

COMPUTATION OF EXTERNAL FLOW AROUND ROTATING BODIES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

L. OKTAY GÖNÇ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILISOPHY
IN
THE DEPARTMENT OF MECHANICAL ENGINEERING

MARCH 2005

Approval of the Graduate School of Natural and Applied Sciences

Prof. Dr. Canan ÖZGEN
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Doctor of Philosophy.

Prof. Dr. S. Kemal İDER
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Doctor of Philosophy.

Dr. Mehmet Ali AK
Co-Supervisor

Prof. Dr. M. Haluk AKSEL
Supervisor

Examining Committee Members

Prof. Dr. Zafer DURSUNKAYA	(METU, ME)	_____
Prof. Dr. M. Haluk AKSEL	(METU, ME)	_____
Prof. Dr. İsmail Hakkı TUNCER	(METU, AEE)	_____
Dr. Mehmet Ali AK	TÜBİTAK-SAGE	_____
Prof. Dr. Haşmet TÜRKOĞLU	(Gazi Un., ME)	_____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name : L. Oktay GÖNÇ
Signature :

ABSTRACT

COMPUTATION OF EXTERNAL FLOW AROUND ROTATING BODIES

GÖNÇ, L. Oktay

Ph. D., Department of Mechanical Engineering

Supervisor: Prof. Dr. M. Haluk AKSEL

Co-Supervisor: Dr. Mehmet Ali AK

March 2005, 236 pages

A three-dimensional, parallel, finite volume solver which uses Roe's upwind flux differencing scheme for spatial and Runge-Kutta explicit multistage time stepping scheme for temporal discretization on unstructured meshes is developed for the unsteady solution of external viscous flow around rotating bodies. The main aim of this study is to evaluate the aerodynamic dynamic stability derivative coefficients for rotating missile configurations.

Arbitrary Lagrangian Eulerian (ALE) formulation is adapted to the solver for the simulation of the rotation of the body. Eigenvalues of the Euler equations in ALE form has been derived. Body rotation is simply performed by rotating the entire computational domain including the body of the projectile by means of rotation matrices. Spalart-Allmaras one-equation turbulence model is implemented to the solver.

The solver developed is first verified in 3-D for inviscid flow over two missile configurations. Then inviscid flow over a rotating missile is tested. Viscous flux computation algorithms and Spalart-Allmaras turbulence model implementation are validated in 2-D by performing calculations for viscous flow over flat plate, NACA0012 airfoil and NLR 7301 airfoil with trailing edge flap. Then ALE formulation is validated in 2-D on a rapidly pitching NACA0012 airfoil. Afterwards three-dimensional validation studies for viscous, laminar and turbulent flow calculations are performed on 3-D flat plate problem. At last, as a validation test case, unsteady laminar and turbulent viscous flow calculations over a spinning M910 projectile configuration are performed. Results are qualitatively in agreement with the analytical solutions, experimental measurements and previous studies for steady and unsteady flow calculations.

Keywords: CFD, Arbitrary Lagrangian Eulerian, Unsteady Aerodynamics, Spalart-Allmaras Turbulence Model, Parallel Processing, Dynamic Stability

ÖZ

DÖNEL CİSİMLER ETRAFINDA DIŞ AKIŞ ÇÖZÜMLEMESİ

GÖNÇ, L. Oktay

Doktora, Makina Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. M. Haluk AKSEL

Ortak Tez Yöneticisi: Dr. Mehmet Ali AK

Mart 2005, 236 sayfa

Dönel cisimler etrafındaki ağdalı dış akışların zamana bağımlı olarak çözülebilmesi için, düzensiz çözüm ağını uzayda Roe'nun yön hassas (upwind) akı ayırımına dayalı yöntemini, zamanda ise Runge-Kutta çok kademeli zamanda ilerleme yöntemini kullanarak ayırıştıran, üç boyutlu, paralel bir sonlu hacim çözücüsü geliştirilmiştir. Çalışmanın temel amacı dönel füze konfigürasyonları için aerodinamik devimsel kararlılık türevi katsayılarının elde edilmesidir.

Gövdenin döndürülmesinin benzetimi için çözücüye Arbitrary Lagrangian Eulerian (ALE) formulasyonu uyarlanmıştır. ALE düzenindeki Euler denklemleri için özdeğerler türetilmiştir. Gövde dönüşü, gövdeyi de içeren çözüm ağının bir bütün olarak ilgili döngü matrisleri sayesinde döndürülmesi ile sağlanmıştır. Bu sayede çözüm ağında herhangi bozulma meydana gelmemiştir. Spalart-Allmaras tek denklemler türbülans modeli çözücüye uyarlanmıştır.

