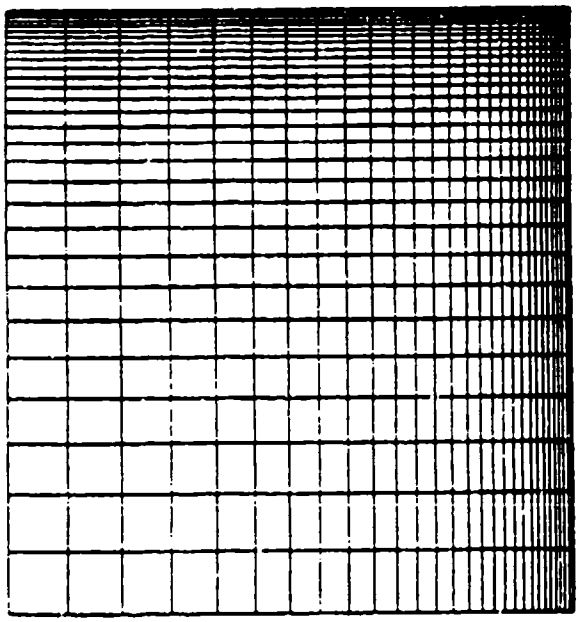
The convergence for the three-dimensional TLNS procedure is shown in Fig. 79 for a optimum CFL number of 20. The initial conditions for this calculation were obtained from the corresponding converged (5 orders of magnitude reduction in the L-2 norm at each cross-plane) PNS solutions. As is seen, for 300 iterations, the L-2 norm drops 7 orders of magnitude, which is about the same rate as a two-dimensional central-differenced ADI solver, as we predicted from the stability analysis given in Section 6.2.1.

6.3 Flowfield Predictions

The test case for three-dimensional flowfield predictions was the supersonic flow through a three-dimensional nozzle with rectangular cross-sections as shown in Fig. 80. The wall contour of this nozzle was chosen to be the same as that of the 272 : 1 axisymmetric contoured nozzle previously given. This nozzle has a constant width of 30 mm. in the y direction. Due to the symmetry conditions, only one



A-A section



B-B section

Figure 77. $5 \times 30 \times 30$ grid of 15^{0} expanding 3-D nozzle for convergence test

