

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE 22 Sep 95		3. REPORT TYPE AND DATES COVERED
4. TITLE AND SUBTITLE An Incompressible 3-D navier-Stokes Method with Adaptive Hybrid Grids			5. FUNDING NUMBERS	
6. AUTHOR(S) Alice Josephine Chen				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) AFIT Students Attending: University of Texas			8. PERFORMING ORGANIZATION REPORT NUMBER 95-017D	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) DEPARTMENT OF THE AIR FORCE AFIT/CI 2950 P STREET, BLDG 125 WRIGHT-PATTERSON AFB OH 45433-7765			10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for Public Release IAW AFR 190-1 Distribution Unlimited BRIAN D. GAUTHIER, MSgt, USAF Chief of Administration			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words)				

19951019 033

DTIC QUALITY INSPECTED 8

14. SUBJECT TERMS			15. NUMBER OF PAGES 205	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT	

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to *stay within the lines* to meet *optical scanning requirements*.

Block 1. Agency Use Only (Leave blank).

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C - Contract	PR - Project
G - Grant	TA - Task
PE - Program Element	WU - Work Unit Accession No.

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. (If known)

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of...; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

DOD - See DoDD 5230.24, "Distribution Statements on Technical Documents."

DOE - See authorities.

NASA - See Handbook NHB 2200.2.

NTIS - Leave blank.

Block 12b. Distribution Code.

DOD - Leave blank.

DOE - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.

NASA - Leave blank.

NTIS - Leave blank.

Block 13. Abstract. Include a brief (*Maximum 200 words*) factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code (*NTIS only*).

Blocks 17. - 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.

Copyright

by

Alice Josephine Chen

1995

**An Incompressible 3-D Navier-Stokes Method
with Adaptive Hybrid Grids**

by

Alice Josephine Chen, BA, BS, MS

Dissertation

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

The University of Texas at Austin

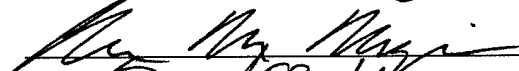
August, 1995

**An Incompressible 3-D Navier-Stokes Method
with Adaptive Hybrid Grids**

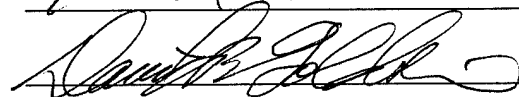
Approved by
Dissertation Committee:











To my husband, Bob

Acknowledgments

I am immensely grateful to my advisor, Dr. John Kallinderis, for providing the proper focus in my research which has enabled me to accomplish this task in the time allotted to me. Through his guidance I have been able to build a strong base in the field of Computational Fluid Dynamics. I am very thankful for the time and effort he has devoted to my research work. There were many times my research appeared to approach a dead end, and he helped direct me toward a promising avenue. I am especially appreciative of his support and understanding of the Air Force's imposed time constraints.

I thank Dr. David Dolling for his classroom instruction for it helped me learn more about aerodynamics. I also thank him, as well as Dr. David Goldstein, Dr. Christine Maziar, and Dr. Kenneth Ball, for their interest in my research and for being members of my dissertation committee.

I especially owe thanks to my office mates: Dr. Vijayan Parthasarathy, Tommy Minyard, Aly Khawaja, Harlan McMorris, and Kengo Nakajima. Kengo taught me the basics of his 2-D code and AIX. As our IBM system administrator, Tommy cheerfully put up with my silly questions on AIX. Tommy and Vijayan were also incredibly helpful the many times I had questions on CFD issues and LaTeX. I also appreciate Vijayan's help on finite element methods, adaptation and

running jobs on the CRAY. Aly and Harlan provided immense amounts of assistance to me on numerous plotting packages. Aly was also very helpful with his insight on the subtle points of prismatic grid generation and Harlan also devoted many hours generating the many FELISA surface and tetrahedra grids I desired. I thank Michaelyn McKelvey for her guidance in academic and administrative affairs. Her caring and affection always brightened my day.

I wish to thank the United States Air Force Academy for sponsoring my academic endeavors and Colonel Michael Smith of the Department of Aeronautics for giving me the chance to earn my doctoral degree.

This work was supported by the Texas Advanced Technology Program (ATP) Grant #003658-413, NSF Engineering Research Centers program grant #CDR-8721512, and NSF Grant ASC-9357677 (NYI program).

The last year and half has been especially tough since the Air Force stationed my husband, Bob, in Washington D.C. and then Alabama, but though he was physically far away, his concern and love were always with me. I thank him for his constant encouragement and support. He helped me keep my sense of humor through the times when I despaired. I thank my parents and siblings for their support during this time. I wish to give a special thanks to my sister Joyce and her husband Jeff for letting me invade their home for the last year and a half.

ALICE JOSEPHINE CHEN

The University of Texas at Austin

August 1995

An Incompressible 3-D Navier-Stokes Method with Adaptive Hybrid Grids

Publication No. _____

Alice Josephine Chen, Ph.D.

The University of Texas at Austin, 1995

Supervisor: John Kallinderis

Accession For	
NTIS CRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification _____	
By _____	
Distribution /	
Availability Codes	
Dist	Avail and/or Special
A-1	

A numerical method has been developed for the unsteady Navier-Stokes equations of incompressible flow in three dimensions. The momentum equations, combined with a pressure correction equation, are solved employing a non-staggered grid which results in a simpler formulation compared to the classical approach of using staggered meshes. The momentum equations are solved explicitly using a finite volume algorithm, while the pressure Poisson equation is discretized using the Galerkin finite element method and implicitly solved. The grid is formed with hybrid (prismatic/tetrahedral) elements. Equation adaptation is utilized, the Navier-Stokes equations are solved in the prismatic region, which includes the viscous region, and the Euler equations are solved in the tetrahedral region, which is inviscid. Adaptive local grid refinement of the prism and tetrahedral cells is employed in order to optimize the mesh to the flow solution. Validation of the algorithm is performed

using experimental and other numerical data, and demonstrate the accuracy and robustness of the unsteady three-dimensional method.

An additional study is conducted using an existing two-dimensional incompressible Navier-Stokes solver. The two-dimensional solver is applied to predict the hydrodynamic forces on a circular cylinder due to reversing flows.

Contents

Acknowledgments	v
Abstract	vii
List of Tables	xiv
List of Figures	xvi
Chapter 1 Introduction	1
1.1 Incompressible versus Compressible Flow Equations	1
1.2 Issues Related to the Numerical Solution of the Incompressible Flow Equations	2
1.2.1 Pressure/Velocity Decoupling	3
1.2.2 Staggered versus Non-Staggered Grids	5
1.2.3 Unstructured versus Structured Meshes	7
1.3 Present Research	8
1.4 Overview of Dissertation	9
Chapter 2 Pressure Correction Numerical Method	11
2.1 3-D Navier-Stokes Equations	11

2.2	Nondimensionalization	13
2.3	Semi-Discrete Form of Governing Equations	14
2.4	Pressure Correction Method	15
Chapter 3	Adaptive Hybrid (Prismatic/Tetrahedral) Grid	18
3.1	Hybrid Mesh	18
3.2	Equation Adaptation for Hybrid Mesh	22
3.3	Spatial Grid Adaptation	22
3.3.1	Detection of Cells to be Adapted	23
3.3.2	Local Surface Grid Embedding for Prismatic Mesh	24
3.3.3	Tetrahedral Embedding	26
Chapter 4	Integration Scheme	28
4.1	Node-Centered Dual Cell	28
4.2	Finite Volume Discretization of the Momentum Equations	31
4.2.1	Inviscid Terms	31
4.2.2	Viscous Terms	32
4.3	Finite Element Discretization of Poisson Equation	34
4.3.1	Isoparametric Transformation	34
4.3.2	Galerkin Method	36
4.3.3	Element and Global Matrix Construction	37
4.4	Boundary Conditions for Unstructured Grids	39
4.4.1	Wall	39
4.4.2	Far Field	40
4.4.3	Symmetry Planes	42
4.5	Initial Conditions	45

4.6	Artificial Dissipation	45
4.7	Time Step Calculation	47
4.8	Solver Computer Requirements	47
Chapter 5	Numerical Issues	52
5.1	Pressure Correction Formulations	52
5.2	Diagonal Dominance	56
5.3	Boundary Conditions for the Poisson Equation	61
5.4	Non-staggered Grid versus Staggered Grid	64
5.5	Discrete Divergence	69
5.5.1	Consistency of Discretized Poisson Equation	69
5.5.2	Error of Discrete Continuity	72
5.6	Analytic Field Function Testing	75
5.6.1	Analytic Field Test of Sphere Mesh	75
5.6.2	Order of Accuracy Evaluation	77
Chapter 6	Validation and Application of the Adaptive Grid Solver	80
6.1	Flow over a Flat Plate at a High Reynolds Number	81
6.2	Steady Flow Around a Cylinder using a Prismatic Mesh	85
6.3	Impulsive Start of a Cylinder using a Prismatic Mesh	94
6.4	Steady 3-D Flow in a Driven Cubic Cavity with a Prismatic Mesh	97
6.5	Steady 3-D Flow Around a Sphere using a Prismatic Mesh	102
6.6	Evaluation of Prismatic Mesh Adaptation Applied to Cubic Cavity Flow	110
6.6.1	Accuracy of Prismatic Adaptation on Cubic Cavity Flow	110
6.6.2	Robustness and CPU Time Savings of Prismatic Adaptation	111

6.7	Equation Adaptation with a Hybrid Mesh for Flow Around a Sphere	117
6.8	Hybrid Mesh Adaptation Applied to Flow Around a Sphere	124
6.8.1	Accuracy of Hybrid Adaptation on Sphere Flow	124
6.8.2	Robustness and CPU Time Savings of Hybrid Adaptation . .	125
6.9	Hybrid Mesh Adaptation for Flow Around Tandem Spheres	133
Chapter 7	Concluding Remarks	138
7.1	Summary	138
7.2	Contributions	139
7.3	Conclusions	139
7.4	Recommendations for Further Research	142
Appendix A	Division of Prism Cells into Tetrahedra Cells	144
Appendix B	Data Structures for Data Arrays	146
B.1	Primitive Variable Pointers	146
B.2	Node Pointers	149
B.3	Cell Pointers	149
B.4	Edge Pointers	150
Appendix C	Data Structures for Prismatic Grid	151
C.1	Face Pointers	152
C.2	Edge Pointers	152
Appendix D	Data Structures for Tetrahedral Grid	155
D.1	Cell Arrays	156
D.2	Face Arrays	156
D.3	Interface Arrays	157

Appendix E Study of Unsteady 2-D Flows	158
E.1 2-D Navier-Stokes Solver	158
E.2 Forces on the Cylinder in Oscillating Flow	159
E.3 Instability and Numerical Simulation	160
E.4 Simulation of Oscillating Flow Around Cylinders	163
E.4.1 Transverse Flow Regime, $KC = 8$, $Re = 1568$	163
E.4.2 Double Pair Flow Regime, $KC = 15$, $Re = 19050$	168
 Appendix F Phase Averaging Procedure for Oscillating Flow Solu-	
tions	174
F.1 Sampling the Time Varying Data	174
F.2 Phase Averaging the Sampled Data	176
 Appendix G Description of 2-D Navier-Stokes Solver	178
G.1 Governing Equations and Pressure Correction Formulation	178
G.2 Finite-Volume Discretization of the Momentum Equations	181
G.3 Finite-Element Discretization of Pressure Correction Equation	181
G.3.1 Artificial Dissipation	183
G.3.2 Time-Step Calculation	184
G.3.3 Boundary Conditions	185
 Bibliography	186
 Vita	204

List of Tables

4.1	Proportions of computing time consumed by parts of the solver. . . .	49
4.2	Proportions of computing time consumed by pressure correction parts of the solver.	50
4.3	Memory requirements of the solver.	51
5.1	Indicators of the level of diagonal dominance for Poisson matrix for- mulations. A ratio greater than 1.0 signifies diagonal dominance. . .	57
5.2	Grid parameters used for sphere meshes for analytic field function analysis.	76
5.3	Grid parameters of sphere meshes for analytic field function grid fine- ness study.	78
6.1	Grid parameters used for flat plate flow at $Re = 10^6$	82
6.2	Grid parameters used for circular cylinder flow at $Re=40$	88
6.3	Comparisons of results for circular cylinder flow at $Re=40$	92
6.4	Grid parameters used for unsteady circular cylinder flow at $Re=40$. .	95
6.5	Grid parameters used for the cubic cavity at $Re = 400$	97
6.6	Prismatic grid parameters used for flow around a sphere.	103

6.7	Surface triangulation used for adaptation accuracy comparison for the cubic cavity at $Re = 400$	114
6.8	Adaptation CPU time comparison for the cubic cavity at $Re = 400$	116
6.9	Grid parameters used for evaluation of equation adaptation for sphere flow at $Re=40$	118
6.10	Equation adaptation CPU time comparison for the sphere at $Re = 40$	119
6.11	Grid parameters used for hybrid mesh adaptation for sphere flow at $Re=100$	127
6.12	Adaptation CPU comparison for the hybrid sphere mesh at $Re = 100$	132
6.13	Adapted hybrid grid parameters used for tandem spheres flow at $Re=100$	134
E.1	Numerical parameters used for oscillating cylinder flow at $KC=8$ and $Re = 1568$	165
E.2	Numerical perturbation parameters used for oscillating cylinder flow at $KC=8$ and $Re = 1568$	165
E.3	Numerical parameters used for oscillating cylinder flow at $KC=15$ and $Re = 19050$	169
E.4	Numerical perturbation parameters used for oscillating cylinder flow at $KC=15$ and $Re = 19050$	169

List of Figures

1.1	Major categories of incompressible flow algorithms. where Ψ : stream function, ω : vorticity, V : velocity, and ϕ : potential	4
3.1	Semi-unstructured prismatic topology. (The unstructured boundary surface is shaded.)	21
3.2	Interface region between prismatic cells and tetrahedral cells.	21
3.3	Local surface grid embedding for prismatic cells: shaded regions illustrate binary and quad triangular face division.	25
3.4	Refinement strategies for tetrahedron: (a) <i>Binary</i> (1:2), (b) <i>Quadtree</i> (1:4) and, (c) <i>Octree</i> (1:8) divisions.	27
4.1	2-D analogies for node-centered dual mesh in the (a) Prismatic region (b) Tetrahedral-Prismatic interface and (c) Tetrahedral region. . . .	29
4.2	Node-centered dual cell surfaces attached to an edge.	30
4.3	2-D analogies for edge-centered dual mesh in the (a) Prismatic region (b) Tetrahedral-Prismatic interface and (c) Tetrahedral region. . . .	33
4.4	Master tetrahedral element in ξ, η, ζ space.	35
4.5	Obtaining the number of the node adjacent to the wall node	40

4.6	Illustration of far field boundary conditions for a hybrid mesh: assigning the cell center value to the tetrahedral face center, and distributing the face center value to the face nodes.	42
4.7	Illustration of a face defined by three far field nodes which does not lie on the far field boundary (shaded region).	43
4.8	Illustration of symmetry boundary conditions: assigning the cell center value to the quadrilateral face center, and distributing the face center value to the face nodes.	44
5.1	Wall pressure coefficient on cylinder for $Re=40$. o Experiment by Grove, — SMAC method, - - - Present work	55
5.2	Comparison of number of Gauss-Seidel iterations needed for convergence of pressure Poisson equation. - - - Finite difference (diagonally dominant) matrix - - Finite element (slightly non-diagonally dominant) matrix — Finite volume (strongly non-diagonally dominant) matrix	58
5.3	Comparison of maximum momentum residual behavior. - - - Finite difference (diagonally dominant) matrix - - Finite element (slightly non-diagonally dominant) matrix — Finite volume (strongly non-diagonally dominant) matrix	59
5.4	Placement of primitive variables on a non-staggered grid.	64
5.5	One-dimensional domain for pressure gradient calculation on a non-staggered grid.	65
5.6	Uniform pressure field in a one-dimensional domain.	66
5.7	Odd-even pressure field in a one-dimensional domain.	66
5.8	Placement of primitive variables on a staggered grid.	66

5.9	One-dimensional domain for pressure gradient calculation on a staggered grid.	67
5.10	Percent error versus percent interior nodes for the divergence calculation of a linear analytic function around a sphere. - - - Prismatic mesh. — Hybrid mesh.	77
5.11	RMS error versus radial step size for the divergence and viscous calculations on a semi-unstructured prismatic sphere mesh. Δ Divergence calculation, $\circ$ Viscous calculation	79
6.1	Boundary conditions for flat plate simulation at $Re = 10^6$	82
6.2	Prismatic mesh with unstructured surface tessellation for flat plate at $Re = 10^6$	83
6.3	U velocity profile at the center of a flat plate ($Re = 10^6$). $\circ$ Blasius result, — Present work	84
6.4	Convergence of the maximum residual in the x-momentum equation for a flat plate ($Re = 10^6$).	84
6.5	Partial prismatic mesh for circular cylinder at $Re = 40$ with “structured” surface triangulation.	86
6.6	Partial prismatic mesh for circular cylinder at $Re = 40$ with “unstructured” surface triangulation.	87
6.7	Flow pathlines around a cylinder at $Re = 40$	88
6.8	Pressure coefficient distribution on cylinder surface for $Re=40$. $\circ$ Experiment by Grove, - - Present work using “structured” grid — Present work using “unstructured” grid	90

6.9	Vorticity distribution on cylinder surface for $Re=40$. $\circ$ Numerical results by Fornberg, - - Present work using “structured” grid, — Present work using “unstructured” grid	90
6.10	Convergence of the drag coefficient on a cylinder at $Re = 40$	92
6.11	Convergence of the maximum residual in the x-momentum equation on a cylinder at $Re = 40$	93
6.12	Growth in time of the separation bubble for an impulsive start of a cylinder $Re = 40$. Δ Experiment by Coutanceau, $\circ$ Experiment by Honji, — Present Work	96
6.13	Velocity contours on the x-z planes in the cubic cavity at $Re = 400$, at $y=0.015$ and $y=0.5$. Contour lines are plotted with velocity increments equal to 0.025.	99
6.14	Velocity contours on the y-z planes in the cubic cavity at $Re = 400$, at $x=0.1, 0.5$, and 0.9 . Contour lines are plotted with velocity increments equal to 0.05.	100
6.15	U velocity profile versus z on the symmetry plane at $x = 0.5$ for the cubic cavity at $Re = 400$. $\circ$ Numerical result by Babu, — Present work	101
6.16	W velocity profile versus x on the symmetry plane at $z = 0.5$ for the cubic cavity at $Re = 400$. $\circ$ Numerical result by Babu — Present work	101
6.17	Prismatic surface and symmetry plane mesh for sphere at $Re = 10$	104

6.18 Drag coefficient of a sphere versus Reynolds number. ·Tic at lower right represents error range of experimental data. - - - Stokes flow, - - - Oseen's equation, — Experimental data from Schlichting and Roos, o Present work with hemisphere grid	106
6.19 Pressure coefficient distribution on the surface of the sphere for $Re=0.1$ (Stokes flow).	107
6.20 Vorticity distribution on the surface of the sphere for $Re=0.1$ (Stokes flow).	107
6.21 Pressure coefficient distribution on the surface of the sphere for low Re . o $Re = 10$, $\triangle Re = 40$ + $Re = 100$, $\times Re = 200$	108
6.22 Vorticity distribution on the surface of the sphere for low Re . o $Re = 10$, $\triangle Re = 40$, + $Re = 100$, $\times Re = 200$	108
6.23 Pathlines on the x-z plane near a sphere at $Re = 200$	109
6.24 Pathlines on the x-y plane near a sphere at $Re = 200$	109
6.25 Coarse grid surface triangulation for cubic cavity at $Re = 400$	111
6.26 Adapted grid surface triangulation for cubic cavity at $Re = 400$. Surface grid shows embedding near the cavity walls.	112
6.27 Globally adapted grid surface triangulation for cubic cavity at $Re = 400$	112
6.28 Adaptation comparison of the u velocity profile versus z on the symmetry plane at $x = 0.5$ for the cubic cavity at $Re = 400$. — Coarse grid, - - - Locally adapted grid, — Globally adapted grid	113
6.29 Adaptation comparison of the w velocity profile versus x on the symmetry plane at $z = 0.5$ for the cubic cavity at $Re = 400$. — Coarse grid, - - - Locally adapted grid, — Globally adapted grid	113

6.30	Demonstration of robustness of prismatic adapter, convergence of the maximum residual of the x-momentum equation for the cubic cavity at $Re = 400$. - - - Locally adapted grid, — Globally adapted grid .	115
6.31	Evaluation of equation adaptation with hybrid mesh. Pressure coefficient distributions on the surface of the sphere for $Re=40$. — “Inviscid” Tetrahedra, - - - “Viscous” Tetrahedra	119
6.32	Evaluation of equation adaptation with hybrid mesh. Vorticity distributions on the surface of the sphere for $Re=40$. — “Inviscid” Tetrahedra, - - - “Viscous” Tetrahedra	120
6.33	Effect of prism/tetrahedra interface on the solution. Pathlines near a sphere for $Re=100$ are unaffected by the change in topology of the grid.	121
6.34	Effect of prism/tetrahedra interface on the solution. Velocity contours near a sphere for $Re=100$ are relatively unaffected by the change in topology of the grid.	122
6.35	Effect of prism/tetrahedra interface on the solution. Pressure contours near a sphere for $Re=100$ are relatively unaffected by the change in topology of the grid.	123
6.36	Pressure coefficient distribution on the surface of a sphere with a hybrid adapted mesh for $Re=100$. - - - Coarse grid, - - - Locally adapted grid, — Fine grid (globally adapted prism region)	127
6.37	Vorticity distribution on the surface of a sphere with a hybrid adapted mesh for $Re=100$. - - - Coarse grid, - - - Locally adapted grid, — Fine grid (globally adapted prism region)	128

6.38	Demonstration of robustness of hybrid adapter, maximum residual in the x-momentum equation versus time for the sphere at $Re = 100$. - - Locally adapted grid — Globally adapted grid	128
6.39	Adapted hybrid mesh and flow pathlines for sphere at $Re=100$. A view of the tessellation on the wall surface and symmetry plane. The hybrid grid is embedded isotropically in the tetrahedral region and directionally in the prismatic region.	129
6.40	Adapted hybrid mesh and corresponding flow velocity contours for sphere at $Re=100$. A view of the tessellation on the wall surface, symmetry plane and an interior equatorial plane. The hybrid grid is embedded isotropically in the tetrahedral region and directionally in the prismatic region.	130
6.41	Adapted hybrid mesh and pressure coefficient contours for sphere at $Re=100$. A view of the tessellation on the wall surface and symmetry plane. The hybrid grid is embedded isotropically in the tetrahedral region and directionally in the prismatic region.	131
6.42	Adapted hybrid grid and flow pathlines for tandem spheres at $Re=100$. A view of the tessellation on the wall surface and symmetry plane. The hybrid grid is embedded isotropically in the tetrahedral region and directionally in the prismatic region.	135

6.43	Velocity vectors on the symmetry plane between tandem spheres at $Re=100$. The flow is separated aft of the first sphere. The closeup view shows a portion of the surface mesh of the fore and aft spheres, the quadrilateral faces of the prismatic cells on the symmetry plane, and the triangular faces of the tetrahedral cells on the symmetry plane. Only the velocity vectors with magnitudes less than 0.015 are shown.	136
6.44	Adapted hybrid grid and flow velocity contours for tandem spheres at $Re=100$. A view of the tessellation on the wall surface, symmetry plane and an interior equatorial plane. The hybrid grid is embedded isotropically in the tetrahedral region and directionally in the prismatic region.	137
A.1	Prism cell division showing split orientation method.	144
C.1	Node and edge numbering system for prism cells.	153
C.2	Boundary edge-based node and prismatic cell numbering system.	154
D.1	Node and edge numbering system for tetrahedral cells.	156
E.1	Numerical perturbation to induce asymmetry.	162
E.2	General shape of lift force coefficient for transverse mode.	164
E.3	Numerical in-line force coefficient versus time for transverse mode, $KC = 8$ and $Re = 1568$. - - - Freestream velocity, — In-line force coefficient	166
E.4	Numerical lift force coefficient versus time for transverse mode, $KC = 8$ and $Re = 1568$. - - - Freestream velocity, — Lift coefficient	166

E.5	In-line force coefficient comparison for transverse mode, $KC = 8$ and $Re = 1568$. ◦ Instantaneous numerical data by Justesen, - - - Freestream velocity, — Phase averaged in-line force coefficient . . .	167
E.6	Lift force coefficient comparison for transverse mode, $KC = 8$ and Re $= 1568$. ◦ Instantaneous numerical data by Justesen, - - - Freestream velocity, — Phase averaged lift coefficient	167
E.7	General Shape of Lift Force Coefficient for Double Pair Mode	168
E.8	Numerical in-line force coefficient versus time for double pair mode, $KC = 15$ and $Re = 19050$. - - - Freestream velocity, — In-line force coefficient	171
E.9	Numerical lift force coefficient versus time for double pair mode, KC $= 15$ and $Re = 19050$. - - - Freestream velocity, — Lift coefficient .	171
E.10	In-line force coefficient comparison for double pair mode, $KC = 15$ and $Re = 19050$. ◦ Experimental phase averaged data by Obasaju, - - - Freestream velocity, — Phase averaged in-line force coefficient .	172
E.11	Lift force coefficient comparison for mirror image double pair mode, $KC = 15$ and $Re = 19050$. ◦ Negative of experimental phase averaged data by Obasaju, - - - Freestream velocity, — Phase averaged lift coefficient	173
E.12	Lift force coefficient comparison for double pair mode, $KC = 15$ and $Re = 19050$. ◦ Experimental phase averaged data by Obasaju, - - - Freestream velocity, — Phase averaged lift coefficient	173
F.1	Sampling of time varying data within specified time intervals. . . .	175
F.2	Weighted averaging of sampled data.	176

Chapter 1

Introduction

Incompressible flows are frequently encountered in engineering applications in the fields of aerodynamics, hydraulics, meteorology and hydrodynamics. Applications of incompressible flows are also found in specialized topics such as stratified flows, turbulence, biological fluid mechanics, analysis of cloud droplets and ice crystals, sediment transport, rotating flows and lubrication theory.

1.1 Incompressible versus Compressible Flow Equations

The incompressible flow equations are a subset of the compressible flow equations. Incompressible flows could be solved using the general compressible flow equations. For compressible flow solvers, the time step used to march the solution in time is proportional to the inverse of the speed of sound. As the incompressible flow limit is approached, the theoretical speed of sound approaches infinity. This results in the time step of compressible flow solvers approaching zero. As a consequence, numerical simulation of incompressible flows with compressible flow solvers becomes prohibitively slow and therefore computationally expensive. Implicit schemes are

often used to increase the time step limitation of explicit time marching methods. However, the increase in time step is practically less than a factor of 5-10 [7] due to the growth in truncation error. Therefore, for both explicit and implicit schemes, solution of the compressible governing equations is prohibitively expensive for incompressible flows. It is advantageous to employ the incompressible Navier-Stokes equations rather than the compressible equations. The incompressible Navier-Stokes equations appear simpler than the compressible form of the Navier-Stokes equations when isothermal flow is assumed because the energy equation is decoupled from the momentum and continuity equations. If the temperature is not of interest, the energy equation does not need to be solved for incompressible flows.

1.2 Issues Related to the Numerical Solution of the Incompressible Flow Equations

A few major issues, related to solving numerically the incompressible flow equations, must be addressed. The infinite theoretical speed of sound in incompressible flows requires special treatment in the solution procedure for the pressure field. The solution of the pressure field is often decoupled from the solution of the velocity field. Several distinct methods have been developed to solve incompressible flow equations which address the decoupling of the pressure and velocity fields. A second issue of importance is the use of staggered or non-staggered grids. The location of the pressure and velocity unknowns have significant consequences on the solver's convergence and solution. A third major issue is the topology of the computational grid employed. Structured grids are easier to use and the flow solver is simpler to develop. Application of structured grid solvers are limited as it is very difficult to

develop a structured grid about a complex three- dimensional surface. Unstructured grids have the capability to cover a complex shape, but unstructured flow solvers typically place substantial demands on computer storage requirements as well as being more difficult to develop. Historical development of these issues is presented as well as discussion of the methodology chosen for the present work.

1.2.1 Pressure/Velocity Decoupling

During the past two decades a significant number of numerical algorithms have been developed for the solution of the incompressible Navier-Stokes equations. In compressible flows, the conservation of mass is given as a partial differential equation for the temporal variation of density. In fact, the set of five equations: continuity, x, y, z-momentum and energy, can be solved explicitly for the unknown density, u,v,w velocities, and energy, since each equation contains the temporal variation of the unknown. The pressure is then determined from the energy equation using an equation of state. The four incompressible equations have the unknowns: u, v, w velocities and pressure. Lack of a pressure term in the continuity equation makes solution of the momentum equations with the divergence-free constraint more difficult. In the case of incompressible flows, conservation of mass acts as a constraint condition that the velocity field needs to satisfy. Furthermore, spatial discretization of the pressure and velocity may produce oscillatory solutions.

The three major categories of algorithms used to solve incompressible flows are shown in Figure 1.1. Historically, the first category (vorticity based formulations) calculates the values of the stream function and vorticity, and determines the velocity and pressure fields afterwards. The other two categories, artificial compressibility and pressure correction, both use the primitive variables, pressure and velocity, as

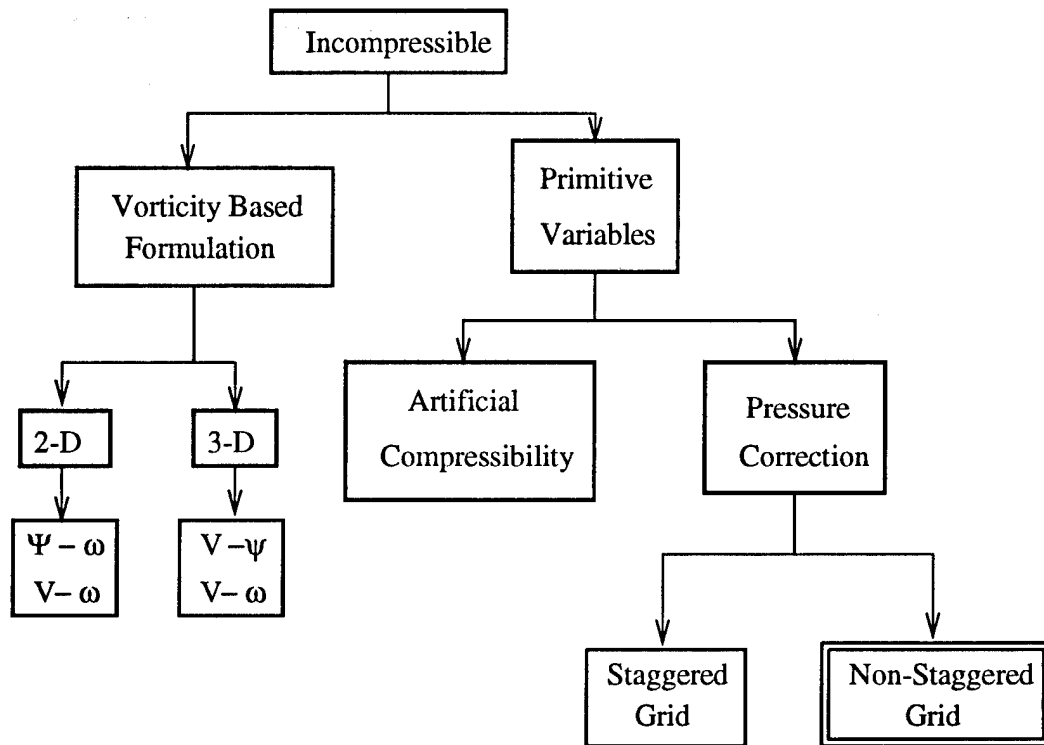


Figure 1.1: Major categories of incompressible flow algorithms.

where Ψ : stream function

ω : vorticity

V : velocity

ϕ : potential

unknowns but are significantly different in design.

The vorticity based formulation completely decouples the velocity and pressure calculations. In two-dimensional calculations, the governing equations are formulated in terms of a stream function and a vorticity [1, 2, 6, 26, 33, 40, 59, 110, 119, 137]. Direct extension of this method to three dimensions is not possible. However, different formulations have been used in three-dimensions, such as the vorticity-velocity approach [96], the vector-potential approach [132], and the vector stream function method [116]. The selection of boundary conditions is often

quite challenging for these methods. The methods also seem computationally expensive since multiple Laplace or Poisson equations need to be solved for each time iteration.

The artificial compressibility approach uses compressible-like governing equations. The artificial compressibility approach in two-dimensions [28, 30, 111, 140] and in three-dimensions [73, 121, 123, 146] adds a time derivative of the pressure to the continuity equation and the incompressible flow field is treated as compressible during the transient stage. Time accuracy of the simulation is usually not preserved, but recent progress has been made in the development of time accurate algorithms in two dimensions [16, 74, 82, 84, 95, 112, 113, 140] and three dimensions [85, 114]. Time accurate algorithms usually require additional iterations in pseudo-time to converge the solution at each time step.

The third class of algorithms, called pressure correction methods, use a Poisson equation for the pressure field [50, 99]. The usual computational procedure is to assume an initial pressure field, and then an iterative process is defined until the continuity equation is satisfied. The present work uses a pressure correction formulation.

1.2.2 Staggered versus Non-Staggered Grids

A major consideration of pressure correction methods is whether the grid is *staggered* or *non-staggered*. A non-staggered grid has velocity components and pressure collocated with each other. In the finite element context using similar approximation functions for the velocity and pressure is analogous to the use of non-staggered grids. The use of the non-staggered grid can produce oscillations in the pressure field. In the finite element context this is related to the inf-sup condition [24, 25].

In order to reject these oscillatory modes staggered grids, or different approximation functions for finite element methods, have been employed by several pressure correction type algorithms in two-dimensions [8, 15, 20, 21, 23, 27, 32, 39, 42, 50, 55, 56, 68, 71, 88, 89, 100, 103, 128, 129, 134, 135] and in three dimensions [19, 41, 47, 48, 51, 52, 60, 67, 69, 70, 72, 87, 127]. Use of these staggered grids with local embedding adaptive grids complicates the treatment of grid interfaces.

On the other hand, employment of non-staggered grids requires some special treatment of the pressure correction formulation to couple the velocity and pressure and eliminate the oscillations. One method modifies the continuity equation in the derivation of the pressure correction equations to include a slight compressibility effect in two-dimensions [58, 80, 124, 141] and three dimensions [10, 49]. A second method uses the momentum interpolation method in two-dimensions [13, 14, 29, 78, 83, 86, 102, 107] and in three-dimensions [79, 122, 147]. The momentum interpolation method assumes the velocities at any control volume face are driven by the pressure difference between the two adjacent nodes on either side of the face in question, which is similar to the basic concept of staggered grids. A linear interpolation is used to evaluate cell face convective and diffusive terms from the cell centered convective and diffusive terms. A third method requires dissipation in the algorithms in two-dimensions [64, 81, 91, 141] and three-dimensions [9, 106, 125, 126, 130]. Stability of both approaches with high Reynolds number flows is an important issue. Staggered and non-staggered grids will be discussed in greater detail in the section on Numerical Issues. The present work uses a non-staggered grid and artificial dissipation to eliminate oscillations.

1.2.3 Unstructured versus Structured Meshes

Simulation of flows around three-dimensional bodies is a major issue in computational fluid mechanics. The generation of a structured hexahedral grid which covers a complex three dimensional body has been proven to be very difficult [11]. Unstructured tetrahedral grids provide flexibility in three-dimensional grid generation since they can cover complicated topologies more easily compared to the hexahedral meshes. Unstructured grids have been used extensively for three-dimensional inviscid flow modeling of compressible flows. Substantial progress is currently being made in large scale viscous flow computations on complex three-dimensional geometries using compressible flow solvers. However, there has been very little analogous work in two-dimensions [81, 143] or in three dimensions [75, 77] using incompressible viscous flow solvers.

For viscous flow solvers, it is critical that the grid have clustered points in the normal direction to any no-slip boundaries. This usually results in high aspect ratio grid cells in the viscous region. It is difficult to generate high aspect ratio cells using an unstructured grid while minimizing the number of cells. Unstructured grids using tetrahedral cells also typically place substantial demands on computer storage requirements because of indirect addressing and arbitrariness of nodal distribution which requires the need for extensive connectivity information. It is therefore imperative that a numerical scheme be storage efficient. Most existing unstructured Navier-Stokes methods need a supercomputer to compute viscous flows on complex three-dimensional geometries. One of the prime objectives of the present work has been to implement a scheme that could fit viscous simulations in a workstation.

The mesh methodology proposed by [66, 90] provides an alternative approach to generate suitable grids for Navier-Stokes computations by using prismatic

elements. The prisms are unstructured in the lateral direction, while they are structured in the normal-to-surface direction. The prisms permit the use of sufficient grid clustering in the normal direction as well as flexibility in covering complex surfaces using their triangular faces. Prismatic data structures also reduce the storage requirements considerably compared to tetrahedral data structures.

The hybrid mesh [63, 138] incorporates both prismatic and tetrahedral elements. Viscous effects are dominant in a relatively small region of the flow domain, in close proximity to the no-slip wall. Inviscid flow features that are dominant away from the body do not exhibit the directionality that is characteristic of viscous stresses. If prismatic cells are used in the viscous portion of the flow, they will fill only a small fraction of the computational domain. Since unstructured grids have the capability to fill a given volume with fewer cells, the inviscid portion of the domain is tessellated with tetrahedra which are generated starting from the triangulation of the outermost surface of the prismatic region.

1.3 Present Research

The primary goal of this dissertation is to develop and evaluate a three-dimensional incompressible Navier-Stokes method using hybrid (prismatic/tetrahedral) grids. The flow domain is discretized with prismatic and tetrahedral elements using a non-staggered grid. The use of the hybrid mesh enables the solver to compute accurate solutions with relatively less memory requirements than a fully unstructured mesh. The use of prismatic cells to discretize the full Navier-Stokes equations and tetrahedra cells to discretize the Euler equations renders the solver equation-adaptive. The present work is the first procedure which computes three-dimensional incompressible viscous flows with hybrid elements.

A secondary goal is to apply grid adaptation to the present method. A hybrid grid adaptation scheme implementing local mesh refinement provides optimum grids for viscous flow computations. Grid refinement is a dual adaptation scheme that couples a three-dimensional isotropic division of tetrahedra and two-dimensional directional division of prisms. The advantage of using local adaptation is examined; namely, yielding accurate results with reduced computing resources.

A tertiary goal is to study the forces on a circular cylinder in an oscillating freestream flow. This study uses an existing two-dimensional incompressible Navier-Stokes solver.

1.4 Overview of Dissertation

A brief summary of the contents of each chapter follows.

Chapter 2 details the pressure correction numerical method used in the present work. The Navier-Stokes equations are non-dimensionalized and discretized using a non-staggered grid. Equations for the pressure correction method are derived and methodology described. The numerical method was developed using the ideas and formulations of many different authors. The synthesis of these different techniques resulted in a numerical method which is unique.

Chapter 3 describes the adaptive hybrid (prismatic/tetrahedral) grid. Equation adaptation as well as the spatial adaptation methodology is detailed and includes a description of the flow feature detection algorithm.

Chapter 4 details the integration scheme developed in the present work. The concept of the dual cell used in the edge-based integration is described, as well as the finite volume discretization of the momentum equations. Finite element discretization of the pressure Poisson equation is described and includes the construction of the

global matrix. Initial conditions are described and special considerations required for boundary conditions on unstructured grids is presented as well as the artificial dissipation methodology, time step calculation, and solver computer requirements.

Chapter 5 details the major numerical issues encountered in the development of this three-dimensional incompressible adaptive Navier-Stokes solver. Pressure correction formulations are discussed, and the importance of diagonal dominance and boundary conditions for the Poisson equation are emphasized. The consequence of non-staggered versus staggered grids is presented as well as an explanation of the presence of a non-zero discrete divergence. Accuracy of spatial discretization is determined by testing with analytic field functions.

Chapter 6 presents the validation test cases of the adaptive hybrid solver. Flow over a flat plate is used to demonstrate the stability of the solver at high Reynolds number. A quasi-two-dimensional cylinder, three-dimensional cubic cavity and sphere are used to demonstrate the validity of the all-prismatic solver for steady flows. An impulsive start of flow around the cylinder is used to validate time accuracy of the solver. The cubic driven cavity case is used to illustrate the usefulness of prismatic adaptation, and the sphere is used to demonstrate hybrid adaptation. A special case of flow over tandem spheres is used to demonstrate the suitability of hybrid adaptation.

Chapter 7 presents a summary of the present work, and includes the contributions of this research, conclusions, and recommendations for further research.

Appendix E presents the study of the hydrodynamic forces on a circular cylinder in an oscillating freestream flow using an existing two-dimensional solver. The simulations are conducted for two oscillating flow regimes and the lift and in-line force coefficients are compared against both numerical and experimental results.

Chapter 2

Pressure Correction Numerical Method

The laws of conservation of mass and momentum for fluid flow are expressed in integral form for a three-dimensional domain. The three-dimensional Navier-Stokes equations are then derived and expressed in non-dimensional form. The semi-discrete form of the governing equations are presented and the pressure correction formulation is described.

2.1 3-D Navier-Stokes Equations

Consider a volume Ω enclosed by a boundary surface $\partial\Omega$ with $\hat{n}$ being the outward normal on the boundary surface. The velocity of the fluid is given by $\vec{V}$, and the fluid in the control volume is assumed to be non-reacting and isothermal.

The law of conservation of mass requires the time rate of decrease of mass of the fluid inside the control volume equal the net mass flow out of the control volume

through the boundary surface. This is shown in equation (2.1) in integral form.

$$\frac{d}{dt} \int_{\Omega} \rho dV + \oint_{\partial\Omega} \rho (\vec{V} \cdot \hat{n}) dS = 0 \quad (2.1)$$

The law of conservation of momentum requires the time rate of change of momentum due to unsteady fluctuations of flow properties inside the control volume plus the net flow of momentum out of the control volume across the boundary surface be balanced by the forces that act on the control volume and its boundary surface. This is shown in equation (2.2) in integral form.

$$\frac{d}{dt} \int_{\Omega} \rho \vec{V} dV + \oint_{\partial\Omega} \rho \vec{V} (\vec{V} \cdot \hat{n}) dS = \sum \vec{F} \quad (2.2)$$

Neglecting the body force terms, the net force acting on the surface of the control volume is given by

$$\sum \vec{F} = \oint_{\partial\Omega} (-p + \tau) \hat{n} ds \quad (2.3)$$

where p and τ are the normal pressure and shear stress tensor, respectively.

The mass and momentum integral equations in conservation form can be written collectively in integral form as

$$\frac{d}{dt} \int_{\Omega} \mathbf{U} dV + \oint_{\partial\Omega} \mathbf{F}(\mathbf{U}) \cdot \hat{n} ds = 0 \quad (2.4)$$

where $\mathbf{F} = (\mathbf{F}_I - \mathbf{F}_V)\hat{i} + (\mathbf{G}_I - \mathbf{G}_V)\hat{j} + (\mathbf{H}_I - \mathbf{H}_V)\hat{k}$. In differential form in Cartesian coordinates the equations can be written as:

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}_I}{\partial x} + \frac{\partial \mathbf{G}_I}{\partial y} + \frac{\partial \mathbf{H}_I}{\partial z} = \frac{\partial \mathbf{F}_V}{\partial x} + \frac{\partial \mathbf{G}_V}{\partial y} + \frac{\partial \mathbf{H}_V}{\partial z} \quad (2.5)$$

The dimensional state vector $\mathbf{U}$, the inviscid flux vectors $\mathbf{F}_I, \mathbf{G}_I, \mathbf{H}_I$, and the viscous flux vectors $\mathbf{F}_V, \mathbf{G}_V, \mathbf{H}_V$, are expressed in terms of the dimensional primitive variables, density ρ , pressure p and x, y, z velocities, u, v , and w . The first entry in the state vectors corresponds to the conservation of mass equation, while the

second, third, and fourth terms correspond to the x, y and z momentum equations, respectively.

$$\mathbf{U}^T = \begin{pmatrix} 0 & u & v & w \end{pmatrix} \quad (2.6)$$

$$\mathbf{F_I}^T = \begin{pmatrix} u & u^2 + \frac{p}{\rho} & uv & uw \end{pmatrix} \quad (2.7)$$

$$\mathbf{G_I}^T = \begin{pmatrix} v & vu & v^2 + \frac{p}{\rho} & vw \end{pmatrix} \quad (2.8)$$

$$\mathbf{H_I}^T = \begin{pmatrix} w & wu & wv & w^2 + \frac{p}{\rho} \end{pmatrix} \quad (2.9)$$

$$\mathbf{F_V}^T = \mu \begin{pmatrix} 0 & \frac{\partial u}{\partial x} & \frac{\partial v}{\partial x} & \frac{\partial w}{\partial x} \end{pmatrix} \quad (2.10)$$

$$\mathbf{G_V}^T = \mu \begin{pmatrix} 0 & \frac{\partial u}{\partial y} & \frac{\partial v}{\partial y} & \frac{\partial w}{\partial y} \end{pmatrix} \quad (2.11)$$

$$\mathbf{H_V}^T = \mu \begin{pmatrix} 0 & \frac{\partial u}{\partial z} & \frac{\partial v}{\partial z} & \frac{\partial w}{\partial z} \end{pmatrix} \quad (2.12)$$

2.2 Nondimensionalization

The laminar governing equations (2.5) are normalized by a characteristic length, L , freestream density ρ_∞ , and freestream speed, U_∞ . Terms in the equations are rearranged into inviscid, viscous, and source terms. The non-dimensional Navier-Stokes equations in three dimensions can be written in Cartesian coordinates in differential form:

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F_I}}{\partial x} + \frac{\partial \mathbf{G_I}}{\partial y} + \frac{\partial \mathbf{H_I}}{\partial z} = \frac{\partial \mathbf{F_V}}{\partial x} + \frac{\partial \mathbf{G_V}}{\partial y} + \frac{\partial \mathbf{H_V}}{\partial z} + \mathbf{S} \quad (2.13)$$

The state vector $\mathbf{U}$, the convective flux vectors $\mathbf{F_I}, \mathbf{G_I}, \mathbf{H_I}$, the viscous flux vectors $\mathbf{F_V}, \mathbf{G_V}, \mathbf{H_V}$, and the source term $\mathbf{S}$ are expressed in terms of the non-dimensional primitive variables, namely, density ρ , x, y, z - velocities and pressure p .

$$\mathbf{U}^T = \begin{pmatrix} 0 & u & v & w \end{pmatrix} \quad (2.14)$$

$$\mathbf{F_I}^T = \begin{pmatrix} u & u^2 & uv & uw \end{pmatrix} \quad (2.15)$$

$$\mathbf{G_I}^T = \begin{pmatrix} v & vu & v^2 & vw \end{pmatrix} \quad (2.16)$$

$$\mathbf{H_I}^T = \begin{pmatrix} w & wu & wv & w^2 \end{pmatrix} \quad (2.17)$$

$$\mathbf{F_V}^T = \frac{1}{Re} \begin{pmatrix} 0 & \frac{\partial u}{\partial x} & \frac{\partial v}{\partial x} & \frac{\partial w}{\partial x} \end{pmatrix} \quad (2.18)$$

$$\mathbf{G_V}^T = \frac{1}{Re} \begin{pmatrix} 0 & \frac{\partial u}{\partial y} & \frac{\partial v}{\partial y} & \frac{\partial w}{\partial y} \end{pmatrix} \quad (2.19)$$

$$\mathbf{H_V}^T = \frac{1}{Re} \begin{pmatrix} 0 & \frac{\partial u}{\partial z} & \frac{\partial v}{\partial z} & \frac{\partial w}{\partial z} \end{pmatrix} \quad (2.20)$$

$$\mathbf{S}^T = \begin{pmatrix} 0 & -\frac{\partial p}{\partial x} & -\frac{\partial p}{\partial y} & -\frac{\partial p}{\partial z} \end{pmatrix} \quad (2.21)$$

$$(2.22)$$

The Reynolds number is defined as

$$Re = \frac{\rho_\infty U_\infty L}{\mu_\infty} \quad (2.23)$$

where ρ_∞ is the freestream density, μ_∞ is the freestream viscosity, U_∞ is the freestream velocity, and L is a characteristic length.