Geliştirilen çözücü ilk olarak iki adet füze konfigurasyonu için üç boyutta ağdasız akış çözümlenmesi gerçekleştirilerek doğrulanmıştır. Dönen füze etrafında ağdasız akış çözümlenmesi yapılmıştır. İki boyutta ağdalı akı değerleri hesaplama algoritmalarının ve Spalart-Allmaras türbülans modeli uygulamasının doğrulanması düz plaka üzerinde, NACA0012 aerodinamik profili ve arkasında flap olan NLR 7301 aerodinamik profili üzerinde gerçekleştirilmiştir. Daha sonra ALE formulasyonu hızlı yunuslama hareketi yapan NACA0012 aerodinamik profili etrafında zamana bağımlı ağdalı akış çözümlenerek doğrulanmıştır. Bundan sonra üç boyutta ağdalı laminar ve türbülanslı akış doğrulama çalışmaları üç boyutlu düz plaka problemi üzerinde gerçekleştirilmiştir. En son olarak, geliştirilen çözücünün onaylanması için, dönel M910 mermi konfigurasyonu etrafındaki ağdalı üç boyutlu türbülanslı akış incelenmiştir. Sonuçlar analitik çözümlenmelerle, deneysel ölçümlerle ve önceki çalışmalarla niceliksel bir uyum içerisinde çıkmıştır.

Anahtar Kelimeler: SAM, Durağan Olmayan Aerodinamik, Spalart-Allmaras Türbülans Modeli, , Paralel İşleme, Devimsel Kararlılık

To My Parents

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to Prof Dr. Ahmet Ş. ÜÇER, Prof. Dr. M. Haluk AKSEL, Prof. Dr. İsmail Hakkı TUNCER and Dr. Mehmet Ali AK, for their endless support, guidance and for helpful discussions we have made throughout the study.

This work has been supported by TÜBİTAK-SAGE. I would like to thank TÜBİTAK-SAGE for providing the computational power and literature sources which were enormously important for this study. I would also like to send my thanks to the TÜBİTAK-SAGE Propulsion Division members for their help and understanding.

I would like to thank my colleague Osman BAŞOĞLU for his unforgettable valuable teamwork and help in my studies.

My special thanks go to Nuray KOT for her endless support and friendship.

Lastly, my deepest thanks go to my family who gave me endless support and love which made this thesis possible.

TABLE OF CONTENTS

PLAGIARISM	iii
ABSTRACT.....	iv
ÖZ	vi
ACKNOWLEDGMENTS	ix
TABLE OF CONTENTS.....	x
LIST OF TABLES	xiv
LIST OF FIGURES	xv
LIST OF SYMBOLS	xxiv
1. INTRODUCTION	1
2. LITERATURE SURVEY	5
2.1. Background on Missile Aerodynamics	5
2.1.1. <i>Stability & Control</i>	8
2.1.1.1. <i>Static Stability</i>	9
2.1.1.2. <i>Dynamic Stability</i>	10
2.2. Numerical Discretization Techniques.....	13
2.2.1. <i>Finite Difference Method (FDM)</i>	13
2.2.2. <i>Finite Element Method (FEM)</i>	13
2.2.3. <i>Finite Volume Method (FVM)</i>	14
2.2.3.1. <i>Cell Vertex Scheme</i>	16
2.2.3.2. <i>Cell Centered Scheme</i>	17

2.3.	Numerical Schemes.....	18
2.3.1.	<i>Spatial Discretization</i>	18
2.3.1.1.	<i>Flux Vector Splitting Type Schemes</i>	20
2.3.1.2.	<i>Flux Difference Splitting (Godunov) Type Schemes</i>	20
2.3.2.	<i>Temporal Discretization</i>	20
2.4.	Computational Grid.....	22
2.4.1.	<i>Structured Grids</i>	23
2.4.2.	<i>Unstructured Grids</i>	24
2.4.3.	<i>Hybrid Grids</i>	26
2.5.	Parallel Processing	27
2.6.	Turbulence	29
2.6.1.	<i>Algebraic (Zero-Equation) Models</i>	30
2.6.2.	<i>One-Equation Models</i>	30
2.6.3.	<i>Two-Equation Models</i>	31
2.6.4.	<i>Other Turbulence Models</i>	32
2.7.	Unsteady Aerodynamics – Implementation of Rotation.....	32
3.	NUMERICAL METHOD.....	35
3.1.	Coordinate System Convention	35
3.2.	Governing Equations.....	35
3.2.1.	<i>Conservation of Mass</i>	37
3.2.2.	<i>Conservation of Momentum</i>	37
3.2.3.	<i>Conservation of Energy</i>	37