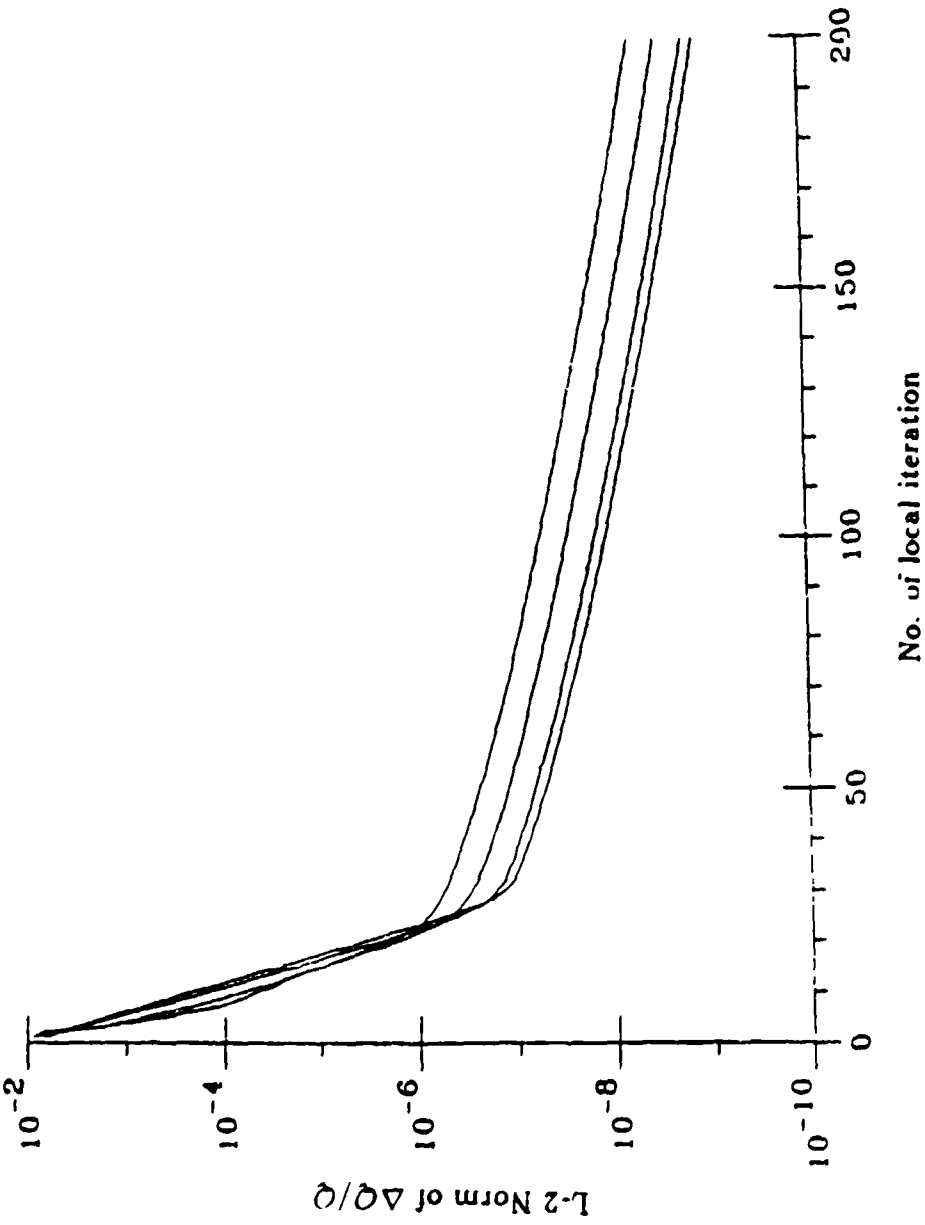


Figure 78. Convergence of 3-D PNS algorithm with $CFL = 20$

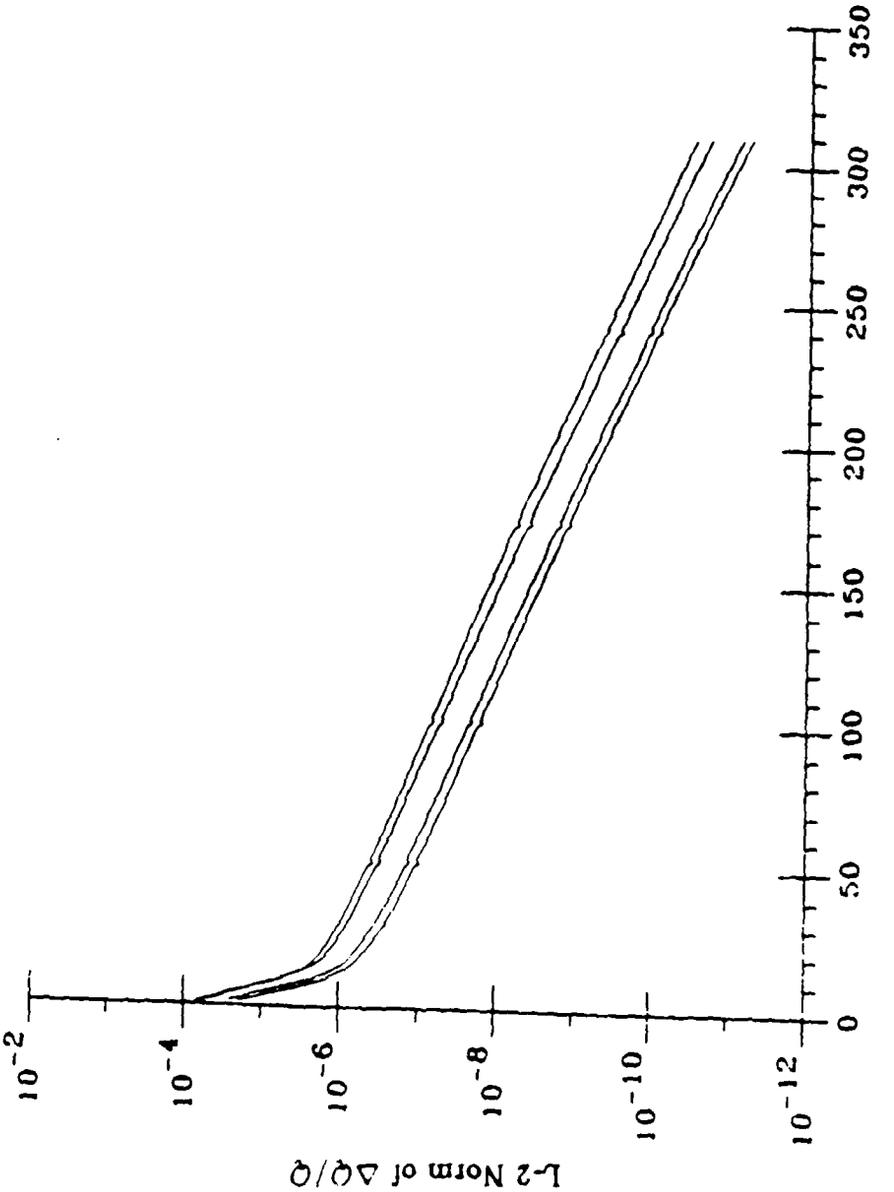


Figure 79. Convergence of 3-D TLNS algorithm with $CFL = 20$

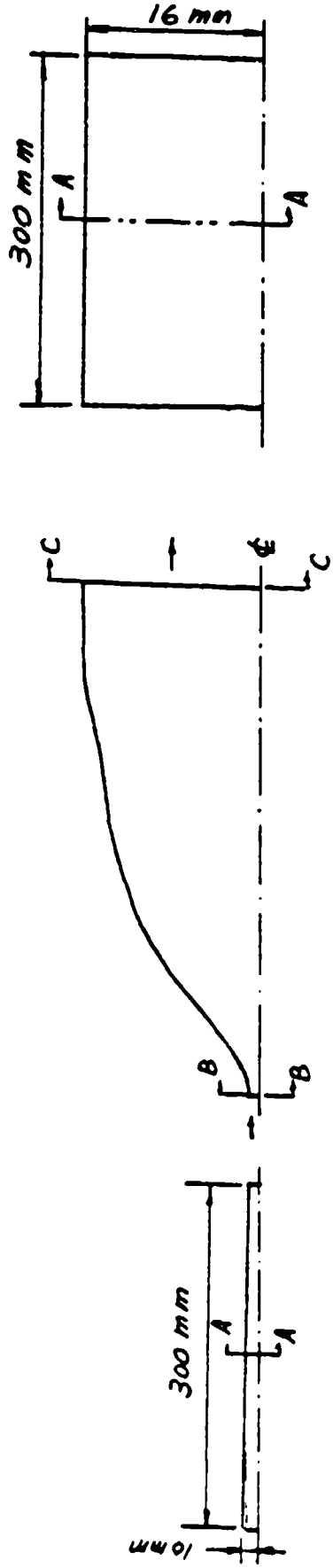


Figure 80. Nozzle geometry of 3-D flowfield prediction

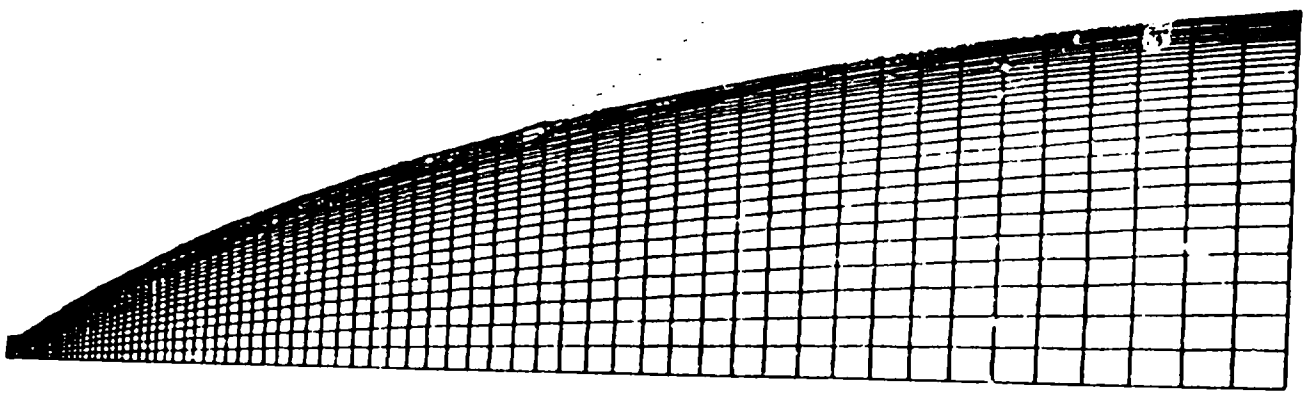
quadrant on the cross-section need to be calculated.

The 75 × 30 × 30 grid system with 75 in the x-direction and 30 × 30 on one quadrant of the cross-plane is shown in Figs. 81-82. Figure 81 shows the grid on the side wall of the nozzle. As is seen, a strong clustering near the top is chosen to resolve the boundary layer. Typical grids on the cross-planes are shown in Fig. 82 for both the inlet and exit planes. This figure also shows strong stretching near the side wall and the top due to the boundary layers in the y and z directions.