2.3 Semi-Discrete Form of Governing Equations

An explicit/implicit marching scheme is adopted for the integration in time of equations (2.13). The velocity values are treated explicitly, while the pressure values are treated implicitly in the momentum equations. The velocity values are marched in time with a forward Euler scheme [50]. The continuity equation is formulated implicitly with the velocity values considered at time level $(n + 1)$. Specifically, the corresponding semi-discrete system is written as follows, where the superscripts

denote the time levels.

$$\mathbf{U}^T = \begin{pmatrix} 0 & u^{n+1} & v^{n+1} & w^{n+1} \end{pmatrix} \quad (2.24)$$

$$\mathbf{F_I}^T = \begin{pmatrix} u^{n+1} & u^n u^n & u^n v^n & u^n w^n \end{pmatrix} \quad (2.25)$$

$$\mathbf{G_I}^T = \begin{pmatrix} v^{n+1} & v^n u^n & v^n v^n & v^n w^n \end{pmatrix} \quad (2.26)$$

$$\mathbf{H_I}^T = \begin{pmatrix} w^{n+1} & w^n u^n & w^n v^n & w^n w^n \end{pmatrix} \quad (2.27)$$

$$\mathbf{F_V}^T = \frac{1}{Re} \begin{pmatrix} 0 & \frac{\partial u^n}{\partial x} & \frac{\partial v^n}{\partial x} & \frac{\partial w^n}{\partial x} \end{pmatrix} \quad (2.28)$$

$$\mathbf{G_V}^T = \frac{1}{Re} \begin{pmatrix} 0 & \frac{\partial u^n}{\partial y} & \frac{\partial v^n}{\partial y} & \frac{\partial w^n}{\partial y} \end{pmatrix} \quad (2.29)$$

$$\mathbf{H_V}^T = \frac{1}{Re} \begin{pmatrix} 0 & \frac{\partial u^n}{\partial z} & \frac{\partial v^n}{\partial z} & \frac{\partial w^n}{\partial z} \end{pmatrix} \quad (2.30)$$

$$\mathbf{S}^T = \begin{pmatrix} 0 & -\frac{\partial P^{n+1}}{\partial x} & -\frac{\partial P^{n+1}}{\partial y} & -\frac{\partial P^{n+1}}{\partial z} \end{pmatrix} \quad (2.31)$$

2.4 Pressure Correction Method

Equations (2.13) through (2.31) cannot be solved directly due to the implicit treatment of the pressure term. An auxiliary velocity vector $\mathbf{U}'^T = (0, u', v', w')$ is introduced, which can be written in the same nondimensional state vector formulation used previously:

$$\frac{\partial \mathbf{U}'}{\partial t} + \frac{\partial \mathbf{F_I}}{\partial x} + \frac{\partial \mathbf{G_I}}{\partial y} + \frac{\partial \mathbf{H_I}}{\partial z} = \frac{\partial \mathbf{F_V}}{\partial x} + \frac{\partial \mathbf{G_V}}{\partial y} + \frac{\partial \mathbf{H_V}}{\partial z} \quad (2.32)$$

In the auxiliary velocity state vector equation (2.32), the pressure term does not exist, and $\vec{U}'$ can be obtained directly since it contains terms only at the known time level (n). However, the solution $\vec{U}'$ does not satisfy the continuity equation. Subtracting (2.32) and from the momentum equations (2.13 - 2.31) rewriting in primitive variables, it is obtained:

$$\vec{u}^{(n+1)} - \vec{u}' = - [\nabla p^{(n+1)}] \Delta t \quad (2.33)$$

Introducing a scalar potential ϕ , such that

$$\vec{u}^{(n+1)} - \vec{u}' = -\nabla \phi, \quad (2.34)$$

the following equation for pressure can be obtained.

$$p^{(n+1)} = \frac{1}{\Delta t} \phi \quad (2.35)$$

Finally, taking the divergence of each side of equation (2.34) and considering the continuity equation (2.24) through (2.26), the following pressure correction Poisson equation is obtained.

$$\nabla^2 \phi = \nabla \cdot \vec{u}' \quad (2.36)$$

Using the ϕ values obtained by equation (2.36), we can correct the velocity and pressure fields using equations (2.34) and (2.35) as follows:

$$\vec{u}^{(n+1)} = \vec{u}' - \nabla \phi \quad (2.37)$$

$$p^{(n+1)} = \frac{1}{\Delta t} \phi \quad (2.38)$$

The present solution procedure is similar to the pressure correction method in [145]. The overall solution procedure corresponding to marching by one-time-step is summarized as follows:

1. Calculate the auxiliary velocity vector $\vec{u}'$ from (2.32) using $\vec{u}^{(n)}$ values.
2. Solve the pressure correction Poisson equation (2.36) and obtain the ϕ values.
3. Correct the pressure and velocity at the $n+1$ time step using equations (2.37) and (2.38).
4. Advance to the next time step.

Chapter 3

Adaptive Hybrid (Prismatic/Tetrahedral) Grid

The capability of the present code to handle complex geometries is expanded with the use of a hybrid grid. The hybrid grid is composed of semi-unstructured prismatic layered cells near the surface, and unstructured tetrahedra cells to fill the remaining portion of the domain. Use of the hybrid grid also takes advantage of equation adaptation. The Navier-Stokes equations are solved in the prismatic region, while only the Euler equations are solved in the tetrahedral region.

3.1 Hybrid Mesh

Simulation of flows around three-dimensional bodies is a major issue in computational fluid mechanics. Geometric and flow-field complexity combine to make three-dimensional computations a pacing item. The generation of a body-conforming grid has proven to be a difficult task [11]. Success of structured grid generation may be extremely dependent on geometry and operator proficiency. Block-structured

schemes exist which, based on extensive user input, break the computational domain into a number of blocks within which hexahedra are constructed. This method requires special treatment to ensure continuity of the mesh across the boundaries of neighboring blocks. These aspects point away from structured grid approaches toward methods which require only suitable definitions of surface and outer boundaries.

A radical alternative to structured meshes is to use tetrahedra. Tetrahedral grids provide flexibility easily in three-dimensional topologies compared to the hexahedral meshes [76, 101, 144]. This does not come without a price; unstructured grids require a great deal more memory than their structured counterparts. They employ pointers to provide connectivity information between cells, faces, edges, and nodes. Additionally, approximately five to six times more tetrahedra than hexahedra are required to fill a given region with a fixed number of nodes. This vast increase in the number of required cells leads directly to impractical memory requirements for three-dimensional viscous flow simulations. Furthermore, resolution of the strong directional flow gradients encountered in viscous flows requires very thin grid elements. It is very expensive to generate tetrahedral cells with high aspect ratios to resolve such gradients.

One solution to the dilemma between hexahedra and tetrahedra is to use a semi-unstructured grid made of prisms. Prismatic cells are composed of triangular faces in the lateral (body-surface) directions and quadrilateral faces in the normal direction as shown in Figure 3.1. Therefore, prisms can provide the geometric flexibility of unstructured grids as well as the orthogonality and high aspect ratio qualities of structured grids. Results have been obtained using prismatic grids that reveal their suitability for resolving viscous flow phenomena [63, 66, 90, 98]. The

prismatic grid requires a set of pointers to define their base triangular mesh combined with a single index for each prism belonging to the same stack [98]. This reduces the memory storage of required pointers to slightly more than a two-dimensional unstructured solver.

Areas between different prismatic layers covering the surfaces of the domain can be quite irregular. Furthermore, relevant flow features do not usually exhibit the strong directionality that are characteristic of viscous stresses. Tetrahedral elements appear to be appropriate for these irregularly shaped regions. Their triangular faces can match the corresponding triangular faces of the prisms. Tetrahedral cells are created from these triangular faces as shown in Figure 3.2. The present work employs two families of grid elements: prismatic grid cells for the viscous region and tetrahedral grid cells elsewhere. In general, the extent of the prismatic region is specified by the user when the grid is created, and as a minimum should enclose the entire boundary layer.

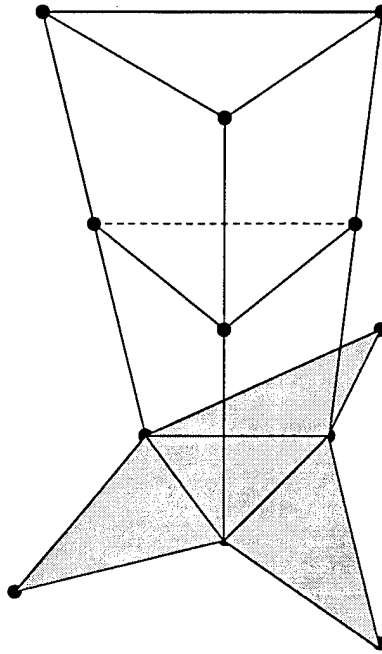


Figure 3.1: Semi-unstructured prismatic topology. (The unstructured boundary surface is shaded.)

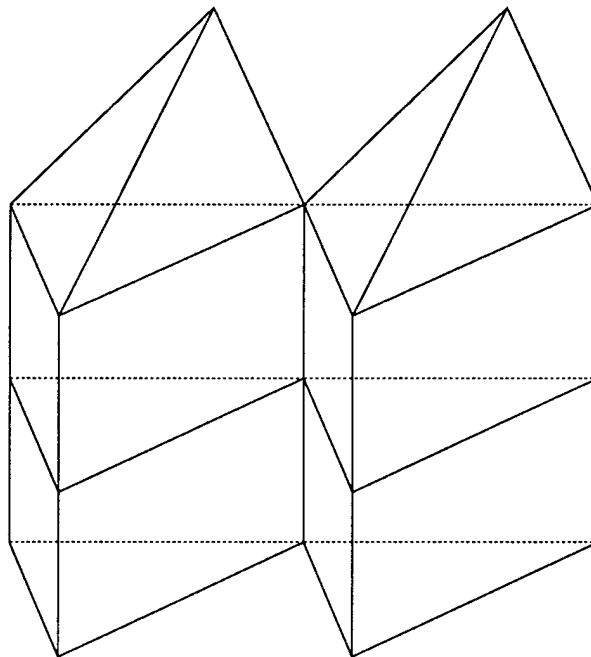


Figure 3.2: Interface region between prismatic cells and tetrahedral cells.

3.2 Equation Adaptation for Hybrid Mesh

The Navier-Stokes equations are the general governing equations for fluid flows. However, often flows can be adequately described with partial forms of the Navier-Stokes equations. For example, inviscid flows can be adequately computed with the Euler equations. The majority of flows consist of two regions, a region near the surface where the magnitude of the viscous terms is appreciable, and an outer inviscid region where viscosity can be neglected. Equation adaptation takes advantage of this property and solves the Navier-Stokes equations in this inner region, while the simpler Euler equations are employed elsewhere.

Evaluation of the viscous terms of the Navier-Stokes equations is usually the most expensive portion of the computer code. For incompressible codes, the solution of the Poisson equation is the most expensive portion of the code, but the resources needed for the viscous terms calculation are quite significant. Omitting the viscous terms for a large portion of the flow field is therefore advantageous in terms of computer memory storage and CPU time.

3.3 Spatial Grid Adaptation

Spatial grid adaptation adjusts the grid by altering the local distribution of points. There are two ways of adapting the grid. The redistribution method moves the existing nodes to cluster the points in regions which require further resolution. The second method, embedding, adds additional nodes in the regions which require further resolution, by dividing the cells in the region. The present work employs adaptive local embedding of an initial mesh according to a method developed in [98].

3.3.1 Detection of Cells to be Adapted

The success of spatial grid adaptation is dependent upon accurately selecting the regions which require finer resolution. Regions are usually selected on the basis of certain flow features. There are several flow parameters which can be used in the flow feature detection algorithm [62, 65]. For the present work, *velocity gradients* and *velocity differences* will be the flow feature parameters used to determine where local embedding should occur. Another variable in local adaptation is the choice of threshold values of the detection parameters for embedding. Regional adaptation can also be used to embed certain regions of the mesh based on x,y,z coordinates.

The flow detection algorithm is edge based. The variation in the flow parameters across the edges is monitored during the adaptive computation. The mean and standard deviation of the velocity gradients and velocity differences across the edges are calculated for the entire flow field. If the value of the detection parameter, φ , at a particular edge is greater than the corresponding average value plus a fraction of its standard deviation, the edge is flagged for embedding:

$$\varphi_{threshold} = \varphi_{mean} + \alpha \varphi_{sd} \quad (3.1)$$

This factor, α , is called the *threshold parameter* and is determined by numerical experimentation. The smaller the threshold parameter, the higher the percentage of flagged edges and more refinement will occur. If the threshold parameter is small enough, global refinement will occur.

If two edges of a face are flagged for refinement, the third edge is also flagged. The method of *extended embedding* may also be used. This technique introduces additional rows of embedded faces at the borders of the regions flagged for division. For example, if one edge of a face has been flagged for embedding, extended embedding

will flag the remaining two edges to be refined. In effect, this extends the embedded region by one “row” of faces. Extended embedding can be repeated multiple times to flag additional rows of cells.

The cell-detection procedure is summarized as follows:

1. Start the flow simulation on a relatively coarse grid and run the calculations until the basic flow features have developed.
2. Using the coarse grid and solution data, loop through all grid edges and calculate the velocity gradients and velocity differences.
3. Calculate the mean and standard deviation of the distributions of the velocity gradients and velocity differences, determine if the value of the velocity gradients or velocity differences of an edge exceeds the threshold value, if so mark the edge.
4. Apply extended embedding if desired.

3.3.2 Local Surface Grid Embedding for Prismatic Mesh

A prismatic mesh is created from a triangular mesh distribution on the surface, and the prismatic cells are grown away from the surface in the normal direction [63, 66]. The prismatic grid is refined by dividing the triangular surface faces. The prismatic cells are then grown from the refined surface mesh as usual. This type of adaptation increases the lateral resolution of the grid, and does not destroy the structure of the prismatic mesh. A surface mesh can initially be relatively coarse. Adaptive embedding subdivides the coarse cells in the regions which require finer resolution. The subdivision principle is illustrated in Figure 3.3. A triangle can be subdivided into two triangles (binary division) or it can be divided into four triangles

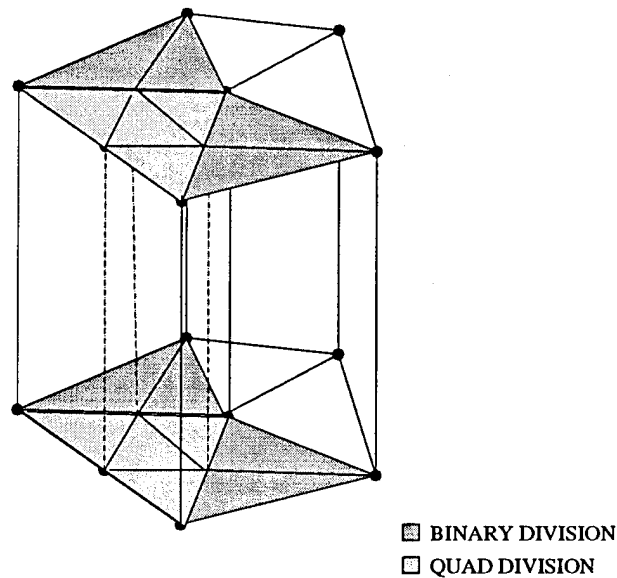


Figure 3.3: Local surface grid embedding for prismatic cells: shaded regions illustrate binary and quad triangular face division.

(quad division). When a triangular face is flagged for embedding, it undergoes quad division. If the neighbor faces are not flagged for refinement, the quad division leaves three hanging nodes on the original face which are eliminated by dividing the neighboring triangular face into two triangles (binary division).

The division of cells produces new nodes into the mesh. The location of these new nodes are interpolated from the two nodes of the edge that is divided. If the two nodes lie on a curved surface, the interpolated new node will usually not lie on the surface. After the surface grid is embedded, the new nodes are moved to a location which is consistent with the curved surface. Interpolated values of the velocities, pressure, and pressure correction (ϕ) parameters are also assigned to the new nodes. Local embedding can be repeatedly applied to obtain multiple levels of adaptation. This results in successively finer locally embedded grids until a region is adequately resolved.

3.3.3 Tetrahedral Embedding

Tetrahedral cells constitute the portion of the grid where inviscid flow features are dominant. These features do not exhibit the directionality that is generally prevalent in viscous stresses. Hence, the tetrahedra are refined isotropically, employing *binary*, *quadtrees* and *octree* divisions as shown in Figure 3.4. As the feature detector flags only the edges for division, there will arise situations where the flagged edges of a tetrahedron may not allow one of these three types of division. To avoid this problem and to simplify the refinement algorithm, such cases are constrained by flagging additional edges in the tetrahedron so that the cell conforms to one of the refinement types shown in Figure 3.4. The tetrahedral cells created by 1:8 *octree* division have an aspect ratio that is comparable to that of the parent cell whereas the 1:2 *binary* and 1:4 *quadtrees* divisions result in skewed cells. To avoid excessive grid skewness, repeated binary and quadtree divisions of tetrahedra that originated from such a refinement in a previous pass is avoided. Furthermore, to avoid sudden changes in grid size, the grid refinement algorithm also limits the maximum difference in embedding level between neighboring cells to two. The hybrid adaptation is completed by splicing the refined prismatic and tetrahedral grids together at the interface. The feature detectors that flag the edges in the prismatic and tetrahedral regions function independently. The following situations may occur:

1. An edge on the wall surface may be flagged for refinement but its *counterpart* in the tetrahedral region may not have been flagged by the tetrahedral feature detector.

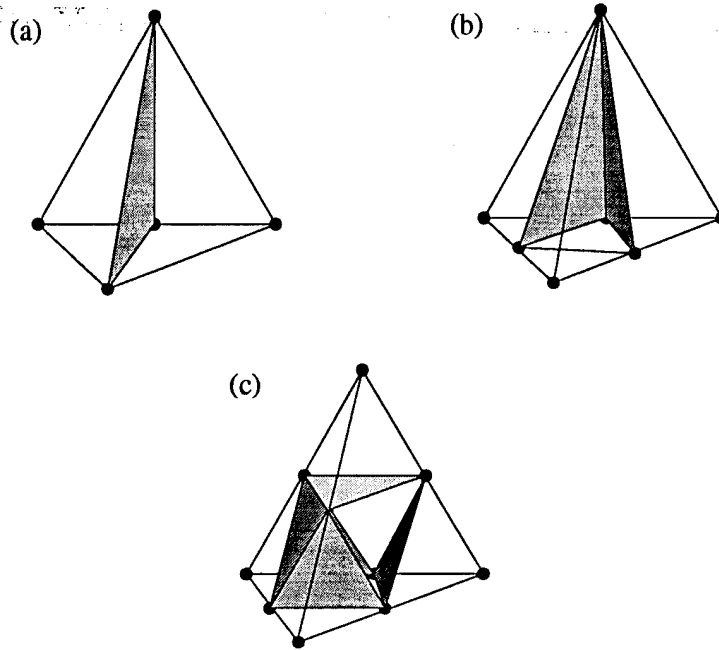


Figure 3.4: Refinement strategies for tetrahedron: (a) *Binary* (1:2), (b) *Quadtree* (1:4) and, (c) *Octree* (1:8) divisions.

2. A tetrahedral edge may be flagged for refinement, but its *footprint* on the wall may not have been flagged by the prismatic feature detector.

In the above cases, mid-edge nodes (hanging) nodes will appear which would then require special treatment by the solver. Hanging nodes are avoided by flagging both cells of a prism-tetrahedron pair at the interface, if at least one cell is flagged for division.

Chapter 4

Integration Scheme

Spatial integration begins by conceptually constructing around each node, N , a dual cell. By integrating over the dual cell, most of the integration can be completed using an efficient edge-based method. Finite volume discretization of the momentum equation is presented in detail as is finite element discretization of the pressure Poisson equation. Initial and boundary conditions for the hybrid grid, artificial dissipation methodology, time step calculation and the solver's computer requirements are also presented.

4.1 Node-Centered Dual Cell

The present 3-D numerical method uses a non-staggered cell-vertex scheme; all the primitive variables are defined at the cell corners (nodes). The cell defined by these corner nodes is called the basic cell. An additional cell can be conceptually defined, and this cell is called a node-centered *dual cell*. The node-centered dual cell represents the control volume over which the integral averages of equation (2.4) are evaluated. The two-dimensional analogy of defining node-centered dual cells

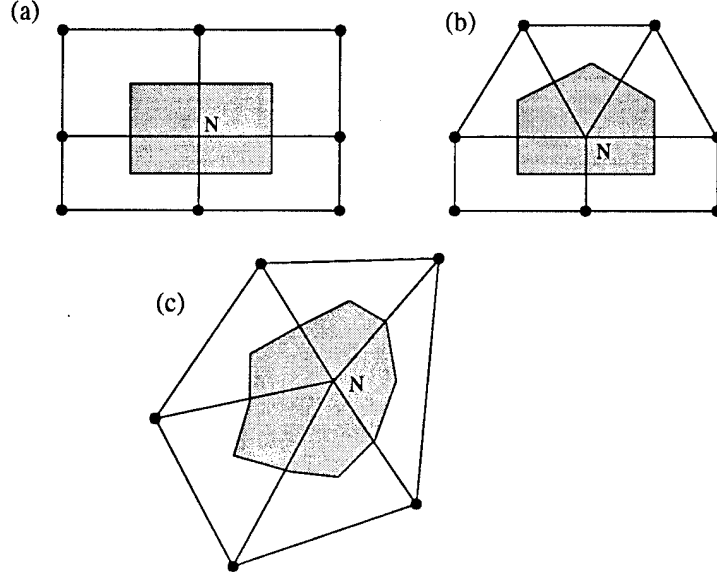


Figure 4.1: 2-D analogies for node-centered dual mesh in the (a) Prismatic region (b) Tetrahedral-Prismatic interface and (c) Tetrahedral region.

for different situations in a triangular-quadrilateral hybrid mesh is illustrated in Figure 4.1. Node-centered dual cells are defined by connecting the mid-points of the edges and centroids of the triangular and/or quadrilateral faces that share the node. Node-centered dual cells for a three-dimensional hybrid grid are constructed along similar lines using the centroids of faces and cells that each node is associated.

The integral in equation (2.4) is written in discrete form as

$$\left(\frac{\partial \mathbf{U}}{\partial t} \right)_N = \frac{-1}{\Omega_N} \sum_f (\mathbf{F})_f \cdot \hat{n}_f S_f \quad (4.1)$$

where the summation f is over all the discrete faces of the node-centered dual mesh that constitute $\partial\Omega_N$. The summation in equation (4.1) can be alternatively computed on an edge-wise basis [136] as

$$\left(\frac{\partial \mathbf{U}}{\partial t} \right)_N = \frac{-1}{\Omega_N} \sum_e (\mathbf{F})_e \cdot \hat{n}_e S_e \quad (4.2)$$

where the summation e is over all the edges that share the node N . The term S_e

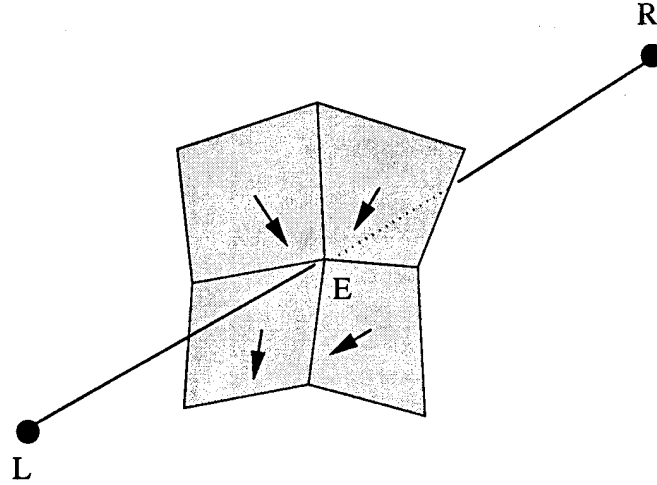


Figure 4.2: Node-centered dual cell surfaces attached to an edge.

represents the node-centered dual edge area associated with each edge and $\hat{n}_e$ is the normal vector of the node-centered dual edge area S_e . The areas S_e are computed using the node-centered dual mesh construction of Figure 4.1, by accumulating the areas of each node-centered dual mesh face that shares the edge e . An example of the node-centered dual edge area associated with a single edge is shown in Figure 4.2. The number of faces connected to an edge depends on the number of edges that shares that node. Thus, the flux summation over the node-centered dual cell surfaces is equivalent to the summation over the edges of the grid and the fluxes through the node-centered dual mesh faces associated with each edge. Details of edge-based operations are described in [136]. The finite volume scheme then proceeds by computing δU s at the nodes by a global sweep over the edges and is thus transparent to whether a node lies in the tetrahedral region, prismatic region or at the interface.

4.2 Finite Volume Discretization of the Momentum Equations

The Navier-Stokes equations of incompressible viscous flow are discretized using the finite volume approach with a non-staggered grid [61]. The equations are given in integral form for a bounded three-dimensional domain Ω as follows:

$$\frac{1}{\Omega} \int_{\Omega} \left[\frac{\partial \mathbf{U}}{\partial t} + \left(\frac{\partial \mathbf{F}_I}{\partial x} + \frac{\partial \mathbf{G}_I}{\partial y} + \frac{\partial \mathbf{H}_I}{\partial z} \right) \right] d\Omega = \frac{1}{\Omega} \int_{\Omega} \left[\left(\frac{\partial \mathbf{F}_V}{\partial x} + \frac{\partial \mathbf{G}_V}{\partial y} + \frac{\partial \mathbf{H}_V}{\partial z} \right) + \mathbf{S} \right] d\Omega \quad (4.3)$$

The state, flux and source vectors have been previously defined in equations (2.24) through (2.31).

4.2.1 Inviscid Terms

In the finite volume approach, the volume integral containing the spatial derivatives in equation (4.3) is transformed to a surface integral using the divergence theorem. Considering the convective flux vector terms, we have:

$$\frac{1}{\Omega} \int_{\Omega} \left(\frac{\partial \mathbf{F}_I}{\partial x} + \frac{\partial \mathbf{G}_I}{\partial y} + \frac{\partial \mathbf{H}_I}{\partial z} \right) d\Omega = \frac{1}{\Omega} \int_{\partial\Omega} (\mathbf{F}_I n_x + \mathbf{G}_I n_y + \mathbf{H}_I n_z) dS \quad (4.4)$$

Convective fluxes are calculated at each node, so the contributions from each face of the node-centered dual cell is accumulated. The surface integral on the right hand side of equation (4.4) is rewritten in discrete form on an edge-wise basis,

$$\frac{1}{\Omega_n} \sum_e (\mathbf{F}_I S_x + \mathbf{G}_I S_y + \mathbf{H}_I S_z)_e \quad (4.5)$$

where the summation is over all the edges, Ω_n is the volume of the node-centered dual cell, and S_x, S_y, S_z are the node-centered dual cell face area projections associated with each edge e in the corresponding coordinate directions. The areas S_e are

computed using the dual mesh construction of Figure 4.1, by accumulating the areas of each node-centered dual cell face that shares the edge e . The flux vectors are needed at the center of the edge e and their values are obtained by averaging the values at the two nodes of each edge.

4.2.2 Viscous Terms

The volume integral containing the spatial derivatives in equation (4.3) is transformed to a surface integral using the divergence theorem. Considering the viscous flux vector terms, we have:

$$\frac{1}{\Omega} \int_{\Omega} \left(\frac{\partial \mathbf{F}_V}{\partial x} + \frac{\partial \mathbf{G}_V}{\partial y} + \frac{\partial \mathbf{H}_V}{\partial z} \right) d\Omega = \frac{1}{\Omega} \int_{\partial\Omega} (\mathbf{F}_V n_x + \mathbf{G}_V n_y + \mathbf{H}_V n_z) dS \quad (4.6)$$

The flux vectors $\mathbf{F}_V$, $\mathbf{G}_V$, and $\mathbf{H}_V$ contain the gradients of the velocities that need to be calculated first at the edge centers. For this edge center calculation, another conceptual dual cell is used [18]. The two-dimensional edge-centered dual cell is shown in Figure 4.3. The edge-centered dual cell is composed of all the cells which share the edge. For an unstructured grid, the number of cells, m , which share an edge will vary.

Equation (4.7) shows the calculation for the $\frac{\partial u}{\partial x}$ term.

$$\frac{1}{\Omega_e} \int_{\Omega} \frac{\partial u}{\partial x} d\Omega = \frac{1}{\Omega_e} \int_{\partial\Omega} u n_x dS = \frac{1}{\Omega_e} \sum_{f=1}^m u S_x \quad (4.7)$$

This summation is over the faces of the edge-centered dual cell, which correspond to selected faces of the basic cells. Ω_e is the volume associated with the edge-centered dual cell, and S_x , S_y , and S_z are the area projections of the edge-centered dual cell face f in the corresponding coordinate directions. Velocities are required at the center of the edge-centered dual face f and their values are obtained by averaging the values at the four nodes of the quadrilateral face or three nodes of the triangular face.

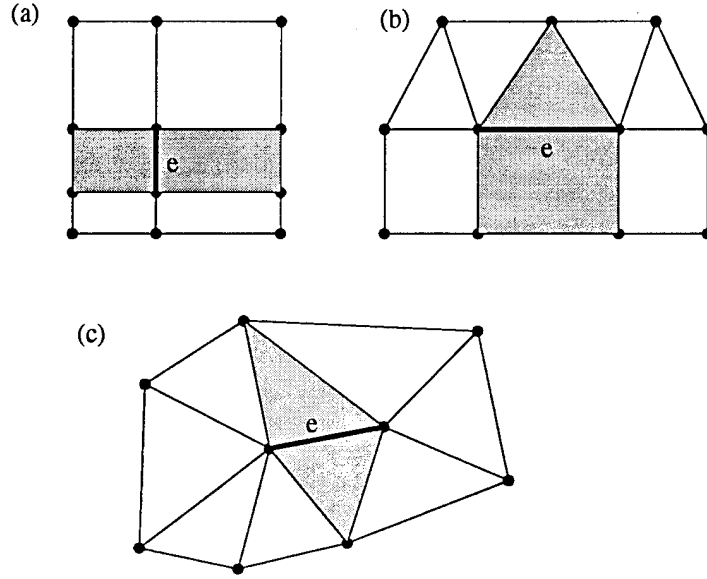


Figure 4.3: 2-D analogies for edge-centered dual mesh in the (a) Prismatic region (b) Tetrahedral-Prismatic interface and (c) Tetrahedral region.

For the hybrid grid, when inviscid tetrahedra are used, the summation is over the edge-centered dual cell faces in the prismatic region. When viscous tetrahedra are used, the summation is over the edge-centered dual cell faces in the entire domain.

The second derivatives (divergence of the velocity gradients) are calculated at the nodes. The right hand side of equation (4.6) is discretized over the node-centered dual cells and equation (4.8) is obtained. The integration is completed using the node-centered dual mesh and edge-based methodology described in the previous section on the inviscid terms. For the hybrid grid, when inviscid tetrahedra are used, the summation is over the prismatic edges. When viscous tetrahedra are used, the summation is over all the edges.

$$\frac{1}{\Omega_n} \sum_e (\mathbf{F}\mathbf{v}S_x + \mathbf{G}\mathbf{v}S_y + \mathbf{H}\mathbf{v}S_z)_e \quad (4.8)$$

4.3 Finite Element Discretization of Poisson Equation

Finite element discretization of the Poisson equation is used because the resulting coefficient matrix is reasonably diagonally dominant. Tetrahedral finite elements are chosen since these elements have attractive properties and the coefficient matrix is simpler to construct than any other element. The hybrid mesh is already composed of a majority of tetrahedral cells, and only requires conceptual division of the prism cells into tetrahedra cells for the formation of the Poisson equation matrix. This prismatic division procedure is described in Appendix A.

4.3.1 Isoparametric Transformation

The pressure correction equation (2.36) is discretized using the Galerkin finite-element approach [12]. The scheme is compact with all operations being restricted to within each grid cell. *Bilinear isoparametric* tetrahedral elements are employed [39, 148]. For tetrahedral cells, the values of ϕ and κ are defined in each element using the following finite-element formulation :

$$\phi = \sum_{i=1}^4 N_i \phi_i, \quad \kappa = \sum_{i=1}^4 N_i \kappa_i \quad (4.9)$$

where ϕ_i are nodal values of ϕ , κ_i are arbitrary constants of the test function κ , and N_i is the interpolating *shape* function associated with the i -th node in x,y,z space.

The x, y, and z coordinates can also be defined in terms of the shape functions:

$$x = \sum_{i=1}^4 N_i x_i, \quad y = \sum_{i=1}^4 N_i y_i, \quad z = \sum_{i=1}^4 N_i z_i \quad (4.10)$$

The master tetrahedral element is shown in Figure 4.4 and the bilinear shape functions for the master tetrahedral element in ξ, η, ζ space are defined as

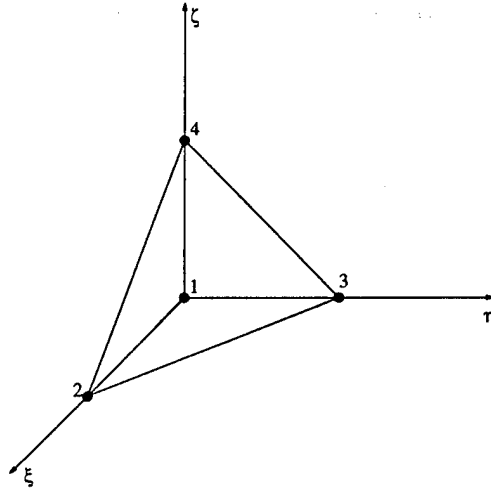


Figure 4.4: Master tetrahedral element in ξ, η, ζ space.

$$N_1 = 1 - \xi - \eta - \zeta \quad (4.11)$$

$$N_2 = \xi$$

$$N_3 = \eta$$

$$N_4 = \zeta$$

Relations are needed to relate the master shape functions in ξ, η, ζ space to the corresponding shape functions in x, y, z space. The transformation needed to supply these relations [148] is shown in equation (4.12).

$$\begin{bmatrix} \frac{dN}{d\xi} \\ \frac{dN}{d\eta} \\ \frac{dN}{d\zeta} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{bmatrix} \begin{bmatrix} \frac{dN}{dx} \\ \frac{dN}{dy} \\ \frac{dN}{dz} \end{bmatrix} \quad (4.12)$$

The matrix in equation (4.12) is the Jacobian of the transformation. The

derivatives of the shape functions in x, y , and z space can be found by inverting equation (4.12).

All the terms of the Jacobian matrix, can be calculated directly since the relationship between physical coordinates and shape functions are known from equations (4.10) through (4.11). Shape function derivatives can be found easily since the shape functions are defined in ξ, η, ζ space. The Jacobian matrix is a 3×3 matrix and can be easily inverted in the numerical code.

4.3.2 Galerkin Method

The Galerkin method is applied to the left hand side of equation (2.36) to obtain the matrix coefficients of the Poisson equation. Rewriting the equation, we have:

$$-\Delta\phi = f(x, y, z) \quad (4.13)$$

where $f(x, y, z)$ is the divergence of the velocity. Multiplying both sides by the test function κ , and integrating over the domain Ω we obtain:

$$\int_{\Omega} (-\kappa\Delta\phi - \kappa f) d\Omega = 0 \quad (4.14)$$

Applying the following vector identity to equation (4.14)

$$-\kappa\Delta\phi = -\nabla \cdot (\kappa\nabla\phi) + \nabla\phi \cdot \nabla\kappa \quad (4.15)$$

we obtain

$$\int_{\Omega} [\nabla\phi \cdot \nabla\kappa - \nabla \cdot (\kappa\nabla\phi) - \kappa f] d\Omega = 0 \quad (4.16)$$

Using the divergence theorem on the second term on the left hand side of equation (4.16), we obtain equation (4.17).

$$\int_{\Omega} -\nabla \cdot (\kappa\nabla\phi) d\Omega = - \int_{\partial\Omega} \frac{\partial\phi}{\partial n} \kappa dS \quad (4.17)$$

This term will disappear since the integral of $\frac{\partial \phi}{\partial n} \kappa$ over the surfaces of consistently defined tetrahedral cells is zero. Therefore the pressure correction equation becomes

$$\int_{\Omega} \nabla \phi \cdot \nabla \kappa d\Omega = \int_{\Omega} \kappa f d\Omega \quad (4.18)$$

The coefficients of ϕ , determined from the left hand side of the equation, become the entries in the Poisson equation matrix.

The element matrix system therefore becomes

$$D_{ij} \phi_j = f_j \quad (4.19)$$

where

$$D_{ij} = \iiint_{\Omega} \left(\frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} + \frac{\partial N_i}{\partial z} \frac{\partial N_j}{\partial z} \right) dx dy dz \quad (4.20)$$

and the derivatives in equation (4.20) are found after inverting the Jacobian in equation (4.12). The integrand is constant over each element and the integration can be completed by multiplying the factor in the parentheses with the volume of the tetrahedral cell.

Instead of the finite element technique, the right hand side vector, f_j , is calculated using the finite volume technique, similar to the edge-based method shown earlier for the inviscid momentum equation terms.

$$f_j = \sum_e (u S_x + v S_y + w S_z)_e \quad (4.21)$$

4.3.3 Element and Global Matrix Construction

We can construct the following global matrix system by assembling the element matrix obtained from equation (4.19)

$$\mathbf{D} \Phi = \mathbf{f} \quad (4.22)$$

where

$$\begin{aligned} \mathbf{D} &= \sum_e D_{ij} \\ \mathbf{f} &= \sum_e f_i \\ \Phi^T &= [\phi_1, \phi_2, \phi_3, \dots] \end{aligned} \tag{4.23}$$

Values for ϕ on the left hand side of equation (4.22) are treated implicitly, which requires the solution of a system of equations. The matrix that is formed after these steps is a symmetric linear matrix. This matrix is slightly modified to handle the implicit boundary conditions, and in general is no longer symmetric.

Solution of the pressure Poisson equation is the most computationally expensive portion of the incompressible flow calculation [81]. The fraction of computational time consumed for solving this equation can be as high as 80 % of the total CPU cost [134]. It is very important that the resulting system of equations be solved in an efficient manner. The use of direct solvers is unattractive because of the large storage requirements and computer effort. The matrix created by the Poisson equation is large and sparse. The full matrix is an n by n matrix, where n is the total number of nodes. The system is solved by the Gauss-Seidel method.

Use of iterative solvers creates the question of when the iteration procedure should be terminated. Often the convergence criterion is chosen as an arbitrarily small number. Performance of the entire algorithm depends on the convergence criteria for the Poisson equation. If the Poisson iteration is terminated before sufficient convergence is achieved, errors are propagated throughout the solution, possibly resulting in divergence or slow convergence. However, it is a waste of computer resources to choose convergence criteria which are too small.

Different values of the Poisson equation convergence criteria were used to

determine its effect on the solution. Increasing the Poisson convergence criteria significantly increased the number of iterations needed to reduce the divergence to an acceptable level. In the present work, when the absolute value of the difference in the value of ϕ between iterations is less than or equal to $5.0e-5$ for any node i , convergence is reached. This convergence criteria is

$$|\phi_i^{k+1} - \phi_i^k| \leq 5.0e-5 \quad (4.24)$$

where k denotes the iteration level.

In the solution of the Poisson equation, it has been found the lagging of boundary values is detrimental [103]. Boundary values must be updated in a manner similar to the interior points. Therefore, the present work uses the iterative solver to solve for all the grid points simultaneously.

4.4 Boundary Conditions for Unstructured Grids

Treatment of boundary conditions is crucial to any numerical simulation. With structured meshes, the boundary conditions can be treated easily using the i, j, k indices. In unstructured meshes, treatment of the boundary conditions are not as straightforward. Treatment of surface, far field, symmetry planes and side walls developed in the present work is presented.

4.4.1 Wall

Surface wall boundary conditions are applied to the three velocity components, as well as to the pressure corrections, ϕ . At the wall, the u , v and w components of velocity are set to zero. The value of $\partial\phi/\partial n$ is also set to zero.

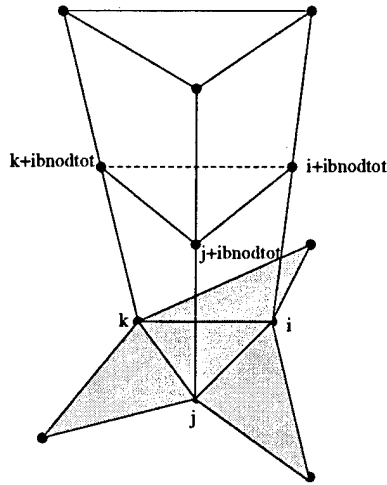


Figure 4.5: Obtaining the number of the node adjacent to the wall node

Since the wall boundary nodes are all located on triangular faces of prismatic cells, the boundary condition $\partial\phi/\partial n$ can be treated very simply. Prismatic layers are grown away from the wall surface in the normal direction so the neighbor node which is normal to a specific wall node is known immediately. For example, suppose the total number of nodes on the wall boundary is $ibndtot$. The node immediately off the wall node, j , in the normal direction is obtained by *adding* $ibndtot$ to the wall node number, as shown in Figure 4.5. The wall boundary is the shaded region. In the process of solving the Poisson equation and satisfying $\partial\phi/\partial n = 0$ at node j , we prescribe, $\phi(j) = \phi(j+ibndtot)$. This is completed for all the nodes on the wall boundary.

4.4.2 Far Field

The far field boundary conditions are applied to the three velocity components, as well as to the pressure corrections, ϕ . At the *far field*, the velocity components are set equal to the freestream values, while the value of $\partial\phi/\partial n$ is set to zero. The

treatment of the far field boundary conditions of the prismatic mesh are different than the treatment of boundary conditions of the hybrid mesh.

Prismatic Grid

For the all-prismatic grid, the far field boundary nodes are all located on triangular faces of the prismatic cells, and the boundary condition $\partial\phi/\partial n$ can be treated similarly to the treatment of the surface wall boundary nodes. In the process of solving the Poisson equation and satisfying $\partial\phi/\partial n = 0$ at far field node j , $\phi(j) = \phi(j\text{-ibnodtot})$ is prescribed. This is completed for all the nodes on the far field boundary.

Hybrid Grid

In the hybrid grid, the tetrahedra cells are grown away from the outer prismatic region to fill a hexahedral domain. The far field boundary lies along the $x = \pm x_{\max}$, $y = y_{\max}$, and $z = \pm z_{\max}$ planes. The far field boundary nodes in a hybrid grid are triangular faces of a tetrahedra. When a tetrahedral cell is on the far field boundary, the cell center value of ϕ , is extrapolated to the center of the triangular face on the boundary. This face center value is distributed to the face nodes. The nodal distribution uses weighting factors which are inversely proportional to the distance the face node is away from the face center. Therefore, the closer the face node is to the face center, the greater the distribution the face node will receive. This procedure is illustrated in Figure 4.6, where the far field boundary surface is the shaded region. In the Poisson equation calculation, this methodology is treated implicitly.

Care must be taken when defining the faces on the far field boundary of the

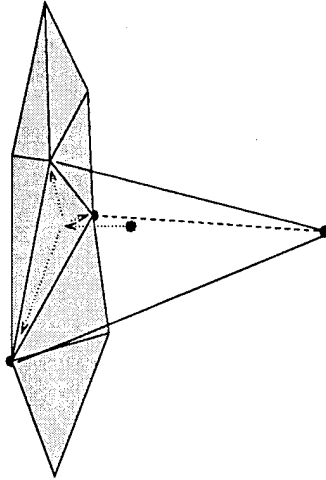


Figure 4.6: Illustration of far field boundary conditions for a hybrid mesh: assigning the cell center value to the tetrahedral face center, and distributing the face center value to the face nodes.

hybrid grid. A face is defined as being on the far field boundary if its three nodes all lie on the same far field boundary *plane*. A face can have all three nodes identified as a far field boundary node, yet the face defined by the three nodes may not lie on the boundary. This is easily seen for a tetrahedral cell which lies at the juncture of two far field planes of the computational domain, as shown in Figure 4.7. The shaded planes lie on the far field boundary.