3.2.4.	<i>Integral Compact Form of Governing Equations</i>	38
3.2.5.	<i>Non-Dimensionalization of Governing Equations</i>	41
3.3.	Spatial Discretization	42
3.3.1.	<i>Inviscid – Convective Fluxes</i>	43
3.3.2.	<i>Viscous – Diffusive Fluxes</i>	48
3.4.	Temporal Discretization	50
3.5.	Turbulence Model	53
3.6.	Boundary Conditions	59
3.6.1.	<i>Far Field Boundary Condition</i>	60
3.6.2.	<i>Wall Boundary Condition</i>	61
3.6.3.	<i>Symmetry Boundary Condition</i>	62
3.6.4.	<i>Turbulence Model Boundary Conditions</i>	63
3.7.	Implementation of Rotation Into Governing Equations	63
3.8.	Grid Movement	71
3.9.	Parallel Processing	75
4.	VERIFICATION AND VALIDATION STUDIES	83
4.1.	3-D Verification Studies for the Euler Solver	83
4.1.1.	<i>Inviscid Flow over Finned Projectile Configuration</i>	83
4.1.2.	<i>Inviscid Flow over Sparrow Missile</i>	90
4.2.	3-D Validation Study for the Euler-ALE Solver	97
4.3.	2-D Validation Studies for the Navier-Stokes / ALE Solver	103
4.3.1.	<i>Laminar Flow over 2-D Flat Plate</i>	104

4.3.2.	<i>Laminar Flow over NLR 7301 Airfoil With Trailing Edge Flap</i>	111
4.3.3.	<i>Laminar Flow over NACA0012 Airfoil.....</i>	118
4.3.4.	<i>Unsteady Aerodynamics Of Rapidly Pitched Airfoil.....</i>	124
4.3.5.	<i>Turbulent Flow over 2-D Flat Plate.....</i>	133
4.3.6.	<i>Turbulent Flow over NACA0012 Airfoil.....</i>	142
4.4.	3-D Validation Studies for the Navier-Stokes Solver	154
4.4.1.	<i>Laminar Flow Over 3-D Flat Plate</i>	154
4.4.2.	<i>Turbulent Flow over 3-D Flat Plate.....</i>	160
5.	VISCOUS UNSTEADY FLOWS OVER A SPINNING PROJECTILE....	167
5.1.	Laminar Flow Solutions over Non-Spinning M910 Projectile	171
5.2.	Laminar Flow Solutions over Spinning M910 Projectile	178
5.3.	Turbulent Flow Solutions over Non-Spinning M910 Projectile	185
5.4.	Turbulent Flow Solutions over Spinning M910 Projectile	194
6.	CONCLUSION & DISCUSSION	201
	REFERENCES.....	206
	APPENDIX A	220
	SAMPLE INPUT DATA	220
	APPENDIX B	221
	GRID FILE FORMAT	221
	APPENDIX C	222
	UNSTRUCTURED GRID GENERATION FOR A PROJECTILE	222
	APPENDIX D	228
	MIMIMUM WALL DISTANCE CALCULATION IN THE SPALART- ALLMARAS TURBULENCE MODEL	228
	CURRICULUM VITAE	232

LIST OF TABLES

Table 1. Nondimensional Variables	42
Table 2. Nondimensional Form of Free Stream Variables	42
Table 3. Solution Matrix for Finned Projectile	86
Table 4. Solution Matrix for Sparrow Missile	92
Table 5. Solution Matrix for Laminar Flow Calculations.....	172
Table 6. Connectivity for Three Tetrahedral Elements Obtained from the Prismatic Element at the Nose of the Projectile	225
Table 7. Connectivity for Three Tetrahedral Elements Obtained from the Prismatic Element at the Base of the Projectile	226
Table 8. Connectivity for Six Tetrahedral Elements Obtained from a Hexahedral Element	227

LIST OF FIGURES

Figure 1. Aerodynamic Moments and Conventional Coordinate System on a Projectile	1
Figure 2. Standard Missile Configuration.....	6
Figure 3. Wrapped Around Fin Configuration.....	7
Figure 4. Illustration of Static Stability [8].....	9
Figure 5. Monotonic & Oscillatory Dynamic Stability [8].....	11
Figure 6. Dynamically Unstable Behaviour [8].....	11
Figure 7. 2-D and 3-D Cell Vertex Median Dual Cells	16
Figure 8. 2-D and 3-D Cell Centered Median Dual Cells.....	17
Figure 9. Sample 3-D Structured Grid	23
Figure 10. Sample 3-D Unstructured Grid.....	25
Figure 11. Sample 3-D Structured - Unstructured Hybrid Grid.....	26
Figure 12. Coordinate System Convention in x-z Plane	36
Figure 13. Coordinate System Convention in x-y Plane.....	36
Figure 14. Coordinate System Convention – Three Dimensional View.....	37
Figure 15. Reconstruction Scheme for Cell-Centered Approach [9].....	47
Figure 16. 3-D Velocity Gradient Computation Stencil at Cell Centroid.....	50
Figure 17. 3-D Face Viscous Stress Calculation Stencil.....	51
Figure 18. Illustration of Right & Left Going Waves and Riemann Invariants.....	61
Figure 19. One Dimensional 1 st Order Finite Volume Description	64
Figure 20. Motion of Fluid across the Cell Boundary	65
Figure 21. Motion of the Cell Boundary with respect to Stationary Fluid	66
Figure 22. Schematic Figure of Rotational Motion of a Missile	72