The inlet Mach number was assumed to be uniformly 1.02 and the gas properties described in Section 3.6 were imposed at the inlet. This resulted in a nozzle Reynolds number based on the throat hydraulic radius of 1.5×10^4 and laminar flow was assumed. The PNS procedure, Eq. (6.13), was utilized to perform the calculation.

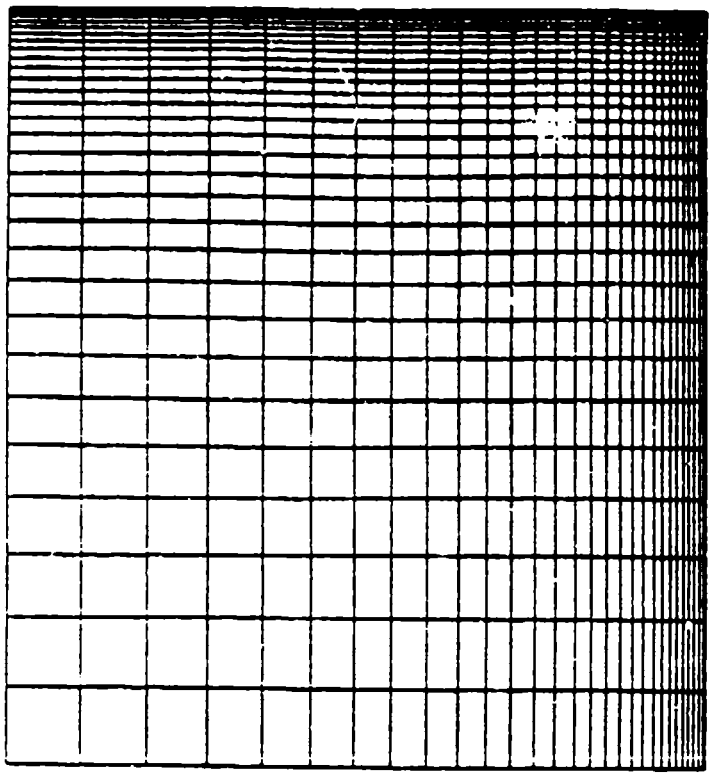
The flowfield results are shown at several locations indicated in Fig. 83. The computed Mach number contours at locations A and B are shown in Fig. 84. This figure shows quite different results from those of the axisymmetric calculations presented in previous chapters due to the three-dimensionality. Although not shown here, these Mach number contours are more similar to corresponding planar two-dimensional results. The wiggles of the contours near the exit and the center plane are possibly due to insufficient grid resolution and the reflection of a weak oblique shock at the center plane.

The streamwise velocity contours on several cross-planes (indicated in Fig. 83 as locations C, D, E, and F) are shown in Figs. 85-87. In these figures, the growing of the boundary layer thickness near the side wall and the top is clearly shown. Figures 88-90 show the cross-stream velocity vector plots at locations C, D, E, and F. Secondary flow patterns and the development of vortices of the three-dimensional boundary layer near the side wall and the top are observed. The secondary flow



A-A section

Figure 81. 75 × 30 surface grid on the side wall of the 3-D nozzle



C-C section exit

B-B section inlet

Figure 82. 30 x 30 grids on the inlet and exit cross-plane.

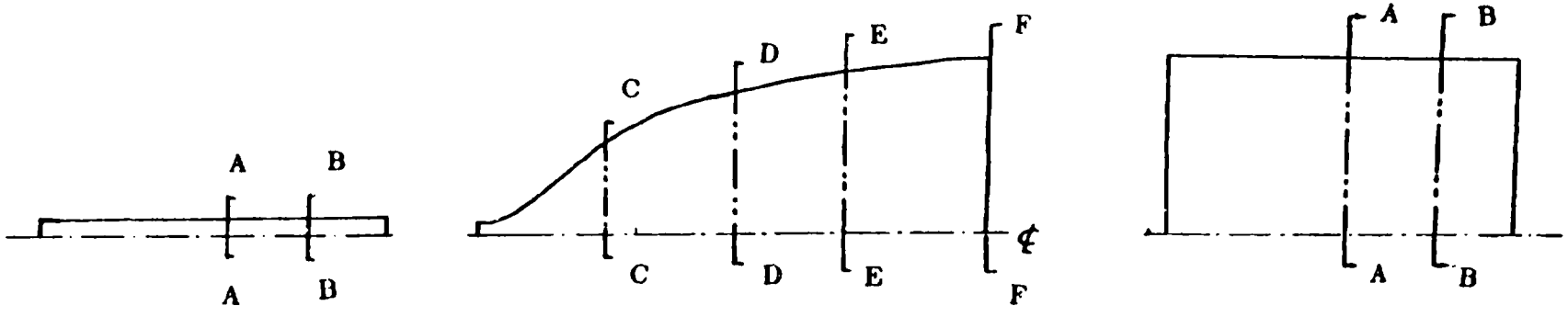
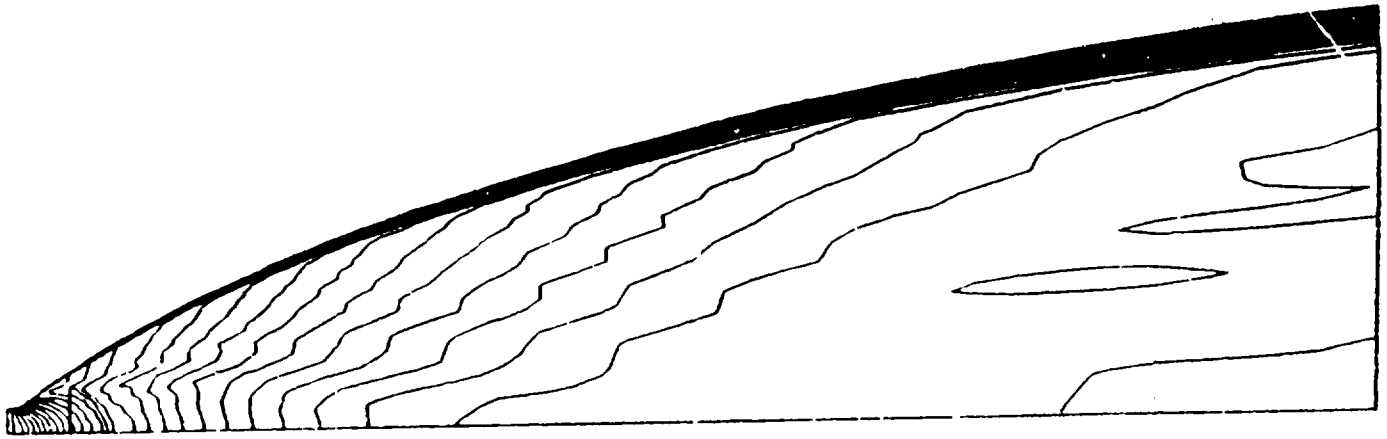
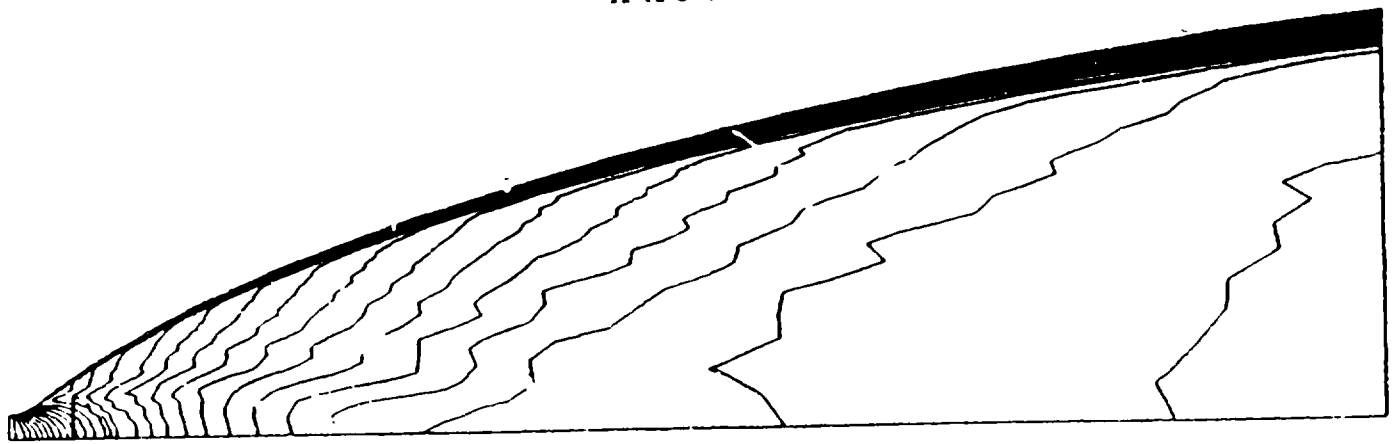