4.4.3 Symmetry Planes

Prismatic Grid

Boundary nodes on a symmetry plane in the prismatic region are nodes on a quadrilateral face. Since the cells are prisms, there is no direct one-to-one correspondence of nodes normal to the nodes on the symmetry plane. The boundary conditions on a symmetry plane are treated in a methodology similar to the hybrid far field nodes.

Symmetry boundary calculations are performed in the following manner.

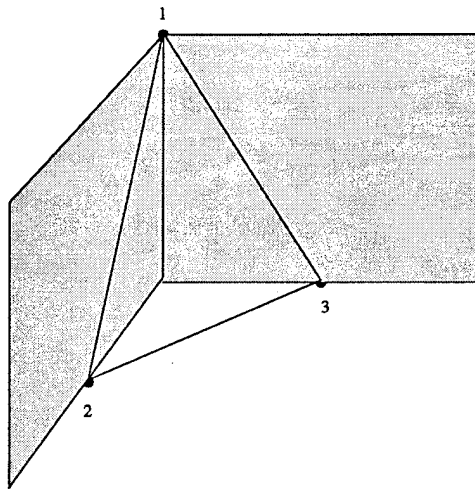


Figure 4.7: Illustration of a face defined by three far field nodes which does not lie on the far field boundary (shaded region).

The cell center value of the x , y , z -velocity or ϕ is determined first by averaging the values from the six cell nodes. The cell center value is extrapolated to the center of the face on the symmetry plane. The face center value is then distributed to the face nodes. The nodal distribution uses weighting factors which are inversely proportional to the distance the face node is away from the face center. Therefore, the closer the face node is to the face center, the greater the distribution the face node will receive. This procedure is illustrated in Figure 4.8. In the figure, the symmetry boundary surface is the shaded region.

In the case of the velocities at a symmetry plane, the velocity normal to the plane is set to zero. The values of the other velocity components are treated as described above. In the case of the pressure correction on a symmetry plane, the values of ϕ at the nodes are handled implicitly and the subsequent equation becomes part of the Poisson matrix.

Boundary conditions for the symmetry planes for the cylinder case were slightly different since the cylinder has a strictly two-dimensional flow. Instead of

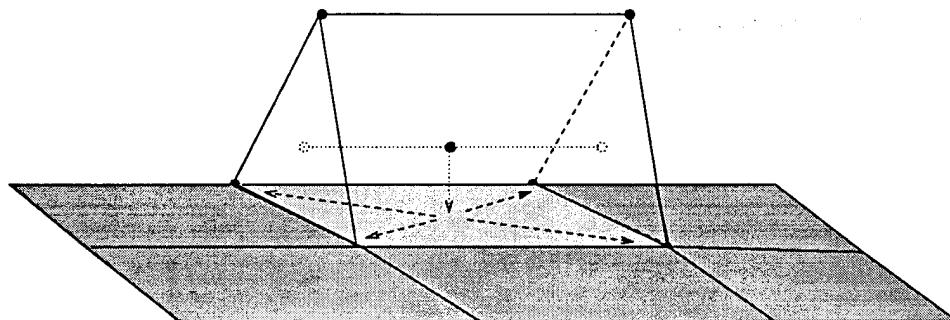


Figure 4.8: Illustration of symmetry boundary conditions: assigning the cell center value to the quadrilateral face center, and distributing the face center value to the face nodes.

extrapolating the ϕ cell center value to the face center, the ϕ value of the (third) node on the triangular face, not on the symmetry edge, is directly distributed to the two nodes on the edge, using a weighting factor inversely proportional to the distance the edge node is away from the projection of the third node on the edge. This reduces the transportation of ϕ values in the radial and circumferential direction. For the cylinder case, ϕ values should only be transferred in the axial (normal) direction for a true symmetry boundary condition.

Hybrid Grid

Symmetry boundary conditions for the tetrahedral portion of the hybrid grid are handled identically to the far field boundary conditions of the hybrid grid. The symmetry plane for the hybrid grid lies at $y=0.0$. When a tetrahedral cell is on the symmetry boundary, the cell center value of the velocity or ϕ , is extrapolated to the center of the triangular face on the boundary. This face center value is distributed to the face nodes. The nodal distribution uses weighting factors which are inversely proportional to the distance the face node is away from the face center. Therefore, the closer the face node is to the face center, the greater the distribution the face

node will receive. In the Poisson equation calculation, this methodology is treated implicitly.

4.5 Initial Conditions

Initial conditions for the flow around a cylinder and sphere were zero velocity at the surface and a linear distribution of velocity to the outer edge of the prisms where the velocity was set equal to the freestream value. For the hybrid mesh, the velocity in the tetrahedral region is initially the freestream value. The pressure ($\propto$ pressure coefficient) was assumed to be a Bernoulli equation type pressure distribution initially, $p = \frac{1}{2}u^2$. Initial conditions for ϕ were $p\delta t$, where δt was an initial representative value of 0.0001.

Initial conditions for the flow over a flat plate were zero pressure and freestream velocity everywhere except for no slip conditions at the wall and the specified velocity at the inlet. Initial conditions for the flow inside a driven cubic cavity were stagnant flow conditions except for the flow at the moving wall. ϕ was assumed to be zero initially.

4.6 Artificial Dissipation

Central space differencing schemes are susceptible to oscillatory modes in the velocity field of high Reynolds number flows. Furthermore, odd-even decoupling of the solution may appear in the pressure field for this non-staggered type of mesh that is employed. To stabilize the calculations for high Reynolds number flows, artificial dissipation is often used explicitly or implicitly [69]. For low Reynolds number flows, artificial dissipation may not be required since the viscous terms are relatively

large and can eliminate the error terms which have been introduced. In the present work, a fourth order smoothing term is added explicitly to the momentum equations in order to suppress odd-even decoupling of the solution [57].

The smoothing operator is cast in a form suitable for adaptive unstructured grids. It is formulated in such a manner as to simulate the implicit dissipation terms of the upwinding schemes without increasing the computational cost of the algorithm [104, 136]. Edge-based operations are used for the calculation of the artificial dissipation term. Consider an edge formed by the nodes L and R.

The equation for the second order contributions at node L is shown in equation:

$$(\delta \mathbf{U}_L)_2 = \sum_{e=1}^n (\mathbf{U}_R - \mathbf{U}_L) \quad (4.25)$$

where e denotes the edges sharing node L. The fourth order smoothing contribution is computed in a similar fashion. Instead of the first difference of state vectors as used in equation (4.25), a difference of the accumulated first difference over the edges sharing a node is used for the fourth order smoothing contribution. The fourth order difference is scaled by the time step at node L, Δt , and node volume, Ω_L . This scaling factor was determined by numerical experimentation and was the most effective of the scaling factors tried.

$$(\delta \mathbf{U}_L)_4 = -\sigma_4 \frac{\Delta t}{\Omega_L} \sum_{e=1}^n [(\delta \mathbf{U}_R)_2 - (\delta \mathbf{U}_L)_2] \quad (4.26)$$

The coefficient σ_4 is an empirical parameter that controls the amount of fourth order smoothing.

Large values of σ_{4u} , σ_{4v} , or σ_{4w} , stabilize the solution but destroy the accuracy. Therefore, special care is required for choosing the value of the smoothing coefficients. Numerical experiments have been carried out to determine optimum values

for the smoothing coefficients. The determined values are such that the solution accuracy is not affected, while the *odd-even* modes are suppressed [61]. Additional dissipation in the pressure Poisson equation was not used. Numerical experimentation showed the artificial dissipation in the momentum equation induced adequate smoothing in the pressure field.

4.7 Time Step Calculation

The stability limitation for the model 1-D convection equation $u_t + cu_y = 0$ is $\frac{c\Delta t}{\Delta y} \leq 1$ (CFL limitation), while the corresponding stability restriction for the 1-D model diffusion equation $u_t = \nu u_{yy}$ is $\frac{\nu\Delta t}{\Delta y^2} \leq \frac{1}{2}$.

In the present central space and forward time differenced scheme, a combination of the two limitations is employed. Specifically,

$$\Delta t = \frac{\omega \cdot vol_i}{|u_i S_{ix}| + |v_i S_{iy}| + |w_i S_{iz}| + \frac{2 \cdot vol_i}{Re(|S_{ix}| + |S_{iy}| + |S_{iz}|)}} \quad (4.27)$$

where vol_i is the dual volume corresponding to node i , u_i , v_i , and w_i are the velocities at node i , S_{ix} , S_{iy} , and S_{iz} are the projected areas of the dual faces for the dual cell at node i , and Re is the Reynolds number. Lastly, ω is the CFL (Courant, Friedrichs, and Lewy) factor [7], and equal to a number around 0.1. The small value for ω reflects the property of projection methods requiring a restrictive time step [13].

4.8 Solver Computer Requirements

The incompressible hybrid solver is described in terms of computing time and memory requirements. The CPU time consumed on a IBM 390 workstation is 0.0002

seconds/node/time iteration. Table 4.1 illustrates the percentage of CPU time which is consumed by the various parts of the solver. These percentages were obtained from averaging the CPU times consumed by the first 100 iterations of a simulation of flow over a sphere using a hybrid mesh. The pressure correction procedure is the most expensive in terms of CPU time, because it contains the solution of the Poisson equation matrix using the Gauss-Seidel method. The first ten time steps of the simulation requires many iterations of the Gauss-Seidel method, however, most of the simulation use one Gauss-Seidel iteration per time step. The percentage of time for the pressure correction method would decrease if data from more iterations were averaged. The percentage of CPU time which is consumed by the pressure correction part of the solver is further examined in Table 4.2. The viscous subroutine is the next most expensive subroutine. For this hybrid mesh, equation adaptation was used. There are 19740 nodes in the prismatic region which are involved in the Navier-Stokes calculation. There are another 10418 nodes solely in the tetrahedral region which are involved in the Euler calculation and not involved in the viscous calculation. If the Navier-Stokes equations were solved over the entire domain, the amount of time consumed by the viscous calculations would increase by 30 %, thereby increasing its percentage of CPU time consumed. It is clear from examination of CPU time, there is a definite advantage in utilizing equation adaptation.

The solver is written in a form which permits vectorization on vector parallel machines. Most of the computationally significant subroutines can be vectorized with the exception of the Gauss-Seidel subroutine, resulting in significant reduction in CPU time. It is possible to force the vectorization of the Gauss-Seidel subroutine, but the resulting matrix solver would probably exhibit convergence properties

Table 4.1: Proportions of computing time consumed by parts of the solver.

Subroutine	CPU Time Proportion (%)
Convective Terms	7.0
Viscous Terms	33.6
Smoothing	2.5
Boundary Conditions	0.5
Time Step Calculation	0.1
Pressure Correction	56.3

between Jacobi and Gauss-Seidel techniques. The savings in CPU time through vectorizing may offset the additional iterations needed from the resulting mixed Jacobi-Gauss-Seidel solver.

Memory requirements for the solver are shown in Table 4.3. The variables requiring the most memory are those associated with the Poisson matrix and vectors. The bytes shown for the Poisson matrix are representative numbers for a prismatic mesh where nodes have a maximum of 20 neighboring nodes. This is a generous number for a prismatic mesh. However, a hybrid mesh may result in four times as many neighbors, thus substantially increasing the amount of storage needed for the Poisson equation. The majority of Poisson equation variables are also double

Table 4.2: Proportions of computing time consumed by pressure correction parts of the solver.

Subroutine	CPU Time Proportion (%)
Divergence of Velocity	4.7
Gauss-Seidel	88.3
Gradient of Phi	5.7
Boundary Conditions	0.8

precision, requiring twice as many bytes compared to the other storage arrays. A smaller, but still significant, memory storage “sink” are the pointers and metrics needed for the tetrahedra cells if the Navier-Stokes equations are solved in this region. Again, the advantage to using equation adaptation is clear, even when based on memory requirements alone.

Table 4.3: Memory requirements of the solver.

Item	Bytes
State Vectors: $\mathbf{U}$ and $\delta\mathbf{U}$, per node	40
Pressure and Time Step, per node	8
Poisson Matrix and Vectors, per node	272
Node Metrics, per node	28
Edge Metrics, per edge	24
Boundary Face Metrics, per boundary face	112
Boundary Edge Metrics, per boundary edge	32
Viscous Tetrahedra, per tetrahedral face	52

Chapter 5

Numerical Issues

In the course of the development of the present numerical method, several numerical issues arose which had significant impact on the direction of the research. These issues include the pressure correction formulation, diagonal dominance of the resulting pressure Poisson equation, boundary conditions, non-staggered versus staggered grids, discrete divergence, and testing subroutines with analytic field functions.

5.1 Pressure Correction Formulations

In the evolution of the present method's development I, as well as others [109], have found that the particular formulation of the Poisson pressure equation has a tremendous impact on the quality of the solution.

The pressure correction methods tested in the present work are variants of Chorin's original projection method which was based on the Hodge projection [30]. Projection methods are a type of fractional step method [14, 30, 145]. These methods obtain an auxiliary velocity vector field by disregarding the solenoidal nature of the velocity, and then project this auxiliary vector field onto the space of

divergence free fields to obtain the divergence free velocity [13, 30]. The numerical form of the projection method computes an auxiliary vector field and applies some type of discrete projection. The projection recovers the divergence free velocity by splitting the velocity field into two parts that are divergence free and curl free [108]. The curl free part is written as the gradient of a potential. The auxiliary vector field is a linear combination of the velocity at the new time level and the gradient of the potential.

$$\vec{u}' = \vec{u}^{(n+1)} + \nabla\phi$$

The first method tried was the Simplified Marker and Cell (SMAC) [4, 5, 71] method, a modification of the Marker and Cell (MAC) [50] method. The SMAC solution procedure follows the explicit/implicit marching scheme in [71, 91]. The overall solution procedure is very similar to the method chosen in the present work except in the choice of the auxiliary velocity vector. The current auxiliary velocity vector was defined in equation (2.32) and is repeated here for clarity.

$$\frac{\partial \mathbf{U}'}{\partial t} + \frac{\partial \mathbf{F}_I}{\partial x} + \frac{\partial \mathbf{G}_I}{\partial y} + \frac{\partial \mathbf{H}_I}{\partial z} = \frac{\partial \mathbf{F}_V}{\partial x} + \frac{\partial \mathbf{G}_V}{\partial y} + \frac{\partial \mathbf{H}_V}{\partial z} \quad (5.1)$$

In the SMAC method, the auxiliary velocity vector equation contains the pressure at the n time level, equation (5.2).

$$\frac{\partial \mathbf{U}'}{\partial t} + \frac{\partial \mathbf{F}_I}{\partial x} + \frac{\partial \mathbf{G}_I}{\partial y} + \frac{\partial \mathbf{H}_I}{\partial z} = \frac{\partial \mathbf{F}_V}{\partial x} + \frac{\partial \mathbf{G}_V}{\partial y} + \frac{\partial \mathbf{H}_V}{\partial z} + \mathbf{S} \quad (5.2)$$

Similarly, the auxiliary velocity state vector can be obtained directly since it contains terms only at the known n time level, and it does not satisfy the continuity equation. Subtracting the auxiliary velocity vector equation (5.2) from the momentum equation (2.13) through (2.31), and rewriting in primitive variables, it

is obtained :

$$\vec{u}^{(n+1)} - \vec{u}' = -\nabla [p^{(n+1)} - p^{(n)}] \Delta t \quad (5.3)$$

Introducing a similar scalar potential ϕ , such that

$$\vec{u}^{(n+1)} - \vec{u}' = -\nabla \phi, \quad (5.4)$$

the following equation for pressure can be obtained:

$$p^{(n+1)} - p^{(n)} = \frac{1}{\Delta t} \phi \quad (5.5)$$

Finally, taking the divergence of each side of equation (5.4) and considering the continuity equation (2.24) through (2.26), the following pressure correction Poisson equation is obtained for the SMAC formulation.

$$\Delta \phi = \nabla \cdot \vec{u}' \quad (5.6)$$

Using the ϕ values obtained by equation (5.6), we can correct the velocity and pressure fields using equations (5.4) and (5.5), derived from the SMAC method, as follows:

$$\vec{u}^{(n+1)} = \vec{u}' - \nabla \phi \quad (5.7)$$

$$p^{(n+1)} = p^{(n)} + \frac{1}{\Delta t} \phi \quad (5.8)$$

The major difference between the present method and the SMAC method is the pressure correction equation, equations (2.38) and (5.8). Equation (2.38), repeated here for convenience,

$$p^{(n+1)} = \frac{1}{\Delta t} \phi \quad (5.9)$$

obtains the corrected pressure field from ϕ , while equation (5.8) yields the corrected pressure field from the previous pressure field and the correction. The SMAC

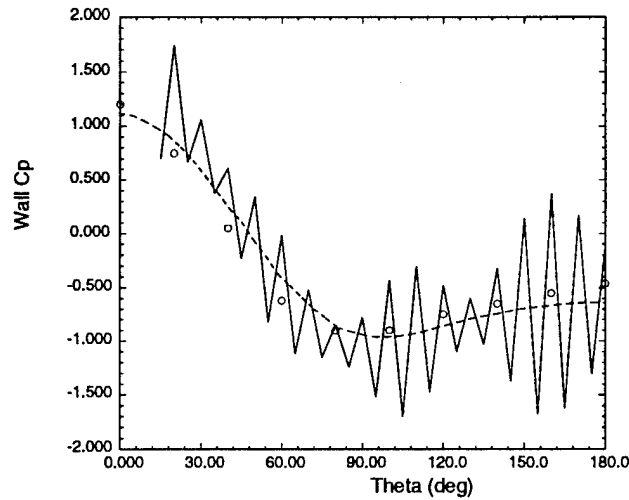


Figure 5.1: Wall pressure coefficient on cylinder for $Re=40$.

- o Experiment by Grove [46]
- SMAC method
- - - Present work

method accumulates the pressure changes. As the iterative accumulation took place using the SMAC method, an oscillatory pressure field developed. The wall pressure coefficient along the circumferential direction of a cylinder at $Re = 40$ is shown in Figure 5.1. On the average, the pressure field was correct, but the oscillations were very severe. The oscillations were a consequence of the use of a non-staggered grid. At this point, the present pressure correction formulation was chosen. Using a similar pressure correction formulation, Rider [108] found omitting the pressure term in the auxiliary velocity equation produces a formulation which has better error characteristics, without a loss of accuracy, compared to a pressure correction method which accumulates the pressure corrections.

5.2 Diagonal Dominance

The form of the matrix corresponding to the resulting discrete system following discretization of the Poisson equation is critical to convergence of iterative methods employed to invert it. Diagonal dominance is the key property of the matrix related to convergence of an iterative method. Matrix A of order n is (strictly row) diagonal dominant [35] if

$$|a_{ii}| > \sum_{j \neq i} |a_{ij}|, i = 1, \dots, n \quad (5.10)$$

Diagonal dominance of the matrix is dependent upon the procedure used to generate the matrix. To examine this property, we used a finite difference, finite element, and finite volume technique to generate the Poisson matrix. To enable the use of the finite difference technique, a structured prismatic grid was used in the analysis, similar to the grid shown in Figure 6.5. The Gauss-Seidel method was used to solve the system of equations for all three cases. The Poisson equation convergence criteria was also identical for all three cases. The quasi-two-dimensional flow around a cylinder at a Reynolds number of 40 was the chosen test case.

An indication of the level of diagonal dominance of the matrix can be seen by examining the ratio

$$R_i = \frac{|a_{ii}|}{\sum_{j \neq i} |a_{ij}|} \quad (5.11)$$

If each row i has a diagonal dominance ratio greater than 1.0, the matrix is diagonally dominant. The diagonal dominance ratio of the resulting matrix for a specific row is examined in Table 5.1 for the finite difference, finite element, and the finite volume technique. The finite difference method matrix had a ratio of 1.0, which is (marginally) diagonally dominant. The finite element method produced a matrix

Table 5.1: Indicators of the level of diagonal dominance for Poisson matrix formulations. A ratio greater than 1.0 signifies diagonal dominance.

	Finite Difference	Finite Element	Finite Volume
Diagonal Dominance Ratio, R_i	1.0	0.91	0.22

with a ratio of 0.91, a slightly non-diagonally dominant matrix. The finite volume discretization produced a matrix with a ratio of 0.22, a strongly non-diagonally dominant matrix.

A comparison of the number of Gauss-Seidel iterations required for the convergence for the three cases is shown in Figure 5.2. For the finite difference (diagonally dominant) matrix, the number of Gauss-Seidel iterations at each time step was initially high, but quickly decreased to one Gauss-Seidel iteration per time step. The finite element (slightly non-diagonally dominant) matrix had a similar convergence behavior. For the finite volume (strongly non-diagonally dominant) matrix, convergence was not obtained. The momentum residuals were also compared and are shown in Figure 5.3. The solution using the finite volume method (strongly non-diagonally dominant) never converged. The convergence behavior of the finite difference and finite element methods are very similar.

For semi-unstructured and unstructured meshes, the finite difference technique is unsuitable because it cannot handle the arbitrariness of node placement. The loss of diagonal dominance makes the standard finite volume technique for the Poisson equation inadequate. For this reason, a finite element technique was used

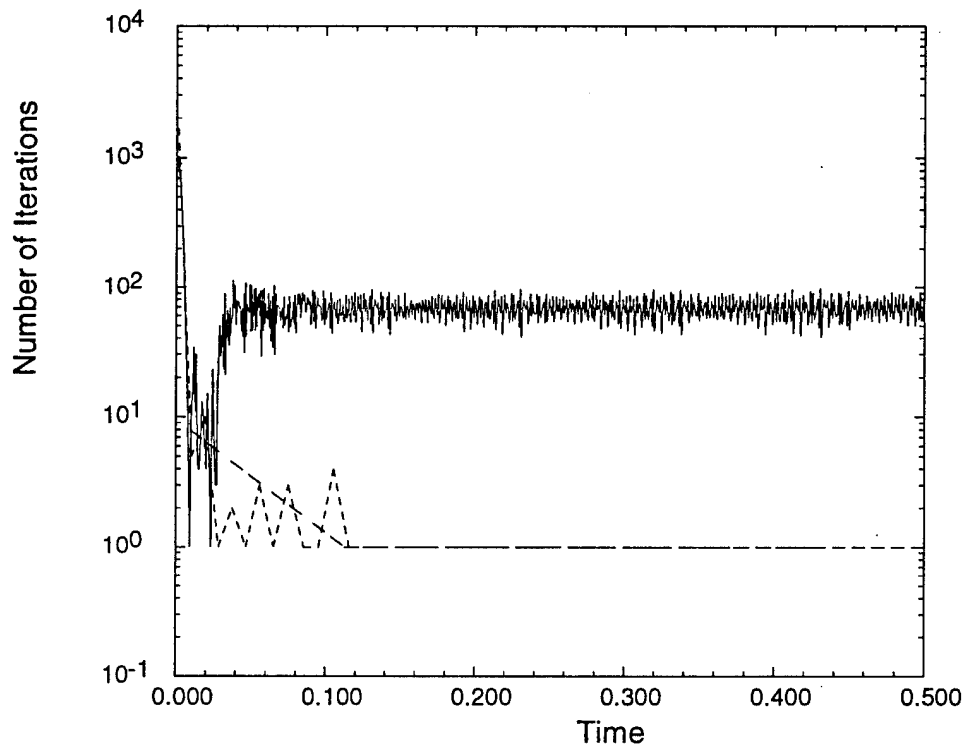


Figure 5.2: Comparison of number of Gauss-Seidel iterations needed for convergence of pressure Poisson equation.

- - - Finite difference (diagonally dominant) matrix
- - Finite element (slightly non-diagonally dominant) matrix
- Finite volume (strongly non-diagonally dominant) matrix

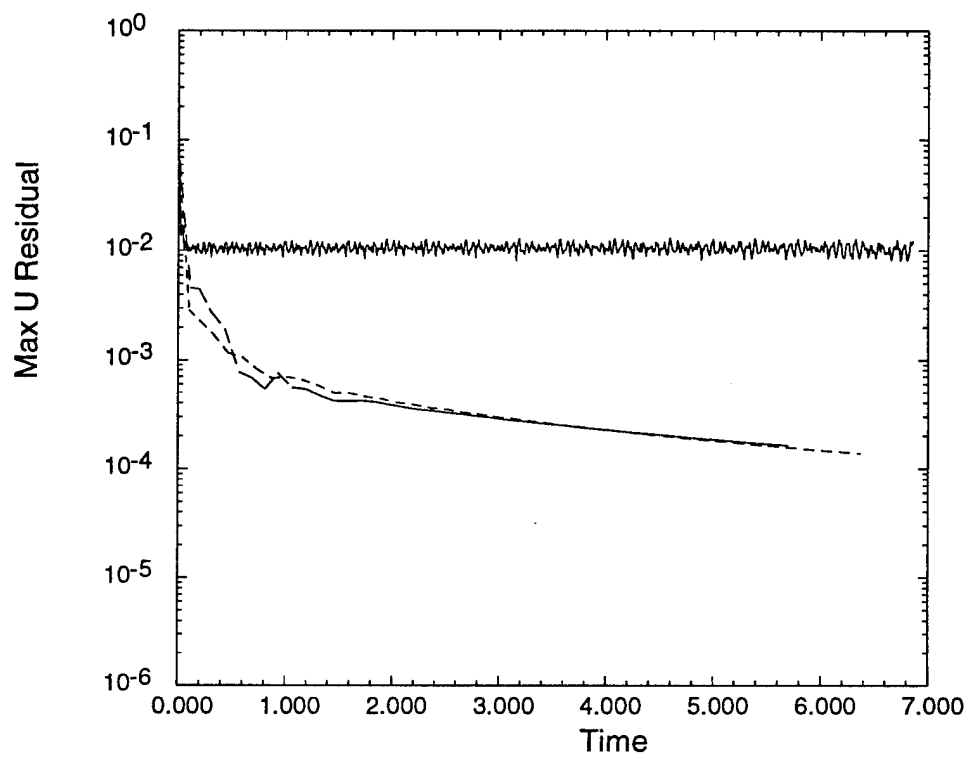


Figure 5.3: Comparison of maximum momentum residual behavior.

- - - Finite difference (diagonally dominant) matrix
- - Finite element (slightly non-diagonally dominant) matrix
- Finite volume (strongly non-diagonally dominant) matrix

for solution of the Poisson equation.

This difficulty with the formation of non-diagonally dominant matrices is usually not an issue for staggered grids used with the finite volume technique. This is related to the number of neighboring nodes needed to compute the Poisson equation for each node. When the finite volume technique is applied to a node in a non-staggered hexahedral mesh, the pressure values at all the neighboring *nodes* are required. The resulting Poisson matrix will have 26 off-diagonal entries in each row of the matrix, and is usually not row diagonally dominant. When the finite volume technique is applied to node in a pressure-staggered (velocity is defined at the nodes, pressure is defined at the cell centers) hexahedral mesh, the pressure values at the neighboring *cell centers* is required. The resulting Poisson matrix will have only 8 off-diagonal entries in each row of the matrix. The matrix resulting from a hexahedral staggered mesh is diagonally dominant [70].

5.3 Boundary Conditions for the Poisson Equation

The pressure correction equation is a Poisson equation and can be written in the following form.

$$\Delta\phi = -f \quad (5.12)$$

The value of the function ϕ on a boundary, can be specified in two different ways. The first type of boundary condition specifies the value of the function ϕ on the boundary surface (Dirichlet boundary conditions):

$$\phi = g(s) \quad (5.13)$$

The second type of boundary condition specifies the value of the normal derivative of ϕ on the boundary surface (Neumann boundary conditions):

$$\frac{\partial\phi}{\partial n} = g(s) \quad (5.14)$$

Dirichlet boundary conditions with the Poisson equation have been used sparingly in the literature. Improper definition of the boundary condition may create local mass sources and create sinks to appear [45]. Most of the methods published in the literature use Neumann boundary conditions for the Poisson equation. The choice of using Dirichlet or Neumann boundary conditions has been discussed in the literature [1, 3, 43].

In the present work, severe pressure oscillations resulted when ϕ was specified. Moving the far field boundary further away from the body did not result in a decrease in the pressure oscillations. Since severe oscillations are often an indication of unsuitable boundary conditions [44], Neumann boundary conditions were used.

For the Neumann problem, g may not be chosen independently of f , but rather as a consequence of Green's first identity. The identity states f and g must

satisfy a compatibility condition,

$$\int_{\Omega} f d\Omega = \int_{d\Omega} g(s) ds \quad (5.15)$$

if a solution is to exist [3, 22, 42]. A property of the Neumann problem is the non-uniqueness of the solution. Although the shape of the solution is unique, the location of this solution is not fixed in space, unlike the Dirichlet problem. There are an infinite number of solutions each differing from the others by an arbitrary additive constant.

One method developed to eliminate the floating of the solution is to choose one point in the flow field and assign to it a specific value, essentially forcing one point to have Dirichlet boundary conditions [134, 135]. The choice of selecting an interior point has also been studied [56]. However, the selection of a Dirichlet point may cause the formation of a singularity at that point. Failure to recompute the value at this point results in the partial differential equation not being satisfied at this point. Thus the link with any solution is lost at this point [38]. Accumulation of nearly all of the round-off error may also occur at this point, sending perturbations throughout the entire flow field [56].

A second method developed to eliminate the floating of the solution also selects one point in the flow field. However, the solution is computed for all points, even the selected point. The resultant approximation to the solution is shifted by subtracting the value of the solution at the selected point from all the points in the flow field [38]. The use of this method results in successive iterations approximating the same solution.

A third method developed to eliminate the floating of the solution uses the compatibility condition, equation (5.15), as an integral constraint on the pressure [22, 69, 103]. This method produces acceptable results, but is computationally more

expensive than the second method.

In the present work, the second method is used to eliminate the floating of the ϕ values, and subsequently the pressure values. One point in the far field is chosen, and the resulting pressure value is subtracted from the entire pressure field each time the pressure is corrected.

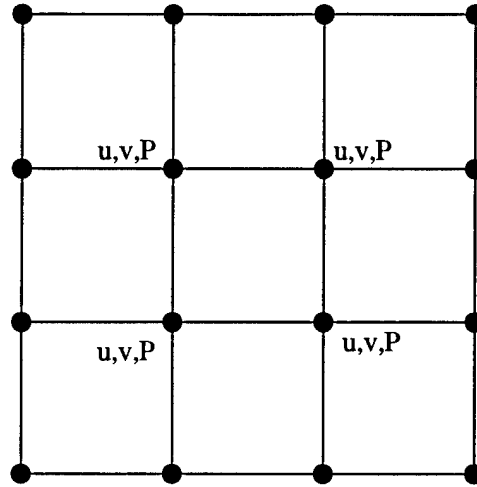


Figure 5.4: Placement of primitive variables on a non-staggered grid.

5.4 Non-staggered Grid versus Staggered Grid

On a non-staggered grid, values of all the primitive variables are defined at each node. Figure 5.4 illustrates a non-staggered grid in two-dimensions. Figure 5.5 illustrates a one-dimensional uniform domain used to calculate the pressure gradient at node i using the finite volume approach on a non-staggered grid. The pressure gradient at node i is found from the difference of the pressure value at faces e and w . Since the pressure is defined at the nodes only (non-staggered grid) the value of the pressure at each face is found by linearly interpolating the values of the pressure at the two adjacent nodes as shown in equation (5.16). However, the equation simplifies and the resulting pressure gradient term will contain the pressure difference between two alternate grid points, and not between adjacent ones.

$$\frac{\partial P_i}{\partial x} = \frac{P_e - P_w}{\Delta x} = \frac{\left(\frac{P_{i+1} + P_i}{2}\right) - \left(\frac{P_i + P_{i-1}}{2}\right)}{\Delta x} = \frac{P_{i+1} - P_{i-1}}{2\Delta x} \quad (5.16)$$

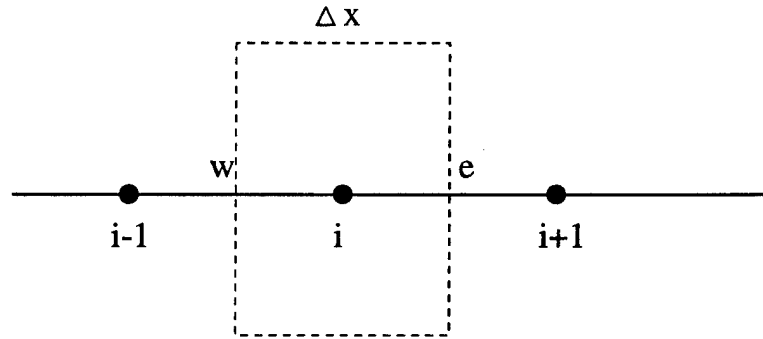


Figure 5.5: One-dimensional domain for pressure gradient calculation on a non-staggered grid.

Comparison of the smooth and odd-even pressure fields on the one-dimensional domain are shown in Figures 5.6 and 5.7. The resulting pressure gradient calculation from equation (5.16) produces the same result for both pressure fields. A consequence of central differencing on a non-staggered grid is that a wavy pressure field will be “felt” like a uniform pressure field by the momentum equation. If oscillatory pressure fields arise during the iterative pressure correction procedure, the oscillations would be preserved until convergence since the momentum equations would be oblivious to their presence.

This difficulty can be eliminated with the use of staggered grids [100]. On a staggered grid, the values of all the primitive variables are defined at different nodes. Figure 5.8 illustrates one example of a staggered grid. The staggering concept may be different than shown in Figure 5.8. The pressure staggered grid [70] defines the velocities at cell nodes and the pressure is defined at the cell center. The arbitrary Lagrangian-Eulerian (ALE) procedure [27, 54, 55] defines the pressure at the nodes and velocity components at the cell center.

In the past, most incompressible flow solvers used staggered grids [47, 48, 52, 67, 70, 72, 123]. A one-dimensional staggered grid analogy to Figure 5.5 is

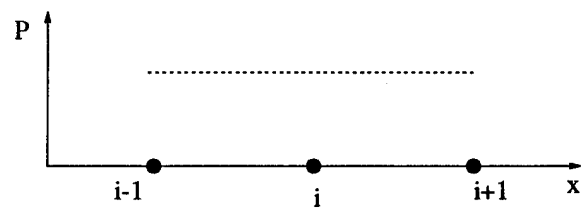


Figure 5.6: Uniform pressure field in a one-dimensional domain.

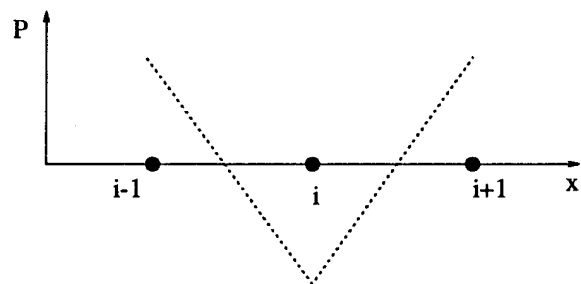


Figure 5.7: Odd-even pressure field in a one-dimensional domain.

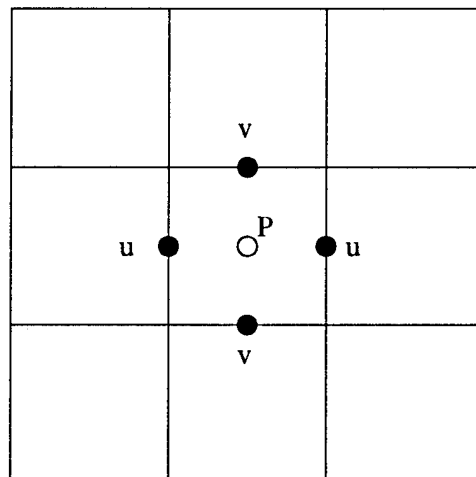


Figure 5.8: Placement of primitive variables on a staggered grid.

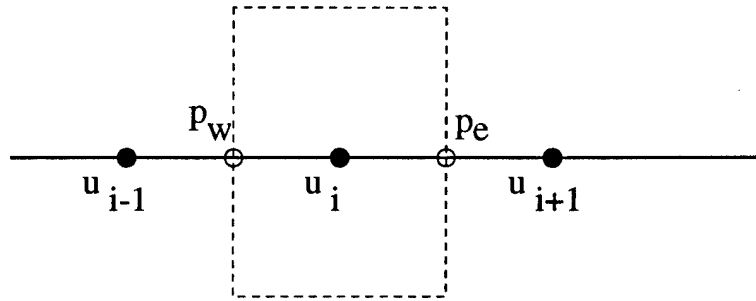


Figure 5.9: One-dimensional domain for pressure gradient calculation on a staggered grid.

illustrated in Figure 5.9. The figure illustrates a one-dimensional uniform domain used to calculate the pressure gradient at node i using the finite volume approach on a staggered grid. The advantage of the staggered grid is that the pressure gradient at node i is calculated from the pressure at the faces e and w , which are known. Therefore, the pressure gradient at node i is calculated from the two adjacent grid points, and will not permit oscillatory pressure fields. A disadvantage of the use of staggered grids is that each staggered variable requires its own control volume. For the staggered grid shown in Figure 5.8, three control volumes would be required. Two control volumes for the u and v velocities and one control volume for the pressure would necessitate an increase in memory storage.

The non-staggered grid requires only one control volume for all the variables. This is an advantage in terms of memory storage and coding complexity. As the popularity of unstructured grids rises, the use of non-staggered grids [10, 49, 106, 122, 126, 130] has become more prevalent. The present work uses hybrid (prismatic/tetrahedral) meshes which led to the choice of non-staggered grid. Care must be taken to ensure an oscillatory pressure field does not develop during the iterative pressure correction process.

A comparison of solutions and convergence properties of a finite volume nu-

merical method with staggered and non-staggered grids [102] concluded the non-staggered version had no disadvantages in *performance* relative to the staggered version. In fact, treatment of boundary conditions and the implementation of higher order differencing schemes was simpler for the non-staggered version. In some cases, the non-staggered version displayed faster convergence. Convergence acceleration techniques, like the multigrid technique, are also easier to apply to the non-staggered grid arrangement.

A disadvantage to non-staggered methods not mentioned by [102], but found in all non-staggered methods [10, 102, 106, 124] is a non-zero value of the divergence of the velocity. Further explanation of this phenomenon is included in the subsection on Discrete Divergence.

5.5 Discrete Divergence

A potentially serious consequence to numerically modeling a flow field is errors introduced to the solution due to the discretization of the governing equations. For the present work, the momentum and pressure Poisson equations are the governing equations of interest. These equations can be discretized independently of one another or the discretization of one equation can be developed using the other equation in a consistent formulation. Using a methodology similar to [124], the discretization procedure of the present pressure correction formulation for one dimension in finite difference form is explained. The 1-D momentum equation is considered.

$$\frac{\partial u}{\partial t} = -\frac{\partial P}{\partial x} - \xi \quad (5.17)$$

where

$$\xi = u \frac{\partial u}{\partial x} - \frac{1}{Re} \frac{\partial^2 u}{\partial x^2} \quad (5.18)$$

The 1-D auxiliary velocity vector is

$$\frac{\partial u'}{\partial t} = -\xi \quad (5.19)$$

5.5.1 Consistency of Discretized Poisson Equation

The discrete Poisson equation will be derived using two methods, to determine the consistency of the formulation. The methods are:

1. The Poisson equation is derived using the momentum and continuity equations. The final form of the Poisson equation is then discretized.
2. The discrete Poisson equation is derived using the discrete momentum and the discrete continuity equations.

The resulting discrete Poisson equations will be compared for consistency using both staggered and non-staggered grids. Central differencing will be used in the discretizations.

Staggered Grid

In the first method, the Poisson equation is derived using the momentum and continuity equations. The Poisson equation in one dimension, which was derived previously and originally defined in equation (2.36) is repeated here for clarity

$$\Delta\phi = \nabla \cdot \vec{u}' \quad (5.20)$$

and rewritten in terms of the pressure variable.

$$\Delta P = \frac{\nabla \cdot \vec{u}'}{\Delta t} \quad (5.21)$$

Using a staggered grid where the pressure is defined at the nodes $(i-1, i, i+1)$ and the velocity is staggered between the nodes $(i+1/2 \text{ and } i-1/2)$, the discrete auxiliary vector at locations $i+1/2$ and $i-1/2$ become:

$$u'_{i+1/2}^{n+1} = u_{i+1/2}^n - \Delta t \xi_{i+1/2}^n \quad (5.22)$$

$$u'_{i-1/2}^{n+1} = u_{i-1/2}^n - \Delta t \xi_{i-1/2}^n \quad (5.23)$$

Substitution of equations (5.22) and (5.23) into equation (5.21), produces the discrete Poisson equation.

$$\left(\frac{P_{i+1}^{n+1} - 2P_i^{n+1} + P_{i-1}^{n+1}}{(\Delta x)^2} \right) + \left(\frac{\xi_{i+1/2}^n - \xi_{i-1/2}^n}{\Delta x} \right) = \frac{1}{\Delta t} \left(\frac{u_{i+1/2}^n - u_{i-1/2}^n}{\Delta x} \right) \quad (5.24)$$

Applying the second method on a staggered grid, the discrete Poisson equation is derived using the discrete continuity and discrete momentum equations. The

discrete momentum equation at locations $i+1/2$ and $i-1/2$ are:

$$u_{i+1/2}^{n+1} = u_{i+1/2}^n - \Delta t \left(\frac{P_{i+1}^{n+1} - P_i^{n+1}}{\Delta x} \right) - \Delta t \xi_{i+1/2}^n \quad (5.25)$$

$$u_{i-1/2}^{n+1} = u_{i-1/2}^n - \Delta t \left(\frac{P_i^{n+1} - P_{i-1}^{n+1}}{\Delta x} \right) - \Delta t \xi_{i-1/2}^n \quad (5.26)$$

The discretized continuity equation is:

$$\frac{u_{i+1/2}^{n+1} - u_{i-1/2}^{n+1}}{\Delta x} = 0 \quad (5.27)$$

Substituting the discrete momentum equations (5.25) and (5.26) into the discrete continuity equation (5.27), it is obtained:

$$\left(\frac{P_{i+1}^{n+1} - 2P_i^{n+1} + P_{i-1}^{n+1}}{(\Delta x)^2} \right) + \left(\frac{\xi_{i+1/2}^n - \xi_{i-1/2}^n}{\Delta x} \right) = \frac{1}{\Delta t} \left(\frac{u_{i+1/2}^n - u_{i-1/2}^n}{\Delta x} \right) \quad (5.28)$$

Comparisons of equations (5.24) and (5.28), show the resulting discrete Poisson equation obtained by both methods are identical on a staggered grid. The staggered grid formulation is consistent.

Non-staggered Grid

In a non-staggered grid, the velocity and pressure are defined at the nodes $(i-1, i, i+1)$. Using the first method to calculate the discrete Poisson equation requires the discrete auxiliary vector at nodes $i+1$ and $i-1$:

$$u_{i+1}^{n+1} = u_{i+1}^n - \Delta t \xi_{i+1}^n \quad (5.29)$$

$$u_{i-1}^{n+1} = u_{i-1}^n - \Delta t \xi_{i-1}^n \quad (5.30)$$

Substitution of equations (5.29) and (5.30) into the Poisson equation (5.21) produces the discrete Poisson equation.

$$\left(\frac{P_{i+1}^{n+1} - 2P_i^{n+1} + P_{i-1}^{n+1}}{(\Delta x)^2} \right) + \left(\frac{\xi_{i+1}^n - \xi_{i-1}^n}{2\Delta x} \right) = \frac{1}{\Delta t} \left(\frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x} \right) \quad (5.31)$$

Applying the second method on a non-staggered grid, the discrete Poisson equation is derived using the discrete continuity and discrete momentum equations.

The discrete momentum equations at nodes $i+1$ and $i-1$ are:

$$u_{i+1}^{n+1} = u_{i+1}^n - \Delta t \left(\frac{P_{i+2}^{n+1} - P_i^{n+1}}{2\Delta x} \right) - \Delta t \xi_{i+1}^n \quad (5.32)$$

$$u_{i-1}^{n+1} = u_{i-1}^n - \Delta t \left(\frac{P_i^{n+1} - P_{i-2}^{n+1}}{2\Delta x} \right) - \Delta t \xi_{i-1}^n \quad (5.33)$$

The discretized continuity equation is:

$$\frac{u_{i+1}^{n+1} - u_{i-1}^{n+1}}{2\Delta x} = 0 \quad (5.34)$$

Substituting the discrete momentum equations (5.32) and (5.33) into the discrete continuity (5.34), it is obtained:

$$\left(\frac{P_{i+2}^{n+1} - 2P_i^{n+1} + P_{i-2}^{n+1}}{(2\Delta x)^2} \right) + \left(\frac{\xi_{i+1}^n - \xi_{i-1}^n}{2\Delta x} \right) = \frac{1}{\Delta t} \left(\frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x} \right) \quad (5.35)$$

Comparisons of equations (5.31) and (5.35), show the resulting discrete Poisson equations obtained by the two methods are not identical on a non-staggered grid. The non-staggered grid formulation is not consistent.

5.5.2 Error of Discrete Continuity

The discrete Poisson equation, (5.28), using staggered grids is obtained using a consistent formulation. It is observed that the right hand side of the equation is the discrete continuity equation, while the left hand side is the difference of the residual of the momentum equation at two different nodes. At steady state, the residual of the momentum equation at all nodes is zero. If the left hand side of the equation is driven to zero, so will the right hand side. Therefore, for a staggered grid, the discrete continuity equation will approach machine zero.