Figure 23. Domain Decomposition and Interface Boundary Concept.....	76
Figure 24. Sample Domain Decomposition for a 2-D Computational Domain.....	76
Figure 25. Data Flow between Executables “MASTER” and “WORKER”	82
Figure 26. Detailed Geometry of the Finned Projectile and the Coordinate Axes (all dimensions are in mm).....	84
Figure 27. Unstructured Volume Mesh around the Finned Projectile	85
Figure 28. Unstructured Mesh Details on the Surface of the Finned Projectile	85
Figure 29. Finned Projectile-Variation of Drag Coefficient w.r.t. Angle of Attack	87
Figure 30. Finned Projectile-Variation of Lift Coefficient w.r.t. Angle of Attack..	88
Figure 31. Finned Projectile-Variation of Pitching Moment Coefficient w.r.t. Angle of Attack.....	89
Figure 32. Detailed Geometry of Sparrow Missile and the Coordinate Axes (all dimensions are in mm).....	91
Figure 33. Unstructured Volume Mesh around Sparrow Missile	91
Figure 34. Unstructured Mesh Details on the Surface of Sparrow	92
Figure 35. Sparrow-Variation of Axial Force Coefficient w.r.t. Angle of Attack...	94
Figure 36. Sparrow-Variation of Normal Force Coefficient w.r.t. Angle of Attack	95
Figure 37. Sparrow-Variation of Pitching Moment Coefficient w.r.t. Angle of Attack	96
Figure 38. Steady-State Non-Dimensional Pressure Contours on the Projectile.....	98
Figure 39. Variation of Roll Moment Coefficient w.r.t. Rotation of the Projectile.	99
Figure 40. Non-Dimensional Pressure Distribution Near Fins ($\theta = 3.5^\circ$)	101
Figure 41. Non-Dimensional Pressure Distribution Near Fins ($\theta = 10.6^\circ$)	101
Figure 42. Non-Dimensional Pressure Distribution Near Fins ($\theta = 21.3^\circ$)	102
Figure 43. Non-Dimensional Pressure Distribution Near Fins ($\theta = 39^\circ$)	102
Figure 44. Non-Dimensional Pressure Distribution Near Fins ($\theta = 60.3^\circ$)	103

Figure 45. Unstructured Viscous Mesh for Laminar Flat Plate Problem.....	105
Figure 46. Mesh Details at the Leading and Trailing Edges of Flat Plate	105
Figure 47. Residual History of Laminar Flow over a 2-D Flat Plate.....	106
Figure 48. Laminar Boundary Layer Development at the Leading Edge of Flat Plate	107
Figure 49. Laminar Boundary Layer Development at the Leading Edge of Flat Plate (Streamlines).....	107
Figure 50. Overall Laminar Boundary Layer Development on the Flat Plate.....	108
Figure 51. Axial Velocity Distribution at 10%, 25%, 50%, 75% and 90% of the Flat Plate in Comparison with Blassius Solution.....	109
Figure 52. Skin Friction Coefficient vs. Axial Location.....	110
Figure 53. Skin Friction Coefficient vs. Reynolds Number	111
Figure 54. Unstructured Viscous Mesh for NLR Airfoil.....	112
Figure 55. Mesh Details between NLR Airfoil and Trailing Edge Flap.....	113
Figure 56. Residual Histories of Laminar Flow Over NLR Airfoil.....	114
Figure 57. Non-Dimensional Pressure Contours and Streamlines on NLR Airfoil ($\alpha = 0^\circ$).....	114
Figure 58. Non-Dimensional Pressure Contours and Streamlines on NLR Airfoil ($\alpha = 6^\circ$).....	115
Figure 59. Non-Dimensional Pressure Contours and Streamlines on NLR Airfoil ($\alpha = 10.1^\circ$).....	115
Figure 60. Non-Dimensional Pressure Contours and Streamlines on NLR Airfoil ($\alpha = 13.1^\circ$).....	116
Figure 61. Pressure Coefficient Distribution on NLR Airfoil and Trailing Flap ($\alpha = 6^\circ$).....	117
Figure 62. Pressure Coefficient Distribution on NLR Airfoil and Trailing Flap ($\alpha = 10.1^\circ$).....	117
Figure 63. Pressure Coefficient Distribution on NLR Airfoil and Trailing Flap ($\alpha = 13.1^\circ$).....	118