Figure 83. Locations of flowfield plots for 3-D nozzle

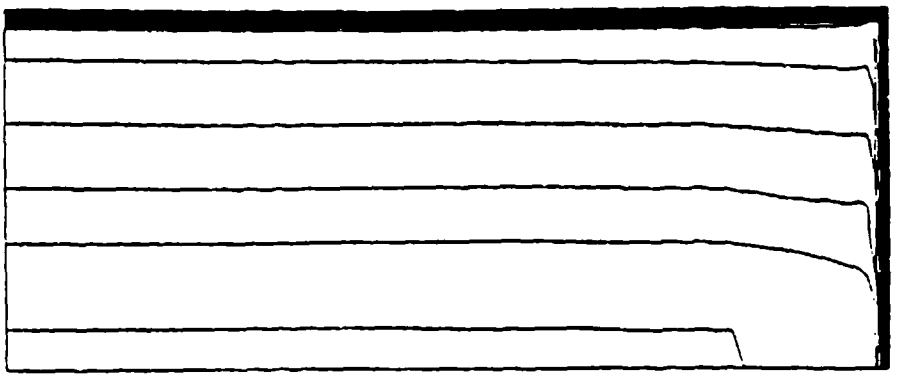


A-A section

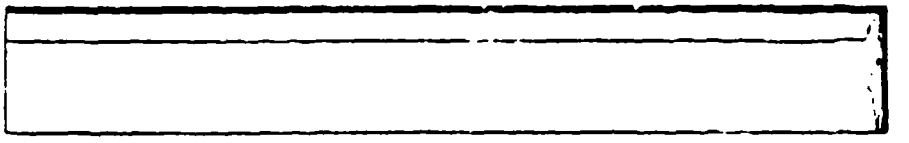


B-B section

Figure 84. Mach number contours at locations *A* and *B*

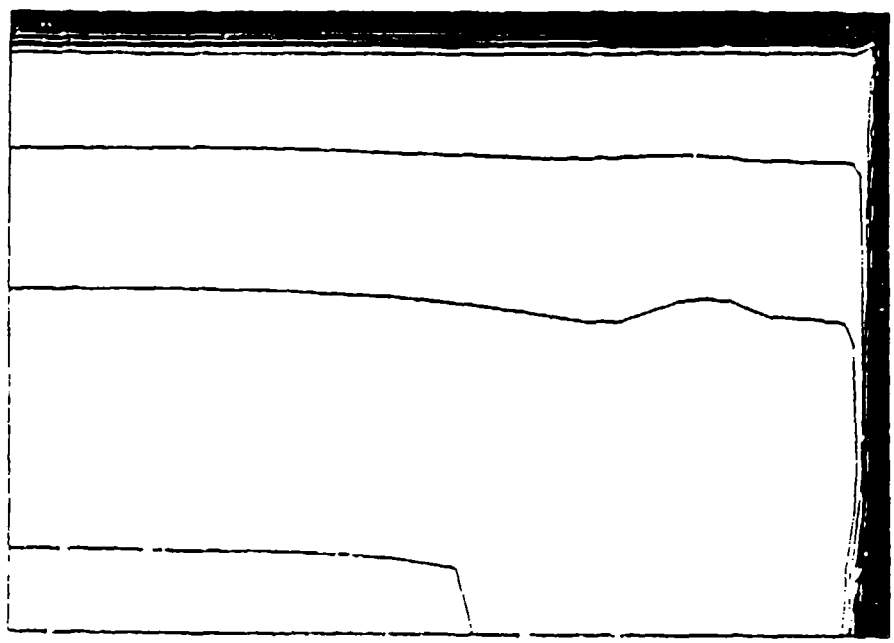


D- D section



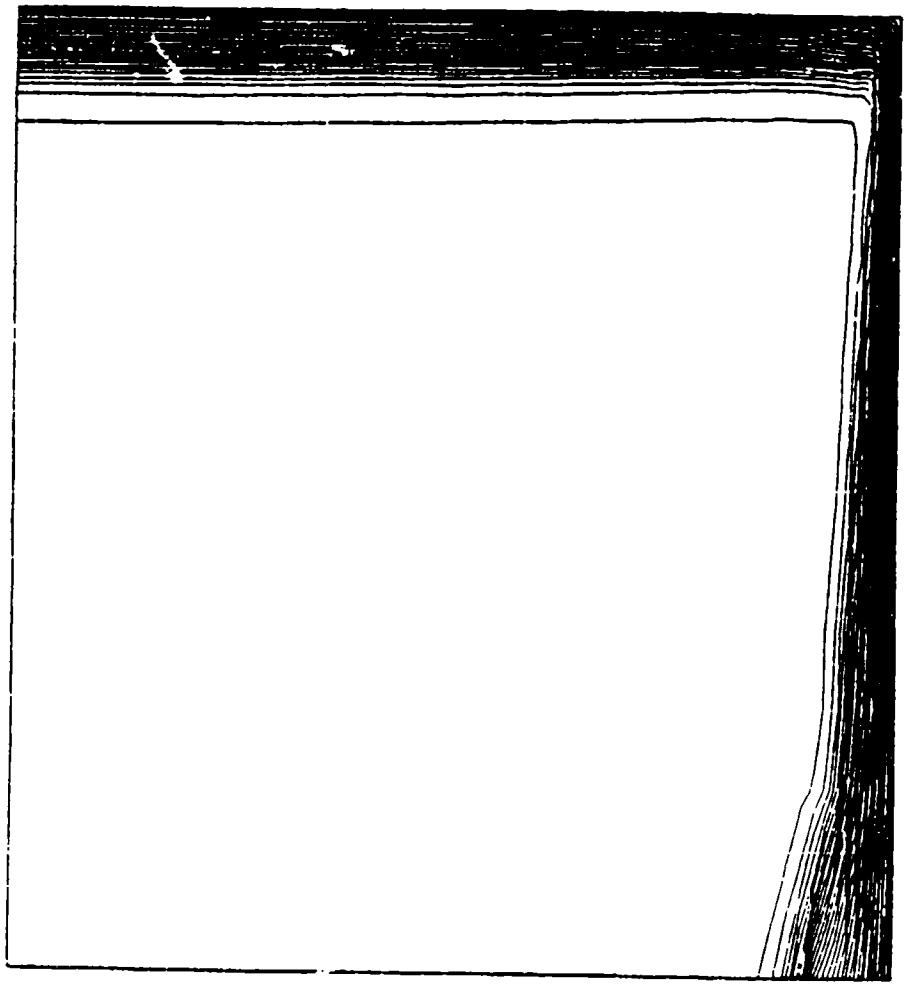
C - C section

Figure 85. Streamwise velocity contours at location C and D



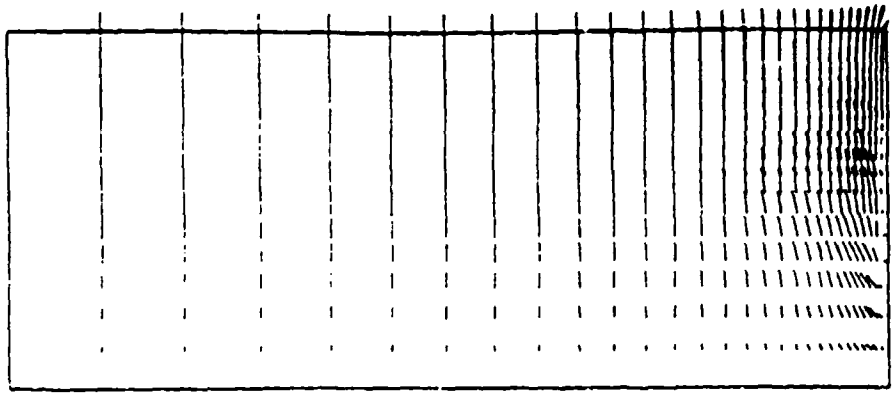
E-E section

Figure 86. Streamwise ve[illegible]ntours at location E

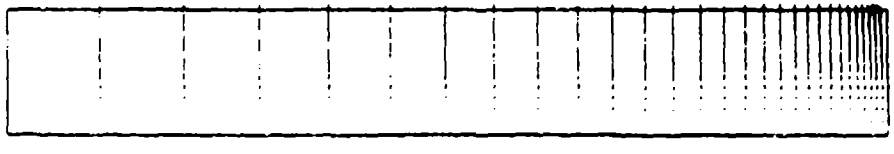


F–F section

Figure 87. Streamwise velocity contours at location F

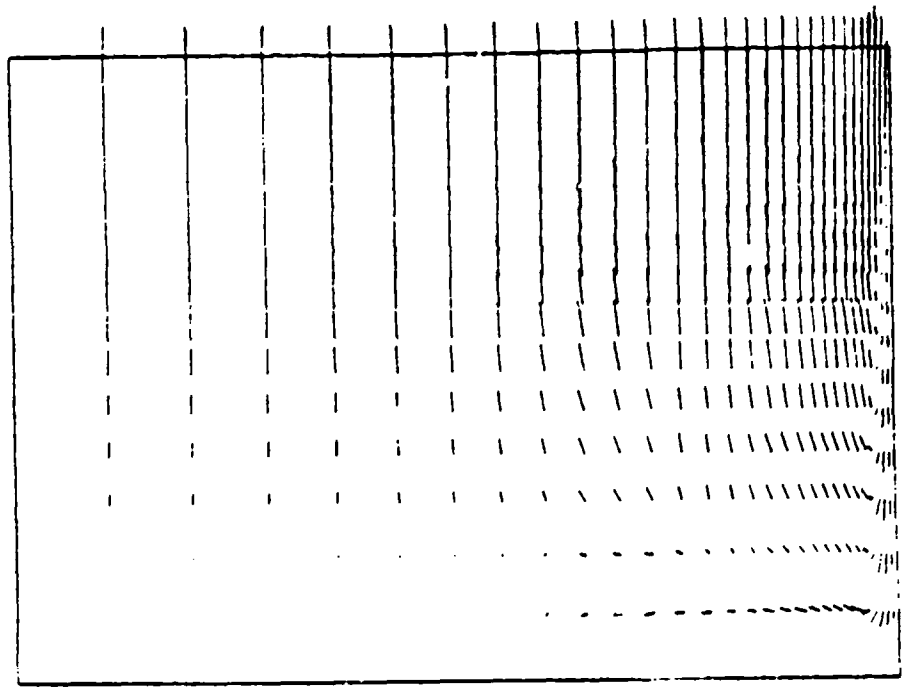


D-D section



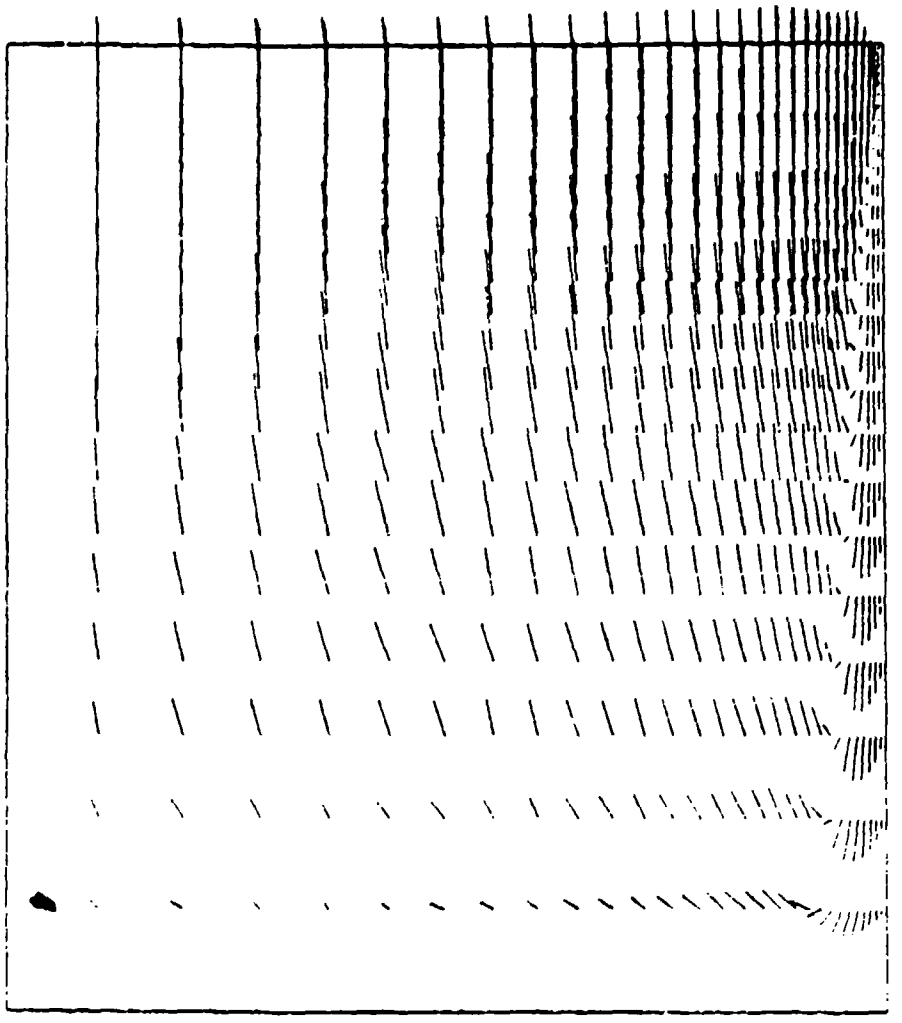
C-C section

Figure 88. Cross-stream velocity vector plots at location C and D



E-E section

Figure 89. Cross-stream velocity vector plots at location E



F F section

Figure 90. Cross-stream velocity vector plots at location *F*

patterns adjacent to the side walls as shown in Figs. 89 and 90 also explains why the boundary layer along the side wall is thicker near the center plane.

CHAPTER 7

SUMMARY

Implicit time-dependent schemes have been successfully applied to solve the compressible thin-layer Navier-Stokes equations in multi-dimensions. Preliminary applications of the implicit algorithm to the one-dimensional Euler equations were studied by using spatial discretizations based on both central-differencing and flux-vector splitting upwind-differencing. Both differencing methods were shown to give rapid convergence and accurate solutions. The Fourier stability analysis has been studied for either differencing method. The results were shown to provide useful information about the convergence criteria. In particular, the explicit-like CFL limitation of the one-dimensional upwind scheme when using approximate Jacobians was successfully predicted from stability analysis and later on confirmed by numerical experiments. The preparatory investigations on one-dimensional flows also provide informative results that are extendible to multi-dimensions.

For two-dimensional calculations, the ADI scheme based on central-differencing was formulated to solve the TLNS equations in a cylindrical coordinate system. The effectiveness of this ADI scheme was tested by calculating typical subsonic, transonic, and supersonic flows through nozzles. The results showed that the convergence rates of subsonic and transonic cases are slow (when compared to one-dimensional calculations) but acceptable. When the flow is predominantly supersonic, the ADI scheme has proven to be inefficient and sometimes even unstable.

Based upon the physical character of viscous supersonic flows, a hybrid discretization composed of central differencing in the streamwise direction and second-

order upwinding in the cross-stream direction was proposed. Stability analyses on a modeled equation were considered for the fully implicit and three approximate factorization procedures based on this hybrid discretization scheme. The results showed that all four algorithms are unconditionally stable for the Burger's equation. Further, the line-relaxation version of the DDADI algorithm gives the eigenvalues of the amplification matrix that approach the fully implicit limit.