The discrete Poisson equations, (5.31) and (5.35), using non-staggered grids are not consistent. The difference between equations (5.31) and (5.35) is the error in the discrete continuity. To examine this error, the pressure term in equation (5.35) will be subtracted from and added to the left hand side of equation (5.31).

$$\left(\frac{P_{i+1}^{n+1} - 2P_i^{n+1} + P_{i-1}^{n+1}}{(\Delta x)^2} \right) - \left(\frac{P_{i+2}^{n+1} - 2P_i^{n+1} + P_{i-2}^{n+1}}{(2\Delta x)^2} \right) +$$

$$\left(\frac{P_{i+2}^{n+1} - 2P_i^{n+1} + P_{i-2}^{n+1}}{(2\Delta x)^2} \right) + \left(\frac{\xi_{i+1}^n - \xi_{i-1}^n}{2\Delta x} \right) = \frac{1}{\Delta t} \left(\frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x} \right) \quad (5.36)$$

The third and fourth terms on the left hand side of the equation are the difference of the residuals of the momentum equation at two different nodes on a non-staggered grid. At steady state, those residuals will be zero. Removing the third and fourth terms from the equation produces

$$\left(\frac{P_{i+1}^{n+1} - 2P_i^{n+1} + P_{i-1}^{n+1}}{(\Delta x)^2} \right) - \left(\frac{P_{i+2}^{n+1} - 2P_i^{n+1} + P_{i-2}^{n+1}}{(2\Delta x)^2} \right) = \frac{1}{\Delta t} \left(\frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x} \right) \quad (5.37)$$

when rearranged becomes

$$\Delta t \left(\frac{-P_{i+2}^{n+1} + 4P_{i+1}^{n+1} - 6P_i^{n+1} + 4P_{i-1}^{n+1} - P_{i-2}^{n+1}}{4(\Delta x)^2} \right) = \left(\frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x} \right) \quad (5.38)$$

which can be rewritten in a more concise form as

$$\frac{\Delta t}{4(\Delta x)^2} \frac{\partial^4 P}{\partial x^4} (\Delta x)^4 = \left(\frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x} \right) \quad (5.39)$$

The right hand side of this equation is the discrete continuity. The left hand side of the equation is the error in the discrete continuity with non-staggered grids at steady state. The discrete continuity residual approaches a term proportional to the fourth derivative of the pressure, the time step and the cube of the grid

spacing, which is the same error found by [124] using a different pressure correction formulation on a non-staggered grid. Since the time step size, Δt , is proportional to the spatial step size, Δx , the order of the discrete divergence error is $(\Delta x)^3$, which is smaller than the spatial discretization error $(\Delta x)^2$.

For non-staggered grids, the Poisson equation is discretized inconsistently relative to the momentum equations. This results in the discrete divergence of the velocity not being driven to machine zero. It has been shown by others the discrete divergence can not be driven to machine zero, and at the same time, still maintain a smooth pressure field [124, 125, 127]. To obtain a smooth pressure field on a non-staggered grid, [125] adds an artificial mass source term to the right hand side of the discrete continuity equation. To enforce mass conservation for internal flows, [17, 122] imposes the known mass flow rate at the inlet and outlet boundaries.

Several papers in the literature which use similar methods (projection methods on non-staggered grids) have claimed their methods conserve mass exactly [107, 147]. These statements have been described as misleading by [108]. Their cell centered methods interpolate the velocities to the cell edges and the divergence is constructed from these values. The discrete divergence for the edge-centered divergences will exactly satisfy continuity. However, the velocities at the cell centers will not in general satisfy continuity exactly. Because the cell centered data are advanced in time, [108] feels it is invalid to state those methods conserve mass exactly. The momentum interpolation method also has difficulty with the velocities not satisfying the continuity equation [54].

5.6 Analytic Field Function Testing

Critical subroutines of the solver are tested using analytic field functions. The divergence calculation is tested using a known analytic function. The finite volume divergence calculation is very similar to the calculation used in the convective and gradient calculations, so the testing of these subroutines will not be undertaken. Since the function being operated on is a known analytic function, the resulting solution is also a known analytic function. This analytic solution is then compared with the solution obtained from the numerical calculation. This testing of the solver using analytic field functions not only shows the basic accuracy of the solver subroutines, but also gives a quantitative evaluation of the suitability of the grid. The location of high error, during the analytic test, can be identified, but if the errors occur in locations of the grid where changes in the flow variables are insignificant, the errors would not occur during an actual flow simulation and the grid is suitable for a simulation. The percent error of the numerical solution is calculated using

$$\%Error = abs \left(\frac{numerical - analytic}{analytic} \right) * 100 \quad (5.40)$$

In addition, the effect of grid size on the accuracy of divergence and viscous calculations is examined with three successively finer grids.

5.6.1 Analytic Field Test of Sphere Mesh

An all prismatic mesh and a hybrid mesh for a sphere are used to demonstrate the accuracy of the numerical calculations. Grid parameters of the meshes are shown in Table 5.2.

Table 5.2: Grid parameters used for sphere meshes for analytic field function analysis.

Type of Grid	Number of Surface Faces	Number of Prism Cells	Number of Tetrahedral Cells	Total Number of Nodes	Radial Step Size
Prism	1346	87490	0	46530	0.01
Hybrid	1346	36342	160054	49195	0.01

Divergence of a Vector

The divergence calculation is tested using a vector which is a linear function in x, y, z .

The analytic linear vector is

$$\mathbf{U} = (x + y + z)\hat{i} + (x + y + z)\hat{j} + (x + y + z)\hat{k} \quad (5.41)$$

The analytic solution is

$$\nabla \cdot \mathbf{U} = 3.0 \quad (5.42)$$

The numerical errors of the divergence calculation of a linear function, using the prismatic mesh and the hybrid mesh, are shown in Figure 5.10. The percentage of interior nodes is plotted against the percent error in increments of 0.1 percent. It is observed that the majority of the interior nodes in the prismatic mesh have an error of less than 1.0%. Maximum error for this mesh was 1.8%. The majority of the nodes in the hybrid mesh have errors less than 0.1%. Nodes contributing the highest errors are located at the interface of the prismatic and tetrahedral regions.

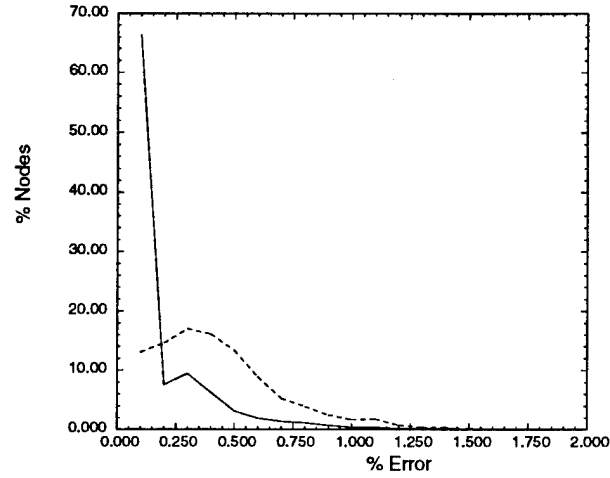


Figure 5.10: Percent error versus percent interior nodes for the divergence calculation of a linear analytic function around a sphere.

- - - Prismatic mesh.

— Hybrid mesh.

5.6.2 Order of Accuracy Evaluation

The effect of grid size on accuracy is demonstrated with three semi-unstructured prismatic meshes around a sphere. Only one quadrant of the sphere is modeled. The medium mesh is created by adding a radial node between every radial node in the coarse mesh and adding a node in the center of the triangular faces in the prismatic cells. One coarse prism cell is subdivided into eight prism cells in order to construct the medium mesh. The fine mesh is created from the medium mesh using the same refinement principles. Specific parameters of the meshes are shown in Table 5.3. The root mean square (rms) error is defined as

$$RMS \ Error = \frac{\left[\sum_{node \ i} (numerical - analytic)^2 \right]^{\frac{1}{2}}}{total \ interior \ nodes} \quad (5.43)$$

The analytic function used in the divergence calculation was shown in equa-

Table 5.3: Grid parameters of sphere meshes for analytic field function grid fineness study.

Type of Grid	Number of Radial Nodes	Number of Surface Nodes	Number of Surface Faces	Total Number of Nodes	Radial Step Size
Coarse	51	87	139	4437	0.08
Medium	101	311	556	31411	0.04
Fine	201	1176	2224	236376	0.02

tion (5.41), while the analytic function used for the viscous calculation is:

$$u = x^2\hat{i} + y^2\hat{j} + z^2\hat{k} \quad (5.44)$$

The analytic solution is

$$\nabla^2 u = 6.0 \quad (5.45)$$

The RMS error of the divergence and viscous calculations using the coarse, medium, and fine grids are shown in Figure 5.11. The slope of the log-log plot shows both calculations are more accurate as the mesh is refined and have second order accuracy.

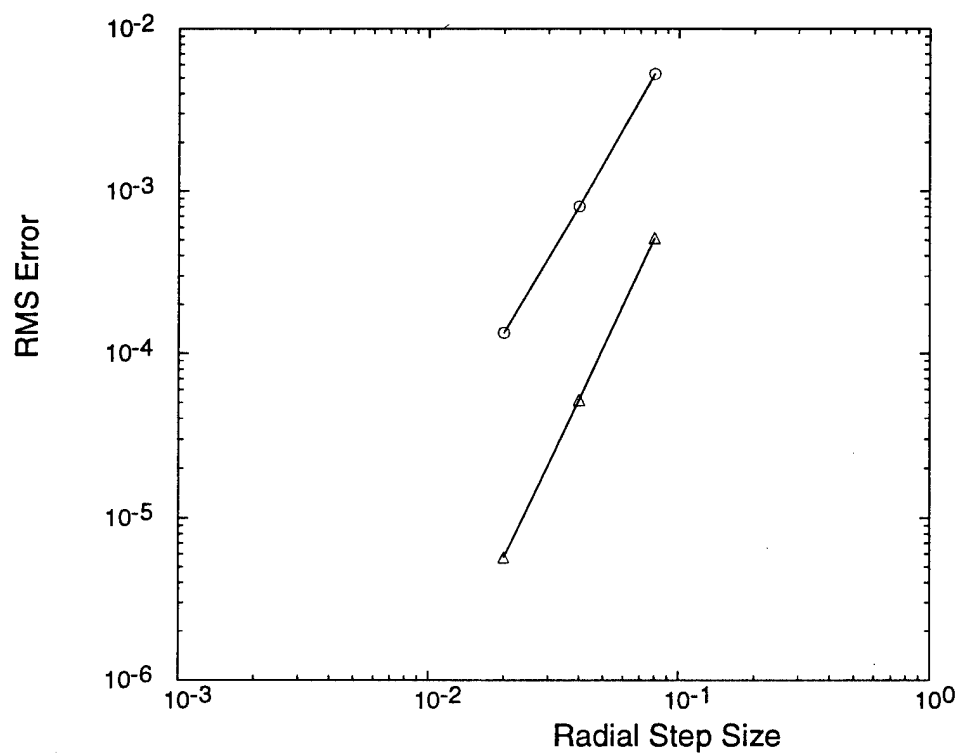


Figure 5.11: RMS error versus radial step size for the divergence and viscous calculations on a semi-unstructured prismatic sphere mesh.

- $\triangle$ Divergence calculation
- $\circ$ Viscous calculation

Chapter 6

Validation and Application of the Adaptive Grid Solver

Validation of the three-dimensional hybrid Navier-Stokes solver for incompressible flows is demonstrated in a piecewise fashion. First, stability of the solver at a high Reynolds number is examined. Spatial accuracy of the solver is investigated on steady quasi-two-dimensional cylinder flow using two different prismatic meshes. Time accuracy is examined with an impulsive start of flow over a cylinder using a prismatic mesh. Three-dimensional aspects of the solver are validated using driven cubic cavity flow, as well as flow over a sphere using prismatic meshes at several Reynolds numbers. The effectiveness of prismatic adaptation is demonstrated with cubic cavity flow. Equation and hybrid grid adaptation are also illustrated with steady flow over a sphere. The effectiveness of hybrid adaptation is also demonstrated with flow over tandem spheres.

6.1 Flow over a Flat Plate at a High Reynolds Number

To examine the *stability* of the algorithm at high Reynolds numbers, the flow over a flat plate is computed at $Re = 10^6$, based on the length of the plate. Grid parameters are shown in Table 6.1 while boundary conditions are shown in Figure 6.1. The computational domain begins at $x=1.0$ with inlet conditions of a Blasius profile. The artificial dissipation factor, σ_4 , was equal to 10^{-6} and the CFL factor was 0.01. The signature of the semi-unstructured prismatic mesh on the plate, as well as on the other boundaries, is shown in Figure 6.2. The plate is tessellated with triangles, while the side boundaries are tessellated with the quadrilateral faces of the prisms.

The u velocity profile at the center of the plate is shown in Figure 6.3. The numerical results coincide with the Blasius velocity profile [120]. The convergence history of the simulation in terms of the maximum residual in the x -momentum equation is shown in Figure 6.4. The maximum residual is defined as the maximum of $|u^{n+1} - u^n|$ over all nodes. From these results, the following conclusions can be drawn:

1. The pressure correction scheme with a non-staggered grid converges well and is stable for flow over a flat plate at a high Reynolds number.
2. The incompressible three-dimensional solver, using a semi-unstructured prismatic mesh, yields an accurate solution when compared to analytical results.

Table 6.1: Grid parameters used for flat plate flow at $Re = 10^6$.

Number of Faces on Wall	Grid Spacing at Wall	Grid Stretching Factor	Number of Prism Layers	Number of Nodes
1774	0.0001	1.125	70	67308

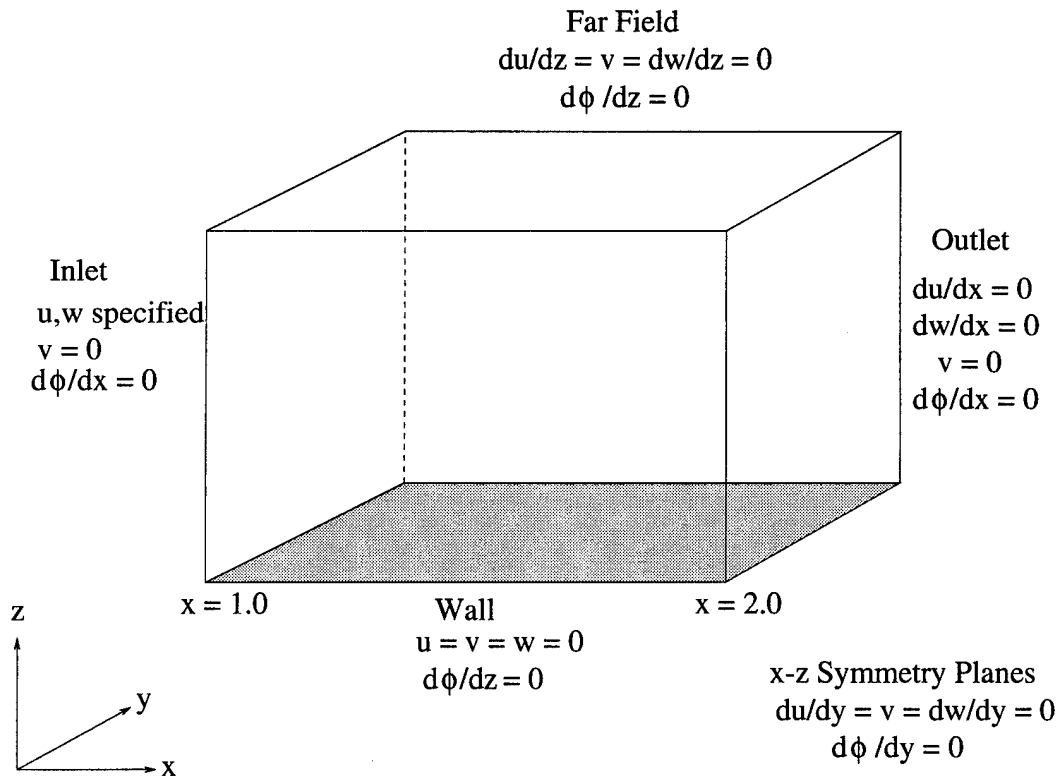


Figure 6.1: Boundary conditions for flat plate simulation at $Re = 10^6$.

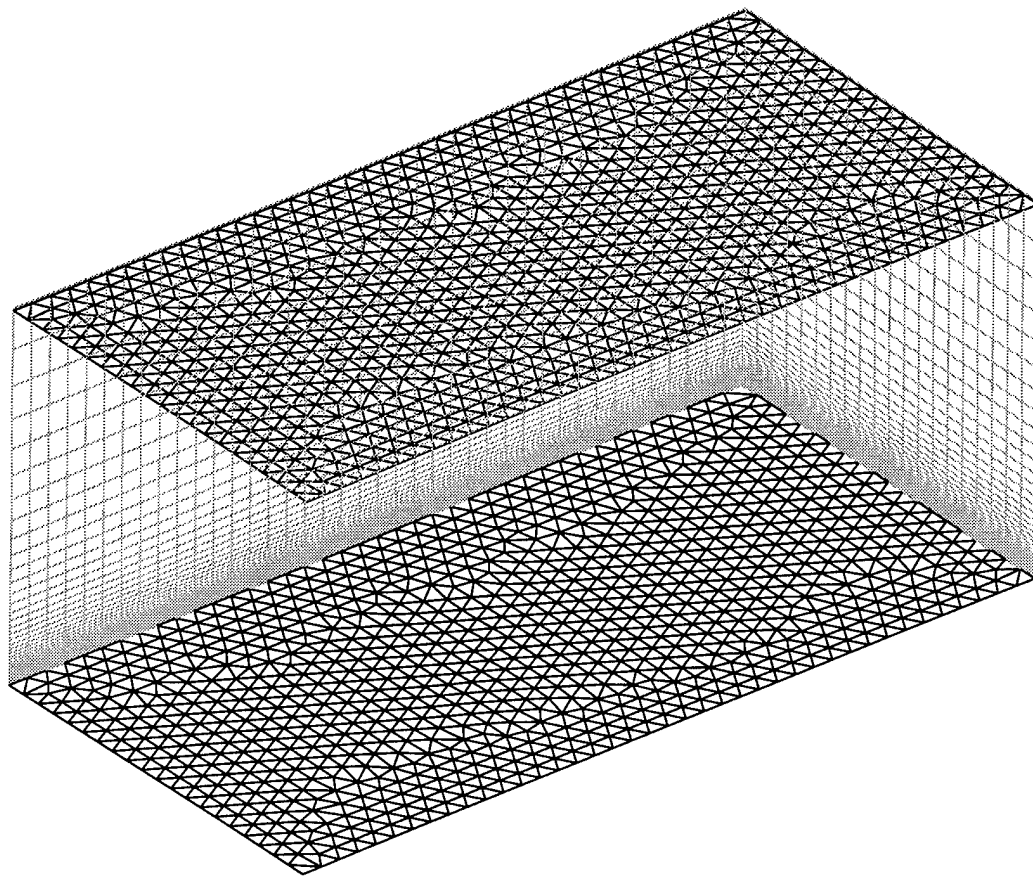


Figure 6.2: Prismatic mesh with unstructured surface tessellation for flat plate at $Re = 10^6$.

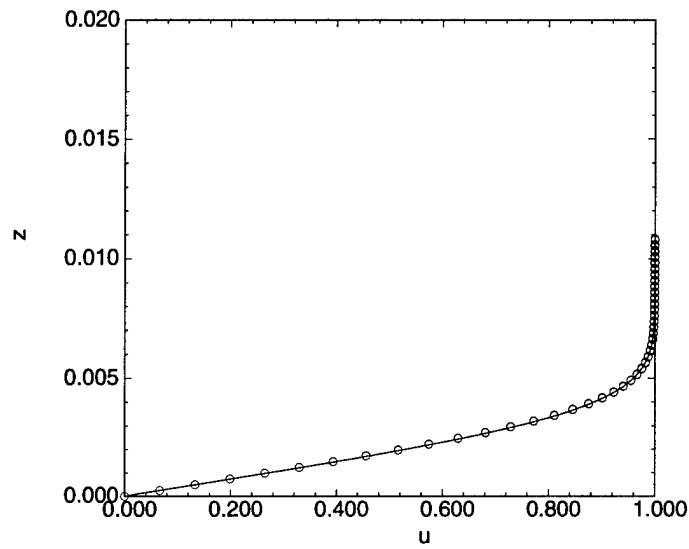


Figure 6.3: U velocity profile at the center of a flat plate ($Re = 10^6$).
 o Blasius result — Present work

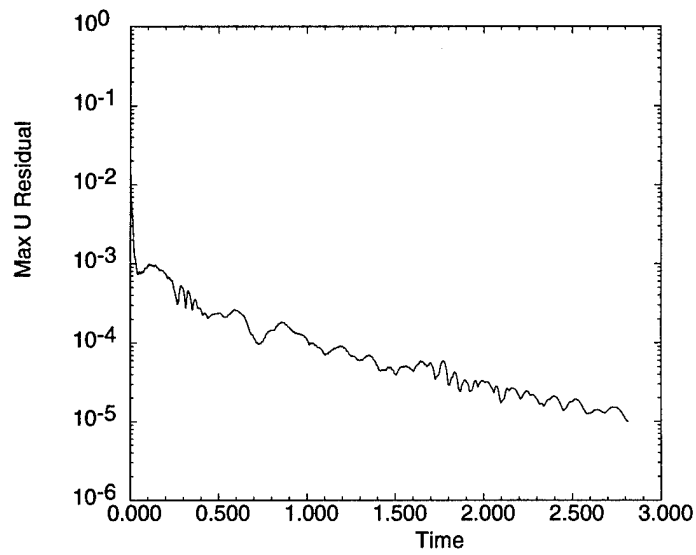


Figure 6.4: Convergence of the maximum residual in the x-momentum equation for a flat plate ($Re = 10^6$).

6.2 Steady Flow Around a Cylinder using a Prismatic Mesh

Steady quasi-two-dimensional flow around a circular cylinder at a Reynolds number of 40, based on the cylinder diameter, is used to examine the spatial accuracy of the solver on a prismatic mesh. At this Reynolds number, the flow is attached to the cylinder over most of its surface, with an attached separation bubble behind the cylinder.

Two meshes were used to demonstrate the validity of the solver. The first mesh has a “structured” surface triangulation, while the second mesh has an “unstructured” surface triangulation. The first mesh is shown in Figure 6.5 and is generated by subdivision of quadrilateral faces along one of their diagonals. The second “unstructured” mesh is shown in Figure 6.6 which is generated via an advancing front-type of surface grid generation. The cylinder axial length is three times the diameter. Grids of varying fineness were tested until the results were no longer mesh dependent. The grid parameters used for this case are shown in Table 6.2. Symmetry planes were defined at both ends of the cylinder. Location of the far field boundary is at approximately 21 diameters. Surface triangular faces were stretched in the axial direction since the flow does not vary along the axis of the quasi two-dimensional cylinder. The artificial dissipation factor, σ_4 , was equal to 10^{-5} and the CFL factor was 0.1. Results from the three-dimensional solver show the flow is basically two-dimensional.

Particle paths for one cross-sectional plane of the cylinder are shown in Figure 6.7. These paths were obtained from the solution obtained from the “unstructured” mesh. The attached flow and separation bubble are clearly shown.

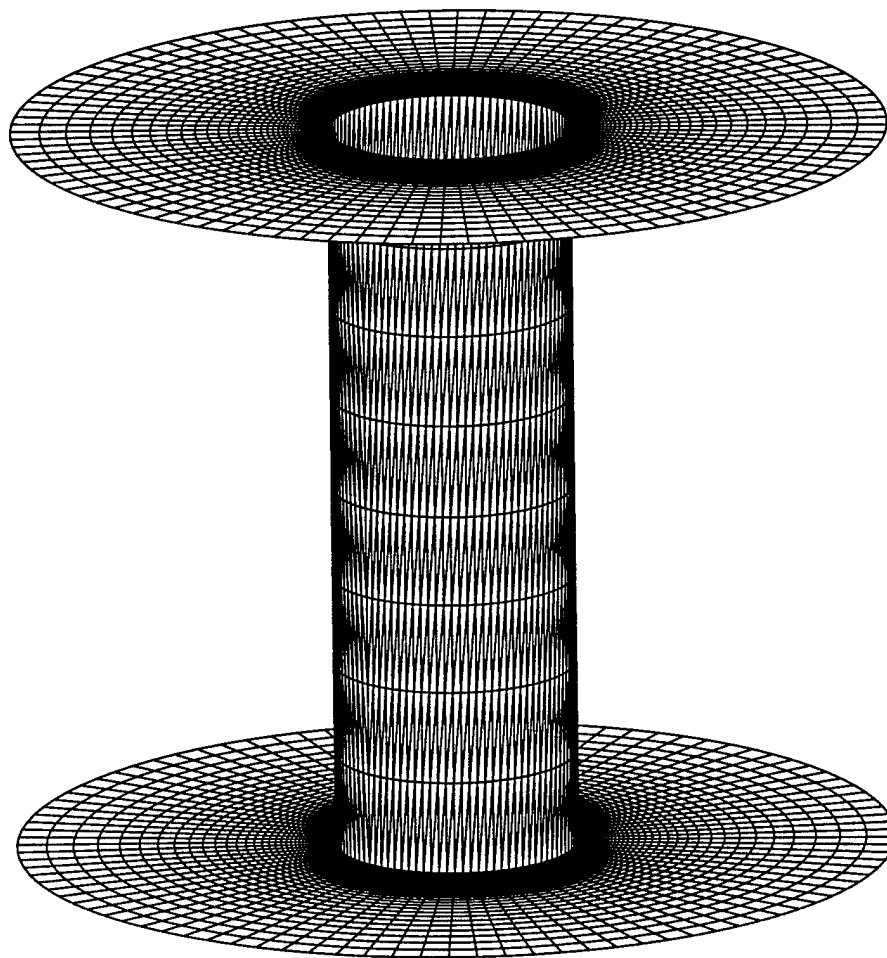


Figure 6.5: Partial prismatic mesh for circular cylinder at $Re = 40$ with “structured” surface triangulation.

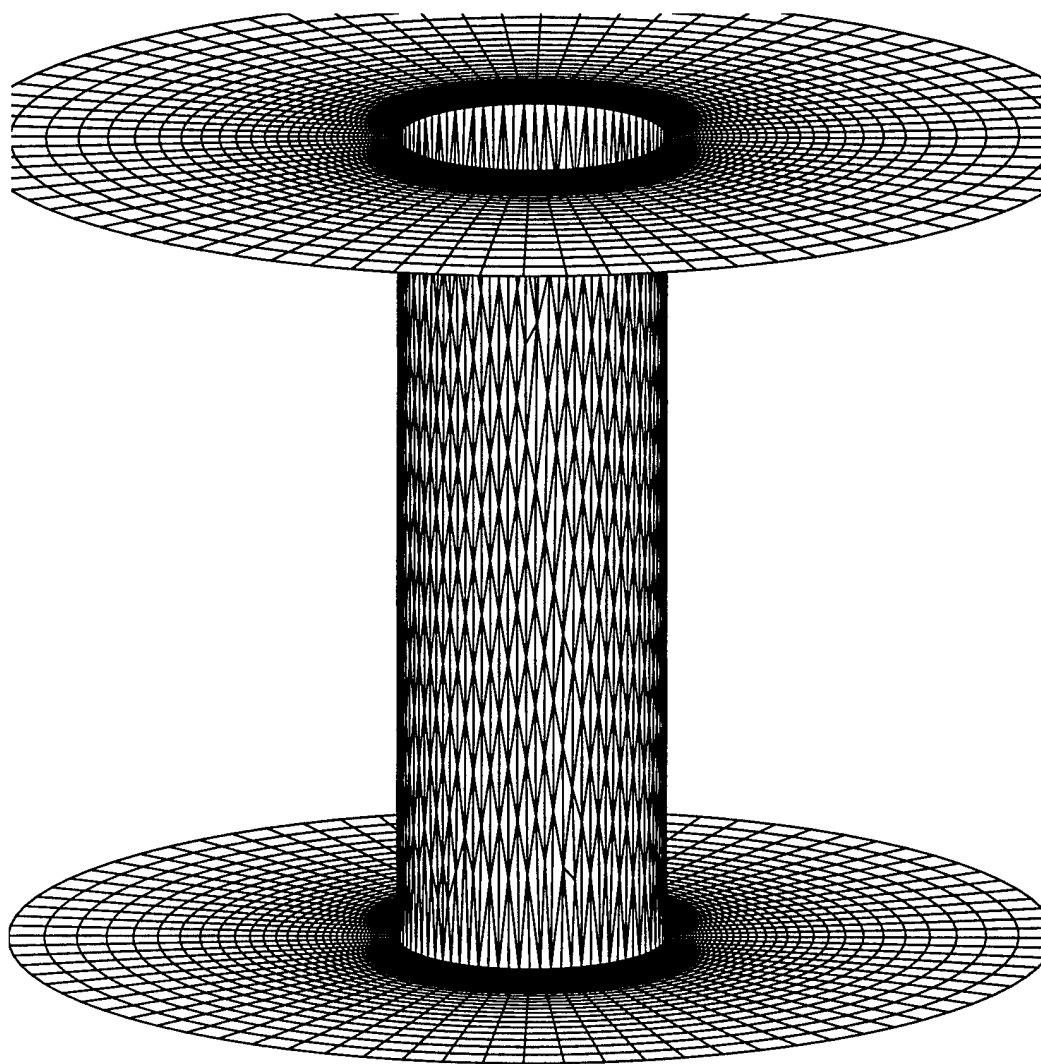


Figure 6.6: Partial prismatic mesh for circular cylinder at $Re = 40$ with “unstructured” surface triangulation.

Table 6.2: Grid parameters used for circular cylinder flow at $Re=40$.

Type of Surface Mesh	Number of Surface Faces	Number of Prismatic Layers	Radial Stretching Factor	Number of Surface Nodes	Total Number of Nodes
“Structured”	1536	65	1.1	864	57024
“Unstructured”	1604	65	1.1	890	58740

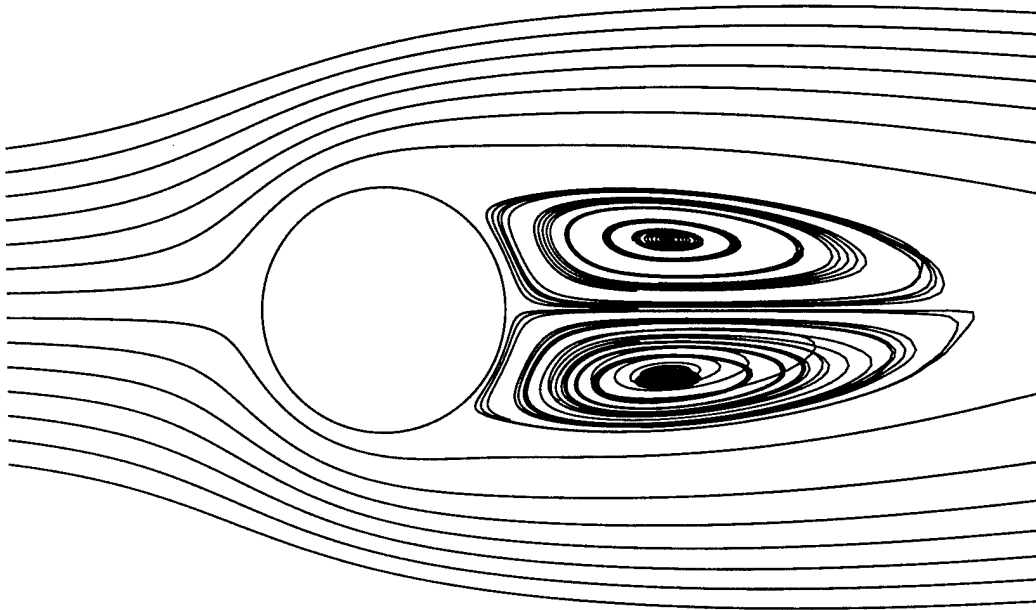


Figure 6.7: Flow pathlines around a cylinder at $Re = 40$.

The pressure coefficient on the wall of the cylinder along the circumferential direction for both grids is shown in Figure 6.8. The pressure coefficient is defined in equation (6.1).

$$C_p = \frac{p - p_\infty}{\frac{1}{2}\rho_\infty U_\infty^2} \quad (6.1)$$

The angle theta on the abscissa is defined as the angle from the forward stagnation point to the aft stagnation point on the cylinder. Numerical data are compared with experimental data by Grove [46]. The figure shows the solution obtained from the “unstructured” mesh compares well with the experimental data and the solution from the “structured” mesh.

Nondimensional vorticity on the wall of the cylinder is shown in Figure 6.9. The nondimensional vorticity is defined in equation (6.2) where D is the diameter of the cylinder and ω_y is the axial component of the vorticity.

$$vorticity = \frac{\omega_y D}{2U_\infty} \quad (6.2)$$

Numerical results are compared with the numerical data by Fornberg [40]. Fornberg’s numerical data were obtained using the stream function-vorticity method with 130 points in the circumferential direction and 114 points in the radial direction. The figure shows the solution obtained from the “unstructured” mesh compares reasonably well with the data from the literature and the solution from the “structured” mesh.

Additional comparisons of the present work on the “unstructured” grid with numerical and experimental results found in the literature [20, 36, 97, 131, 133] are shown in Table 6.3. The separation angle is defined as the angle from the rear stagnation line at which the attached boundary layer separates from the cylinder. The other parameters are the nondimensional length of the bubble, and the drag

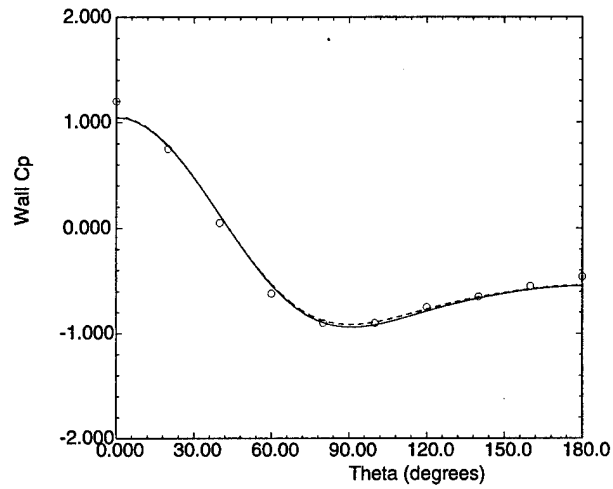


Figure 6.8: Pressure coefficient distribution on cylinder surface for $Re=40$.

- Experiment by Grove [46]
- - Present work using "structured" grid
- Present work using "unstructured" grid

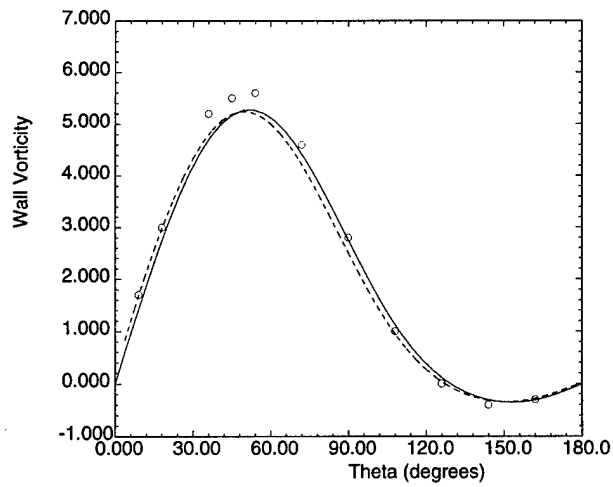


Figure 6.9: Vorticity distribution on cylinder surface for $Re=40$.

- Numerical results by Fornberg [40]
- - Present work using "structured" grid
- Present work using "unstructured" grid

coefficient which is defined as follows:

$$C_d = \frac{\text{Drag}}{\frac{1}{2}\rho_\infty U_\infty^2 A} \quad , \quad (6.3)$$

where A is the frontal area of the cylinder (diameter times length). The drag is calculated by integrating the pressure and shear stresses acting on the faces of the cells on the cylinder surface. The convergence of the drag coefficient and the maximum residual of the x-momentum equation are shown in Figures 6.10 and 6.11 for the case using the “unstructured” grid. From these results, the following conclusions can be drawn:

1. A grid composed of prismatic cells can yield accurate results when compared to experimental data and other numerical data from the literature.
2. The pressure correction scheme converges well and yields stable solutions with fully unstructured prisms.
3. The use of a non-staggered grid with artificial dissipation produces smooth pressure and vorticity fields.
4. Fully unstructured prisms yield the same accuracy as “structured” prismatic meshes.

Table 6.3: Comparisons of results for circular cylinder flow at $Re=40$.

	Separation Angle (deg)	Bubble Length/ Cylinder Diameter	Drag Coefficient
Present Work	53	~ 2.0	1.49
Other Computations and Experiments	50 - 53	1.8 - 2.5	1.44 - 1.70

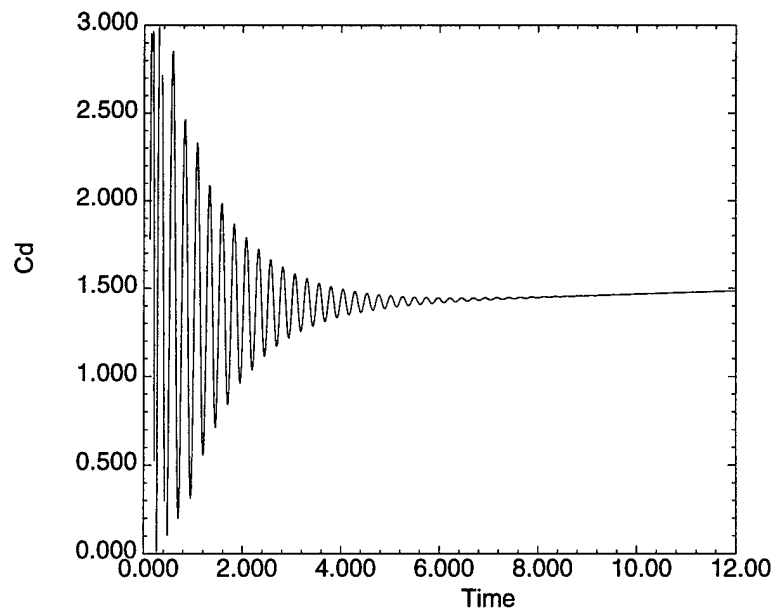


Figure 6.10: Convergence of the drag coefficient on a cylinder at $Re = 40$.

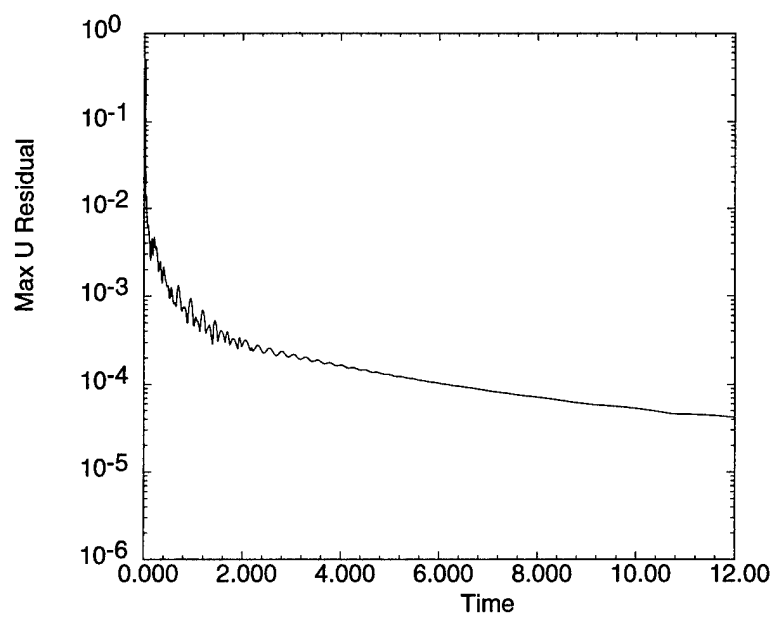


Figure 6.11: Convergence of the maximum residual in the x-momentum equation on a cylinder at $Re = 40$.

6.3 Impulsive Start of a Cylinder using a Prismatic Mesh

Impulsive start of a circular cylinder which attains a steady speed corresponding to a Reynolds number of 40 is simulated in order to examine the *time* accuracy of the solver with a prismatic mesh. Grids of varying fineness were tested until the results were no longer mesh dependent. The grid parameters used for this case are shown in Table 6.4. The surface mesh is the same fully-unstructured surface mesh used in the previous section. Symmetry planes were defined at both ends of the cylinder. The artificial dissipation factor, σ_4 , was equal to 10^{-4} and the CFL factor was 0.1.

Initial conditions of the impulsive start simulation is potential flow [43]. As the simulation begins, the nodes on the aft stagnation line of the cylinder are monitored and the node closest to the cylinder to register a positive velocity defines the outermost position of the separation bubble. The beginning of the separation bubble is defined at the rear of the cylinder surface. The separation bubble length, L , is nondimensionalized by the cylinder diameter, D , and the time is the nondimensional time, tU_∞/D . The growth of the separation bubble in time is shown in Figure 6.12. Experimental results from Coutanceau [37] and Honji [53] are shown for comparison.

Results from the present work match the unsteady portion of the data, however, it slightly underpredicts the length of the steady state value of the separation bubble. Experimental results also show a significant variation in the steady state value of the separation bubble size, and other numerical data [37] and Table 6.3 show more variation. Other numerical results [33] report a slight decrease in the steady state value of the separation bubble as the grid is refined. The present work

found a similar trend.

From these results, the following conclusion can be drawn: for this impulsive start, the scheme is time accurate when compared to experimental data.

Table 6.4: Grid parameters used for unsteady circular cylinder flow at $Re=40$.

Number of Surface Faces	Number of Prismatic Layers	Radial Stretching Factor	Grid Spacing at Wall	Total Number of Nodes	Radius of Far Field Boundary
1604	111	1.045	0.01	99680	~ 20

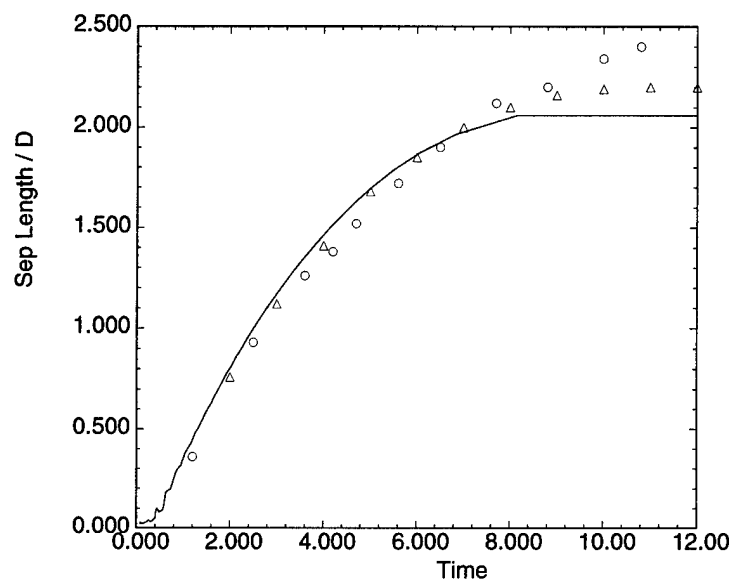


Figure 6.12: Growth in time of the separation bubble for an impulsive start of a cylinder $Re = 40$.

$\triangle$ Experiment by Coutanceau [37]

$\circ$ Experiment by Honji [53]

— Present Work

6.4 Steady 3-D Flow in a Driven Cubic Cavity with a Prismatic Mesh

Steady flow within a driven cubic cavity is used to demonstrate the three-dimensional capability of the solver on a prismatic mesh. A cubic cavity is modeled with sides of length one. The lid (at $z = 1.0$) is moving with a steady velocity in the positive x direction. Since the flow is symmetric about the $y=0.5$ plane, only half the cubic cavity is actually modeled with one plane being a symmetry plane. The flow at Reynolds number of 400, based on the length of the sides of the cubic cavity, is modeled. No artificial dissipation was used in this case and the CFL factor was 0.1. Grids of varying fineness were tested until the results were no longer mesh dependent. The grid parameters are shown in Table 6.5.

Two velocity contours of the flow on x - z planes at $y=0.015$ and $y=0.5$ are shown in Figure 6.13 to give a qualitative view of the three dimensionality of the flow. The cavity lid is the top surface and is moving obliquely from right to left. The grid mesh on the bottom wall is shown, with the clustered points near walls. The grid mesh on the $x=0$ side wall and the $y=0$ side wall are also shown. Velocity contours in three y - z planes at $x=0.1$, 0.5, and 0.9 are shown in Figure 6.14. The

Table 6.5: Grid parameters used for the cubic cavity at $Re = 400$.

Number of Faces on Wall	Total Number of Nodes	Number of Layers	Smallest Grid Step Size	Largest Grid Step Size
2500	67626	50	0.01	0.025

velocity contours do not show the presence of small vortices in the corners of the cavity, however, velocity vector plots, which are not shown, indicate the vortices are present.

Two velocity profiles on the symmetry plane of the cubic cavity are plotted for comparison of the solution accuracy with numerical data by Babu [10]. Figure 6.15 shows the u velocity versus the z coordinate at $x = 0.5$, and Figure 6.16 shows the w velocity versus the x coordinate at $z = 0.5$. Both plots show the present work compares well with the numerical data from the literature. The pressure boundary conditions for this cubic cavity flow required special consideration. The solid walls of the cavity required boundary conditions of

$$\frac{\partial \phi}{\partial n} = \frac{\Delta t}{Re} \frac{\partial^2 u^n}{\partial n^2} \quad (6.4)$$

where n represents the normal direction to the specific wall. Derivatives were either forward or backward differenced, depending on the location of the wall. From these results, the following conclusions can be drawn:

1. The simulated flow within a three-dimensional geometry, using a grid composed of prismatic cells, contains the physical characteristics of the expected flow.
2. The solver yields accurate results when compared to other numerical data from the literature.