Figure 64. Unstructured Viscous Mesh for Laminar Flow over NACA0012 Airfoil	119
Figure 65. Viscous Mesh Detail nearby NACA0012 Airfoil.....	120
Figure 66. Mesh Details at the Leading and Trailing Edges of NACA0012 Airfoil	120
Figure 67. Residual History for Laminar Flow over NACA0012 Airfoil	121
Figure 68. Laminar Boundary Layer Development on NACA0012 Airfoil, M=0.5	122
Figure 69. Mach Number Contours for Laminar Flow Over NACA0012 Airfoil, Mavripilis and Jameson [101].....	122
Figure 70. Mach Number Contours for Laminar Flow Over NACA0012 Airfoil, Present Study.....	123
Figure 71. Pressure Coefficient Comparison for Laminar Flow over NACA0012 Airfoil, Re = 5,000	124
Figure 72. Residual History for Steady State NACA0012 Solution.....	125
Figure 73. Steady State Mach Number Distribution Around NACA0012 Airfoil	126
Figure 74. Laminar Boundary Layer Development on NACA0012 Airfoil, M=0.2	127
Figure 75. Lift Coefficient Variation w.r.t. Incidence Angle of Rapidly Pitched NACA0012 Airfoil	128
Figure 76. Drag Coefficient Variation w.r.t Incidence Angle of Rapidly Pitched NACA0012 Airfoil	128
Figure 77. Streamlines of NACA0012 Airfoil at $\alpha = 17.7^\circ$	129
Figure 78. Streamlines of NACA0012 Airfoil at $\alpha = 26.7^\circ$	130
Figure 79. Streamlines of NACA0012 Airfoil at $\alpha = 34.4^\circ$	130
Figure 80. Streamlines of NACA0012 Airfoil at $\alpha = 34.4^\circ$ (After rotation stopped)	131
Figure 81. Vortex Shedding Behind Pitching NACA0012 Airfoil	132
Figure 82. Unstructured Viscous Mesh for Turbulent Flat Plate Problem	134

Figure 83. Mesh Details at the Leading Edge of Flat Plate	134
Figure 84. Residual History of Turbulent Flow over a 2-D Flat Plate.....	135
Figure 85. Turbulent Boundary Layer Development at the Middle of Flat Plate..	136
Figure 86. Variation of Turbulent Viscosity Variable over the Turbulent Flat Plate	136
Figure 87. Variation of Turbulent Viscosity Variable at the Middle of Flat Plate (Close-up View).....	137
Figure 88. Comparison of Laminar and Turbulent Boundary Layers over Flat Plate	137
Figure 89. Turbulent Axial Velocity Distribution at 10%, 25%, 50%, 75% and 90% of the Flat Plate in Comparison with the Approximate Solution.....	138
Figure 90. Non-Dimensional Velocity Profile for Turbulent Flow Over a Flat Plate and Identification of Different Regions within the Turbulent Boundary Layer [103].....	140
Figure 91. Turbulent Skin Friction Coefficient vs. Axial Location.....	141
Figure 92. Unstructured Viscous Mesh for Turbulent Flow over NACA0012 Airfoil	143
Figure 93. Viscous Mesh Detail nearby NACA0012 Airfoil.....	143
Figure 94. Mesh Details at the Leading and Trailing Edges of NACA0012 Airfoil	144
Figure 95. Residual History of Turbulent Flow over a NACA0012 Airfoil.....	144
Figure 96. Mach Number Contours for Turbulent Flow Over NACA0012 Airfoil, Present Study.....	145
Figure 97. Mach Number Contours for Turbulent Flow Over NACA0012 Airfoil, FLUENT Solver.....	146
Figure 98. Turbulent Viscosity Contours for Flow over NACA0012 Airfoil, Present Study	147
Figure 99. Turbulent Viscosity Contours for Flow over NACA0012 Airfoil, FLUENT Solver.....	147

Figure 100. Velocity Profiles at the Upper and Lower Surfaces of NACA0012 Airfoil, $x/L = 0.2$	148
Figure 101. Velocity Profiles at the Upper and Lower Surfaces of NACA0012 Airfoil, $x/L = 0.5$	149
Figure 102. Velocity Profiles at the Upper and Lower Surfaces of NACA0012 Airfoil, $x/L = 0.8$	149
Figure 103. Velocity Profiles at the Trailing Edge of NACA0012 Airfoil.....	150
Figure 104. Turbulent Viscosity Profiles at the Upper and Lower Surfaces of NACA0012 Airfoil, $x/L = 0.2$	151
Figure 105. Turbulent Viscosity Profiles at the Upper and Lower Surfaces of NACA0012 Airfoil, $x/L = 0.5$	151
Figure 106. Turbulent Viscosity Profiles at the Upper and Lower Surfaces of NACA0012 Airfoil, $x/L = 0.8$	152
Figure 107. Turbulent Viscosity Profiles at the Trailing Edge of NACA0012 Airfoil	152
Figure 108. Pressure Coefficient Comparison for Turbulent Flow over NACA0012 Airfoil, $Re = 1.86 \times 10^6$	153
Figure 109. 3-D Unstructured Mesh for Flat Plate	155
Figure 110. 3-D Unstructured Mesh Detail at the Leading Edge of Flat Plate.....	156
Figure 111. Residual History for Solution of 3-D Flat Plate Problem.....	156
Figure 112. Laminar Boundary Layer Development at the Leading Edge of 3-D Flat Plate ($y = 0.025m$).....	157
Figure 113. Axial Velocity Distribution at 25% of the 3-D Plate.....	158
Figure 114. Axial Velocity Distribution at 50% of the 3-D Plate.....	159
Figure 115. Axial Velocity Distribution at 75% of the 3-D Plate.....	159
Figure 116. Axial Velocity Distribution at 90% of the 3-D Plate.....	160
Figure 117. 3-D Unstructured Mesh for Flat Plate – Turbulent.....	162
Figure 118. 3-D Unstructured Mesh Detail at the Leading Edge of Flat Plate – Turbulent.....	162