On the basis of encouraging stability results, four algorithms indicated above were then applied to solve the TLNS equations for flows through nozzles. Of the three approximate techniques, the DDADI scheme suggested by Lombard [19] is shown to require the least number of iterations, but in terms of CPU time, the PNS-ADI scheme developed in this study is as fast as the DDADI scheme. The standard ADI factorization arising from traditional ADI schemes [8,9,10] proves to be the most inefficient in terms of both the number of iterations and the CPU time required. The direct method devised from the physics of high Reynolds number viscous flows seems to be particularly suited for supersonic problems, but rapid convergence of the DDADI and the PNS-ADI schemes allows them to surpass the direct method in terms of CPU time required. However, the direct method has proven to be more robust than any approximate schemes by noting that the CFL number can be as high as 10^{10} without losing stability. Numerical experiments regarding boundary conditions and Jacobian matrices have indicated that implicit boundary conditions toge er with true Jacobians of the split flux play a decisive role on convergence.

The solutions of present upwind-central differencing algorithms were compared to those of MOC calculations, excellent agreements on the wall pressure distribution and Mach number contours were demonstrated.

For the first time, proper downstream boundary conditions are applied for the subsonic portion of the outflow. These downstream boundary conditions were shown to be capable of allowing supersonic solutions to respond to nozzle back pressure conditions as they should do in realistic situation. The calculations with recirculation and reentry flows at the exit plane caused no difficulty, and the results showed that different exit pressures would alter the nozzle boundary layer characteristics near the exit plane. The extrapolation conditions that are normally used were shown to give solutions corresponding to one specific back pressure condition. A series of results showing the effects of variations in back pressures, wall temperatures, and nozzle Reynolds numbers are given for both a conical nozzle and a high expansion ratio contoured nozzle. The effects of turbulence on supersonic nozzle flows with separation were investigated by solving the Reynolds averaged Navier-Stokes equations with the Baldwin and Lomax model. The global characteristics of turbulent flows were properly resolved by using this algebraic turbulence model.

The results of testing on the global mass conservation indicate that global mass errors can be kept below 1% when fully conservative form is used, while quasi conservative form may give a global error as large as 30% even in flowfields without discontinuities.