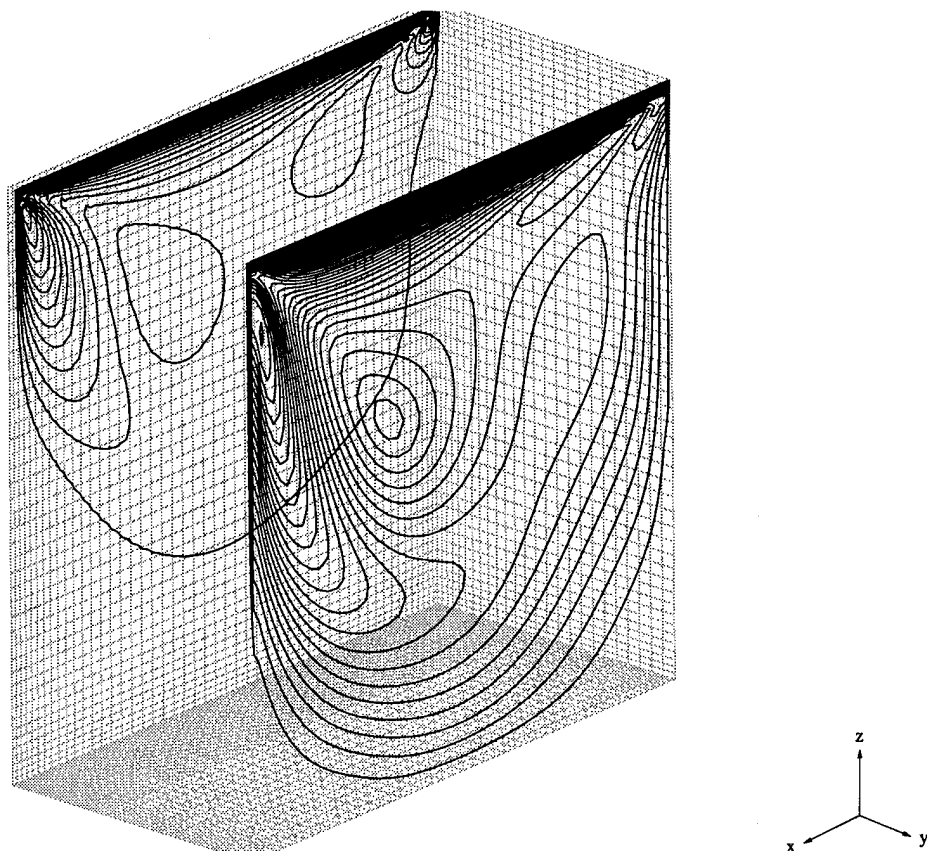


Figure 6.13: Velocity contours on the x-z planes in the cubic cavity at $Re = 400$, at $y=0.015$ and $y=0.5$. Contour lines are plotted with velocity increments equal to 0.025.

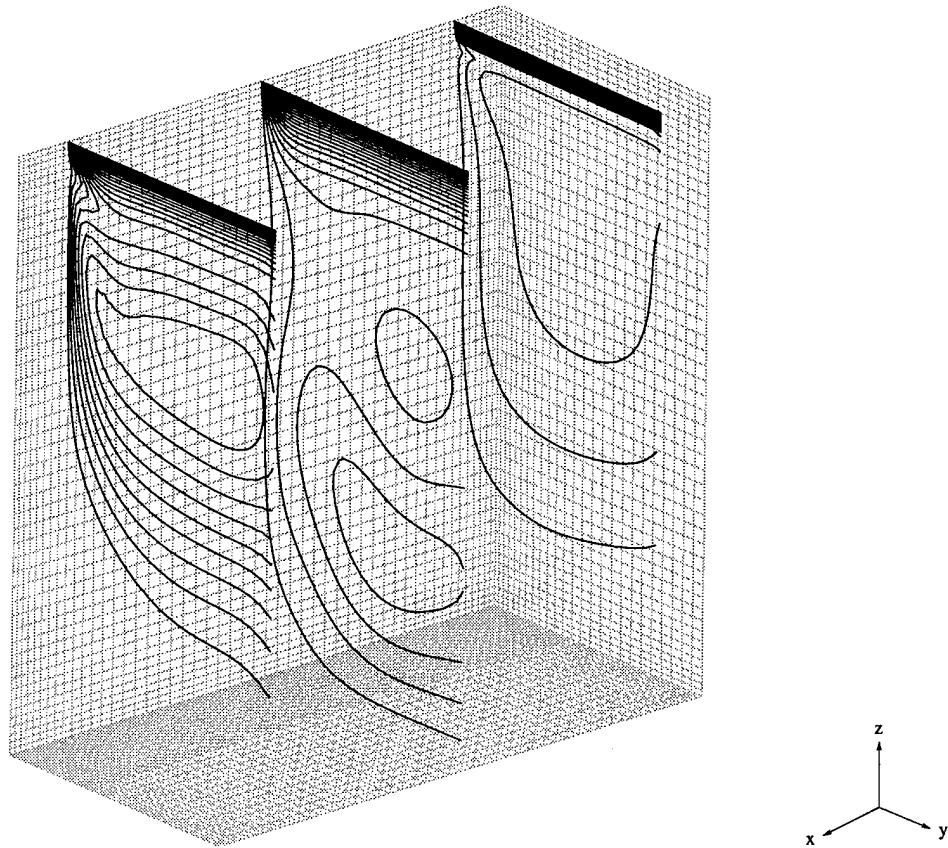


Figure 6.14: Velocity contours on the y-z planes in the cubic cavity at $Re = 400$, at $x=0.1, 0.5$, and 0.9 . Contour lines are plotted with velocity increments equal to 0.05 .

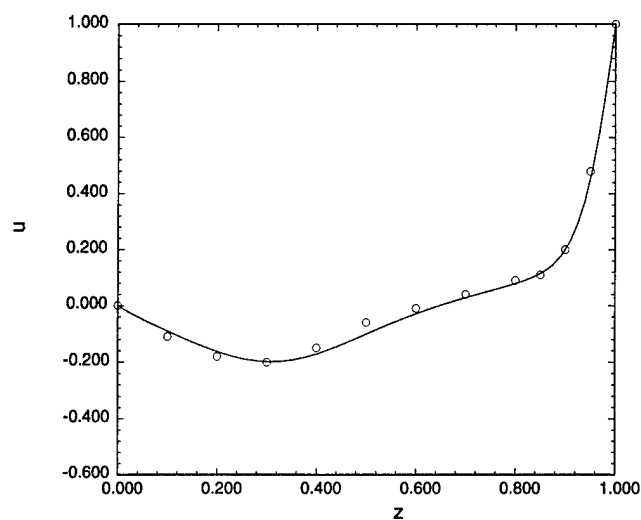


Figure 6.15: U velocity profile versus z on the symmetry plane at $x = 0.5$ for the cubic cavity at $Re = 400$.

○ Numerical result by Babu [10]

— Present work

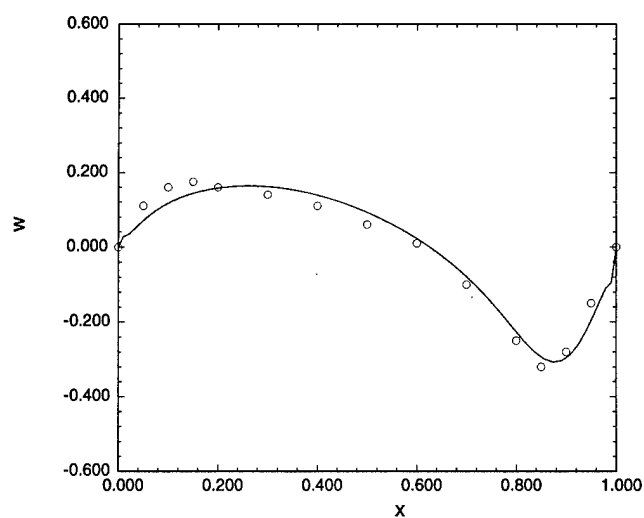


Figure 6.16: W velocity profile versus x on the symmetry plane at $z = 0.5$ for the cubic cavity at $Re = 400$.

○ Numerical result by Babu [10]

— Present work

6.5 Steady 3-D Flow Around a Sphere using a Prismatic Mesh

Steady flow around a sphere is used to demonstrate the three-dimensional capability of the solver on a prismatic mesh. The spherical flow is calculated for several Reynolds numbers, based on the diameter of the sphere. At very low Reynolds numbers, the viscous terms dominate and the convective terms are negligible (Stokes flow). As the Reynolds number increases, convective terms become significant and Oseen developed a correction to the equations for Stokes flow. When the Reynolds number exceeds 1.0, both Stokes and Oseen's flow equations are invalid, and the resulting flow is steady and attached. An attached axisymmetric ring eddy appears when the Reynolds number is approximately 20 [139]. The ring eddy remains steady and attached until the Reynolds number reaches 130. A slight oscillation in the wake [92, 117] develops at this Reynolds number. Experiments [117] have shown periodic hairpin vortex shedding begins at approximately a Reynolds number of 300. These vortices produce an unsteady laminar wake. When the Reynolds number is increased to 800, the wake becomes turbulent and the structure of the vortices become unclear.

At low Reynolds numbers, below 300, the flow over a sphere is basically steady and symmetric. Therefore, the flow simulations at these Reynolds numbers used a half sphere on a symmetry plane. Grids of varying fineness were tested until the results were no longer mesh dependent. Grid parameters used for the sphere calculations are shown in Table 6.6. All the grids had 1346 triangles on the surface mesh and used a radial stretching factor of 1.1. The convergence criterion for the maximum residual in the x-momentum equation was 10^{-5} . The artificial dissipation factor, σ_4 , was equal to 10^{-4} and the CFL factor was 0.1. The prismatic mesh for

Table 6.6: Prismatic grid parameters used for flow around a sphere.

Reynolds Number	Number of Prismatic Layers	Grid Spacing at Wall	Total Number of Nodes	Radius of Far Field Boundary
0.1	57	0.02	40890	~ 20
1.0	65	0.01	46530	~ 21
10.0	65	0.01	46530	~ 21
40.0	65	0.01	46530	~ 21
100.0	65	0.01	46530	~ 21
200.0	65	0.01	46530	~ 21

the Reynolds number equal to 10.0 case is shown in Figure 6.17.

Classic experimental drag coefficient versus Reynolds number data from Schlichting [120], as well as other experimental data with an error range [115], are used for comparison, as well as the analytic results from Stokes flow around a sphere and Oseen's equation for a sphere [97, 120]. The drag coefficient in the present work is defined by the following equation:

$$C_d = \frac{\text{Drag}}{\frac{1}{2}\rho_{\infty}U_{\infty}^2 A} \quad (6.5)$$

where A is the frontal area of the sphere (πr^2) where r is the radius of the sphere.

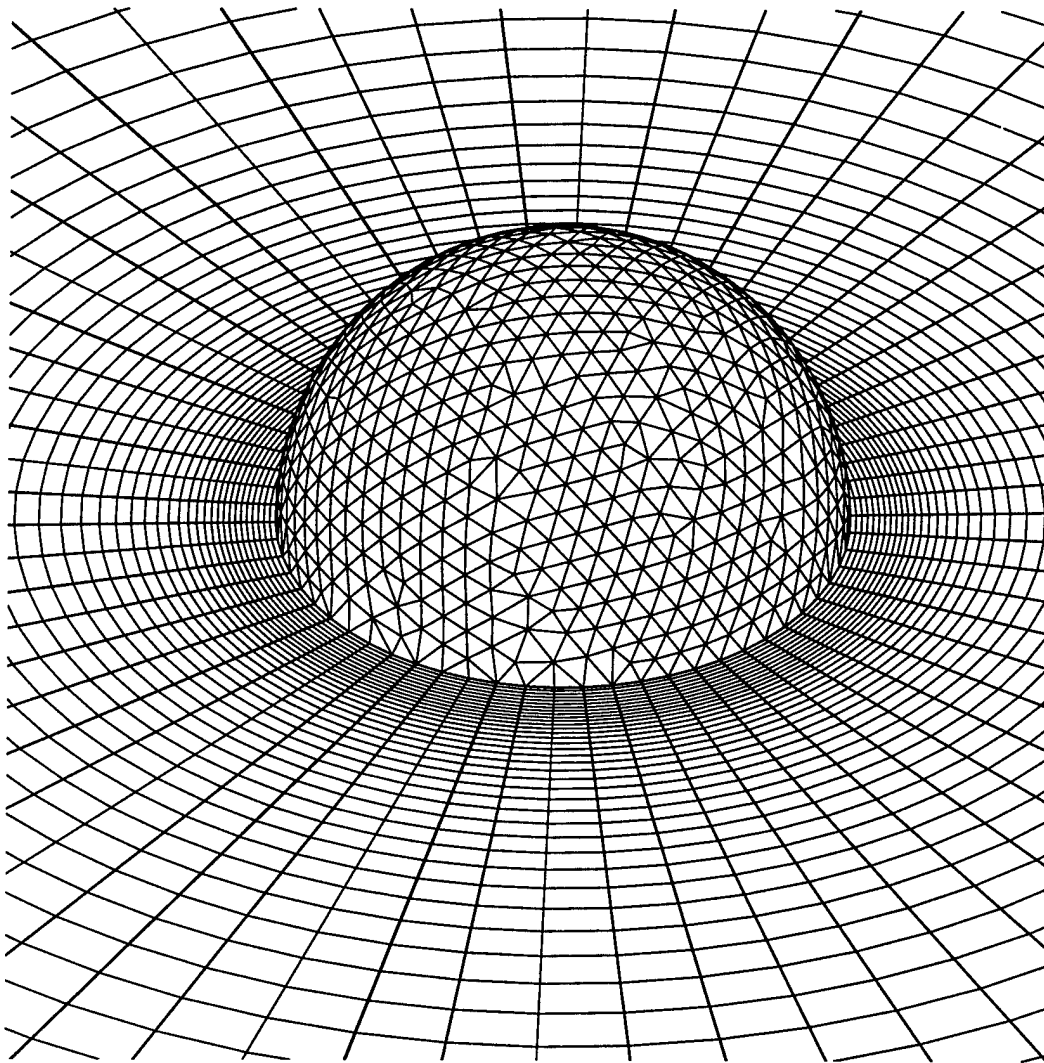


Figure 6.17: Prismatic surface and symmetry plane mesh for sphere at $Re = 10$.

The drag coefficient for Stokes flow is given as follows, while the coefficient corresponding to Oseen's solution is given by the following equation.

$$C_{dStokes} = \frac{24.0}{Re} \quad (6.6)$$

$$C_{dOseen} = \frac{24.0}{Re} \left(1 + \frac{3}{16} Re \right) \quad (6.7)$$

Results of the drag coefficient comparisons are shown in Figure 6.18. Numerical results agree quite well with the theoretical results and fall within the scatter of the experimental results.

The wall pressure coefficient and vorticity are plotted in Figures 6.19 and 6.20, respectively, for the $Re=0.01$ case. The pressure coefficient shows an antisymmetric profile which is characteristic of Stokes flow. The vorticity shows a symmetric profile which is also characteristic of Stokes flow, ie., no separation has occurred. The wall pressure coefficient and vorticity are plotted in Figures 6.21 and 6.22, respectively, for the $Re=10, 40, 100$, and 200 cases. Both figures show separation occurring on the aft side of the sphere when the Reynolds number is greater than 40 . The pathlines around the sphere are illustrated for the Reynolds number equal to 200 case in Figures 6.23 and 6.24. Two orthogonal plane cuts are shown to illustrate the presence of the ring eddy on the rear of the sphere. From these results, the following conclusions can be drawn:

1. The incompressible solver yields accurate solutions, for flow over a three-dimensional body, when compared to experimental data.
2. The solver yields expected results over a range of Reynolds numbers which includes Stokes flow and separated flows.

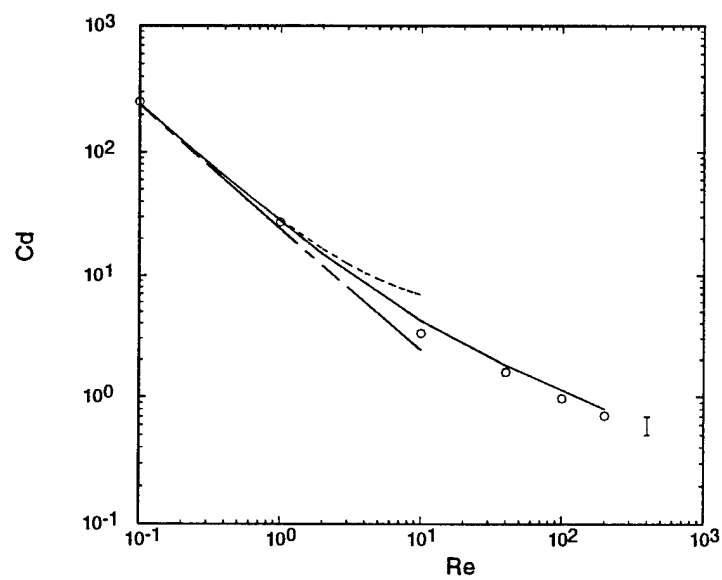


Figure 6.18: Drag coefficient of a sphere versus Reynolds number.
 Tic at lower right represents error range of experimental data.
 - - - Stokes flow
 - - - - Oseen's equation
 — Experimental data from Schlichting [120] and Roos [115]
 o Present work with hemisphere grid

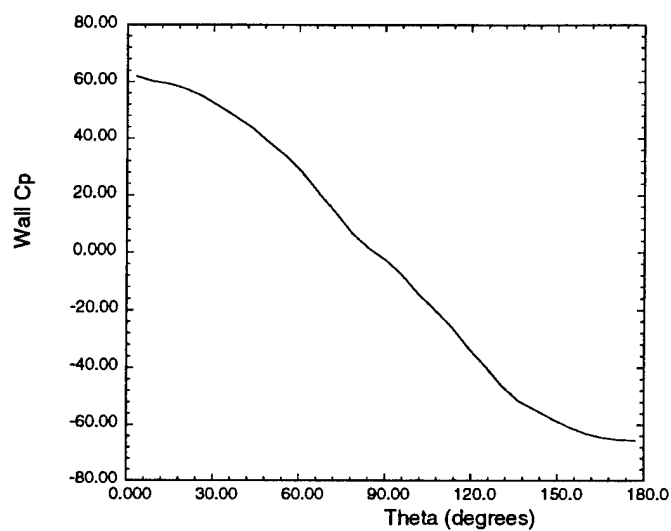


Figure 6.19: Pressure coefficient distribution on the surface of the sphere for $Re=0.1$ (Stokes flow).

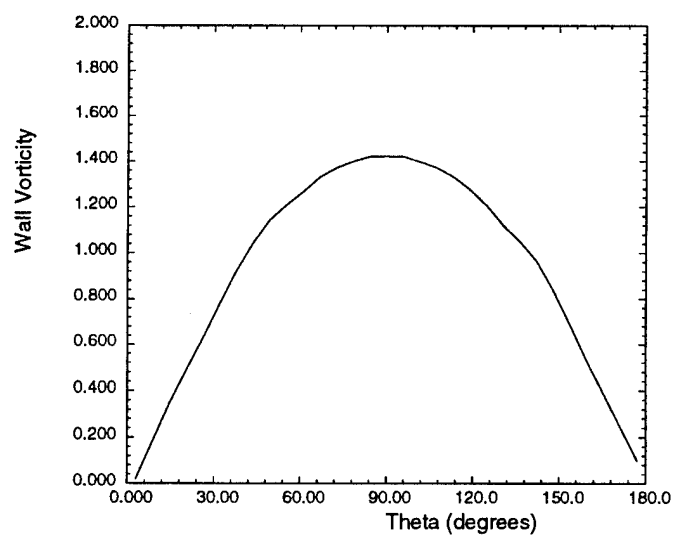


Figure 6.20: Vorticity distribution on the surface of the sphere for $Re=0.1$ (Stokes flow).

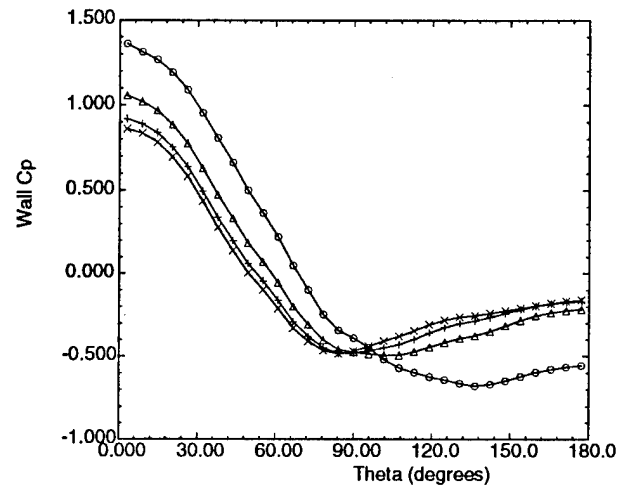


Figure 6.21: Pressure coefficient distribution on the surface of the sphere for low Re.

$\circ$ Re = 10 $\triangle$ Re = 40
 $+$ Re = 100 $\times$ Re = 200

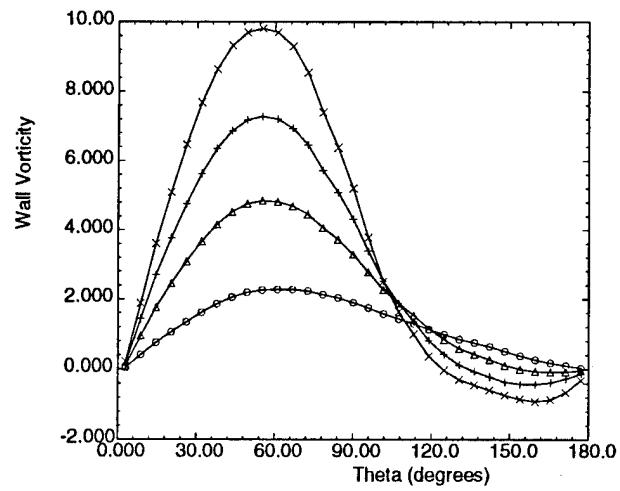


Figure 6.22: Vorticity distribution on the surface of the sphere for low Re.

$\circ$ Re = 10 $\triangle$ Re = 40
 $+$ Re = 100 $\times$ Re = 200

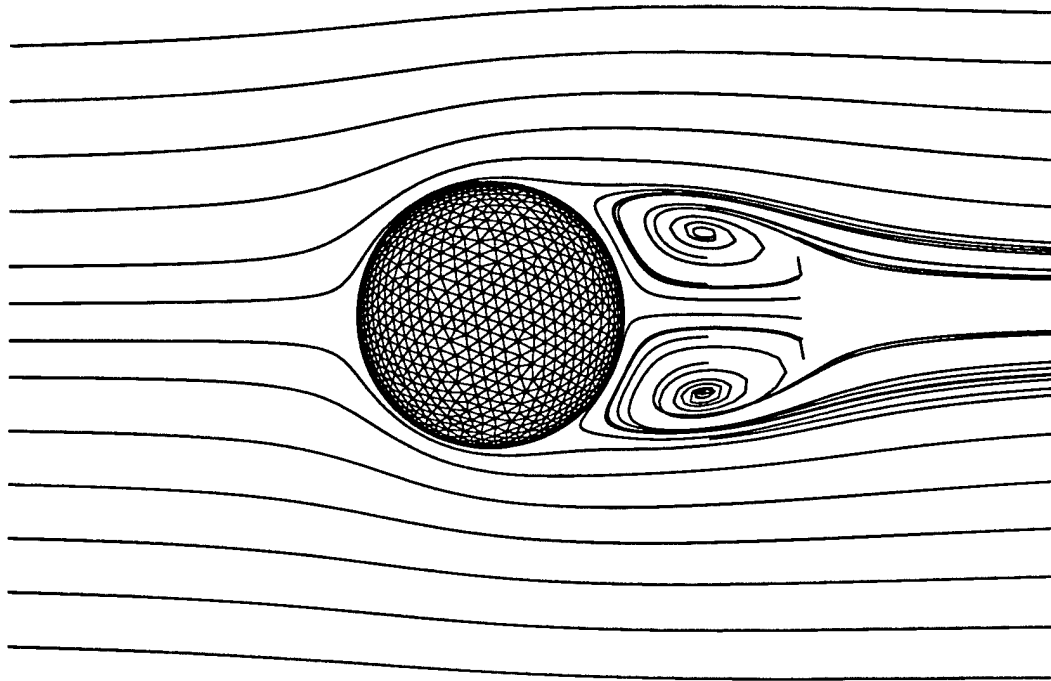


Figure 6.23: Pathlines on the x-z plane near a sphere at $Re = 200$.

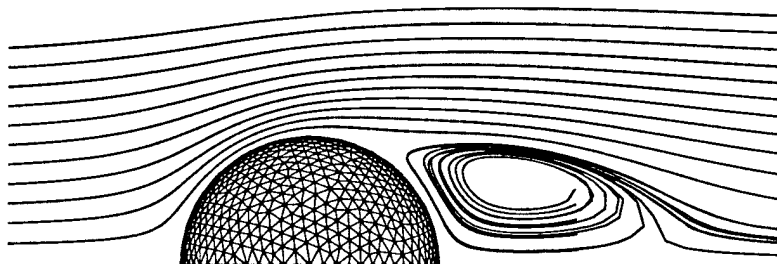


Figure 6.24: Pathlines on the x-y plane near a sphere at $Re = 200$.

6.6 Evaluation of Prismatic Mesh Adaptation Applied to Cubic Cavity Flow

Prismatic adaptation will be evaluated by examining the accuracy of the solution with local embedding compared to the solution from an unembedded coarse grid. The solution obtained with local embedding should be the same as the solution obtained by the corresponding globally embedded mesh. Robustness of the scheme will also be examined.

6.6.1 Accuracy of Prismatic Adaptation on Cubic Cavity Flow

Steady flow in a cubic cavity at a Reynolds number of 400 is calculated using coarse, adapted, and globally adapted prismatic grids. The cubic cavity is modeled with sides of length one. The lid (at $z = 1.0$) is moving with a steady velocity in the positive x direction. Since the flow is symmetric about the $y=0.5$ plane, only half the cubic cavity is actually modeled with one plane being a symmetry plane. All three grids have the same point distribution in the z direction as shown in Table 6.5. Only the surface triangulation is different in the three cases. The adapted grid was obtained from the coarse grid using the solution after 5000 iterations. The CFL factor for all three cases was 0.1, and no artificial dissipation was used. The grid parameters for the three grids are shown in Table 6.7. The coarse, adapted, and globally adapted grids are shown in Figures 6.25, 6.26, and 6.27, respectively. In Figure 6.26, directional embedding of the prisms is focused near the side-walls of the cavity. This is due to the flow decelerating near the wall boundaries. The boundary which is not adapted is a symmetry plane. The solid walls of the cavity had boundary conditions of $\frac{\partial \phi}{\partial n} = 0.0$ where n represents the normal direction to the

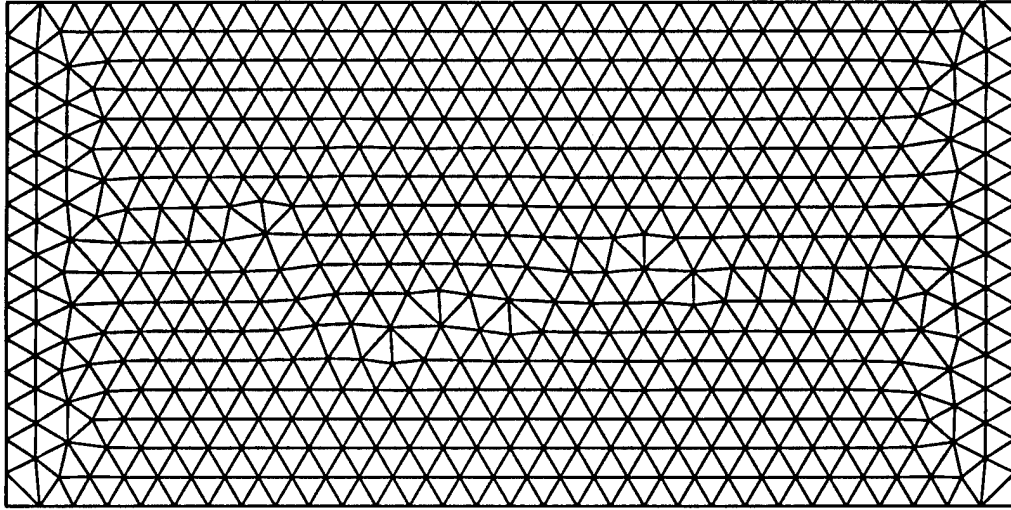


Figure 6.25: Coarse grid surface triangulation for cubic cavity at $Re = 400$.

specific wall.

Two velocity profiles on the symmetry plane of the cubic cavity are plotted for comparison of the solution accuracy. Figure 6.28 shows the u velocity versus the z coordinate at $x = 0.5$, and Figure 6.29 shows the w velocity versus the x coordinate at $z = 0.5$. There is no significant difference between the three solutions in Figure 6.28, but the coarse grid gives a very poor solution for the w velocity in Figure 6.29.

6.6.2 Robustness and CPU Time Savings of Prismatic Adaptation

Robustness of the prismatic adapter is demonstrated in Figure 6.30. The maximum residual in the x -momentum equation is plotted versus time for the adapted and globally adapted cases. The large spike in Figure 6.30 occurs due to the linearly interpolated solution from the initial coarse grid nodes to the newly created ones due to adaptation. The flow solver converges relatively quickly after the restart to

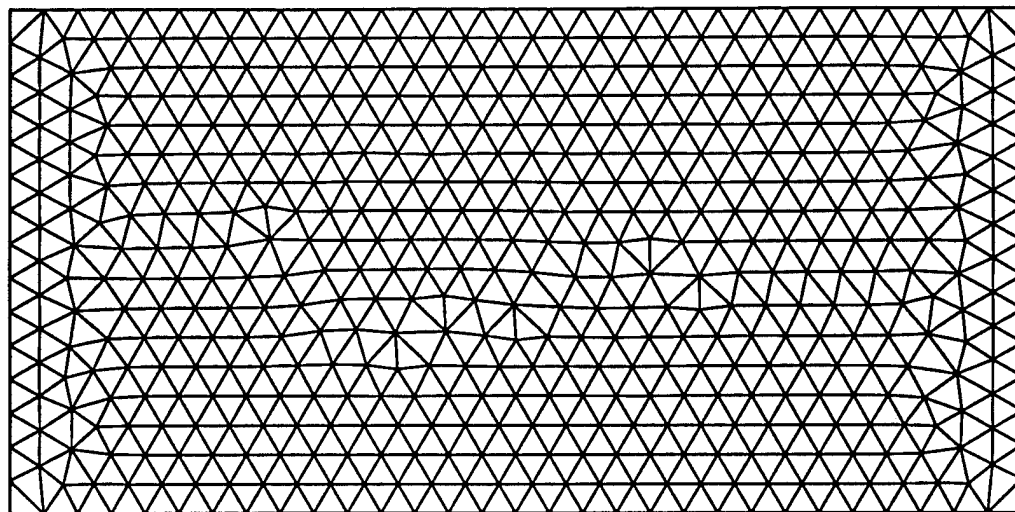


Figure 6.25: Coarse grid surface triangulation for cubic cavity at $Re = 400$.

specific wall.

Two velocity profiles on the symmetry plane of the cubic cavity are plotted for comparison of the solution accuracy. Figure 6.28 shows the u velocity versus the z coordinate at $x = 0.5$, and Figure 6.29 shows the w velocity versus the x coordinate at $z = 0.5$. There is no significant difference between the three solutions in Figure 6.28, but the coarse grid gives a very poor solution for the w velocity in Figure 6.29.

6.6.2 Robustness and CPU Time Savings of Prismatic Adaptation

Robustness of the prismatic adapter is demonstrated in Figure 6.30. The maximum residual in the x -momentum equation is plotted versus time for the adapted and globally adapted cases. The large spike in Figure 6.30 occurs due to the linearly interpolated solution from the initial coarse grid nodes to the newly created ones due to adaptation. The flow solver converges relatively quickly after the restart to

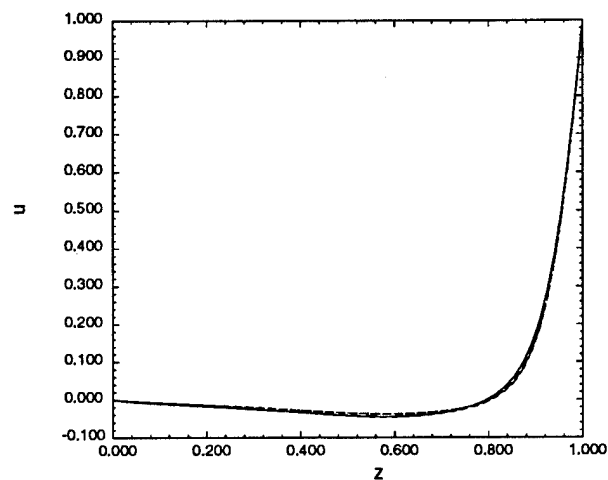


Figure 6.28: Adaptation comparison of the u velocity profile versus z on the symmetry plane at $x = 0.5$ for the cubic cavity at $Re = 400$.

— — Coarse grid - - - Locally adapted grid
 — Globally adapted grid

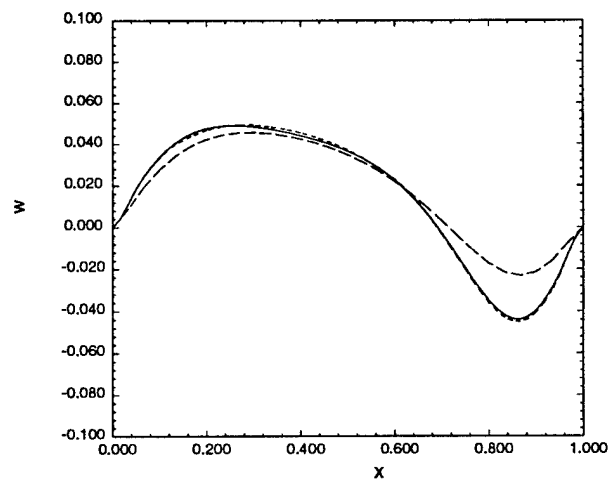


Figure 6.29: Adaptation comparison of the w velocity profile versus x on the symmetry plane at $z = 0.5$ for the cubic cavity at $Re = 400$.

— — Coarse grid - - - Locally adapted grid
 — Globally adapted grid

Table 6.7: Surface triangulation used for adaptation accuracy comparison for the cubic cavity at $Re = 400$.

Type of Grid	Number of Surface Faces	Total Number of Nodes
Coarse	994	27693
Locally Adapted	3068	82518
Globally Adapted	3976	106029

residual levels on the order of 10^{-5} .

Another criterion used to evaluate an algorithm is efficiency. Computing time to reach steady state using the locally adapted grid as well as the globally adapted grid are compared. The CPU time for the adapted grid consists of the CPU time for the initial 5000 iterations with the coarse grid, and the CPU time for the adapted grid for the final 5000 iterations. The globally adapted fine grid CPU time is for 10000 iterations which were required to reach steady state conditions. The maximum residuals in the x-momentum equation were on the order of 10^{-6} . The CPU time for the locally adapted grid and the globally adapted grid are shown in Table 6.8. The CPU time for the locally adapted grid case will vary depending on the time the coarse grid is adapted and the time the simulation is terminated. In this test case, there is more than 50% CPU time savings using the locally adapted grid for the same solution accuracy,

From these results, the following conclusions can be drawn:

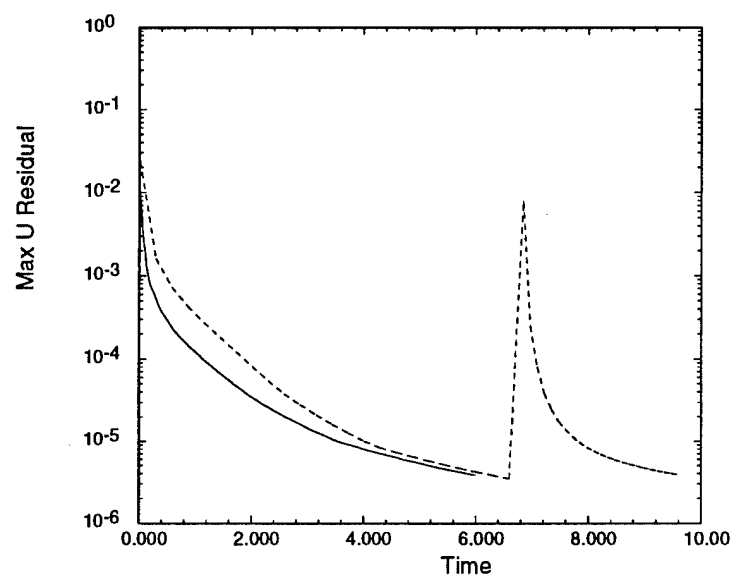


Figure 6.30: Demonstration of robustness of prismatic adapter, convergence of the maximum residual of the x-momentum equation for the cubic cavity at $Re = 400$.

- - - Locally adapted grid

— Globally adapted grid

Table 6.8: Adaptation CPU time comparison for the cubic cavity at $Re = 400$.

Type of Grid	Number of Cells	CPU Time (sec)
Locally Adapted	153400	31300
Globally Fine	198800	64100

1. The solver with locally adapted prismatic cells yields a stable solution.
2. The locally adapted prismatic mesh yields the same accuracy with reduced computing resources compared to a globally adapted prismatic mesh.

6.7 Equation Adaptation with a Hybrid Mesh for Flow Around a Sphere

Equation adaptation of the solver is evaluated for the case of flow over a sphere at $Re = 40$ using a hybrid mesh. Two cases are considered here. The first case solved the Euler equations in the tetrahedral region (“inviscid” tetrahedra), while the second case solved the Navier-Stokes equations in the tetrahedral region (“viscous” tetrahedra). The full Navier-Stokes equations are employed over the prism region for both cases. The grid parameters used for this study are shown in Table 6.9. Both simulations terminated after 6000 iterations with the maximum residuals in the x-momentum equation being on the order of 10^{-4} . The CFL factor is 0.1, and the momentum artificial dissipation parameter, σ_4 , is 10^{-4} . The pressure coefficient and vorticity are examined along the symmetry plane (equator) of the sphere in Figures 6.31 and 6.32. The angle theta is defined as zero at the front stagnation point and 180 degrees at the rear stagnation point of the sphere. It is observed that the solutions corresponding to the two cases are nearly identical. The CPU times for the two simulations are shown in Table 6.10. The initial case with “inviscid tetrahedra” requires about 60 % of the time required by the case that does not use equation adaptation. It should be noted that the hybrid (prismatic/tetrahedral) mesh makes equation adaptation simple to implement.

The effect of the interface between the prisms and the tetrahedra on the solution is examined next. Flow around the sphere at $Re = 100$ is considered. The prismatic region was reduced so that the separation bubble would no longer lie solely in the “viscous” prism region. Figure 6.33 shows the pathlines around the sphere. It is observed that the pathlines are smooth across the interface. Figures 6.34 and

Table 6.9: Grid parameters used for evaluation of equation adaptation for sphere flow at $Re=40$.

Number of Surface Faces	Number of Prismatic Layers	Radial Stretching Factor	Grid Spacing at Wall	Number of Prism Nodes	Total Number of Nodes	Number of Tetrahedral Cells
1346	27	1.1	0.01	19740	49195	160054

6.35 illustrate the velocity and pressure coefficient contours on the symmetry plane.

The contours are relatively smooth across the interface.

From these results, the following conclusions can be drawn:

1. The solver with equation adaptation and a hybrid (prism/tetrahedra) mesh yields stable solutions.
2. The solver with equation adaptation yields nearly the same results with reduced computing resources compared to the case with solves the Navier-Stokes equations over the entire mesh.
3. The change of topology of the grid (prism/tetrahedra interface) has little effect on the solution.

Table 6.10: Equation adaptation CPU time comparison for the sphere at $Re = 40$.

Type of Tetrahedra	CPU Time (sec)
"Inviscid"	29800
"Viscous"	49700

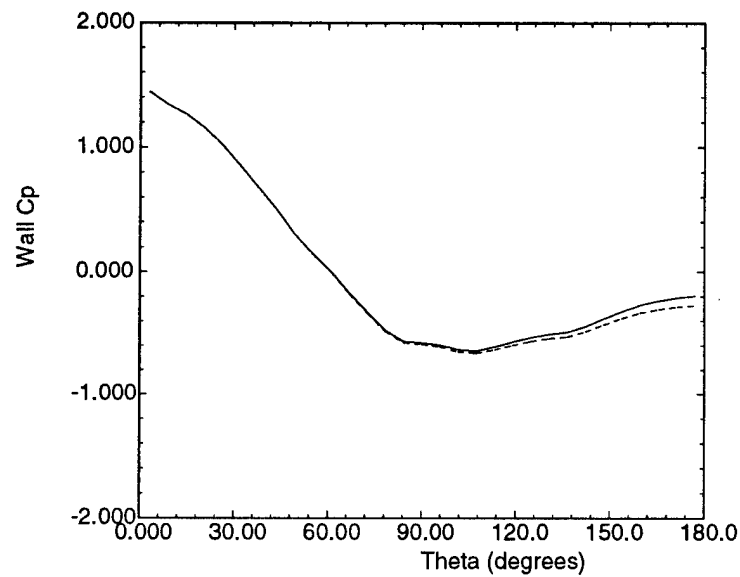


Figure 6.31: Evaluation of equation adaptation with hybrid mesh. Pressure coefficient distributions on the surface of the sphere for $Re=40$.

— "Inviscid" Tetrahedra
 - - - "Viscous" Tetrahedra

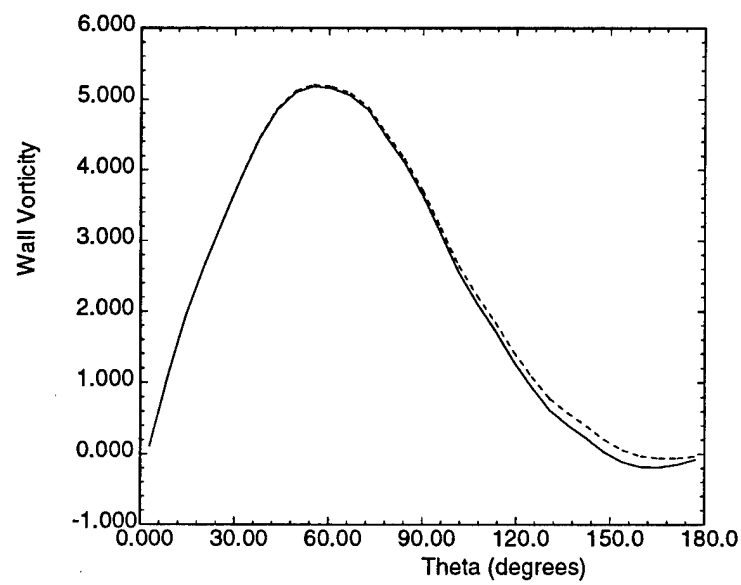


Figure 6.32: Evaluation of equation adaptation with hybrid mesh. Vorticity distributions on the surface of the sphere for $Re=40$.

— “Inviscid” Tetrahedra
- - - “Viscous” Tetrahedra

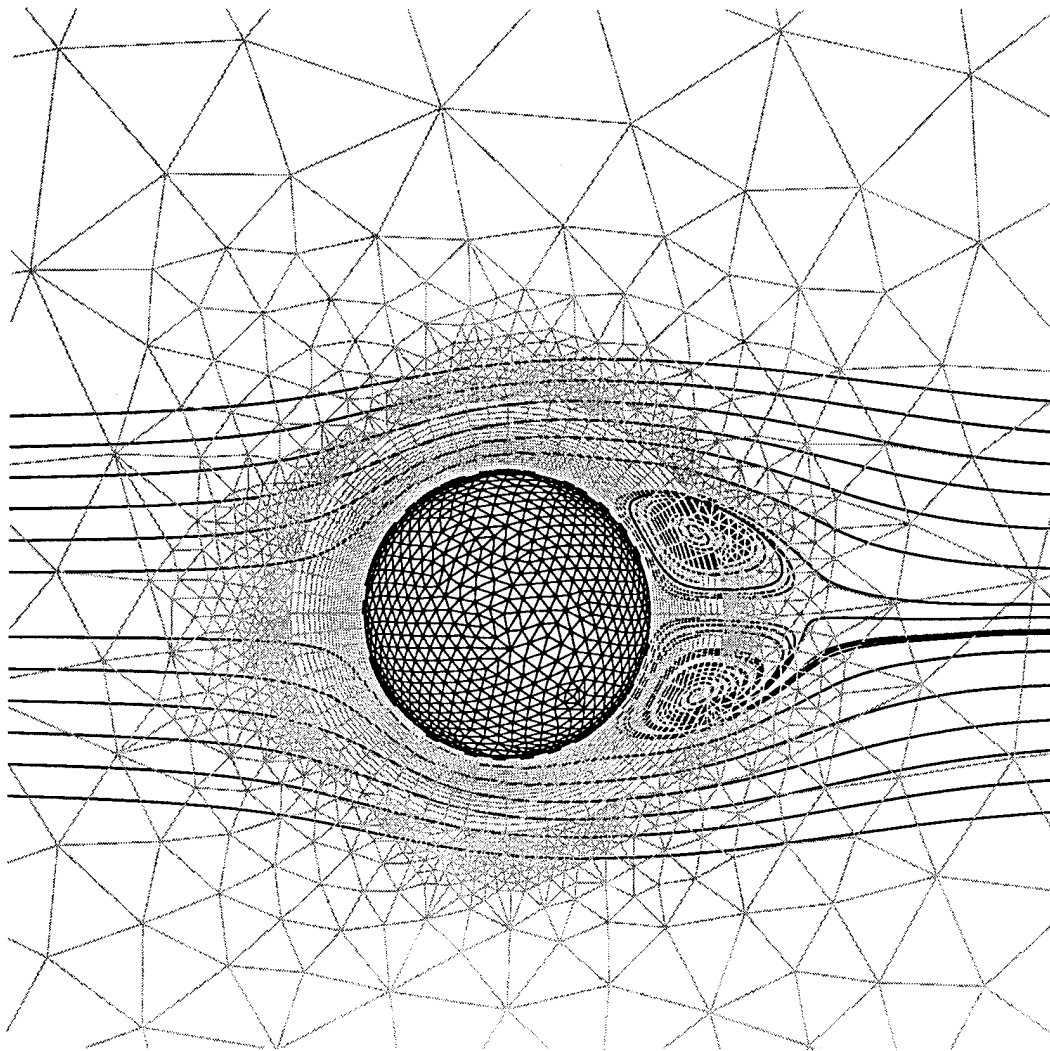


Figure 6.33: Effect of prism/tetrahedra interface on the solution. Pathlines near a sphere for $Re=100$ are unaffected by the change in topology of the grid.