Figure 119. Residual History for Solution of Turbulent 3-D Flat Plate Problem..	163
Figure 120. Turbulent Boundary Layer Development at the Middle of 3-D Flat Plate (y = 0.025m).....	163
Figure 121. Variation of Turbulent Viscosity Variable at the Middle of 3-D Flat Plate (y = 0.025m).....	164
Figure 122. Axial Velocity Distribution at 25% of Turbulent 3-D Plate.....	165
Figure 123. Axial Velocity Distribution at 50% of Turbulent 3-D Plate.....	165
Figure 124. Axial Velocity Distribution at 75% of Turbulent 3-D Plate.....	166
Figure 125. Axial Velocity Distribution at 90% of Turbulent 3-D Plate.....	166
Figure 126. Detailed Geometry of M910 Projectile and the Coordinate Axes (all dimensions are in mm).....	168
Figure 127. Unstructured Volume Mesh on M910 Projectile.....	170
Figure 128. Unstructured Mesh Details on the Surface of M910 Projectile.....	170
Figure 129. Unstructured Mesh Details on M910 Projectile in Vertical Direction to Body Axis	171
Figure 130. Convergence Histories of the Axial Force Coefficient for Non-Spinning Projectile, Laminar Flow Calculations.....	172
Figure 131. Residual Histories for Non-Spinning Projectile Laminar Flow Calculations.....	173
Figure 132. Mach Number Contours for Non-Spinning Projectile Laminar Flow Calculations, pitch plane, M = 0.4, Re = $9.32 \times 10^{+6}$	173
Figure 133. Mach Number Contours for Non-Spinning Projectile Laminar Flow Calculations, yaw plane, M = 0.4, Re = $9.32 \times 10^{+6}$	174
Figure 134. Streamlines Colored by Mach Number for Non-Spinning Projectile Laminar Flow Calculations, M = 0.4, Re = $9.32 \times 10^{+6}$	174
Figure 135. Laminar Boundary Layer Development on the Surface of M910 Projectile, M = 0.4, Re = $9.32 \times 10^{+6}$	175
Figure 136. Mach Number Contours for Non-Spinning Projectile Laminar Flow Calculations, pitch plane, M = 0.6, Re = $1.40 \times 10^{+7}$	175

Figure 137. Mach Number Contours for Non-Spinning Projectile Laminar Flow Calculations, pitch plane, $M = 1.2$, $Re = 2.80 \times 10^{+7}$	176
Figure 138. Zero angle of attack Drag Force Coefficient vs Mach Number	177
Figure 139. Roll Moment Coefficient Variation with Rotation for Laminar Flow, $M = 0.4$, $Re = 9.32 \times 10^{+6}$	179
Figure 140. Laminar Flow Streamlines on M910 Projectile, $\theta = 0.1^\circ$	181
Figure 141. Laminar Flow Streamlines on M910 Projectile, $\theta = 1^\circ$	181
Figure 142. Laminar Flow Streamlines on M910 Projectile, $\theta = 3^\circ$	182
Figure 143. Laminar Flow Streamlines on M910 Projectile, $\theta = 6^\circ$	182
Figure 144. Laminar Flow Streamlines at the Base of M910 Projectile, $\theta = 0.1^\circ$	183
Figure 145. Laminar Flow Streamlines at the Base of M910 Projectile, $\theta = 1^\circ$...	183
Figure 146. Laminar Flow Streamlines at the Base of M910 Projectile, $\theta = 3^\circ$...	184
Figure 147. Laminar Flow Streamlines at the Base of M910 Projectile, $\theta = 6^\circ$...	184
Figure 148. Convergence History of the Drag Force Coefficient for Non-Spinning Projectile, Turbulent Flow Calculations	186
Figure 149. Convergence History of the Roll Moment Coefficient for Non-Spinning Projectile, Turbulent Flow Calculations	186
Figure 150. Convergence History of the Yaw Moment Coefficient for Non-Spinning Projectile, Turbulent Flow Calculations	187
Figure 151. Mach Number Contours for Non-Spinning Projectile, Turbulent Flow Calculations, pitch plane, $M = 0.4$, $Re = 9.32 \times 10^{+6}$, $\alpha = 3^\circ$	188
Figure 152. Turbulent Viscosity Contours for Non-Spinning Projectile, pitch plane, $M = 0.4$, $Re = 9.32 \times 10^{+6}$	188
Figure 153. Turbulent Viscosity Contours for Non-Spinning Projectile, pitch plane, $M = 0.4$, $Re = 9.32 \times 10^{+6}$, $\alpha = 3^\circ$, Reduced Contours	189
Figure 154. Turbulent Boundary Layer Development on the Surface of M910 Projectile, $M = 0.4$, $Re = 9.32 \times 10^{+6}$	189
Figure 155. Variation of Turbulent Viscosity Variable on the Surface of M910 Projectile, $M = 0.4$, $Re = 9.32 \times 10^{+6}$	190