Along with the development of Navier-Stokes algorithms, the parabolized procedures were also investigated. Unlike the traditional approach, the PNS procedure was obtained from the time-dependent general flux-vector split TLNS equations, for which the streamwise flux vector has been split into two parts corresponding to downstream and upstream characteristics. By omitting the parts with upstream characteristics, the whole equation set was made parabolic in the streamwise direction. With this approach, a distinct PNS formulation can be obtained for each type of flux-vector splitting considered. Two examples were chosen for demonstra-

tion. The traditional PNS formulation is obtained by using a pressure gradient splitting. The use of characteristic-based flux splitting yields a PNS algorithm that includes only the downstream characteristics. Stability results showed that this characteristic-based PNS algorithm is stable for space-marching and numerical results indicated that it provides solutions that are identical to the classical pressure gradient split PNS formulation and in excellent agreement with the TLNS solutions.

One advantage of the present PNS algorithm when compared to non-iterative space-marching procedures is that the current approach requires no safety factor. Comparisons of the pressure gradient splitting PNS calculations with the TLNS solutions show that the introduction of a safety factor deteriorates the solution accuracy. The necessary local iterations on each ξ plane for the time-iterative PNS algorithm result in more computational time than the traditional non-iterative PNS procedure. However, it has been shown that this CPU time deficit is partially offset in that the local iterations allow the ξ derivative to be formulated in a conservative form so that variable step sizes in ξ can be used.

The global pressure iteration procedure in the traditional PNS approaches has been interpreted as a TLNS procedure. These mathematically and physically well-posed TLNS procedures based on approximate factorization are suggested instead of the traditional global pressure iteration procedures based on an arbitrary relaxation of the pressure field.

Numerical algorithms for computing viscous swirling nozzle flows have also been studied by using time-iterative implicit schemes. The implicit ADI and the PNS-ADI algorithms are utilized to solve transonic and supersonic swirling flows, respectively. These algorithms prove to be equally efficient for swirling two-dimensional calculations. The combined effects of viscosity and swirling on the flowfield and the integral nozzle performance are investigated for transonic and supersonic flows

through mild to high expansion ratio nozzles. Viscous calculations are performed for three nozzle geometries previously investigated by Dutton [34]. These results validate the inviscid assumptions for high Reynolds number flows and show how rapidly nozzle performance deteriorates with the Reynolds number. For the high expansion ratio contoured nozzle and the plug nozzle, the combined effects of swirling and viscosity have significant influence on the flowfields and the nozzle performance.

Finally, the algorithms developed for axisymmetric two-dimensional flows are extended to solve the three-dimensional TLNS equations. These three-dimensional algorithms are based upon the DDADI splitting for the streamwise flux vector and an additional approximate factorization of the left hand side operator. The optimum CFL number reduces to the order of 10 and it gives slow convergence due to this approximate factorization. Both the PNS and TLNS procedures are formulated for three-dimensional calculations. Typically, acceptable convergence can be achieved by 40 local iterations for the PNS algorithm. Although this convergence rate is not competitive with that of two-dimensional algorithms, these three-dimensional algorithms prove to be robust and are insensitive to grid-stretching along the streamwise direction.

As a final comment, although all the results shown in this study are computations of internal flowfields of nozzles, all numerical algorithms developed here are applicable to external flows as well.

BIBLIOGRAPHY

1. Shapiro, A. H., *The Dynamics and Thermodynamics of Compressible Fluid Flow*, Vol. I, Ronald Press, New York, 1953.
2. Schlichting, H., *Boundary Layer Theory*, 7th ed., translated by J. Kestin, McGraw Hill, New York, 1979.
3. Anderson, D. A., Tannehill, J. C., and Pletcher, R. H., *Computational Fluid Mechanics and Heat Transfer*, McGraw Hill Book Co., New York, 1984.
4. Hoffman, R. J., Hetrick, M. A. Jr., Nickerson, G. R., and Jarossy, F. J., *Plume Contamination Effects Prediction: CONTAM III Computer Program- Vols. I, II and III*, Science Applications Inc., AFRPL-TR-82-033, December 1982.
5. Kushida, R., Hermel, J., Apfel, S., and Zydowicz, M., "Performance of High-Area-Ratio Nozzle for a Small Rocket Thruster," *J. Propulsion and Power*, Vol.3, July-August 1987, pp.329-333.
6. Moretti, G., Abbett, M., "A Time-Dependent Computational Method for Blunt Body Flows," *AIAA J.*, Vol. 4, 1966, pp.2136-2141.
7. MacCormack, R. W., "The Effect of Viscosity in Hypervelocity Impact Cratering," AIAA Paper 69-354, Cincinnati, Ohio, 1969.
8. Beam, R. M. and Warming, R. F., "An Implicit Finite-Difference Algorithm for Hyperbolic System in Conservation Law Form," *J. of Comp. Phys.*, Vol. 22, 1976, pp.87-100.
9. Beam, R. M. and Warming, R. F., "An Implicit Factored Scheme for the Compressible Navier-Stokes Equations," *AIAA J.*, Vol. 16, 1978, pp. 393-401.
10. Briley, W. R. and McDonald H., "On the Structure and Use of Linearized Block Implicit Schemes," *J. of Comp. Phys.*, Vol.34, 1980, pp. 54-77.

11. Steger, J. L., "Implicit Finite-Difference Simulation of Flow about Arbitrary Geometries with Application to Airfoils," AIAA Paper 77-665, Albuquerque, New Mexico, 1977.

12. Baldwin, B. S. and Lomax, H., "Thin Layer Approximations and Algebraic Model for separated Turbulent Flows," AIAA Paper 78-257, 1978.

13. Pulliam, T. H., "Euler and Thin-Layer Navier-Stokes Codes: ARC2D, ARC3D," Univ. of Tennessee Space Institute, Pub. E02-4005-023-84, 1984.

14. Lombard, C. K., Oliger, J., Yang, J. Y., and Davy, W. C., "Conservative Supra-Characteristics Method for Splitting the Hyperbolic Systems of Gasdynamics with Computed Boundaries for Real and Perfect Gases," AIAA Paper 82-0837, June 1982.

15. MacCormack, R. W., "Current Status of Numerical Solutions of the Navier-Stokes Equations," AIAA Paper 85-0032, Jan. 1985.

16. Steger, J. L., Warming, R. F., "Flux Vector Splitting of the Inviscid Gas Dynamic Equations with Application to Finite Difference Methods," *J. Comp. Phys.*, Vol. 40, 1981, pp.263-293.