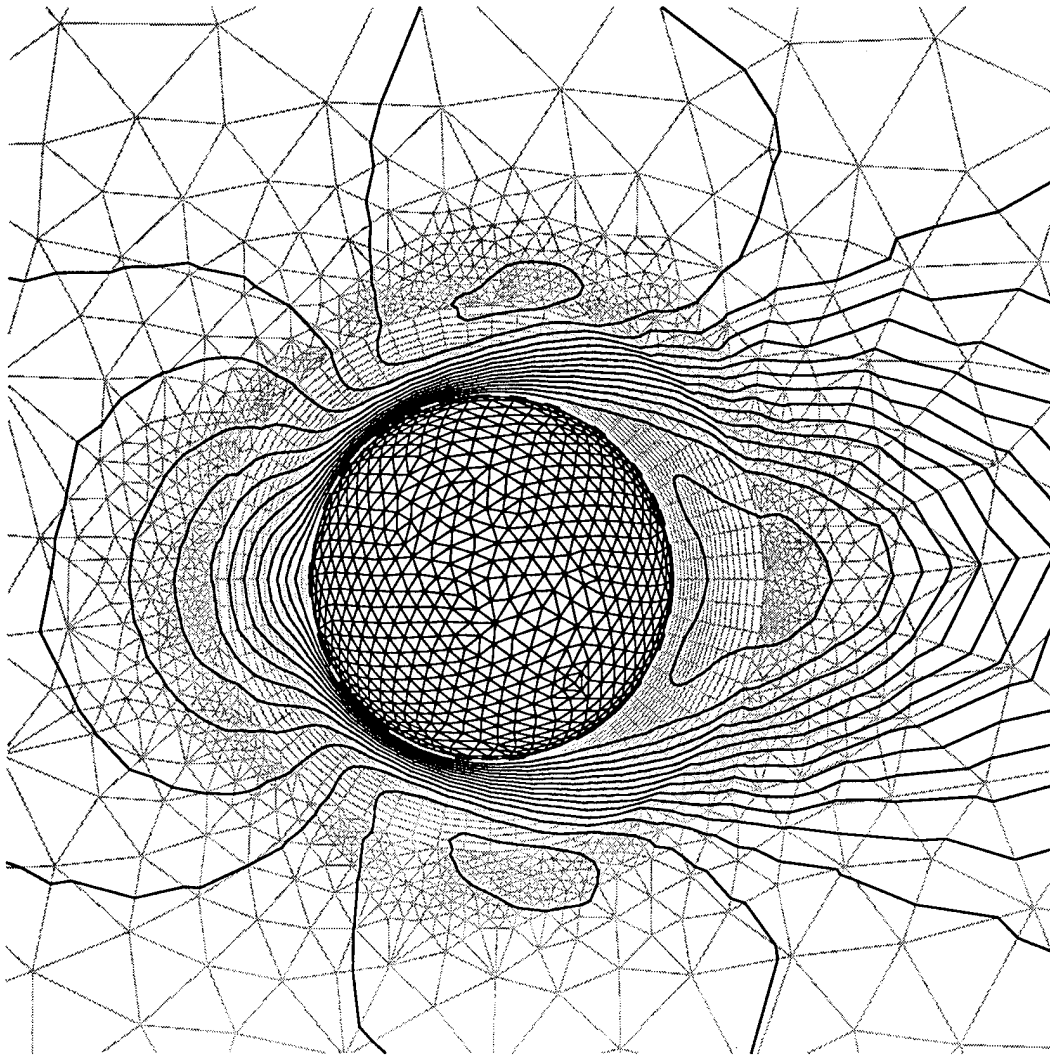


Figure 6.34: Effect of prism/tetrahedra interface on the solution. Velocity contours near a sphere for $Re=100$ are relatively unaffected by the change in topology of the grid.

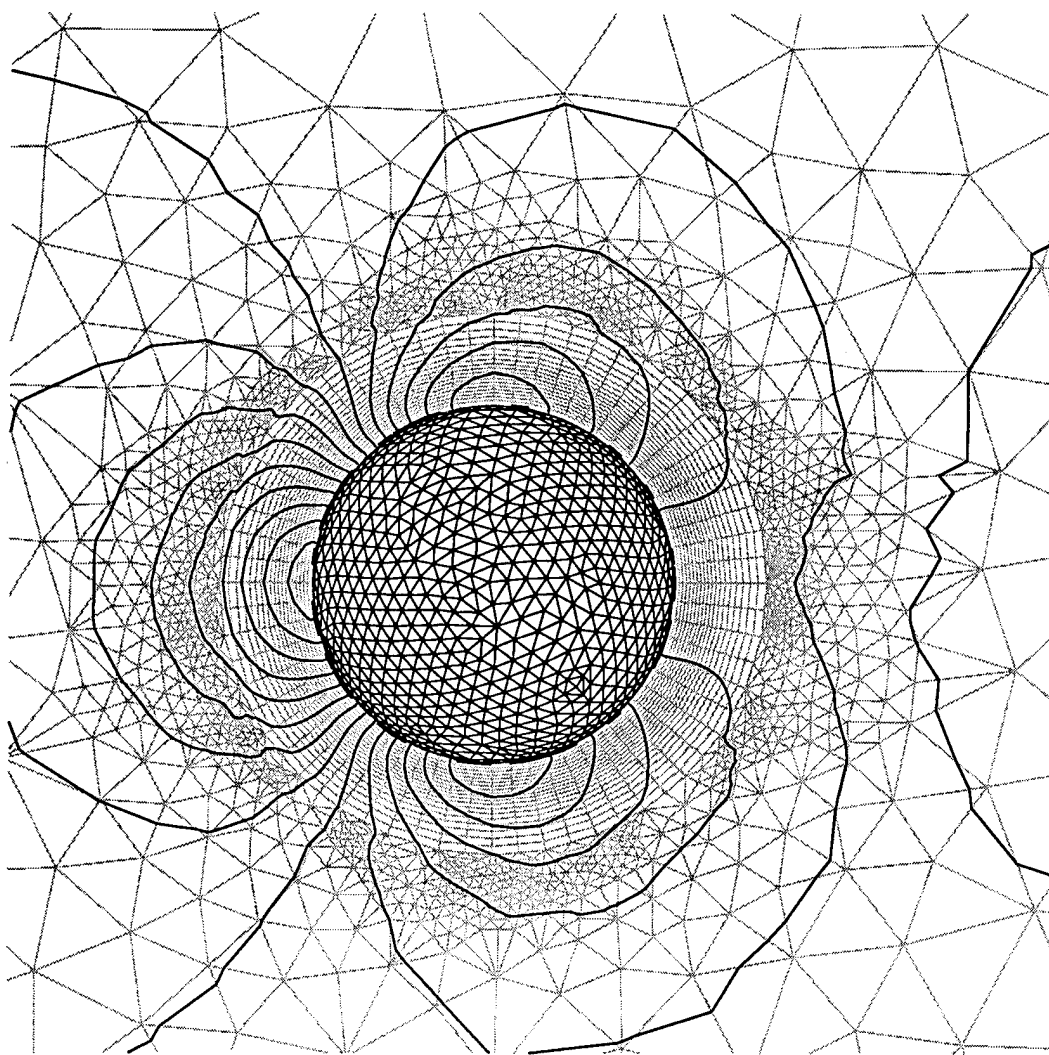


Figure 6.35: Effect of prism/tetrahedra interface on the solution. Pressure contours near a sphere for $Re=100$ are relatively unaffected by the change in topology of the grid.

6.8 Hybrid Mesh Adaptation Applied to Flow Around a Sphere

Steady flow over a sphere is used to demonstrate the advantages of hybrid mesh adaptation. The hybrid mesh consists of “viscous” prisms near the surface of the sphere, and “inviscid” tetrahedra filling the rest of the computational domain. The robustness of the scheme will also be presented.

6.8.1 Accuracy of Hybrid Adaptation on Sphere Flow

Steady flow over a sphere at a Reynolds number of 100 is used to demonstrate the accuracy of hybrid mesh adaptation. The parameters of the coarse, locally adapted, and globally fine grids used for this study are shown in Table 6.11. The globally fine mesh was created by globally refining the prismatic region of the coarse grid. Refinement occurred on portions of the tetrahedral mesh due to the refinement of the prisms. In the prism region, all three grids had 36 layers, with the grid step size at the wall being 0.01, and a stretching factor of 1.1. The artificial dissipation factor, σ_4 , was equal to 10^{-4} and the CFL factor was 0.1.

Wall pressure coefficient and vorticity are shown in Figures 6.36 and 6.37. The pressure coefficient solution for the adapted mesh and the fine mesh are nearly identical. However, the vorticity distribution corresponding to the adapted and fine meshes have some differences. The two solutions agree over the rear portion of the sphere, while differences are observed near $\theta = 60$ degrees. An examination of the adapted mesh shows the rear of the sphere is adapted, while the region near $\theta = 60$ degrees was not adapted. At low Reynolds numbers, the sphere actually requires a globally fine mesh. However, this comparison of adapted and fine grids still shows

the benefit of adaptation.

The adapted grid is shown in Figure 6.39 along with pathlines around the sphere. The figure shows the tessellation on the wall surface and on the symmetry plane. It is clearly seen that embedding in the tetrahedral region is focused at the rear of the sphere. The prismatic region is also directionally refined near the upstream and downstream sections of the body. This is due to the flow accelerating from the upstream stagnation point and the flow downstream separating causing significant flow gradients in the lateral directions which are detected by the directional adaptive algorithm. Pathlines show the ring eddy formed at the rear of the sphere, and the cells in the tetrahedral region of the hybrid mesh. Figure 6.40 shows a different view of the adapted hybrid mesh along with velocity contours. Three surfaces are shown. The first is the sphere surface, the second is the symmetry plane, and the third is a plane cutting through the interior of the grid, normal to the symmetry plane. Despite the drastic changes in topology at the interface, the contours lines across it are relatively smooth. Figure 6.41 shows the pressure coefficient contours on the symmetry plane. The pressure field is relatively smooth as well as the contour lines across the prismatic/tetrahedral interface.

6.8.2 Robustness and CPU Time Savings of Hybrid Adaptation

Robustness of the hybrid adapter is demonstrated in Figure 6.38. The maximum residual of the x-momentum equation is plotted versus time for the locally adapted and globally adapted cases. The large spike in Figure 6.38 is due to the linearly interpolated solution from the initial coarse grid nodes to the newly created ones due to adaptation. The flow solver converges relatively quickly after the restart to residual levels on the order of 10^{-4} .

Another criteria used to evaluate an algorithm is efficiency. The CPU time to reach steady state for the locally adapted grid and the globally adapted grid are compared in Table 6.12. The CPU time for the locally adapted grid case will vary depending on the time the coarse grid is adapted and the time the simulation is terminated. For this case, there is more than 40 % CPU time savings using the locally adapted grid.

From these results, the following conclusions can be drawn:

1. The solver with hybrid adaptation yields stable solutions.
2. The locally adapted hybrid mesh yields approximately the same results with reduced computing resources compared to a fine mesh.
3. The incompressible pressure correction method with non-staggered grids yields smooth pressure and velocity fields.
4. The velocity field and flow pathlines are unaffected by the change in grid topology at the prism and tetrahedra interface.

Table 6.11: Grid parameters used for hybrid mesh adaptation for sphere flow at $Re=100$.

Type of Grid	Number of Surface Faces	Total Number of Nodes	Number of Prism Cells	Number of Tetrahedral Cells
Coarse	332	13372	11952	35063
Locally Adapted	1095	31981	39420	59401
Fine	1328	36495	47808	59796

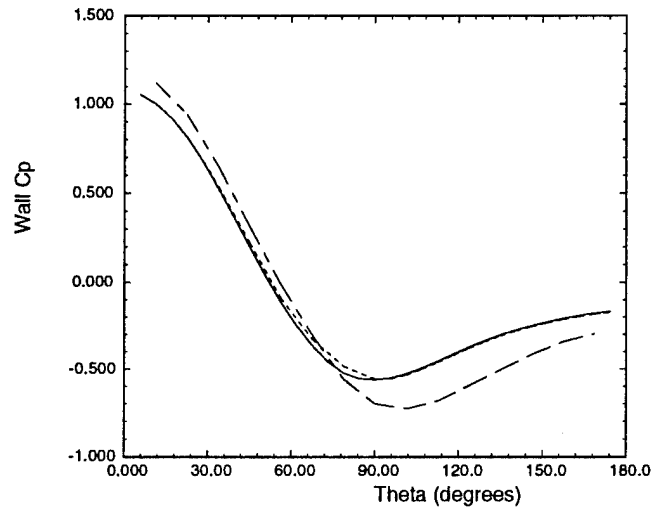


Figure 6.36: Pressure coefficient distribution on the surface of a sphere with a hybrid adapted mesh for $Re=100$.

- - - Coarse grid - - - Locally adapted grid
 — Fine grid (globally adapted prism region)

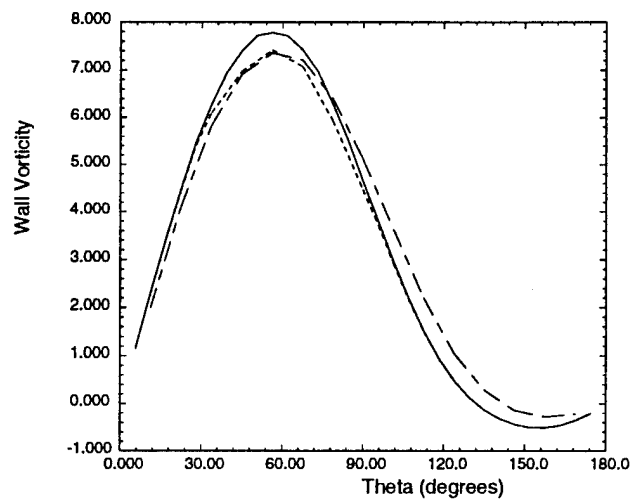


Figure 6.37: Vorticity distribution on the surface of a sphere with a hybrid adapted mesh for $Re=100$.

- - - Coarse grid - - - Locally adapted grid
 — Fine grid (globally adapted prism region)

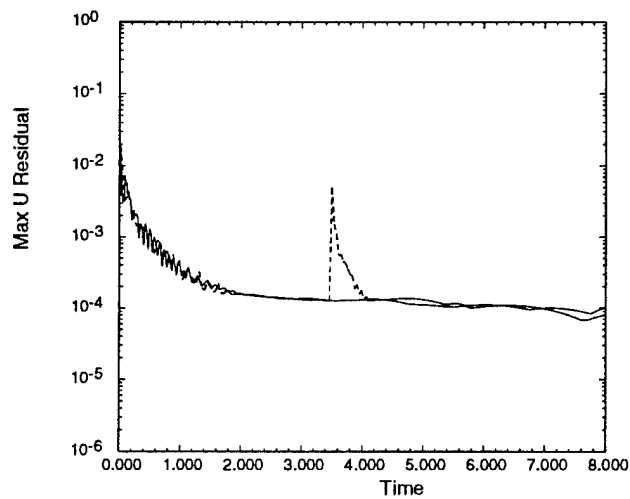


Figure 6.38: Demonstration of robustness of hybrid adapter, maximum residual in the x-momentum equation versus time for the sphere at $Re = 100$.

- - - Locally adapted grid — Globally adapted grid

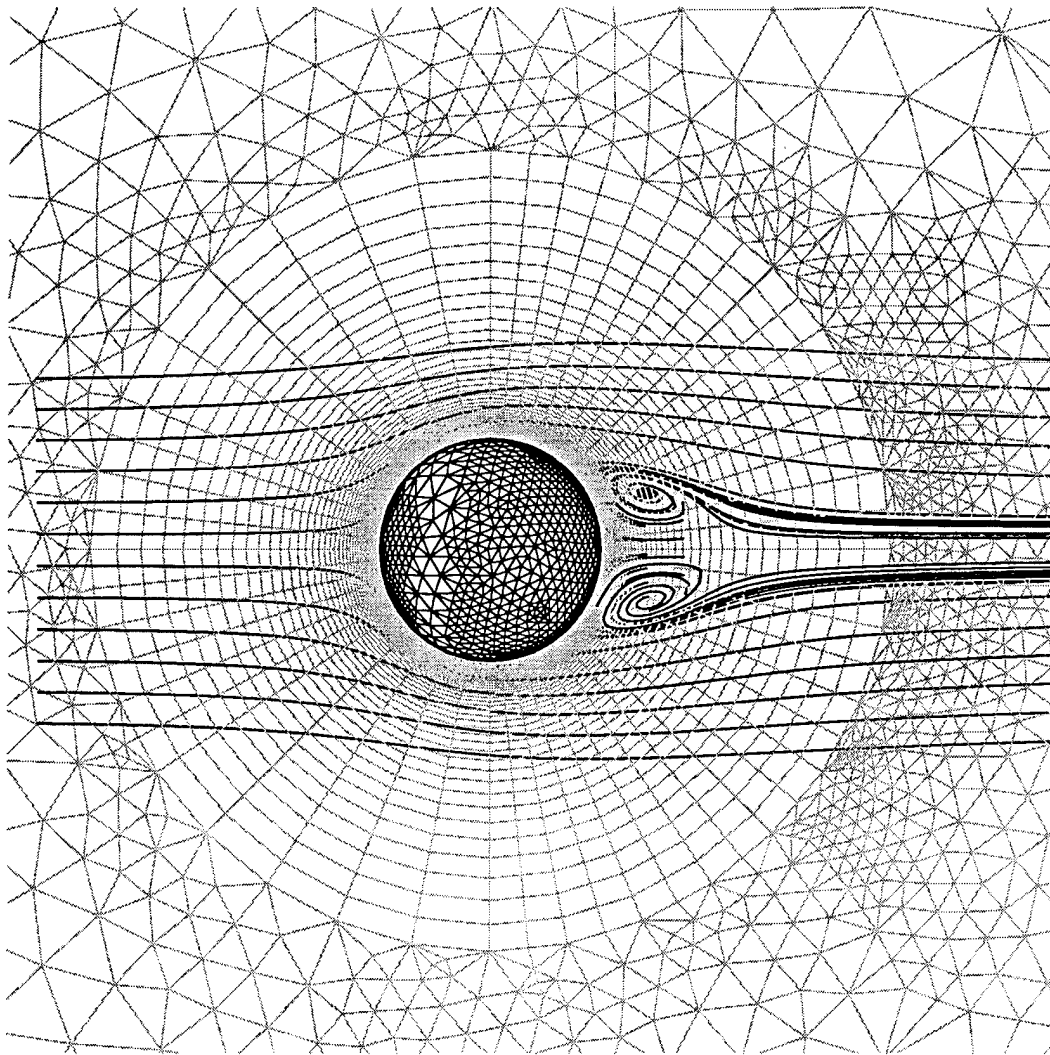


Figure 6.39: Adapted hybrid mesh and flow pathlines for sphere at $Re=100$. A view of the tessellation on the wall surface and symmetry plane. The hybrid grid is embedded isotropically in the tetrahedral region and directionally in the prismatic region.

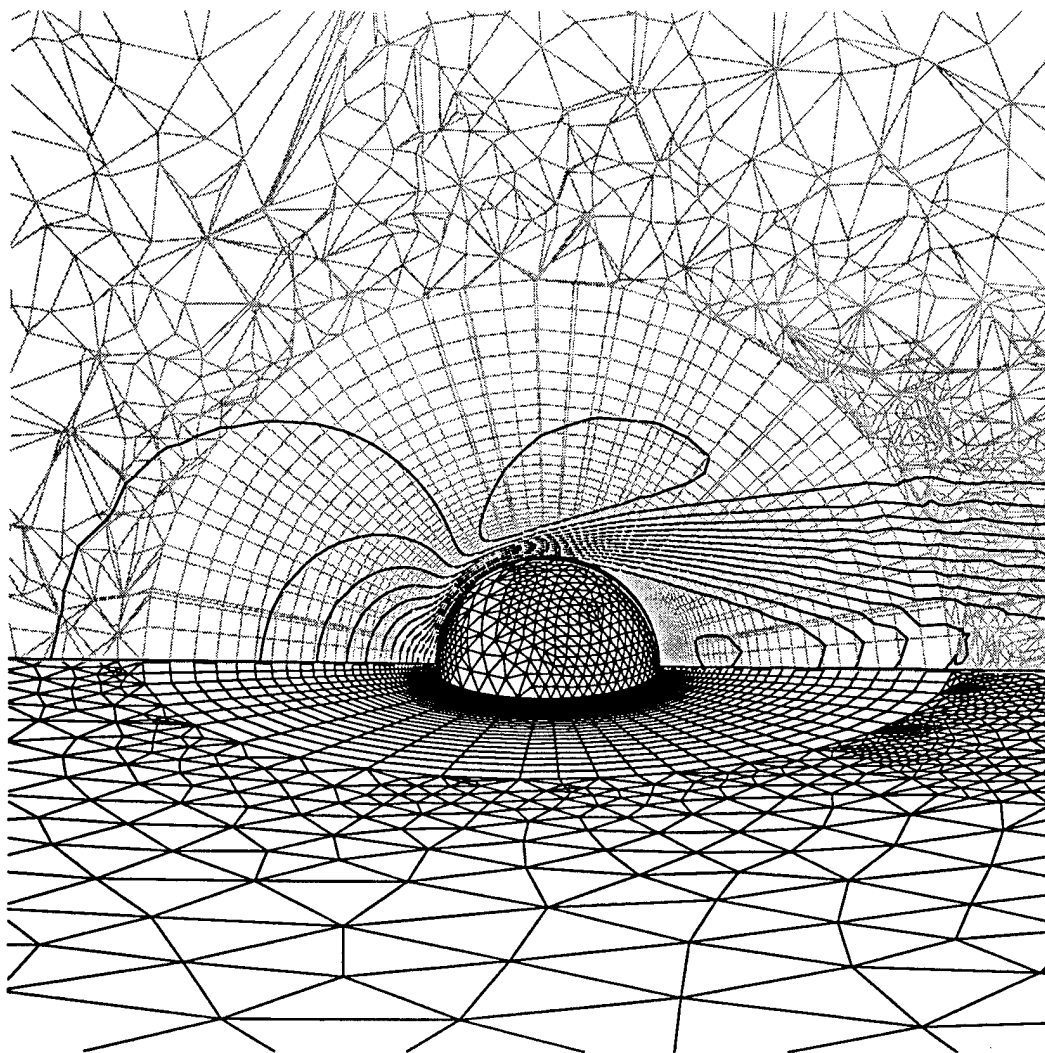


Figure 6.40: Adapted hybrid mesh and corresponding flow velocity contours for sphere at $Re=100$. A view of the tessellation on the wall surface, symmetry plane and an interior equatorial plane. The hybrid grid is embedded isotropically in the tetrahedral region and directionally in the prismatic region.

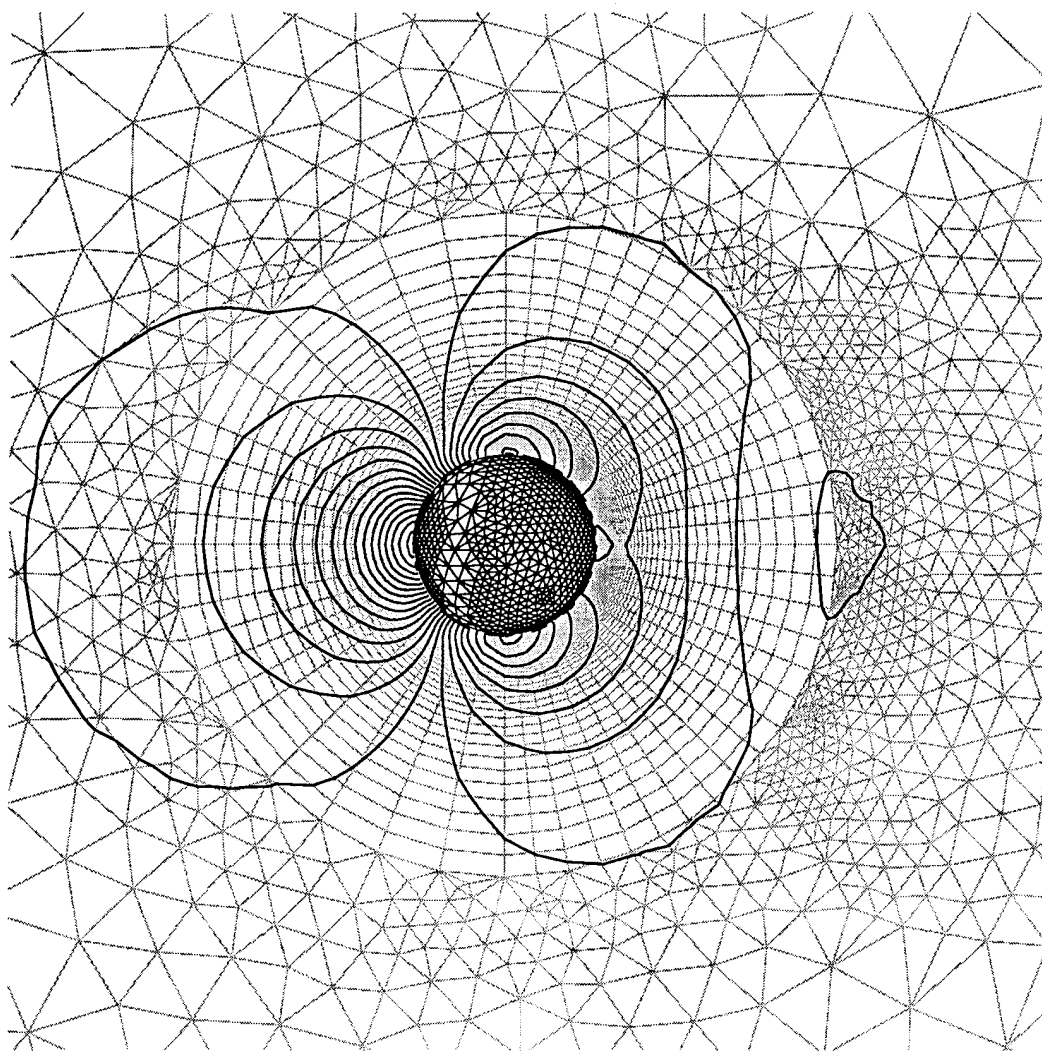


Figure 6.41: Adapted hybrid mesh and pressure coefficient contours for sphere at $Re=100$. A view of the tessellation on the wall surface and symmetry plane. The hybrid grid is embedded isotropically in the tetrahedral region and directionally in the prismatic region.

Table 6.12: Adaptation CPU comparison for the hybrid sphere mesh at $Re = 100$.

Type of Grid	Number of Nodes	CPU Time (sec)
Locally Adapted	31981	11510
Globally Fine	36495	19720

6.9 Hybrid Mesh Adaptation for Flow Around Tandem Spheres

Hybrid mesh adaptation is used to simulate the steady flow around two spheres in a tandem (in-line) arrangement. A symmetry plane is used to model half of the flow domain at $Re = 100$. The grid parameters are shown in Table 6.13. The grid had a step size at the wall of 0.01 with a stretching factor of 1.1. The surface triangulation and symmetry plane of the hybrid grid are shown in Figure 6.42. The prismatic region shows directional adaptation on the fore and aft portions of both spheres due to the flow accelerating from the upstream stagnation point and the flow downstream separating or decelerating. Embedding in the tetrahedral region is focused at the rear of the second sphere as well as the region between the two spheres. The artificial dissipation factor, σ_4 , was equal to 10^{-5} , the CFL factor was 0.05, and the maximum residual in the x-momentum equation was on the order of 10^{-5} .

The drag coefficient on the first sphere was 1.26, while the drag coefficient on the second sphere was 0.8. The drag coefficient for a isolated sphere is approximately 1.1. The first sphere apparently “shields” the second sphere from the freestream flow, resulting in lower drag on the second sphere. This shielding phenomenon was seen experimentally [142] for tandem cylinders in oscillating flows. The presence of the second sphere inhibits the formation of the vortex aft of the first sphere, although the flow in this region does appear to be separated. This can be seen when examining the pathlines shown in Figure 6.42. The shielding of the second sphere by the first sphere results in the flow around the second sphere to be representative of lower Reynolds number flow. In fact, the flow past the second sphere does not separate

Table 6.13: Adapted hybrid grid parameters used for tandem spheres flow at $Re=100$.

Number of Surface Faces	Number of Prism Layers	Number of Prism Nodes	Number of Tetrahedra Nodes	Number of Prism Cells	Number of Tetrahedra Cells
3470	10	18070	74309	34700	413938

when it reaches the aft portion of the sphere. A closeup view of the region between the two spheres is shown in Figure 6.43. The surface triangulation of the fore and aft spheres are shown on the left and right edges of the figure. The quadrilateral faces of the prismatic region are shown as well as the triangular faces of the tetrahedral cells on the symmetry plane. The velocity vectors in this region are shown and the flow is separated aft of the first sphere. To simplify the figure, only velocity vectors with magnitudes less than 0.015 are shown. A different view of the adapted hybrid mesh, as well as of the flow velocity contours are shown in Figure 6.44.

From these results, the following conclusions can be drawn:

1. The hybrid mesh is suitable to cover multiple three-dimensional bodies.
2. The adapted hybrid mesh yields a stable solution for this two-body problem.
3. The presence of the second sphere inhibits the formation of the vortex aft of the first sphere.
4. The presence of the first sphere "shields" the second sphere from the main flow resulting in reduced drag.

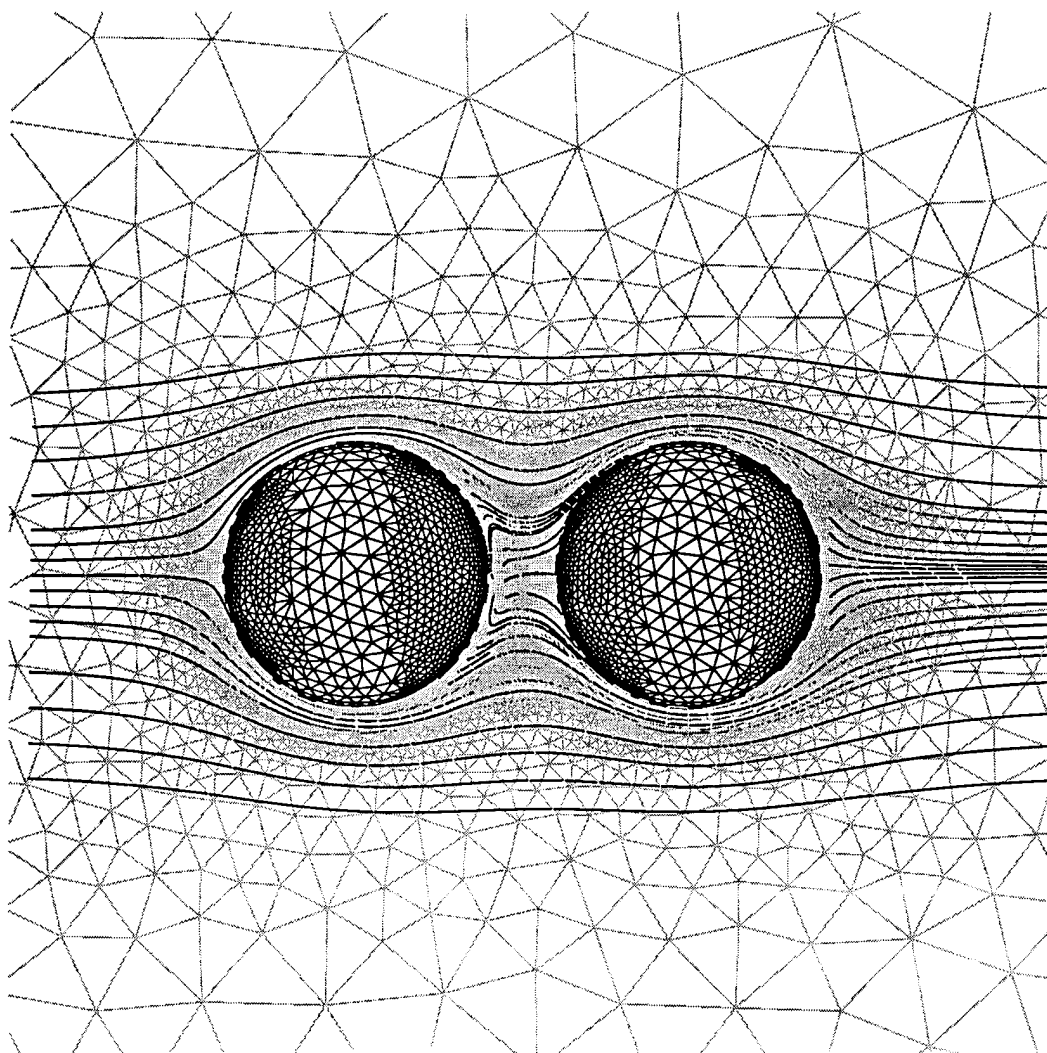


Figure 6.42: Adapted hybrid grid and flow pathlines for tandem spheres at $Re=100$. A view of the tessellation on the wall surface and symmetry plane. The hybrid grid is embedded isotropically in the tetrahedral region and directionally in the prismatic region.

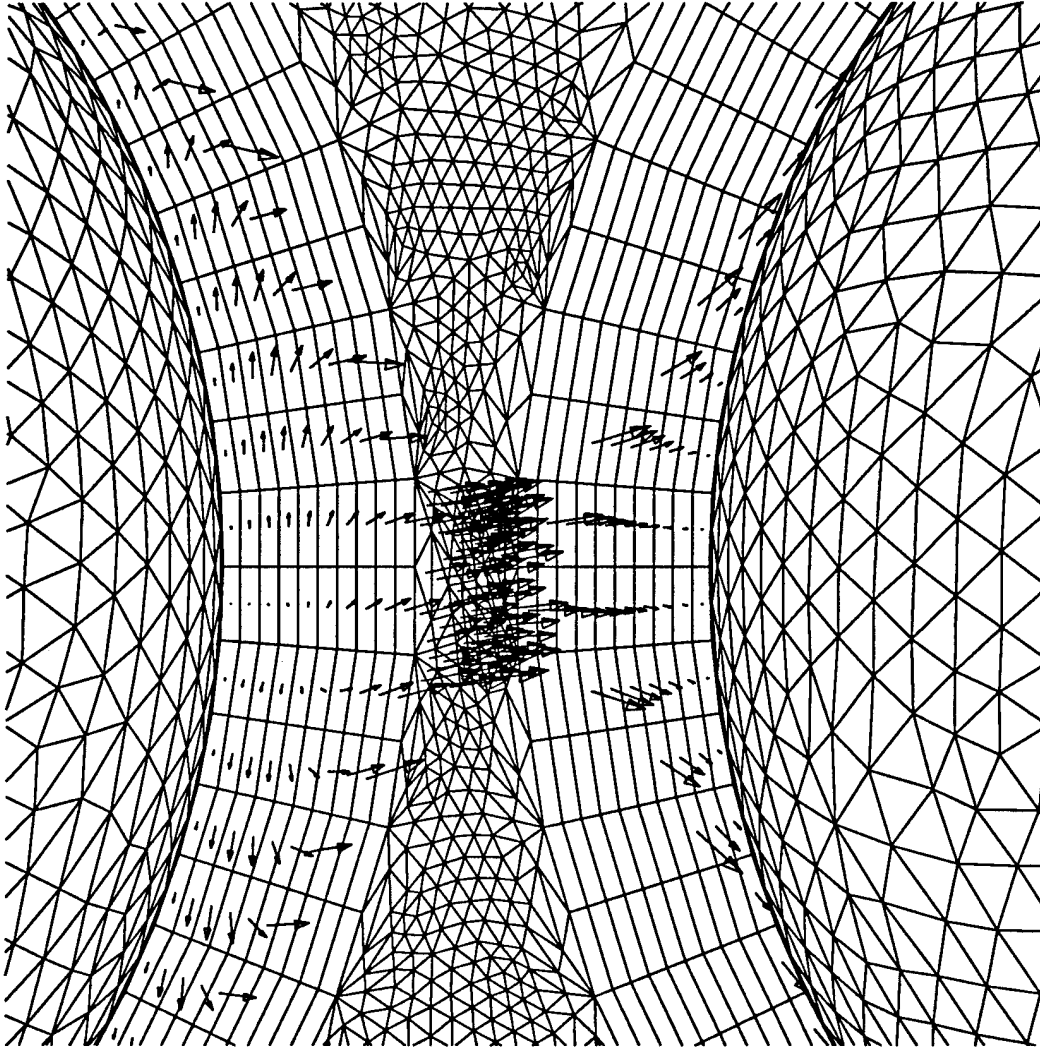


Figure 6.43: Velocity vectors on the symmetry plane between tandem spheres at $Re=100$. The flow is separated aft of the first sphere. The closeup view shows a portion of the surface mesh of the fore and aft spheres, the quadrilateral faces of the prismatic cells on the symmetry plane, and the triangular faces of the tetrahedral cells on the symmetry plane. Only the velocity vectors with magnitudes less than 0.015 are shown.

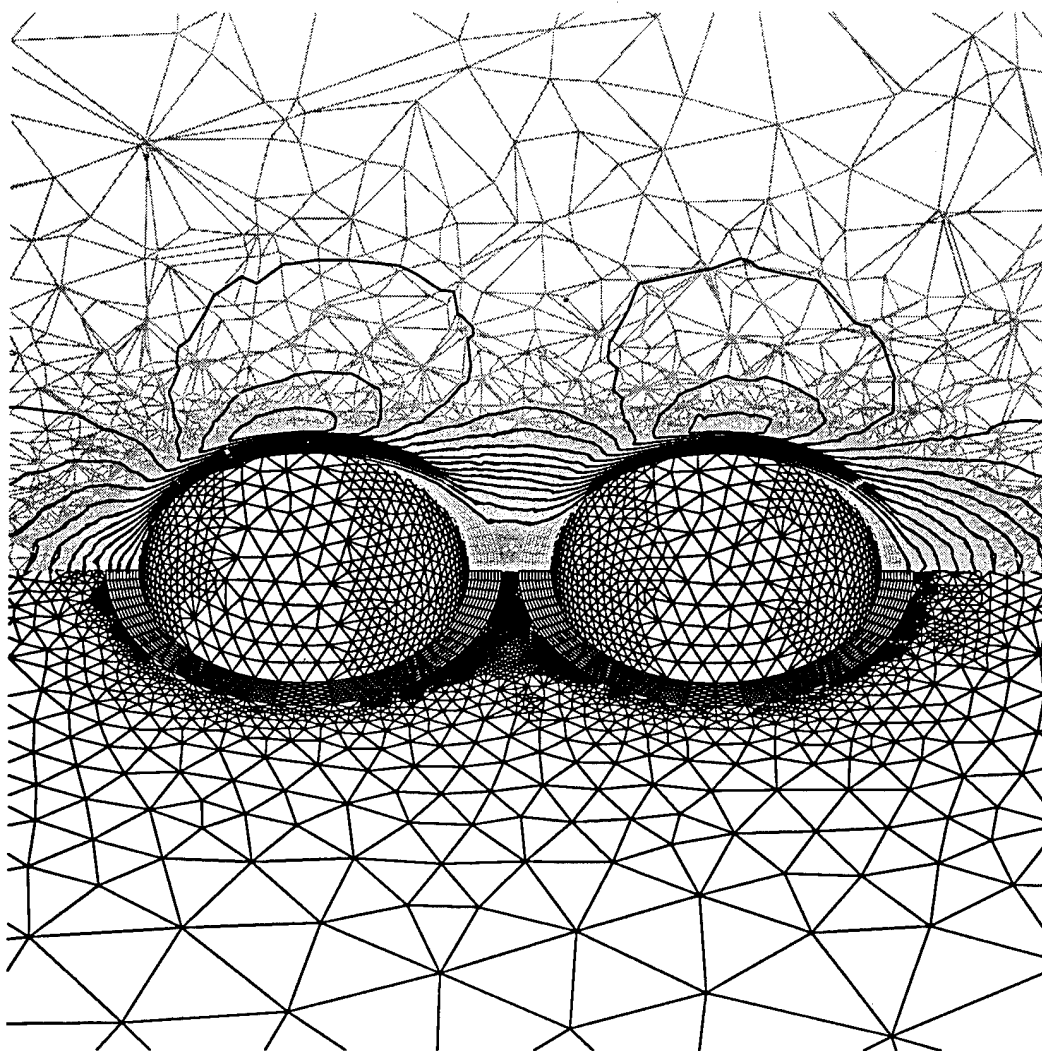


Figure 6.44: Adapted hybrid grid and flow velocity contours for tandem spheres at $Re=100$. A view of the tessellation on the wall surface, symmetry plane and an interior equatorial plane. The hybrid grid is embedded isotropically in the tetrahedral region and directionally in the prismatic region.

Chapter 7

Concluding Remarks

7.1 Summary

The main goal of the present work was to develop an adaptive numerical algorithm for the simulations of three-dimensional viscous incompressible flows. The method should be stable at high Reynolds numbers, accurate, and flexible to use with combined prismatic and tetrahedral grids.

A hybrid adaptive solver for three-dimensional incompressible viscous flow simulations was developed and validated. The hybrid grid, comprising of prisms and tetrahedra, proved to be appropriate for these computations. Equation adaptation resulted in appreciable reduction in computing time. The interface between the prisms and the tetrahedra did not affect accuracy. The semi-structured nature of the prismatic grids reduced the memory requirements considerably as opposed to using a fully unstructured grid. A dual grid adaptation algorithm was implemented in conjunction with the solver. The algorithm enabled the solver to obtain accurate results adaptively using relatively fewer grid nodes.

7.2 Contributions

The contributions of the present work can be categorized into three main areas. These are integration scheme, unstructured mesh, and adaptation.

1. A time accurate finite volume/finite element integration scheme for solving the incompressible Navier-Stokes equations on hybrid grids was developed. An important feature of the integration scheme is the strategy of performing all the operations on an edge-wise basis which makes it computationally efficient, especially for hybrid grids which consist of different types of elements. This is due to the fact that performing the operations on an edge-wise basis enables distinction between different types of cells to be ignored and hence makes the scheme very general.
2. The solver has been developed using non-staggered grids rather than the classical staggered grid approach. Artificial dissipation used in conjunction with non-staggered grids produced non-oscillatory solutions.
3. This is the first incompressible flow method with prismatic and hybrid grids.
4. An equation adaptation scheme was coupled with the hybrid mesh such that the Navier-Stokes equations are solved over the prism region and the simpler Euler equations are employed over the tetrahedral region.
5. This is the first incompressible flow method with hybrid adaptation.

7.3 Conclusions

The following conclusions can be drawn from the present work.

1. *The incompressible three-dimensional Navier-Stokes solver is relatively accurate on stretched and skewed unstructured grids. The scheme is also time accurate.* Comparisons were made with existing experimental observations where possible, and other numerical observations in other cases. Surface pressure and vorticity distributions compared well with the literature, as well as steady state sphere drag results. Simulation of impulsive start of flow around a cylinder demonstrated time accuracy.
2. *The scheme is stable at high Reynolds numbers.* Stability at high Reynolds numbers is a concern for all Navier-Stokes solvers. Flow over a flat plate at a high Reynolds number was completed to demonstrate stability of the solver.
3. *The upwind-like artificial dissipation model was effective in eliminating the odd-even modes in the solution.* Pressure correction methods using non-staggered grids are noted for odd-even modes arising in the solution. The fourth order artificial dissipation successfully suppressed these modes.
4. *Use of equation adaptation, viscous prisms and inviscid tetrahedra, can yield accuracy that is comparable to solving the Navier-Stokes equations over the entire domain.* Equation adaptation was demonstrated on a hybrid sphere mesh. Solutions from both grids were nearly identical.
5. *Use of equation adaptation, viscous prisms and inviscid tetrahedra, provides efficiency by yielding substantial CPU time and memory savings compared to solving the Navier-Stokes equations over the entire domain.* Equation adaptation was demonstrated on a hybrid sphere mesh. The CPU time of the viscous tetrahedra computation was 67 % greater than the analogous computation with inviscid tetrahedra. This is a significant savings in computation

time. A further advantage to inviscid tetrahedra is the memory saved by not having to specify extra pointers and area projections of the tetrahedral faces which are needed for viscous calculations.

6. *Use of spatial adaptation can yield accuracy that is comparable to the equivalent globally fine mesh.* Prismatic spatial adaptation was demonstrated with a three-dimensional cavity flow, while hybrid adaptation was demonstrated with sphere flow. Both test cases illustrated the locally adapted and globally fine meshes produced similar results.
7. *Use of spatial adaptation provides efficiency by yielding substantial CPU time savings relative to the equivalent globally fine mesh.* Prismatic spatial adaptation was demonstrated with a three-dimensional cavity flow, while hybrid adaptation was demonstrated with sphere flow. Both test cases illustrated the adapted mesh produced the same solution in significantly less CPU time than the globally fine mesh. Memory requirements are also reduced when there are fewer nodes or cells.
8. *The developed Navier-Stokes scheme is robust. However, it could be prohibitively expensive for periodic time-accurate computations.* The scheme converges well for the stretched and skewed unstructured grids as evidenced by the test cases on adapted and hybrid meshes. However, a limitation of the scheme lies in the small CFL factor necessary for stability of this explicit time marching scheme.
9. *The Poisson matrix is memory intensive.* Unstructured grids created around complex geometries frequently contain nodes with a large number of neighboring points. This causes a substantial increase in memory requirements for

the storage of the matrix coefficients of the Poisson equation.

7.4 Recommendations for Further Research

Based on the present work on simulating viscous incompressible flows, the following recommendations are given for future development.