Figure 156. Comparison of Laminar and Turbulent Boundary Layers over M910 Projectile, $M = 0.4$, $Re = 9.32 \times 10^{+6}$	191
Figure 157. Mach Number Contours for Non-Spinning Projectile, Turbulent Flow Calculations, pitch plane, $M = 0.6$, $Re = 1.40 \times 10^{+7}$	192
Figure 158. Mach Number Contours for Non-Spinning Projectile, Turbulent Flow Calculations, pitch plane, $M = 1.2$, $Re = 2.80 \times 10^{+7}$	192
Figure 159. Drag Force Coefficient at 3° angle of attack vs Mach Number	193
Figure 160. Roll Damping Coefficient vs Mach Number at $\alpha = 3^\circ$	195
Figure 161. Magnus Moment Coefficient Derivative vs Mach Number at $\alpha = 3^\circ$	197
Figure 162. Turbulent Flow Streamlines on M910 Projectile, $\theta = 0.1^\circ$	199
Figure 163. Turbulent Flow Streamlines on M910 Projectile, $\theta = 1^\circ$	199
Figure 164. Turbulent Flow Streamlines on M910 Projectile, $\theta = 2^\circ$	200
Figure 165. 2-D Structured Computational Mesh Around a Projectile	222
Figure 166. 3-D Structured Computational Mesh Around a Projectile	223
Figure 167. Hexahedral and Prismatic Elements on 3-D Structured Mesh	224
Figure 168. Generating Tetrahedral Elements from the Prismatic Element at the Nose of the Projectile	225
Figure 169. Generating Tetrahedral Elements from the Prismatic Element at the Base of the Projectile	226
Figure 170. Generating Tetrahedral Elements from a Hexahedral Element	227
Figure 171. Minimum Wall Distance Calculation for Highly Stretched Elements Near Wall Boundary	230
Figure 172. Minimum Wall Distance Calculation for Highly Stretched Elements Near Wall Boundary - Corrected	231

LIST OF SYMBOLS

A	Flux Jacobian
ALE	Arbitrary Lagrangian Eulerian
c	Speed of Sound
C_A	Axial Force Coefficient
C_D	Drag Force Coefficient
C_{f_x}	Skin Friction Coefficient
CFD	Computational Fluid Dynamics
CFL	Courant Number
C_l	Rolling Moment Coefficient
C_{l_p}	Roll Damping Moment Derivative
C_{l_0}	Static Roll Producing Moment Coefficient
C_L	Lift Force Coefficient
C_m	Pitching Moment Coefficient
C_{m_q}	Pitch Damping Derivative
C_{m_α} , C_{n_β}	Static Stability Moment Derivative
$C_{m_{\beta p}}$, $C_{n_{\alpha p}}$	Magnus Moment Derivative
C_{m_β} , C_{n_α}	Out of Plane Static Stability Moment Derivative
C_n	Yawing Moment Coefficient
C_N	Normal Force Coefficient
C_p	Pressure Coefficient
C_Y	Side Force Coefficient

C_{Y_p}	Magnus Force Coefficient
DES	Detached-Eddy Simulation
E	Total Energy per Volume
\vec{F}	Convective Flux Vector
F, G, H	Flux Vectors in x, y and z directions
$\vec{i}, \vec{j}, \vec{k}$	Unit Vectors in Cartesian Coordinates
k	Reduced Frequency
L	Length
L	Left Eigenvector
L, R	Left and Right State of a Cell
M	Mach Number
n	Iteration Number
\vec{n}	Unit Normal Vector associated with a face
p	Pressure
p	Roll Rate
q	Pitch Rate
r	Yaw Rate
Pr	Prandtl Number
R	Universal Gas Constant
R	Residual
RANS	Reynolds Averaged Navier-Stokes
R^\pm	Riemann Invariants
$ R $	Rotation Matrices
Re	Reynolds Number; $Re = \frac{\mu VL}{\rho}$

q	Conduction Heat Flux
\vec{Q}	Diffusive – Viscous Flux Vector
x, y, z	Cartesian Coordinates
y^+	Non-Dimensional Normal Distance
u, v, w	Cartesian Velocities in x, y and z directions
U^+	Non-Dimensional Velocity
\vec{U}	Conservative Variable Vector
\vec{V}	Velocity Vector; $\vec{V} = u\vec{i} + v\vec{j} + w\vec{k}$
$\vec{V}_{c.v.}$	Grid Velocity Vector; $\vec{V}_{c.v.} = u_{c.v.}\vec{i} + v_{c.v.}\vec{j} + w_{c.v.}\vec{k}$
S	Cell Boundary Surface
t	Time
T	Temperature