17. Chakravarthy, S. R., "Relaxation Methods for Unfactored Implicit Upwind Schemes," AIAA Paper 84-0165, Jan. 1984.

18. Yoon, S. and Jameson, A.,"An LU-SSOR Scheme for the Euler and Navier-Stokes Equations," AIAA Paper 87-0600, Jan. 1987.

19. Lombard, C. K., Venkatapathy, E., and Bardina, J., "Universal Single Level Implicit Algorithm for Gasdynamics," AIAA Paper 84-1533, June 1984.

20. Chang, C.-L., Kronzon, Y., Merkle, C. L., "Time-Iterative Solutions of Viscous Supersonic Flows," AIAA Paper 87-1289, Jun. 1987.

21. Thomas, J. L. and Walters, R. W., "Upwind Relaxation Algorithm for the Navier-Stokes Equations," *AIAA J.*, Vol. 25, April 1987, pp.527-534.

22. van Leer, B. "Towards the Ultimate Conservative Difference Scheme IV. A New Approach to Numerical Convection," *J. Comp. Phys.*, Vol. 23, 1977, pp.276-299.

23. Newsome, R. W., Walters, R. W., and Thomas, J. L., "An Efficient Iteration Strategy for Upwind/Relaxation Solutions to the Thin-Layer Navier-Stokes Equations," AIAA Paper 87-1113, June 1987.

24. Vichnevetsky, R. and Bowles, J. B., *Fourier Analysis of Numerical Approximations of Hyperbolic Equations*, SIAM studies in Applied Mathematics, 1982.

25. Shirazi, S. A. and Truman, C. R., "Comparison of Algebraic Turbulence Models for PNS Predictions of Supersonic Flow past a Sphere-Cone," AIAA Paper 87-0544, Jan. 1987.

26. Vigneron, Y. C., Rackich, J. V., and Tannehill, J. C., "Calculation of Supersonic Viscous Flow over Delta Wings with Sharp Subsonic Leading Edges," AIAA Paper 78-1137, July 1978.

27. Lawrence, S. L. and Tannehill, J. C., "An Upwind Algorithm for the Parabolized Navier-Stokes Equations," AIAA Paper 86-1117, May 1986.

28. Power, G. D. and Anderson, O. L., "An Assessment of a Parabolic Analysis for Axisymmetric Internal Flows in Rocket and Turbomachinery Ducts," AIAA Paper 86-1598, June 1986.

29. Gielda, T. and McRae, D., "An Accurate, Stable, Explicit, Parabolized Navier-Stokes Solver for High Speed Flows," AIAA Paper 86-1116, May 1986.

30. Buckley, P. L., Craig, R. R., Davis, D. L., and Schwartzkopf, K. G., "The Design and Combustion Performance of Practical Swirlers for Integral Rocket /Ramjets," *AIAA J.*, Vol. 21, May 1983, pp. 733-740.

31. Carpenter, P. W. and Johannesen, N. H.,"An Extension of One-Dimensional Theory to Inviscid Swirling Flow Through Choked Nozzles," *Aeronautical Quarterly*, Vol. 26, May 1975, pp. 71-87.

32. Kornblum, B. T., Thompson, H. D., and Hoffman, J. D.,"An Analytical Investigation of Swirl on Annular Propulsive Nozzles," *J. Propulsion and Power*, Vol. 2, March-April 1986, pp. 155-160.

33. Hoffman, J. D., Thompson, H. D., and Marcum, D. L., "An Analytical Investigation of the Effects of Swirler Design on the Performance of Annular Propulsive Nozzles," AIAA Paper 86-0587, Jan. 1986.

34. Dutton, J. C., "Swirling Supersonic Nozzle Flow," *J. Propulsion and Power*, Vol. 3, No. 4, 1987, pp. 342-349.

35. Vinokur, M., "Conservation Equations of Gas-Dynamics in Curvilinear Coordinate system,"*J. Comp. Phys.*, Vol. 14, 1974, pp.105-125.

36. Rai, M. M. and Chaussee, D. S., "New Implicit Schemes and Implicit boundary conditions," AIAA Paper 83-0123, Jan. 1983.

37. Chakravarthy, S. R., "Euler Equations-Implicit Schemes and Implicit Boundary Conditions," *AIAA J.*, Vol. 21, 1982, pp. 1565-1571.

38. Hildebrand, F. B., *Methods of Applied Mathematics* 2nd. ed. Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1965, pp.19-22.

39. Pulliam, T. H. Steger, J. L., "Recent Improvements in Efficiency, Accuracy, and Convergence for Implicit Approximate Factorization Algorithms," AIAA paper 85-0360, Jan. 1985.

40. Jespersen, D. C. and Pulliam, T. H., "Flux Vector Splitting and Approximate Newton Methods,"AIAA Paper, 83-1899, July 1983.

41. Buning, P. G. and Steger, J. L., "Solution of Two-Dimensional Euler Equation with Generalized Coordinate Transformation Using Flux Vector Splitting," AIAA Paper 82-0971, June 1982.

42. Merkle, C. L. and Athavale, M., "Time-Accurate Unsteady Incompressible Flow Algorithms Based on Artificial Compressibility," AIAA Paper 87-1137, Jun. 1987.

43. Schiff, L. B. and Steger, J. L., "Numerical Solutions of Steady Supersonic Viscous Flow," AIAA Paper 79-0130, Jan. 1979.

44. Rackich, J. V., "Iterative PNS Method for Attached Flows with Upstream Influence," AIAA Paper 83-1955, Danvers, Mass., 1983.

45. Lin, A. and Rubin, S. G., "Three Dimensional Supersonic Viscous Flow over a Cone at Incidence," *AIAA J.*, Vol. 20, No.11., 1982, pp.1500-1507.

46. Davis, R. T., Barnett, M. and Rakich, J. V., "The Calculation of Supersonic Viscous Flow Using the Parabolized Navier-Stokes Equations," *Computers & Fluids*, Vol. 14, No. 3, 1986, pp.197-224.

47. Barnett, M. and Davis, R. T., "A Procedure for the Calculation of Supersonic Flows with Strong Viscous-Inviscid Interaction," AIAA Paper 85-0166, Jan. 1985.

48. Thompson, D. S. and Anderson, D. A., "A Pseudo-Unsteady Approach for Predicting Steady Supersonic Flows," AIAA Paper 87-0541, Jan. 1987.

VITA

Chau-Lyan Chang was born July 4, 1958, in Taipei, Taiwan, The Republic of China. He received a B.S. degree in Mechanical Engineering in June 1980 and an M.S. degree in Mechanical Engineering in June 1982 from National Taiwan University. From October 1982 to August 1984, the author was a Mechanical Engineer during a two-year military service for the Navy, The Republic of China. From September 1984 to the present, he has been employed as a Research Assistant by the Department of Mechanical Engineering, The Pennsylvania State University.