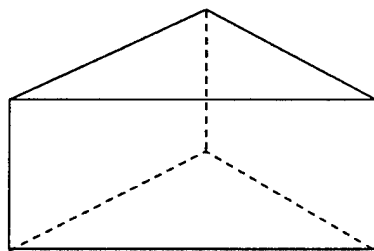
1. Reduction in computation time would allow the scheme to solve a wider range of steady and unsteady problems at higher Reynolds numbers.
 - (a) With explicit time marching, the time step is restricted by the CFL condition. Implicit schemes are not restricted by the CFL condition and larger time steps can be employed. A semi-implicit scheme may also remove the severe CFL limitation.
 - (b) A comprehensive comparison of several matrix solvers (ICCG, GMRES) may produce substantial savings in computation time.
 - (c) Application of a multigrid technique for the Poisson equation could be used to accelerate the convergence.
 - (d) Computing requirements may be reduced by performing several momentum time steps before correcting the velocity back to a divergence free state by solving the pressure Poisson equation.
 - (e) Application of the solver on parallel computers would decrease overall time to reach a solution.
 - (f) Local time-stepping would allow larger time steps and may produce faster convergence for steady flow computations.

2. A tradeoff between memory and CPU time can be examined for the generation and solution of the Poisson equation. Currently the Poisson matrix entries are calculated initially and stored in arrays. Since the finite element generation of the Poisson matrix is relatively simple, calculating the matrix coefficients in the iterative solver procedure will consume more CPU, but will reduce memory requirements permitting flow simulations about larger and more complex geometries.
3. Addition of a turbulence model will permit the application of the method to higher Reynolds numbers flows.
4. The non-zero discrete divergence issue needs further investigation.

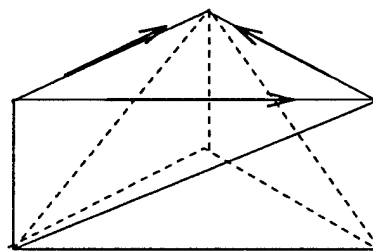
Appendix A

Division of Prism Cells into Tetrahedra Cells

Each prism is divided into three tetrahedral elements using the method of [34]. To have a conservative scheme, the tetrahedral cells created from the prismatic cells must be consistent with neighboring tetrahedral cells. To ensure consistency, the method of [34] assigns a split orientation to each quadrilateral face of the prism, as shown in Figure A.1. These orientations are represented by arrows on the upper triangular face. There are eight possible orientations of the arrows, but two of those



Prism Cell



Split Orientation

Figure A.1: Prism cell division showing split orientation method.

orientations do not result in division of the prism. These orientations are when the arrows are joined nose to tail in a clockwise or counter-clockwise direction around the triangular face. The division of prisms must be completed consistently and the two non-permissible orientations must be avoided. The scheme developed by [34] is described below:

1. For each node, find all the edges containing the node. Assign an orientation to all these edges. This direction points to the current node.
2. If an orientation has been assigned, take no action.

This scheme will guarantee only permissible orientations are assigned. For the present work, the first layer of prisms are divided using this split orientation method. Each layer of prisms repeats the pattern of division developed for the first layer.

Appendix B

Data Structures for Data Arrays

B.1 Primitive Variable Pointers

Primitive state variables and pressure values at the grid points are represented by the following pointers.

$U(node, i), i=1,3$	state variables corresponding to x, y, and z momentum equations
$U(node, i), i=4,6$	change-in-time of the state variables corresponding to x, y, and z momentum equations
$P(node)$	pressure
$DTNOD(node)$	time step
$VAR_N(node)$	value of the u state variable at the previous time step

Arrays for the pressure correction portion of the solver are extensive. The Poisson matrix is stored in five arrays, one array holds the diagonal terms, a second array holds the terms in the lower diagonal belonging to nodes in the prism region, and a third array holds the terms in the upper diagonal belonging to nodes in the

prism region. Two more arrays hold the terms in the lower and upper diagonal belonging to nodes in the tetrahedral and interface region. Although the Gauss-Seidel iterative scheme does not require this data architecture, it would permit easy transition to a more advanced matrix solver. Additional arrays are necessary to hold the neighboring node information for the matrix. The divergence of the velocity and the solution vector are additional arrays. The splitting of the arrays into prism and tetrahedral regions was used to save on memory storage. The number of neighboring nodes for a node in the prism region is usually less than 20. However, due to the unstructuredness of the tetrahedral region, the number of neighboring nodes for a node in the tetrahedral region can be larger than 80. The 2-D arrays for nodes in the prism region can be dimensioned much smaller than the corresponding 2-D arrays in the tetrahedral region.

DIAG(<i>node</i>)	diagonal terms of the Poisson equation matrix
PHI(<i>node</i>)	solution vector of the Poisson equation
DIVNODE(<i>node</i>)	divergence of the velocity vector.
INUM_U(<i>node</i>)	quantity of neighbor nodes of primary node. These neighbor nodes are greater than the primary node
INUM_L(<i>node</i>)	quantity of neighbor nodes of primary node. These neighbor nodes are lower than the primary node

$\text{INDEX_U}(i, \text{node}_{\text{prism}})$	node neighbor number of primary node in prism region. These neighbor nodes are greater than the primary node. The index i must be equal to or less than the number in $\text{INUM_U}(\text{node})$
$\text{INDEX_UT}(i, \text{node}_{\text{tet}})$	similar to $\text{INDEX_U}(i, \text{node})$, except primary node is in tetrahedral region.
$\text{INDEX_L}(i, \text{node}_{\text{prism}})$	node neighbor number of primary node in prism region. These neighbor nodes are lower than the primary node. The index i must be equal to or less than the number in $\text{INUM_L}(\text{node})$
$\text{INDEX_LT}(i, \text{node}_{\text{tet}})$	similar to $\text{INDEX_L}(i, \text{node})$, except primary node is in tetrahedral region.
$\text{AICCG_U}(i, \text{node}_{\text{prism}})$	Poisson matrix entry of that neighbor node stored in the $\text{INDEX_U}(i, \text{node})$ location for the primary node in prism region.
$\text{AICCG_UT}(i, \text{node}_{\text{tet}})$	similar to $\text{AICCG_U}(i, \text{node})$, except primary node is in tetrahedral region.
$\text{AICCG_L}(i, \text{node}_{\text{prism}})$	Poisson matrix entry of that neighbor node stored in the $\text{INDEX_L}(i, \text{node})$ location for the primary node in prism region.
$\text{AICCG_LT}(i, \text{node}_{\text{tet}})$	similar to $\text{AICCG_L}(i, \text{node})$, except primary node is in tetrahedral region.

B.2 Node Pointers

Node pointers relate properties associated with each node in the mesh. This pointer is oblivious to whether the node lies in the prismatic or tetrahedral region.

$XYZ(0:node, i), i=1,3$	xyz coordinates of each node. $i=1$ refers to the x coordinate, $i=2$ refers to the y coordinate, and $i=3$ refers to the z coordinate
$VOLNOD(0:node)$	dual cell volume of each node
$ABSAREA_DUAL_NOD(i, 0:node), i=1,3$	absolute value of the xyz area face projections for the dual cell associated with each node. This metric is used only in the time step calculation

B.3 Cell Pointers

Cell pointers relate properties associated with each cell in the mesh. This pointer is oblivious to whether the cell is a prism or tetrahedra.

$VOLCEL(0:cell)$	cell volume associated with each cell
------------------	---------------------------------------

B.4 Edge Pointers

Edge pointers relate properties associated with each edge in the mesh. These pointers are oblivious to whether the edge is the prismatic or tetrahedral region.

<code>AREA_EDGE(<i>i</i>,0:<i>edge</i>)</code> , <i>i</i> =1,3	node-centered dual cell face xyz area projections associated with each edge
<code>IEDGNOD(<i>i</i>,0:<i>edge</i>)</code> , <i>i</i> =1,2	nodes associated with each edge
<code>IUPD(0:<i>edge</i>,<i>i</i>)</code> , <i>i</i> =1,3	working array used in the coloring the edges and nodes
<code>FCV(0:<i>edge</i>,<i>i</i>)</code> , <i>i</i> =1,3	working array used in the the flux calculations
<code>GCV(0:<i>edge</i>,<i>i</i>)</code> , <i>i</i> =1,3	working array used in the the flux calculations
<code>HCV(0:<i>edge</i>,<i>i</i>)</code> , <i>i</i> =1,3	working array used in the the flux calculations

Appendix C

Data Structures for Prismatic Grid

The prismatic grid consists of triangular faces that cover the body surface, while quadrilateral faces extend in the direction normal to the surface, thus constituting successive layers of cells. The prismatic grid requires a set of pointers to define a two dimensional triangular mesh combined with a single index for each prismatic cell that belongs to the same stack of prisms. Since each stack of prisms is formed from a triangular face on the body surface, a prismatic cell may be referenced by the triangular face number and an index indicating the layer of which the cell constitutes a part.

IBFACTOT	number of faces on the body
IBNODTOT	number of nodes on the body
IBEDGTOT	number of edges on the body
ICELTOT	total number of prism cells
ILAYTOT	number of prismatic layers

NODTOT	total number of nodes in the mesh
IEDGTOT.PRISM	total number of edges in the mesh

C.1 Face Pointers

The triangulation on the boundary surface that constitutes the building block for the prismatic grid is represented by the following sets of pointers.

IBFACNOD_TRI(*i,face*), *i*=1,3 three surface nodes associated
with the surface triangular
face

Using the semi-structure of the grid, the nodes that constitute a face on layer *i*, Figure C.1, are given by:

$$\begin{aligned} N1 &= NB1 + (i - 1) * IBNODTOT \\ N2 &= NB2 + (i - 1) * IBNODTOT \\ N3 &= NB3 + (i - 1) * IBNODTOT \end{aligned} \quad (C.1)$$

C.2 Edge Pointers

The prism edges can be numbered locally using the semi-structure of the grid. IEDGTOT.VERT is the number of vertical edges in the prismatic region. IEDGTOT.VERT is equal to IBNODTOT*NLAYERS. The edges that constitute a cell on layer *i*, Figure C.1 are given by:

$$\begin{aligned} E1 &= EB1 + IEDGTOT.VERT + (i - 1) * IBEDGTOT \\ E2 &= EB2 + IEDGTOT.VERT + (i - 1) * IBEDGTOT \end{aligned} \quad (C.2)$$

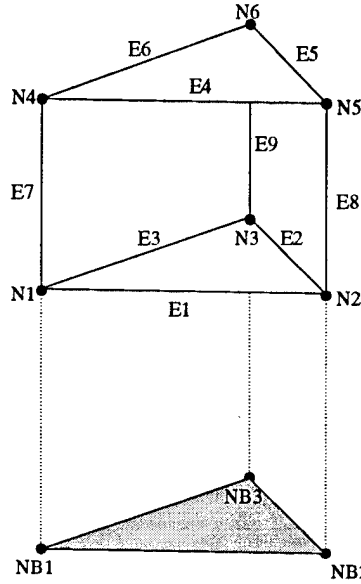


Figure C.1: Node and edge numbering system for prism cells.

$$E3 = EB3 + IEDGTOT_VERT + (i - 1) * IBEDGTOT$$

$$E4 = EB1 + IEDGTOT_VERT + (i) * IBEDGTOT$$

$$E5 = EB2 + IEDGTOT_VERT + (i) * IBEDGTOT$$

$$E6 = EB3 + IEDGTOT_VERT + (i) * IBEDGTOT$$

$$E7 = N1$$

$$E8 = N2$$

$$E9 = N3$$

Note, these are edge numbers based on the local edge numbering. The global edge numbers obtained by hashing the edges will be different than the local numbering based on the prismatic structure. To relate the two numbering systems, the pointer `IEDG_GLOBAL(0:edge)` is used: given a local edge number, the global edge number is produced. This is not a necessary pointer, but it is used in a tradeoff for saving computation time at the expense of memory storage.

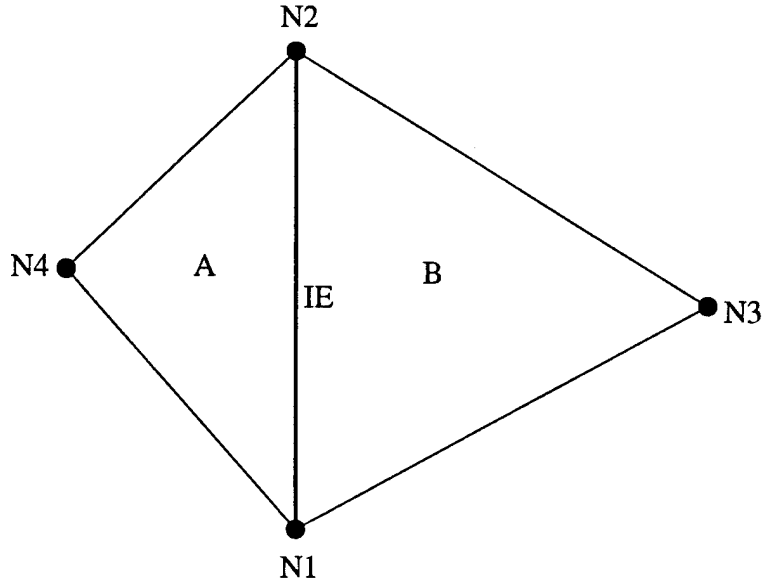


Figure C.2: Boundary edge-based node and prismatic cell numbering system.

The pointers associated with the edges on the boundary surface are illustrated in Figure C.2. The nodes of each edge on the boundary surface are represented by $IBEDGNOD(i,bedge)$, $i=1,2$. The neighboring faces (and cells) of the edge are given by $IBEDG_CELLS(i,bedge)$, $i=1,2$.

$$IBEDGNOD(1,IE) = N1 \quad (C.3)$$

$$IBEDGNOD(2,IE) = N2$$

$$IBEDG_CELLS(1,IE) = A$$

$$IBEDG_CELLS(2,IE) = B$$

Appendix D

Data Structures for Tetrahedral Grid

The cell and face counters for the tetrahedral portion of the grid are listed below.

The nodes and edges of the tetrahedral cell are numbered as shown in Figure D.1.

ITCELTOT	number of tetrahedra cells
IEDGTOT	total number of edges in the mesh
ITCELTOT_DIV	number of tetrahedra cells, including the divided prisms
IFSYMTOT_TET	number of faces on the symmetry boundary
IFFFDTOT_TET	number of faces on the far field boundary
IFACTET_TOT	number of tetrahedral faces. Nonzero if viscous tetrahedra are used.

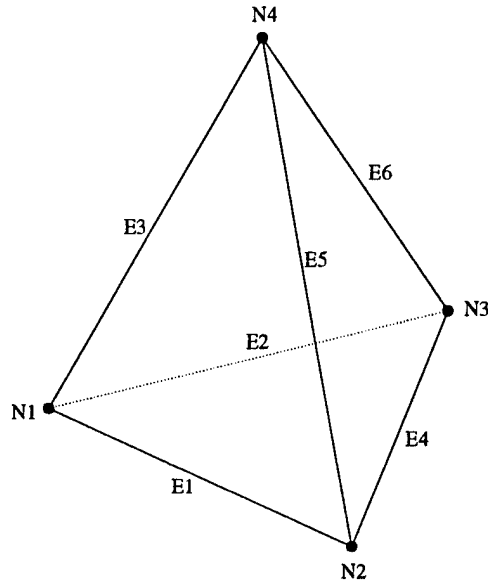


Figure D.1: Node and edge numbering system for tetrahedral cells.

D.1 Cell Arrays

The tetrahedral portion of the hybrid mesh requires pointers relating the nodes with their corresponding cell. The volume of each tetrahedra is also stored in its array. Not only are these nodes and volumes stored for the tetrahedral portion of the mesh, but they are also stored for the tetrahedral cells created by the division of the prisms for the finite element discretization of the Poisson equation.

ICELNOD($i, cell$), $i=1,4$ four nodes associated with the tetrahedra

VOLTET($cell$) volume of the tetrahedra

D.2 Face Arrays

Cell-to-face and edge-to-face pointers are required for the case of viscous tetrahedra. If inviscid tetrahedra are used IFACTET.TOT is equal to zero, although there are tetrahedral faces present.

ITFAC_CEL(<i>0:face</i>)	primary cell associated with a face
ITFAC_NOD(<i>i,face</i>), <i>i</i> =1,3	three nodes of the face
ITFAC_EDG(<i>i,face</i>), <i>i</i> =1,6	the first three indices correspond to the three edges, not on the face, of the primary cell. The second three indices correspond to the three edges, not on the face, of the secondary cell
AREA_TET_FAC(<i>i,face</i>), <i>i</i> =1,3	xyz area projections of the face

D.3 Interface Arrays

A few arrays are necessary to define the pointers needed in the interface region between the prisms and the tetrahedra. These pointers are required only if viscous tetrahedra are used.

IBFAC_TCEL(<i>face</i>)	tetrahedral cell number associated with the prismatic boundary face
ITFAC_EDG(<i>i,face</i>), <i>i</i> =1,3	indices which correspond to the three edges, not on the face, of the tetrahedra cell on the interface.

Appendix E

Study of Unsteady 2-D Flows

Flow around circular cylinders is of significant engineering interest. The corresponding flow fields are quite complex, involving such flow phenomena as boundary layer separation, vortex formation and shedding, as well as vortex/vortex interaction. Study of two-dimensional oscillatory flows is of importance to the case of wave-induced forces on cylindrical structures. It is a first step to understanding the complex three-dimensional wave-structure interaction.

The present study applies an existing two-dimensional incompressible Navier-Stokes solver to examine oscillating flows around cylinders. The flow will be defined and results of the two-dimensional study are presented.

E.1 2-D Navier-Stokes Solver

In the present work, a finite-volume/finite-element numerical scheme has been used to solve the unsteady Navier-Stokes equations of incompressible flow in two dimensions. Quadrilateral elements are used. The momentum equations, combined with a Simplified Marker and Cell (SMAC) [4, 5] type of pressure correction equation,

are solved employing a non-staggered grid where all of the dependent variables are defined at the cell corners. The momentum equations are solved using a finite-volume numerical scheme, while the pressure correction equation is solved using a finite-element numerical scheme. The solution is advanced in time with an explicit/implicit marching scheme. The numerical code has been validated in a previous study for a flat plate, steady uniform flow around a circular cylinder at Reynolds numbers of 40, and uniform flow around a circular cylinder at Reynolds numbers of 16200 and 10^5 [64, 91]. Further details on the 2-D solver are included in Appendix G.

E.2 Forces on the Cylinder in Oscillating Flow

Experimental and numerical results of a cylinder in oscillating flow show the in-line and lift coefficients have a periodic nature, corresponding to the periodicity of the freestream oscillatory flow, which has the form of $U_\infty = U_o \cos \omega t$. Time history of the in-line and lift coefficients are usually phase averaged so that the in-line and lift force during a single averaged period can be examined. In the present work, the numerical simulations were run for a minimum of ten periods, while the results for periods six through ten were phase averaged. The time history results show steady periodic results after the fifth period.

In the present work, the in-line and lift force coefficients, C_X and C_L respectively, are defined as follows

$$C_X = \frac{XF}{\frac{1}{2}\rho U_\infty^2 D} \quad (\text{E.1})$$

$$C_L = \frac{YF}{\frac{1}{2}\rho U_\infty^2 D} \quad (\text{E.2})$$

where XF is the in-line force, YF is the lift force, ρ is the density, U_∞ is the freestream

speed, and D is the cylinder diameter. The periodic in-line force is dominated by the freestream oscillating flow, whereas the periodic lift force is sensitive to the movements of the vortices as they separate from the cylinder and are swept around the cylinder when the flow reverses. As a rough rule, the fundamental lift frequency is $(m+1)$ times the oscillation frequency, where m is the number of vortices shed per half cycle [142]. Therefore, the lift force becomes a strong indicator of the particular flow regime the cylinder is experiencing.

Numerical results from the two-dimensional code will be compared against experimental results. Although the experiments were designed to be two-dimensional, in actuality, the shed vortex sheet is irregular and has three-dimensional coherent structures superimposed on the two-dimensional coherent structures [53]. The two-dimensional numerical simulation cannot resolve any three-dimensional behavior, however, the major characteristics of the in-line and lift force curves should compare reasonably well. Minor deviations of the numerical results from the experimental results is expected.

In the following sections, the unsteady flow simulations are presented, and comparisons with experimental data and other numerical results are performed. Both oscillating flow simulations were calculated on a CRAY C90.

E.3 Instability and Numerical Simulation

Unsteadiness in the form of the Karman vortex street occurs during any physical experiment of flow around a circular cylinder. Conceptually, the origin of the destruction of the symmetric pattern can be explained by the presence of multiple sources of perturbation in the physical model. If the Reynolds number is less than 40, those effects are dissipated by the viscous stresses. However, at higher Reynolds

numbers, they are not damped and the flow is unsteady.

In the case of the numerical simulation of unsteady flow past a circular cylinder all these destabilizing effects are absent. Since the geometry of the flow and the initial and the boundary conditions are symmetric, the simulation of the Navier-Stokes equations leads to a symmetric solution even for values of Reynolds number greater than 40. The truncation and round-off errors, as well as those due to the numerical approximation schemes, are perturbation factors which could eventually generate vortex shedding [20, 21, 58, 59]. It is well known that the vortex shedding can be generated in the case of high Reynolds number flow if there are no or little explicit/implicit dissipation terms. However, in such cases, the solution will be numerically unstable.

The most reasonable way to generate vortex shedding should be to introduce into the numerical model the same perturbations that occur during a physical experiment, if these perturbations were accurately prescribed. Unfortunately, it is not possible to know the characteristics of these perturbations in detail. In numerical simulations, an asymmetry is usually introduced into the initial solution in order to destroy the symmetric flow pattern. Different initial movements of the cylinder have been employed in the literature in order to introduce the asymmetry [20, 59, 89]. One such method is to rotate the cylinder for a short time. It has been found that the final periodic flow pattern is the same for different types of initial perturbation of numerical simulations [20]. This fact indicates that the periodic character of the flow appearing *beyond a critical Reynolds number* is an intrinsic property of the Navier-Stokes equations and does not depend on the nature of the initial perturbations. These conclusions are based on comparisons made by varying the magnitude and timing of the perturbations. It was not reported if the sign of the perturbation

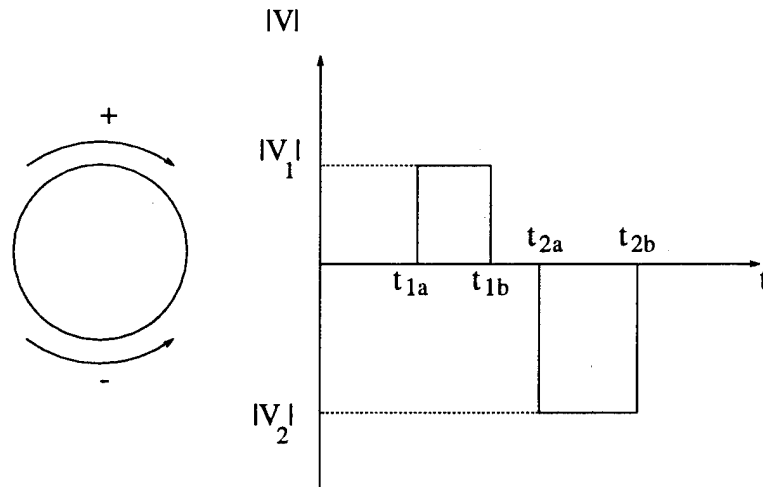


Figure E.1: Numerical perturbation to induce asymmetry.

was also varied.

In the present work, the perturbation introduced corresponds to a clockwise rotation of the cylinder followed by a resting period, and then a counterclockwise rotation, shown in Fig E.1. Moving the surface of the cylinder causes shear stress in the flow field due to the viscosity of the fluid. Using this kind of perturbation, vorticity generated through shear stress in the region around the cylinder is shed downstream. During the perturbation, one side of the surface of the rotating cylinder is moving in the direction of the freestream flow and a lesser amount of vorticity is produced compared to that generated on the other side, which is moving against the flow. This uneven production of vorticity destroys the symmetric pattern of the flow and causes unsteadiness in the flow field. The cylinder is slightly rotated only during this initial perturbation. During the rest of the simulation the cylinder is stationary.

E.4 Simulation of Oscillating Flow Around Cylinders

Oscillating flows around cylindrical bodies are quite complex and significantly different flow patterns can be created. These distinctive flow patterns are known to be a function of the Reynolds Number (Re) and Keulegan-Carpenter Number (KC) [118], where U_o is the maximum fluid velocity, T is the period of the oscillating flow, ρ_∞ is the freestream density, μ_∞ is the freestream viscosity and D is the cylinder diameter.

$$Re = \frac{\rho_\infty U_o D}{\mu_\infty} \quad (E.3)$$

$$KC = \frac{U_o T}{D} \quad (E.4)$$

Often the frequency parameter is used to categorize oscillating flows, β , which is the product of KC and Re . This study of oscillating flows is limited to two distinct flow regimes, the transverse regime and the double pair regime.

E.4.1 Transverse Flow Regime, $KC = 8$, $Re = 1568$

The flow regime, when the KC number is between 7 and 12, is called a transverse street [142], and the wake is comprised of the shedding of one large vortex during each half cycle. The transverse street is composed of a street or jet of vortices moving away from the cylinder, roughly perpendicular to the oscillation direction.

The lift force in the transverse flow regime has a distinctive shape, which is qualitatively shown in Fig E.2. Since there is one vortex shed per half cycle ($m=1$), the fundamental lift frequency is twice that of the oscillation frequency, and can be identified by the four extrema in one complete period. Each negative peak in the lift force is caused by the growth and shedding of a large vortex in each half cycle. Positive peaks are induced by the return of the most recently shed vortex towards

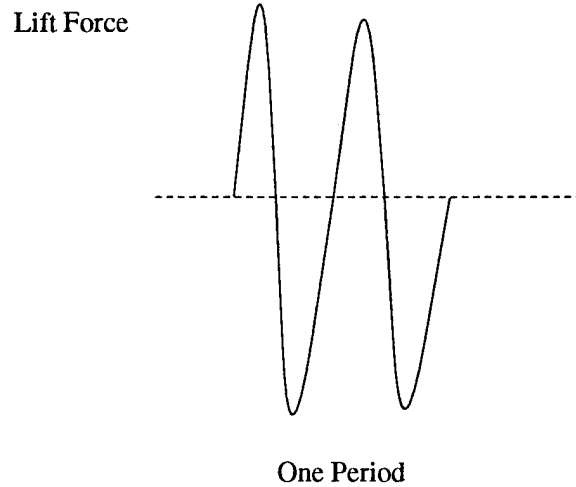


Figure E.2: General shape of lift force coefficient for transverse mode.

the cylinder just after flow reversal [142].

Numerical parameters used for this case of $KC = 8$, and $Re = 1568$ are shown in Table E.1. The far field boundary is located at approximately 33 diameters. Perturbation parameters used to induce asymmetry into the flow are shown in Table E.2. Time history of the in-line and lift force coefficients are shown in Figures E.3 and E.4. Phase averaged in-line and lift force coefficients are compared with numerical results from the literature [59]. These comparisons are shown in Figures E.5 and E.6. The phase averaging procedure is described in Appendix F. The numerical results from the literature are results of a stream function- vorticity calculation which used 80 nodes in the radial direction and 128 nodes in the circumferential direction. The results from the literature were for one instantaneous period and were not phase averaged. The two results are quite similar and the lift force shows the flow is representative of the transverse mode.

Table E.1: Numerical parameters used for oscillating cylinder flow at $KC=8$ and $Re = 1568$.

Number of Radial Nodes	Number of Circumferential Nodes	Radial Stretching Factor	Near to Wall Step Size	σ_{4u} σ_{4v} σ_{4w}	σ_{4p}
96	144	1.08	0.001	1.0e-3	5.0e-3

Table E.2: Numerical perturbation parameters used for oscillating cylinder flow at $KC=8$ and $Re = 1568$.

t_{1a}	t_{1b}	t_{2a}	t_{2b}	V_1	V_2
2.0	2.3	2.5	2.8	0.25	0.25

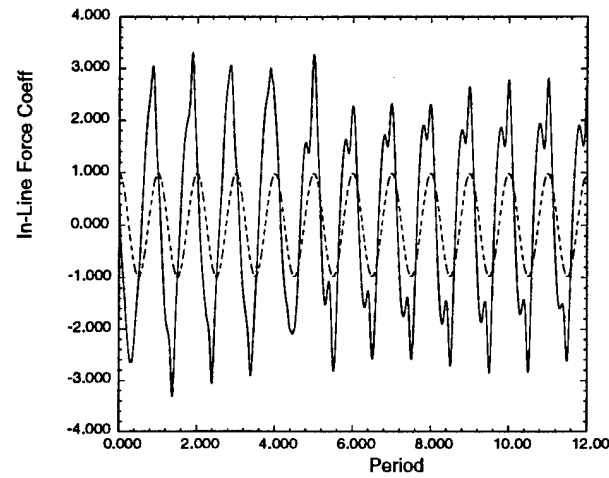


Figure E.3: Numerical in-line force coefficient versus time for transverse mode, $KC = 8$ and $Re = 1568$.

- - - Freestream velocity
 — In-line force coefficient

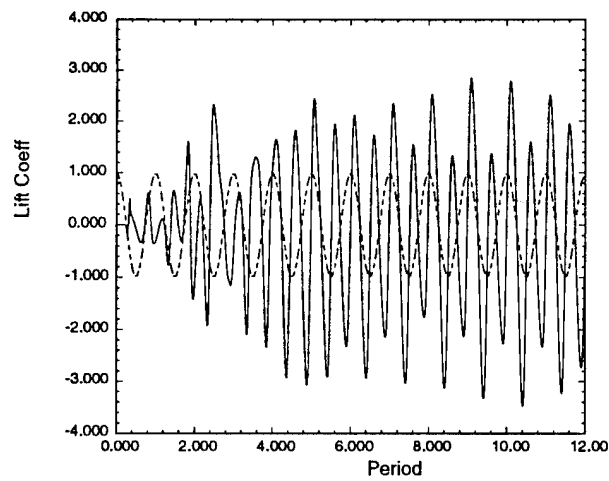


Figure E.4: Numerical lift force coefficient versus time for transverse mode, $KC = 8$ and $Re = 1568$.

- - - Freestream velocity
 — Lift coefficient

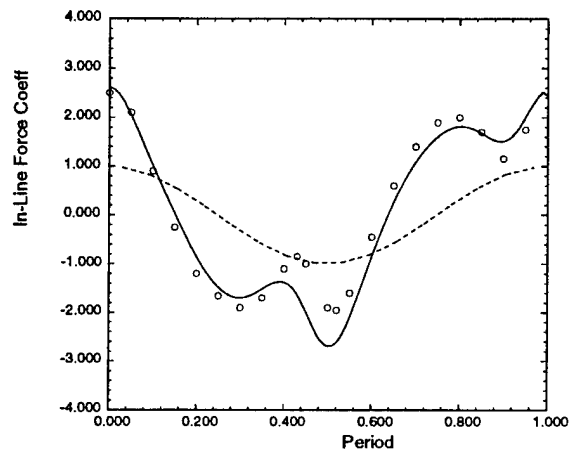


Figure E.5: In-line force coefficient comparison for transverse mode, $KC = 8$ and $Re = 1568$.

- Instantaneous numerical data by Justesen [59]
- - - Freestream velocity
- Phase averaged in-line force coefficient

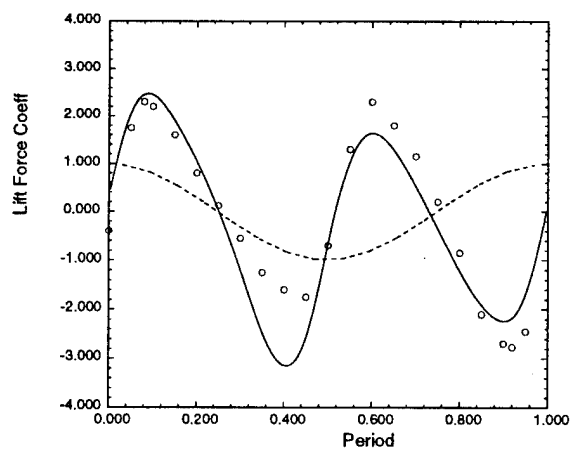
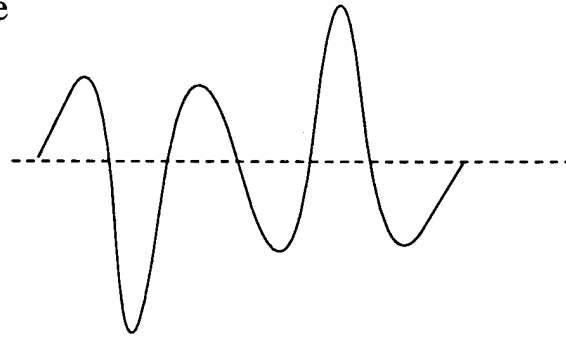


Figure E.6: Lift force coefficient comparison for transverse mode, $KC = 8$ and $Re = 1568$.

- Instantaneous numerical data by Justesen [59]
- - - Freestream velocity
- Phase averaged lift coefficient

Lift Force



One Period

Figure E.7: General Shape of Lift Force Coefficient for Double Pair Mode

E.4.2 Double Pair Flow Regime, $KC = 15$, $Re = 19050$

The flow regime, when the KC number is between 15 and 24, is called a double pair [142], and the wake is comprised of the shedding of two large vortices in each half cycle, and the formation of two pairs in each cycle. A series of vortex pairs convect away from the cylinder in opposite directions and from opposing diagonal points on the cylinder.

The lift force in the double pair flow regime has a distinctive shape, which is qualitatively shown in Fig E.7. Since there are two vortices shed per half cycle ($m=2$), then the fundamental lift frequency is three times that of the oscillation frequency, and can be identified by the six extrema in one cycle.

Numerical parameters used for the case of $KC = 15$ and $Re = 19050$ are shown in Table E.3. The far field boundary is located at approximately 30 diameters. Perturbation parameters are listed in Table E.4. Time history of the in-line and lift force coefficients are shown in Figures E.8 and E.9.

Obasaju, et al,[93] have found *mirror image* modes for certain flow regimes.

Table E.3: Numerical parameters used for oscillating cylinder flow at $KC=15$ and $Re = 19050$.

Number of Radial Nodes	Number of Circumferential Nodes	Radial Stretching Factor	Nearest to Wall Step Size	σ_{4u} σ_{4v} σ_{4w}	σ_{4p}
111	144	1.1	0.0002	5.0e-3	1.0e-3

Table E.4: Numerical perturbation parameters used for oscillating cylinder flow at $KC=15$ and $Re = 19050$.

t_{1a}	t_{1b}	t_{2a}	t_{2b}	V_1	V_2
3.0	3.5	3.5	4.0	0.25	0.25

These modes and their corresponding mirror image modes both generate lift of roughly the same magnitude but of opposite signs. Experimentally, the flow randomly switches from one mode to the other. When the experimental results are phase averaged, the lift coefficient curve averages to zero. To obtain useful information, Obasaju separates the results into the regular mode results and the mirror image mode results. Numerically, the random switching of flow modes from one to the other does not occur. However, we found the mirror image mode can be produced by reversing the numerical perturbation used to induce asymmetry. Results from our original perturbation technique, shown in Figure E.1, corresponded to Obasaju's mirror image mode. Obasaju's original mode resulted from the opposite numerical perturbation, a counterclockwise rotation being initiated first, followed by a resting period, and then a clockwise rotation concluding the perturbation. Phase averaged in-line and lift force coefficients are compared with Obasaju's experimental results. These comparisons are shown in Figures E.10 and E.11. The lift force coefficient is the mirror image mode, so we have plotted our results against the negative of Obasaju's result. The lift coefficient of our normal mode is shown in Figure E.12 and is compared against the actual data from Obasaju. The numerical results are quite similar to the experimental results and are indicative of the double pair regime.

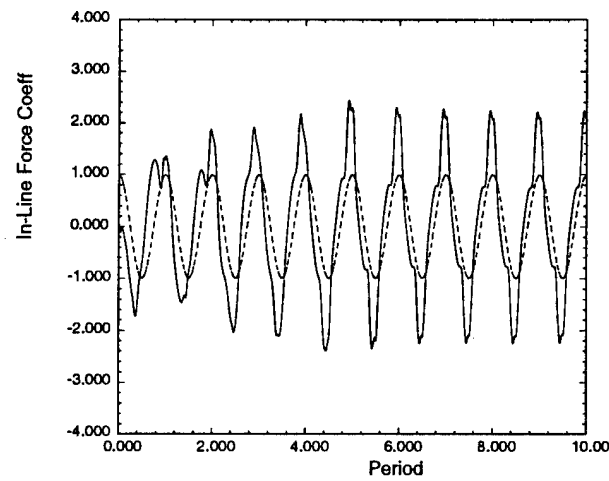


Figure E.8: Numerical in-line force coefficient versus time for double pair mode, $KC = 15$ and $Re = 19050$.

- - - Freestream velocity
 — In-line force coefficient

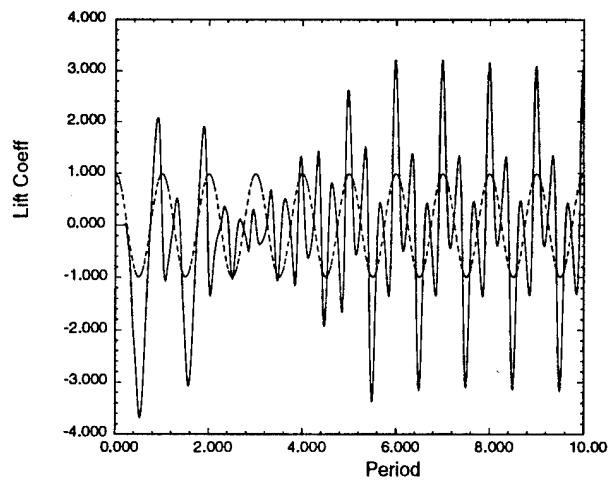


Figure E.9: Numerical lift force coefficient versus time for double pair mode, $KC = 15$ and $Re = 19050$.

- - - Freestream velocity
 — Lift coefficient

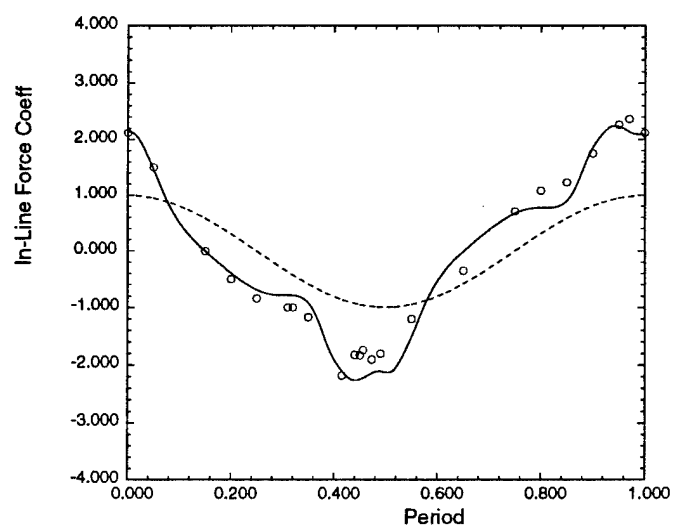


Figure E.10: In-line force coefficient comparison for double pair mode, $KC = 15$ and $Re = 19050$.

- Experimental phase averaged data by Obasaju [93]
- - - Freestream velocity
- Phase averaged in-line force coefficient

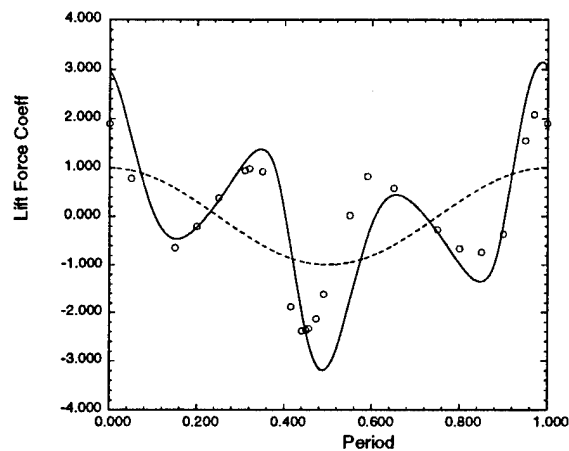


Figure E.11: Lift force coefficient comparison for mirror image double pair mode, $KC = 15$ and $Re = 19050$.

- Negative of experimental phase averaged data by Obasaju [93]
- - - Freestream velocity
- Phase averaged lift coefficient

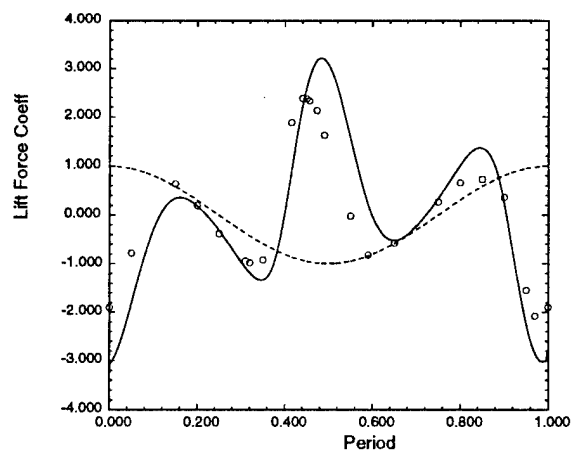


Figure E.12: Lift force coefficient comparison for double pair mode, $KC = 15$ and $Re = 19050$.

- Experimental phase averaged data by Obasaju [93]
- - - Freestream velocity
- Phase averaged lift coefficient

Appendix F

Phase Averaging Procedure for Oscillating Flow Solutions

The time varying velocity and force coefficient solutions for oscillating flows around cylinders are quite complex. However, the solutions, as well as the freestream flow, may have a periodic nature. After several periods, the velocities and force coefficients often reach a “steady periodic” solution. Once the velocities or force coefficients reach this steady periodic point in time, the results can be *phase averaged*. Phase averaging takes the solution over several periods and averages the data at corresponding phases of the period. Once the time varying solution is phase averaged, the solution can be displayed for a single averaged period.

F.1 Sampling the Time Varying Data

The phase averaging procedure begins with sampling of the data. The minimum time at which the phase averaging will begin, t_{min} , and the maximum time at which to end the phase averaging, t_{max} , must be specified. Examining the in-line force

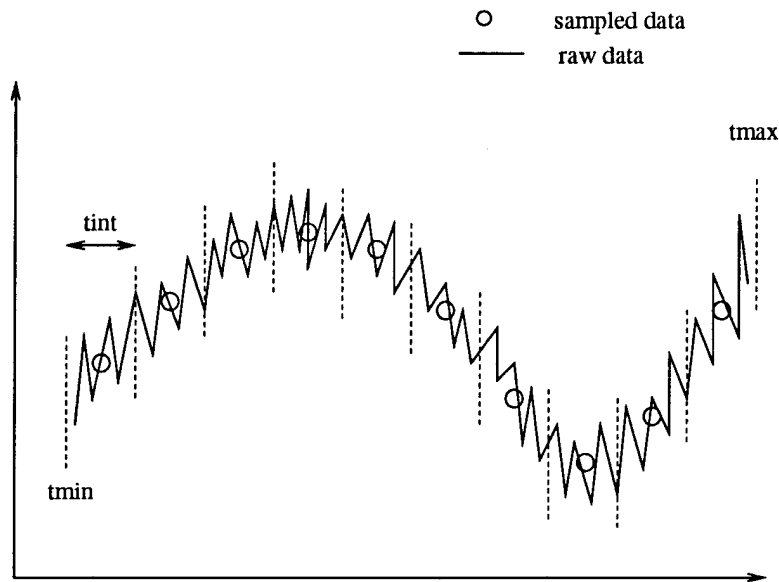


Figure F.1: Sampling of time varying data within specified time intervals.

coefficient at $KC=8$, Figure E.3, shows this solution is steady and periodic after the fifth period. Therefore, sampling from the sixth to the twelfth period is appropriate in this case. Since each period is 8 time units, $t_{min} = 48.0$ and $t_{max} = 96.0$. The time increment or interval for the integration, t_{int} , must also be specified. This interval for the sampling is often determined by numerical experimentation, but a value of 0.1 is usually a good choice.

The sampling procedure begins at the value of time = t_{min} . The values of all data points which fall between $t_1 = t_{min}$ and $t_2 = t_{min} + t_{int}$ are averaged. The value of time in this interval is also averaged. The multiple data points within this sampling interval are replaced by the averaged single data point. The next interval is defined, from $t_1 = t_{min} + t_{int}$ and $t_2 = t_{min} + 2*t_{int}$, and the data sampled. This procedure is repeated for successive sampling intervals until $t_2 = t_{max}$. This sampling procedure is illustrated in Figure F.1.

The averaging procedure in the sampling process is actually a weighted av-

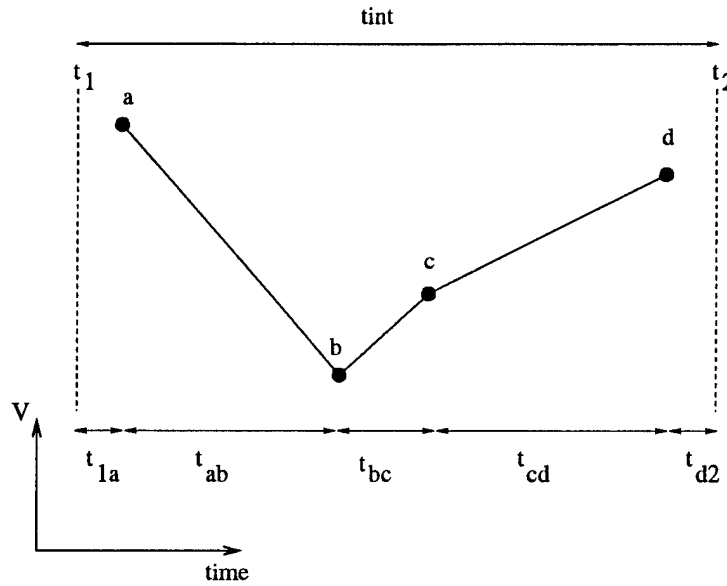


Figure F.2: Weighted averaging of sampled data.

eraging. Each data point is weighted with its own associated percent increment in time. For an interval i , which contains the data points a,b,c, and d, the sampled velocity for the interval is shown in Figure F.2. The final sampled velocity for the i th interval is shown in Equation F.1.

$$V_{sample}(i) = V_a \frac{t_{1a}}{t_{int}} + \frac{V_a + V_b}{2} \frac{t_{ab}}{t_{int}} + \frac{V_b + V_c}{2} \frac{t_{bc}}{t_{int}} + \frac{V_c + V_d}{2} \frac{t_{cd}}{t_{int}} + V_d \frac{t_{d2}}{t_{int}} \quad (F.1)$$

F.2 Phase Averaging the Sampled Data

Once the data is sampled, the phase averaging process can begin. Since the sampling time interval, t_{int} , as well as the period is known, the number of sampling time intervals within one period of time can be determined. The sampled data is processed by averaging the entries in the first time interval for all the sampled periods. For example, if there are 5 periods, with 10 sampling intervals in each

period, there are a total of 50 sampled intervals. The data at intervals 1, 11, 21, 31, and 41 are averaged to obtain the phase averaged data in the first interval of the phase averaged period. This averaging is continued over the remaining intervals in the period. For the $KC = 8$ example mentioned previously, the resulting phase averaged data is shown in Figure E.5.

Appendix G

Description of 2-D Navier-Stokes Solver

G.1 Governing Equations and Pressure Correction Formulation

The governing equations are the following non-dimensionalized laminar Navier-Stokes equations of incompressible flow. The equations were non-dimensionalized using the freestream velocity and the cylinder diameter.

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F_I}}{\partial x} + \frac{\partial \mathbf{G_I}}{\partial y} = \frac{\partial \mathbf{F_V}}{\partial x} + \frac{\partial \mathbf{G_V}}{\partial y} + \mathbf{S} \quad (\text{G.1})$$

The state vector $\mathbf{U}$, the convective flux vectors $\mathbf{F_I}, \mathbf{G_I}$, the viscous flux vectors $\mathbf{F_V}, \mathbf{G_V}$, and the source term $\mathbf{S}$ are expressed in terms of the primitive variables, namely, density ρ , and x, y velocities and pressure P . The first entry in the vectors correspond to the continuity equation, while the second and third terms correspond

to the x and y momentum equations, respectively.

$$\mathbf{U}^T = \begin{pmatrix} 0 & u & v \end{pmatrix} \quad \mathbf{S}^T = \begin{pmatrix} 0 & -\frac{\partial P}{\partial x} & -\frac{\partial P}{\partial y} \end{pmatrix}$$

$$\mathbf{F_I}^T = \begin{pmatrix} u & u^2 & uv \end{pmatrix} \quad \mathbf{G_I}^T = \begin{pmatrix} v & vu & v^2 \end{pmatrix}$$

$$\mathbf{F_V}^T = \frac{1}{Re} \begin{pmatrix} 0 & \frac{\partial u}{\partial x} & \frac{\partial v}{\partial x} \end{pmatrix} \quad \mathbf{G_V}^T = \frac{1}{Re} \begin{pmatrix} 0 & \frac{\partial u}{\partial y} & \frac{\partial v}{\partial y} \end{pmatrix}$$

An explicit/implicit marching scheme is adopted for integration in time of the above equations. The velocity values are treated explicitly, while the pressure values are treated implicitly in the momentum equations. The velocity values are marched in time with a forward Euler scheme [50]. The continuity equation is formulated implicitly with the velocity values considered at time level $(n + 1)$.

The momentum equation cannot be solved directly due to the implicit treatment of the pressure term. An auxiliary velocity vector $\vec{u}'$ is introduced, which satisfies the following equation. The superscripts denote the time levels.

$$\frac{\partial \mathbf{U}^{n+1}}{\partial t} + \frac{\partial \mathbf{F_I}^n}{\partial x} + \frac{\partial \mathbf{G_I}^n}{\partial y} = \frac{\partial \mathbf{F_V}^n}{\partial x} + \frac{\partial \mathbf{G_V}^n}{\partial y} + \mathbf{S}^{n+1} \quad (\text{G.2})$$

In the auxiliary velocity vector equation, the pressure term is treated explicitly and $\vec{u}'$ can be obtained directly. However, the solution $\vec{u}'$ does not satisfy the continuity equation. Subtracting the auxiliary velocity vector equation (G.2) from the momentum equation (G.1), it is obtained :

$$\vec{u}^{(n+1)} - \vec{u}' = - \left[\nabla \left(p^{(n+1)} - p^{(n)} \right) \right] \Delta t \quad (\text{G.3})$$

Introducing a scalar potential ϕ , such that

$$\vec{u}^{(n+1)} - \vec{u}' = -\nabla \phi, \quad (\text{G.4})$$

the following equation for pressure can be obtained:

$$p^{(n+1)} - p^{(n)} = \frac{1}{\Delta t} \phi \quad (\text{G.5})$$

Finally, taking the divergence of each side of equation (G.4) and considering the continuity equation, the following pressure correction Poisson equation is obtained:

$$\Delta \phi = -\nabla \cdot \vec{u}' \quad (\text{G.6})$$

In this equation, the values of ϕ on the left hand side are treated implicitly, which requires the inversion of a matrix. Using the ϕ values obtained by the above equation, we can correct the velocity and pressure fields using equations (G.4) and (G.5) as follows:

$$\vec{u}^{(n+1)} = \vec{u}' - \nabla \phi \quad (\text{G.7})$$

$$p^{(n+1)} = p^{(n)} + \frac{1}{\Delta t} \phi \quad (\text{G.8})$$

The above solution procedure follows the explicit/implicit marching scheme in [71, 91]. The overall solution procedure corresponding to marching by one-time-step is summarized as follows:

1. Calculate the auxiliary velocity vector $\vec{u}'$ from (G.2) using $\vec{u}^{(n)}$ and $p^{(n)}$ values.
2. Solve the pressure correction Poisson equation (G.6) and obtain the ϕ values.
3. Calculate $\vec{u}^{(n+1)}$ and $p^{(n+1)}$ using (G.7) and (G.8).
4. If $\nabla \cdot \vec{u}^{(n+1)} < \epsilon$ where ϵ is the tolerance for divergence, advance to next time step. If not, consider $\vec{u}^{(n+1)}$ as $\vec{u}'$ and repeat steps 2 and 3.

G.2 Finite-Volume Discretization of the Momentum Equations

The Navier-Stokes equations of incompressible viscous flow are discretized using the finite volume approach with a nonstaggered grid. The equations are given in integral form for a bounded two-dimensional domain Ω as follows:

$$\int_{\Omega} \frac{\partial U}{\partial t} d\Omega + \int_{\Omega} \left(\frac{\partial \mathbf{F}_I}{\partial x} + \frac{\partial \mathbf{G}_I}{\partial y} \right) d\Omega = \int_{\Omega} \left(\frac{\partial \mathbf{F}_V}{\partial x} + \frac{\partial \mathbf{G}_V}{\partial y} \right) d\Omega + \int_{\Omega} (\mathbf{S}) d\Omega \quad (\text{G.9})$$

Since only a two-dimensional domain is considered, the faces of the quadrilaterals are assumed to have unit depth. The volume integral containing the spatial derivatives in equation (G.9) is transformed to a surface integral using the divergence theorem.

Considering the convective flux vector terms alone, we have:

$$\int_{\Omega} \left(\frac{\partial \mathbf{F}_I}{\partial x} + \frac{\partial \mathbf{G}_I}{\partial y} \right) d\Omega = \int_{\partial\Omega} (\mathbf{F}_I n_x + \mathbf{G}_I n_y) dS$$

which in discrete form is expressed by,

$$\sum_{f=1}^4 (\mathbf{F}_I S_x + \mathbf{G}_I S_y)_f \quad (\text{G.10})$$

where the summation is over the four faces of the quadrilateral and S_x, S_y are the area projections of the face f in the corresponding co-ordinate directions. The flux vectors are considered at the center of the face f and their values are obtained by averaging the values at the face-nodes.

G.3 Finite-Element Discretization of Pressure Correction Equation

The pressure correction equations are discretized using the Galerkin finite-element approach. The scheme is compact with all operations being restricted to within each

grid-cell. *Bilinear isoparametric* quadrilateral elements are employed [39].

For quadrilaterals, the values of u , v and ϕ are defined in each element using the following finite-element formulation :

$$u = \sum_{i=1}^4 N_i u_i, \quad v = \sum_{i=1}^4 N_i v_i, \quad \phi = \sum_{i=1}^4 N_i \phi_i \quad (\text{G.11})$$

where u_i, v_i and ϕ_i are nodal values of u , v and ϕ and N_i is the interpolating *shape* function associated with the i -th node.

Integrating equation (G.6) over the each element domain Ω using the Galerkin method, the following equation is obtained :

$$\iint_{\Omega} N_i (\phi_{,xx} + \phi_{,yy}) \, dx \, dy = - \iint_{\Omega} N_i (u'_{,x} + v'_{,y}) \, dx \, dy \quad (\text{G.12})$$

Applying Gauss's theorem, we can get following element matrix system :

$$D_{ij} \phi_j = K_{ij}^x u'_j + K_{ij}^y v'_j \quad (\text{G.13})$$

where

$$K_{ij}^x = \iint_{\Omega} N_i \frac{\partial N_j}{\partial x} \, dx \, dy, \quad K_{ij}^y = \iint_{\Omega} N_i \frac{\partial N_j}{\partial y} \, dx \, dy$$

$$D_{ij} = \iint_{\Omega} \left(\frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} \right) \, dx \, dy$$

We can construct the following global matrix system by assembling the element matrix obtained in (G.13)

$$\mathbf{D} \Phi = \mathbf{f} \quad (\text{G.14})$$

where

$$\mathbf{D} = \sum_e D_{ij}, \quad \Phi^T = [\phi_1, \phi_2, \phi_3, \dots]$$

$$\mathbf{f} = \sum_e [K_{ij}^x u'_j + K_{ij}^y v'_j]$$

The values for ϕ on the left hand side of (G.14) are treated implicitly, which requires solution of a system. The matrix that requires inversion is a symmetric linear matrix. The system is solved by the ICCG (*Incomplete Cholesky Conjugate Gradient*) iterative method [94]. The rate of convergence of the iterative process depends on the numbering of nodes [75]. In order to accelerate convergence, the elements of the matrix are renumbered using the RCM (*Reverse Cuthill-Mckee*) method [94].

A typical number of iterations required by the ICCG method for the Poisson equation is 3, while 5 iterations are typically required for convergence of the ϕ values. Convergence criteria for the pressure corrections is $\nabla \cdot \vec{u} < 10^{-2} \cdot U/L$ where U and L are representative velocity and length.

G.3.1 Artificial Dissipation

Central space differencing schemes are susceptible to oscillatory modes in the velocity field of high Reynolds number flows. Furthermore, odd-even decoupling of the solution may appear in the pressure field for this non-staggered type of mesh that is employed.

In the present work, a fourth order smoothing term is added explicitly to the momentum equations in order to suppress odd-even decoupling of the solution [57]. Furthermore, fourth order dissipation is added to the pressure correction equation in order to stabilize the solution and suppress oscillations in the pressure field.

The smoothing operator is cast in a form suitable for adaptive unstructured grids. All operations are split in such a way that no information is required from outside of each cell. Each grid node receives contributions from each one of its surrounding cells. The operator is formed in two steps. The second order difference

operator is formed in the first step. The second order distributions to cell-corners (j) for the momentum equations are as follows :

$$D_j^2(u) = \left(\sum_{i=1}^4 u_i \right) - 4u_j \quad (\text{G.15})$$

The second step duplicates the first, replacing state variables by second order differences from the first step. The fourth order smoothing distributions are:

$$-D_j^4(u) = \left(\sum_{i=1}^4 D_i^2(u) \right) - 4D_j^2(u) \quad (\text{G.16})$$

The fourth-order difference terms of u and v are multiplied by empirical coefficients $\sigma_4(u)$ and $\sigma_4(v)$. Large values of σ_4 stabilizes the solution but destroys the accuracy. Therefore, special care is required for choosing the value of σ_4 . Numerical experiments have been carried out to determine optimum values for the smoothing coefficient. The determined values are such that the solution accuracy is not affected, while the *odd-even* modes are suppressed [61].

Similarly, fourth order difference terms of pressure are added to the right hand side of the Poisson equation (G.14) as follows :

$$\mathbf{D}\Phi = \mathbf{f} + \mathbf{d} \quad (\text{G.17})$$

where

$$\mathbf{d}^T = \left[\sigma_4(p) \mathbf{D}_j^4(\mathbf{p}_j^{(n)}) \right]$$

Generally, the value of $\sigma_4(p)$ is different from $\sigma_4(u)$ and $\sigma_4(v)$. In the present work, the values of $\sigma_4(u)$ and $\sigma_4(v)$ are the same, while the value of $\sigma_4(p)$ is different.

G.3.2 Time-Step Calculation

Using central space and forward time differencing, the stability limitation for the model 1-D convection equation $u_t + cu_y = 0$ is $\frac{c\Delta t}{\Delta y} \leq 1$ (CFL limitation), while the

corresponding stability restriction for the 1-D model diffusion equation $u_t = \nu u_{yy}$ is $\frac{\nu \Delta t}{\Delta y^2} \leq \frac{1}{2}$.

In the present scheme, a combination of the two limitations is employed. Specifically,

$$\Delta t = \omega \min \left\{ \frac{\Delta m}{|u| + \frac{\nu}{\alpha \Delta m}}, \frac{\Delta l}{|v| + \frac{\nu}{\alpha \Delta l}} \right\}, \quad (\text{G.18})$$

where $\Delta m, \Delta l$ are the cell dimensions in the m, l local cell-directions, u, v are the corresponding velocity components, ν is the kinematic viscosity coefficient, and $\alpha = \frac{1}{2}$ is the diffusion stability limitation. Lastly, ω is a safety factor, and equal to 0.9.

G.3.3 Boundary Conditions

Two types of conditions have been applied for the cases considered in the present work: *wall* and *far field*. The boundary conditions are applied to the two velocity components, as well as to the pressure corrections. At the *wall*, the u and v components of velocity are set to zero. The value of $\partial \phi / \partial n$ is also set to zero. At the *farfield*, the velocity components are set equal to the free-stream values, while the value of $\partial \phi / \partial n$ is set to zero.

An added complication to the boundary conditions exist because the pressure correction equation is a Poisson equation, and derivative boundary conditions are used. In this case, there are an infinite number of solutions for the pressure field differing only by arbitrary additive constants. In these circumstances the iterative solution method may never converge unless the pressure correction ϕ is specified at one point in the flow [100]. In the present work, ϕ at a single upstream point on the farfield boundary is specified to be zero.

Bibliography

- [1] S. Abdallah. *Numerical Solutions for the Pressure Poisson Equation with Neumann Boundary Conditions Using a Non-staggered Grid, I.* Journal of Computational Physics, 70:182-192, 1987.
- [2] S. Abdallah. *Numerical Solutions for the Incompressible Navier-Stokes Equations in Primitive Variables Using a Non-staggered Grid, II.* Journal of Computational Physics, 70:193-202, 1987.
- [3] S. Abdallah and J. Dreyer. *Dirichlet and Neumann Boundary Conditions for the Pressure Poisson Equation of Incompressible Flow.* International Journal for Numerical Methods in Fluids, 8:1029-1036, 1988.
- [4] A.A. Amsden and F.H. Harlow. *The SMAC Method: A Numerical Technique for Calculating Incompressible Fluid Flows.* LA-4370, Los Alamos Scientific Laboratory, May 1970.
- [5] A.A. Amsden and F.H. Harlow. *A Simplified MAC Technique for Incompressible Fluid Flow Calculations.* Journal of Computational Physics, 6:322-325, 1970.
- [6] P. Anagnostopoulos, G. Iliadis, and J. Rasoul. *Numerical Solution of Oscillatory Flow Around a Circular Cylinder at Low Reynolds and Keulegan-Carpenter*

- Numbers. Finite Elements in Fluids. Pineridge Press, 1993, pages 258-267.*
- [7] D.A. Anderson, J.C. Tannehill, and R.H. Pletcher. *Computational Fluid Mechanics and Heat Transfer*. Hemisphere Publishing Company, McGraw-Hill Book Company, 1984, pages 10, 503.
 - [8] N. Baba and H. Miyata. *Higher-Order Accurate Difference Solutions of Vortex Generation from a Circular Cylinder in an Oscillatory Flow*. Journal of Computational Physics, 69:2,362-396, 1987.
 - [9] S. Babu and R.H. Pletcher. *Time-Accurate Prediction of Three-Dimensional Incompressible Flows with Free Surfaces*. AIAA Paper 95-0079, 1995.
 - [10] V. Babu and S.A. Korpela. *Numerical Solution of the Incompressible Three-Dimensional Navier-Stokes Equations*. Computers & Fluids, 23:5,675-691, 1994.
 - [11] T.J. Baker. *Developments and Trends in Three Dimensional Mesh Generation*. Applied Numerical Mathematics, 5:275-304, 1989.
 - [12] E.B. Becker, G.F. Carey and J.T. Oden. *Finite Elements: An Introduction, Volume I*. Prentice-Hall, Inc, 1981.
 - [13] J.B. Bell, P. Colella, H.M. Glaz. *A Second-Order Projection Method for Viscous, Incompressible Flow*. AIAA Paper 87-1176, 1987.
 - [14] J.B. Bell, P. Colella, L.H. Howell. *An Efficient Second-Order Projection Method for Viscous, Incompressible Flow*. AIAA Paper 91-1560-CP, 1991.
 - [15] J.B. Bell, J.M. Solomon, and W.G. Szymczak. *An Second-Order Projection Method for the Incompressible Navier-Stokes Equations on Quadrilateral Grids*. AIAA Paper 89-1967-CP, 1989.

- [16] A. Belov, L. Martinelli, A. Jameson. *A New Implicit Algorithm with Multigrid for Unsteady Incompressible Flow Calculations*. AIAA Paper 95-0049, 1995.
- [17] E. Blosch, W. Shyy, and R. Smith. *The Role of Mass Conservation in Pressure-Based Algorithms*. Numerical Heat Transfer, Part B, 24:415-429, 1993.
- [18] M.E. Braaten and S.D. Connell. *A 3-D Unstructured Adaptive Multigrid Scheme for the Navier-Stokes Equations*. GE Research and Development Center, Mechanical Systems Laboratory, 94CRD146, Class 1, August 1994.
- [19] M.E. Braaten and W. Shyy. *Study of Pressure Correction Methods with Multigrid for Viscous Flow Calculations in Nonorthogonal Curvilinear Coordinates*. Numerical Heat Transfer, 11:417-442, 1987.
- [20] M. Braza, P. Chassaing and H.H. Minh. *Numerical Study and Physical Analysis of the Pressure and Velocity Fields in the Near Wake of a Circular Cylinder*. Journal of Fluid Mechanics, 165:79-130, 1986.
- [21] M. Braza and H.H. Minh. *Direct Numerical Simulation of Certain 2-D Transition Features of Flow Past a Circular Cylinder*. Proceedings of the Eighth International Conference on Offshore Mechanics and Arctic Engineering, Vol II, 1989.
- [22] W.R. Briley. *Numerical Method for Predicting Three-Dimensional Steady Viscous Flow in Ducts*. Journal of Computational Physics, 14:8-28, 1974.
- [23] C.H. Bruneau and C. Jouron. *An Efficient Scheme for Solving Steady Incompressible Navier-Stokes Equations*. Journal of Computational Physics, 89:389-413, 1990.

- [24] G.F. Carey and J.T. Oden. *Finite Elements: A Second Course, Volume II*. Prentice-Hall, Inc., New Jersey, 1983.
- [25] G.F. Carey and J.T. Oden. *Finite Elements: Fluid Mechanics, Volume VI*. Prentice-Hall, Inc., New Jersey, 1986.
- [26] K.S. Chang and J.Y. Sa. *Patterns of Vortex Shedding from an Oscillating Circular Cylinder*. AIAA Journal, 30:5,1331-1336, 1992.
- [27] S. Chippada, B. Ramaswamy, M.F. Wheeler, L.C. Cowsar, and D.M. Tetzlaff. *Two-Dimensional Modeling of Flow and Sedimentation*. Finite Elements in Fluids, Pineridge Press, 1993, pages 796-805.
- [28] D. Choi and C.L. Merkle. *Application of Time-Iterative Schemes to Incompressible Flow*. AIAA Journal, 23:1518-1524, 1985.
- [29] S.K. Choi, H.Y. Nam, and M. Cho. *Use of Staggered and NonStaggered Grid Arrangements for Incompressible Flow Calculations on Nonorthogonal Grids*. Numerical Heat Transfer, Part B, 25:193-204, 1994.
- [30] A.J. Chorin. *Numerical Solution of Incompressible Flow Problems*. Studies in Numerical Analysis, 2:64-71, 1968.
- [31] A.J. Chorin. *A Numerical Method for Solving Incompressible Viscous Flow Problems*. Journal of Computational Physics, 2:12-26, 1967.
- [32] K.A. Cliffe and D.A. Lever. *A Comparison of Finite Element Methods for Solving Flow Past a Sphere*. Journal of Computational Physics, 62:321-330, 1986.
- [33] W.M. Collins and S.C.R. Dennis. *Flow Past an Impulsively Started Circular Cylinder*. Journal of Fluid Mechanics, 60:1,105-127, 1973.

- [34] S.D. Connell and M.E. Braaten. *Semi-Structured Mesh Generation for 3D Navier-Stokes Calculations*. GE Research and Development Center, Mechanical Systems Laboratory, 94CRD154, Class 1, August 1994.
- [35] S.D. Conte and C. de Boor. *Elementary Numerical Analysis: An Algorithmic Approach, 3rd Ed.* McGraw-Hill, 1980.
- [36] M. Coutanceau and R. Bouard. *Experimental Determination of the Main Features of the Viscous Flow in the Wake of a Circular Cylinder in Uniform Translation. Part 1. Steady Flow* Journal of Fluid Mechanics, 79:2,231-256, 1977.
- [37] M. Coutanceau and R. Bouard. *Experimental Determination of the Main Features of the Viscous Flow in the Wake of a Circular Cylinder in Uniform Translation. Part 2. Unsteady Flow* Journal of Fluid Mechanics, 79:2,257-272, 1977.
- [38] A.E. Fanning and T.J. Mueller. *Note: On the Solution of a Neumann Problem for an Inhomogeneous Laplace Equation*. Journal of Computational Physics, 13:450-454, 1973.
- [39] C.A.G. Fletcher. *Computational Galerkin Method*. Springer-Verlag, 1984.
- [40] B. Fornberg. *A Numerical Study of Steady Viscous Flow Past a Circular Cylinder*. Journal of Fluid Mechanics, 98:4,819-855, 1980.
- [41] C.J. Freitas, R.L. Street, A.N. Findikakis, and J.R. Koseff. *Numerical Simulation of Three-Dimensional Flow in a Cavity*. International Journal for Numerical Methods in Fluids, 5:561-575, 1985.
- [42] K.N. Ghia, W.L. Hankey Jr. and J.K. Hodge. *Use of Primitive Variables in the Solution of Incompressible Navier-Stokes Equations*. AIAA Journal, 17:3,298-301, 1979.

- [43] P.M. Gresho. *Some Current CFD Issues Relevant to the Incompressible Navier-Stokes Equations*. Computer Methods in Applied Mechanics and Engineering, 87:201-252, 1991.
- [44] P.M. Gresho and R.L. Lee. *Don't Suppress the Wiggles - They're Telling You Something!* Computers and Fluids, 9:223-253, 1981.
- [45] P.M. Gresho and R.L. Sani. *On Pressure Boundary Conditions for the Incompressible Navier-Stokes Equations*. International Journal for Numerical Methods in Fluids, 7:1111-1145, 1987.
- [46] A.S. Grove, F.H. Shair, E.E. Petersen, and A. Acrivos. *An Experimental Investigation of the Steady Separated Flow Past a Circular Cylinder*. Journal of Fluid Mechanics, 19:1,60-81, 1964.
- [47] K. Goda. *A Multistep Technique with Implicit Difference Schemes for Calculating Two- or Three-Dimensional Cavity Flows*. Journal of Computational Physics, 30:76-95, 1979.
- [48] M. Hafez and M. Soliman. *A Velocity Decomposition Method for Viscous Incompressible Flow Calculations, Part II*. AIAA Paper 89-1966 CP, 1989.
- [49] M. Hafez and M. Soliman. *Numerical Solution of the Incompressible Navier-Stokes Equations in Primitive Variables on Unstaggered Grids*. AIAA Paper 91-1561 CP, 1991.
- [50] F.H. Harlow and J.E. Welch. *Numerical Calculation of Time-Dependent Viscous Incompressible Flow with Free Surface*. Physics of Fluids, 8:12,2182-2189, 1965.

- [51] P. He and M. Salcudean. *A Numerical Method for 3D Viscous Incompressible Flows Using Non-Orthogonal Grids*. International Journal for Numerical Methods in Fluids, 18:449-469, 1994.
- [52] C.W. Hirt and J.L. Cook. *Calculating Three-Dimensional Flows Around Structures and over Rough Terrain*. Journal of Computational Physics, 10:324-340, 1972.
- [53] H. Honji. *Streaked Flow Around an Oscillating Circular Cylinder*. Journal of Fluid Mechanics, 107:509-520, 1981.
- [54] Y.H. Hwang. *Arbitrary Domain Velocity Analysis for the Incompressible Navier-Stokes Equations*. Journal of Computational Physics, 110:134-149, 1994.
- [55] Y.H. Hwang. *Solution of the Steady Navier-Stokes Equations by the Pressure Correction Method in the ALE Grid*. Numerical Heat Transfer, Part B, 23:237-255, 1993.
- [56] C.P. Jackson. *The Effect of the Choice of the Reference Pressure Location in Numerical Modelling of Incompressible Flow*. International Journal for Numerical Methods in Fluids, 4:147-158, 1984.
- [57] A. Jameson, W. Schmidt and E. Turkel. *Numerical Solutions of the Euler Equations by Finite-Volume Methods Using Runge-Kutta Time-Stepping Schemes*. AIAA Paper 81-1259, 1981.
- [58] Y. Jiang, C.P. Chen, and P.K. Tucker. *Multigrid Solution of Unsteady Navier-Stokes Equations Using a Pressure Method*. Numerical Heat Transfer, Part A, 20:81-93, 1991.

- [59] P. Justesen. *A Numerical Study of Oscillating Flow Around a Circular Cylinder*. Journal of Fluid Mechanics, 222:157-196, 1991.
- [60] K. Kakuda, N. Tosaka, and H. Kato. *Petrov-Galerkin Finite Element Approach Using Exponential Functions for Three-Dimensional Incompressible Viscous Flow Problems*. Finite Elements in Fluids, Pineridge Press, 1993, pages 204-213.
- [61] Y. Kallinderis. *A Finite Volume Navier-Stokes Algorithm for Adaptive Grids*. International Journal for Numerical Methods in Fluids, 15:193-217, 1992.
- [62] Y. Kallinderis and J.R. Baron. *A New Adaptive Algorithm for Turbulent Flows*. Computers and Fluids Journal, 21:1,77-96, 1992.
- [63] Y. Kallinderis, A. Khawaja, and H. McMorris. *Hybrid Prismatic/Tetrahedral Grid Generation for Complex Geometries*. AIAA Paper 95-0211, 1995.
- [64] Y. Kallinderis and K. Nakajima. *A Finite-Element Method for the Incompressible Navier-Stokes Equations with Adaptive Hybrid Grids*. AIAA Paper 93-3005, 1993.
- [65] Y. Kallinderis and P. Vijayan. *An Adaptive Refinement-Coarsening Scheme for 3-D Unstructured Meshes*. AIAA Journal, 31:8,1440-1447, 1993.
- [66] Y. Kallinderis and S. Ward. *Prismatic Grid Generation for 3-D Complex Geometries*. AIAA Journal, 31:10,1850-1856, 1993.
- [67] T. Kawamura and T. Hayashi. *Computation of Flow around a Yawed Circular Cylinder*. JSME International Journal, Series B, 37:2,229-236, 1994.

- [68] K.M. Kelkar and S.V. Patankar. *Numerical Prediction of Vortex Shedding Behind a Square Cylinder*. International Journal for Numerical Methods in Fluids, 14:327-341, 1992.
- [69] J. Kim and P. Moin. *Application of a Fractional-Step Method to Incompressible Navier-Stokes Equations*. Journal of Computational Physics, 59:308-323, 1985.
- [70] S.W. Kim. *Calculations of Separated 3-D Flows with a Pressure-Staggered Navier-Stokes Equations Solver*. NASA CR 187065, 1991.
- [71] S.W. Kim and T.J. Benson. *Comparison of the SMAC, PISO and Iterative-Advancing Schemes for Unsteady Flows*. Computers & Fluids, 21:3,435-454, 1992.
- [72] S.W. Kim and T.J. Benson. *Calculation of a Circular Jet in Crossflow with a Multiple-Time-Scale Turbulence Model*. International Journal of Heat and Mass Transfer, 35:10,2357-2365, 1992.
- [73] D. Kwak, J.L.C. Chang, S.P. Shanks and S.R. Chakravarthy. *A Three-Dimensional Incompressible Navier-Stokes Flow Solver Using Primitive Variables*. AIAA Journal, 24:390-396, 1986.
- [74] S.Y. Lin and T.M. Wu. *An Adaptive Multigrid Finite-Volume Scheme for Incompressible Navier-Stokes Equations*. International Journal for Numerical Methods in Fluids, 17:687-710, 1993.
- [75] R. Lohner. *A Fast Finite Element Solver for Incompressible Flows*. AIAA Paper 90-0398, 1990.
- [76] R. Lohner and P. Parikh. *Generation of Three-Dimensional Unstructured Grids by the Advancing-Front Method*. AIAA Paper 88-0515, 1988.

- [77] H. Luo, J.D. Baum, and R. Löhner. *A Finite Volume Scheme for Hydrodynamic Free Boundary Problems on Unstructured Grids*. AIAA Paper 95-0668, 1995.
- [78] S. Majumdar. *Role of Underrelaxation in Momentum Interpolation for Calculation of Flow with NonStaggered Grids*. Numerical Heat Transfer, 13:125-132, 1988.
- [79] S. Majumdar, W. Rodi, and J. Zhu. *Three-Dimensional Finite-Volume Method for Incompressible Flows with Complex Boundaries*. Journal of Fluids Engineering, Transactions of the ASME, 114, 496-503, 1992.
- [80] M.L. Mansour and A. Hamed. *Implicit Solution of the Incompressible Navier-Stokes Equations on a Non-staggered Grid*. Journal of Computational Physics, 86:147-167, 1990.
- [81] D. Martin and R. Löhner. *An Implicit Linelet-Based Solver For Incompressible Flows*. AIAA Paper 92-0668, 1992.
- [82] Y.P. Marx. *Time Integration Schemes for the Unsteady Incompressible Navier-Stokes Equations*. Journal of Computational Physics, 112:182-209, 1994.
- [83] M.C. Melaaen. *Analysis of Fluid Flow in Constricted Tubes and Ducts Using Body-Fitted Non-staggered Grids*. International Journal for Numerical Methods in Fluids, 15:895-923, 1991.
- [84] C.L. Merkle and M. Athavale. *Time-Accurate Unsteady Incompressible Flow Algorithms Based on Artificial Compressibility*. AIAA Paper 87-1137, 1987.
- [85] W. E. Milholen II and N. Chokani. *Low Mach Number Performance of Three-Dimensional Compressible Navier-Stokes Codes*. AIAA Paper 95-0767, 1995.

- [86] T.F. Miller and F.W. Schmidt. *Use of a Pressure-Weighted Interpolation Method for the Solution of the Incompressible Navier-Stokes Equations on a Nonstaggered Grid System*. Numerical Heat Transfer, 14:213-233, 1988.
- [87] A. Mizukami and M. Tsuchiya. *A Finite Element Method for the Three-Dimensional Non-steady Navier-Stokes Equations*. International Journal for Numerical Methods in Fluids, 4:349-357, 1984.
- [88] A. Mukhopadhyay, G. Biswas, and T. Sundararajan. *Numerical Investigation of Confined Wakes Behind a Square Cylinder in a Channel*. International Journal for Numerical Methods in Fluids, 14:1473-1484, 1992.
- [89] S. Murashige, T. Kinoshita and M. Hinatsu. *Direct Calculations of the Navier-Stokes Equations for Forces Acting on a Cylinder in Oscillatory Flow*. Proceedings of the Eighth International Conference on Offshore Mechanics and Arctic Engineering, Vol II, 1989.
- [90] K. Nakahashi. *Optimum Spacing Control of the Marching Grid Generation*. AIAA Paper 91-0103, 1991.
- [91] K. Nakajima. *Incompressible Navier-Stokes Methods with Hybrid Adaptive Grids*. MS Thesis, Department of Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin, May 1993.
- [92] I. Nakamura. *Steady Wake Behind a Sphere*. The Physics of Fluids, 19:1,5-8, 1976.
- [93] E.D. Obasaju, P.W. Bearman, and J.M.R. Graham. *A Study of Forces, Circulation and Vortex Patterns Around a Circular Cylinder in Oscillating Flow*. Journal of Fluid Mechanics, 196:467-494, 1988.

- [94] T. Oguni, T. Murata, T. Miyoshi, J.J. Dongarra and H. Hasegawa. *Matrix Computation Software*. Maruzen, 1991.
- [95] H. Ok and D.S. Eberhardt. *Solution of Unsteady Incompressible Navier-Stokes Equations Using an LU Decomposition Scheme*. AIAA 10th Computational Fluid Dynamics Conference, Honolulu, Hawaii, pages 955-956, June 1991.
- [96] G.A. Osswald, K.N. Ghia, and U. Ghia. *A Direct Algorithm for Solution of Incompressible Three-Dimensional Unsteady Navier-Stokes Equations*. AIAA Paper 87-1139, 1987.
- [97] R.L. Panton. *Incompressible Flow*. John Wiley & Sons, 1984.
- [98] V. Parthasarathy, Y. Kallinderis, and K. Nakajima. *Hybrid Adaptation Method and Directional Viscous Multigrid with Prismatic-Tetrahedral Meshes*. AIAA Paper 95-0670, 1995.
- [99] S. Patankar and D. Spalding. *A Calculation Procedure for Heat, Mass and Momentum Transfer in Three-Dimensional Parabolic Flows*. Int. J. Heat and Mass Transfer, 15:1787-1806, 1972.
- [100] S.V. Patankar. *Numerical Heat Transfer and Fluid Flow*. Hemisphere, 1980.
- [101] J. Peraire, J. Peiro, L. Formaggia, K. Morgan and O.C. Zienkiewicz. *Finite Element Euler Computations in Three Dimensions*. AIAA Paper 88-0032, 1988.
- [102] R.M. Perić, G. Kessler and G. Scheuerer. *Comparison of Finite-Volume Numerical Methods with Staggered and Collocated Grids*. Computers & Fluids, 16:4,389-403, 1988.

- [103] D.A. Peshkin and R.K. Cooper. *On the Pressure Neumann Problem in the Navier-Stokes Equations*. AIAA Paper 93-3342, 1993.
- [104] T.H. Pulliam. *Artificial Dissipation Models for the Euler Equations*. AIAA Paper 85-0438, 1985.
- [105] G.D. Raithby and G. E. Schneider. *Numerical Solution of Problems in Incompressible Fluid Flow: Treatment of the Velocity-Pressure Coupling*. Numerical Heat Transfer, 2:417-440, 1979.
- [106] C.M. Rhie. *A Pressure Based Navier-Stokes Solver Using the Multigrid Method*. AIAA Paper 86-0207, 1986.
- [107] C.M. Rhie and W.L. Chow. *Numerical Study of the Turbulent Flow Past an Airfoil with Trailing Edge Separation*. AIAA Journal, 21:11,1525-1532, 1983.
- [108] W.J. Rider. *Filtering Nonsolenoidal Modes in Numerical Solutions of Incompressible Flows*. Technical Report LA-UR-3014, Los Alamos National Laboratory, 1994. Available on World Wide Web at <http://www.c3.lanl.gov/cic3/publications/main.shtml>.
- [109] W.J. Rider, D.B. Kothe, S.J. Mosso, J.H. Cerutti and J.I. Hochstein. *Accurate Solution Algorithms for Incompressible Multiphase Flows*. AIAA Paper 95-0699, 1995.
- [110] Y. Rimon and S.I. Cheng. *Numerical Solution of a Uniform Flow over a Sphere at Intermediate Reynolds Numbers*. The Physics of Fluids, 12:5,949-959, 1969.
- [111] S.E. Rogers. *A Comparison of Implicit Schemes for the Incompressible Navier-Stokes Equations with Artificial Compressibility*. AIAA Paper 95-0567, 1995.

- [112] S.E. Rogers, D. Kwak. *Upwind Differencing Scheme for the Time-Accurate Incompressible Navier-Stokes Equations*. AIAA Journal, 28:2,253-262, 1990.
- [113] S.E. Rogers, D. Kwak, and U. Kaul. *On the Accuracy of the Pseudocompressibility Method in Solving the Incompressible Navier-Stokes Equations*. Applied Mathematics and Modelling, 11:35-44, 1987.
- [114] S.E. Rogers, D. Kwak, and C. Kiris. *Steady and Unsteady Solutions of the Incompressible Navier-Stokes Equations*. AIAA Journal, 29:4,603-610, 1991.
- [115] F.W. Roos and W.W. Willmarth. *Some Experimental Results on Sphere and Disk Drag*. AIAA Journal, 9:2,285-291, 1971.
- [116] A. Rubel and G. Volpe. *Biharmonic Vector Stream Function Formulation and Multigrid Solution For a Three-Dimensional Driven-Cavity Stokes Flow*. AIAA Paper 89-1968-CP, 1989.
- [117] H. Sakamoto and H. Haniu. *A Study on Vortex Shedding From Spheres in a Uniform Flow*. Journal of Fluids Engineering; Transactions of the ASME, 112:386-392, 1990.
- [118] T. Sarpkaya. *Force on a Circular Cylinder in Viscous Oscillatory Flow at Low Keulegan-Carpenter Numbers*. Journal of Fluid Mechanics, 165:61-71, 1986.
- [119] T. Sarpkaya and C. Putzig. *Vortex Trajectories Around a Circular Cylinder in Oscillatory Plus Mean Flow*. Proceedings of the Offshore Mechanics and Arctic Engineering Conference. Vol 1-A Offshore Technology:69-77, 1992.
- [120] H. Schlichting. *Boundary-Layer Theory*. McGraw-Hill, Inc, 1987.

- [121] C. Sheng, L. Taylor and D. Whitfield. *Multiblock Multigrid Solution of Three-Dimensional Incompressible Turbulent Flows About Appended Submarine Configurations*. AIAA Paper 95-0203, 1995.
- [122] K.M. Smith, W.K. Cope, and S.P. Vanka. *A Multigrid Procedure for Three-Dimensional Flows on Non-Orthogonal Collocated Grids*. International Journal for Numerical Methods in Fluids, 17:887-904, 1993.
- [123] W.Y. Soh and S.A. Berger. *Fully Developed Flow in a Curved Pipe of Arbitrary Curvature Ratio*. International Journal for Numerical Methods in Fluids, 7:733-755, 1987.
- [124] F. Sotiropoulos, and S. Abdallah. *The Discrete Continuity Equation in Primitive Variable Solutions of Incompressible Flow*. Journal of Computational Physics, 95:212-227, 1991.
- [125] F. Sotiropoulos, and S. Abdallah. *A Primitive Variable Method for the Solution of Three-Dimensional Incompressible Viscous Flows*. Journal of Computational Physics, 103:336-349, 1992.
- [126] F. Sotiropoulos, W.J. Kim and V.C. Patel. *A Computational Comparison of Two Incompressible Navier-Stokes Solvers in Three-Dimensional Laminar Flows*. Computers Fluids, 23:4,627-646, 1994.
- [127] J.C. Strikwerda and Y.M. Nagel. *A Numerical Method for the Incompressible Navier-Stokes Equations in Three-Dimensional Cylindrical Geometry*. Journal of Computational Physics, 78:64-78, 1988.
- [128] D. Tafti. *A Study of Higher-Order Spatial Finite-Difference Formulations for the Incompressible Navier-Stokes Equations*. 11th AIAA Computational Fluid

Dynamics Conference, Vol 2, Orlando FL, July 1993, A Collection of Technical Papers, Part 2.

- [129] N. Takemitsu. *Finite Difference Method to Solve Incompressible Fluid Flow*. Journal of Computational Physics, 61:499-518, 1985.
- [130] Y. Takemoto and Y. Nakamura. *A Three Dimensional Incompressible Solver*. Lecture Notes in Physics, Tenth International Conference on Numerical Methods in Fluid Dynamics, Proceedings of the Conference Held at the Beijing Science Hall, Beijing, China, June 23-27, 1986. Springer-Verlag, New York.
- [131] H. Tokunaga, N. Satofuka and T. Yoshikawa. *Direct Simulation of High Reynolds Number Flows Using a New Integro-Differential Solver*. AIAA Paper 87-1175, 1987.
- [132] H. Tokunaga, K. Yoyeda, and N. Satofuka. *Direct Simulations of Three-Dimensional Flows Using Generalized Vector Potential Method*. AIAA Paper 91-1610-CP, 1991.
- [133] D.J. Tritton. *Experiments on the Flow Past a Circular Cylinder at Low Reynolds Numbers*. Journal of Fluid Mechanics, 6:547-567, 1960.
- [134] J.P. Van Doormaal and G.D. Raithby. *Enhancements of the SIMPLE Method For Predicting Incompressible Fluid Flows*. Numerical Heat Transfer, 7:147-163, 1984.
- [135] J.A. Vieceilli. *A Computing Method for Incompressible Flows Bounded by Moving Walls*. Journal of Computational Physics, 8:119-143, 1971.

- [136] P. Vijayan and Y. Kallinderis. *A 3-D Finite-Volume Scheme for the Euler Equations on Adaptive Tetrahedral Grids*. Journal of Computational Physics, 113:249-267, 1994.
- [137] X. Wang and C. Dalton. *Oscillating Flow Past a Rigid Circular Cylinder: A Finite-Difference Calculation*. Journal of Fluids Engineering, Transactions of the ASME, 113:377-383, 1991.
- [138] S. Ward and Y. Kallinderis. *Hybrid Prismatic/Tetrahedral Grid Generation for Complex 3-D Geometries*. AIAA Paper 93-0669, 1993.
- [139] F.M. White. *Viscous Fluid Flow, Second Edition*. McGraw-Hill, Inc, New York, NY, 1991.
- [140] M. Williams. *The Helmholtz Pressure Method with Algorithms for Nonlinear Fluid-Structure Interaction and Unstructured Meshes* AIAA Paper 91-1565-CP, 1991.
- [141] P.T. Williams and A.J. Baker. *A Well-Posed Weak Statement Algorithm for the Pressure Poisson Equation for Incompressible Flows*. AIAA Paper 93-3340-CP, 1993.
- [142] C.H.K. Williamson. *Sinusoidal Flow Relative to Circular Cylinders*. Journal of Fluid Mechanics, 155:141-174, 1985.
- [143] J. Wu, Z.J. Zhu, J. Szmelter and O.C. Zienkiewicz. *Error Estimation and Adaptation in Navier-Stokes Incompressible Flows*. Comp. Mech., 6:259-70, 1990.
- [144] M.A. Yerry and M. Shephard. *Automatic 3-D Mesh Generation by the Modified Octree Technique*. International Journal of Numerical Methods in Engineering, 20:1965-1990, 1984.

- [145] R.W. Yeung and P. Ananthakrishnan. *Oscillation of a Floating Body in a Viscous Fluid*. Journal of Engineering Mathematics, 26:211-230, 1992.
- [146] S. Yoon, D. Kwak, L. Chang. *LU-SGS Implicit Algorithm for Three-Dimensional Incompressible Navier-Stokes Equations with Source Term*. AIAA Paper 89-1964-CP, 1989.
- [147] Y. Zang, R.L. Street, and J.R. Koseff. *A Non-staggered Grid, Fractional Step Method for Time-Dependent Incompressible Navier-Stokes Equations in Curvilinear Coordinates*. Journal of Computational Physics, 114:18-33, 1994.
- [148] O.C. Zienkiewicz, and R.L. Taylor. *The Finite Element Method, Fourth Edition, Volume 1 Basic Formulation and Linear Problems*. McGraw-Hill Book Company, 1989.

Vita

Alice was born on November 20, 1958 in Chattanooga, Tennessee, to Paul and Catherine Chen. She graduated from *Carlsbad High School*, Carlsbad, CA in 1976. She received a Bachelor of Arts degree in Chemistry at the *University of California; San Diego* in 1980. She also earned a commission as an officer in the United States Air Force (USAF) in 1980. In 1983 she married Robert D. Hunt, an officer in the USAF. Alice's first assignment in the Air Force was to the *Air Force Institute of Technology* (AFIT), where she earned a Bachelor of Science degree in Aeronautical Engineering in 1982. Her next assignment was to Eglin AFB, FL, where she was a Store Separations Engineer for the F-16 and F-4 Aircraft. She returned to AFIT in 1984 and earned her Master of Science degree in 1986. The experimental portion of her master's thesis work was completed at the *von Karman Institute for Fluid Dynamics* in Rhode-St-Genese, Belgium. Her next assignment was at the Air Force Operational Test and Evaluation Center at Kirtland AFB, NM, where she was the F-15E Aircraft and LANTIRN System Reliability and Maintainability Manager. In 1989 she was assigned to the United States Air Force Academy (USAF), Colorado Springs, CO, as an instructor in the Department of Aeronautics. In 1991 she was promoted to the rank of Assistant Professor at USAFA. She entered the Graduate School of *The University of Texas at Austin* in August 1992.

Permanent Address: 14900 Sierra Way

Anchorage, AK 99516

This dissertation was typeset with $\text{\LaTeX} 2_{\epsilon}$ ¹ by the author.

¹ $\text{\LaTeX} 2_{\epsilon}$ is an extension of $\text{\LaTeX}$. $\text{\LaTeX}$ is a collection of macros for $\text{\TeX}$. $\text{\TeX}$ is a trademark of the American Mathematical Society. The macros used in formatting this dissertation were written by Dinesh Das, Department of Computer Sciences, The University of Texas at Austin.