Greek Letters

α	Angle of Attack
$\alpha_{1,2,3}$	Runge-Kutta Constants
β	Angle of Side Slip
δ	Fin Deflection Angle
γ	Specific Heat Ratio, C_p/C_v
Ω	Finite Volume – Domain of Integration
Ω	Rotation of Projectile Around Flow Trajectory
θ	Rotation Angle Around Projectile Axis
$\theta^{(y)}, \theta^{(z)}$	Directions Cosine Angles

ϖ	Rotation of Projectile Around its Axis
ϕ	Rotation Angle Around Flow Trajectory
τ	Shear Stress Tensor
$ \Lambda $	Matrix of Eigenvectors
Λ	Eigenvectors
Δ	Difference
ν	Kinematic Viscosity
$\tilde{\nu}$	Spalart-Allmaras Turbulence Model Working Variable
μ	Absolute Viscosity
ρ	Density
$\vec{\nabla}$	Gradient Operator

Subscripts

center	Cell Center
c.v.	Control Volume
eul	Eulerian
face	Cell Face
inf	Free Stream Conditions
int	Interior
L	Reference Length
lgr	Lagrangian
node	Cell Node
ref	Reference Conditions
rel	Relative
t, turb	Turbulent

∞ Free Stream Conditions

Superscripts

\cdot Time Derivative

x,y,z Cartesian Coordinates

CHAPTER 1

INTRODUCTION

The accurate prediction of missile aerodynamics is a major task for the missile industry. Determination of the aerodynamic forces and moments is very important for the prediction of the motion of projectiles (Figure 1). The rates of change of these forces or moments with respect to linear or angular velocity components, namely “stability derivatives”, play an important role in the dynamical analysis of the projectiles. So they must be obtained using various techniques.

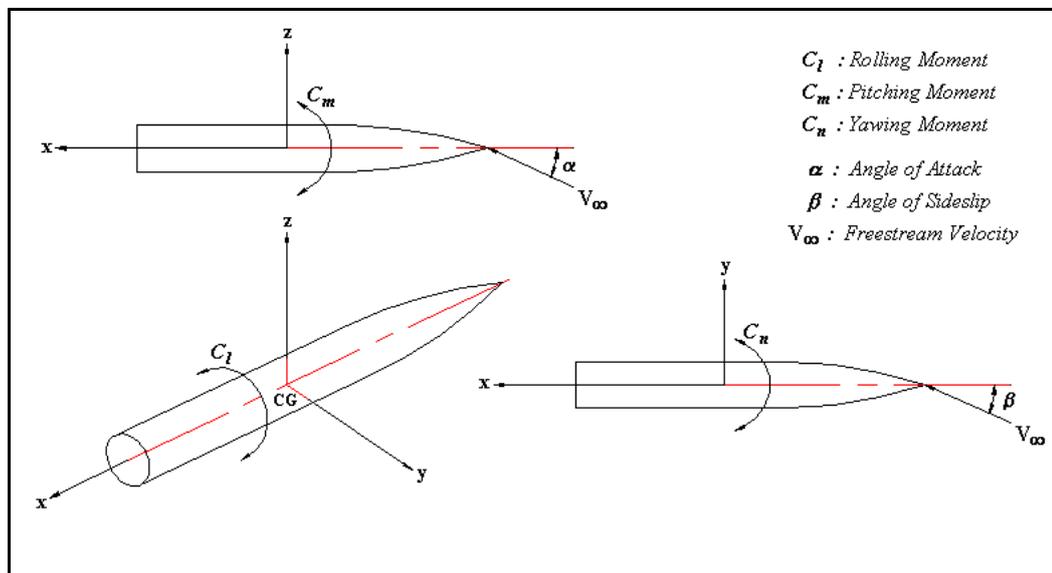


Figure 1. Aerodynamic Moments and Conventional Coordinate System on a Projectile

Although rotation of the projectile is not a must to maintain the stability of a statically stable projectile, rolling the projectile can minimize the effect of aerodynamic and inertial asymmetries of the free as well as powered flight trajectory [1]. But for rolling projectiles the combination of body spin and incidence creates a small force at right angles of the lift vector, called the Magnus force, and a resulting Magnus moment which may disturb the dynamic stability [2].

Stability derivatives can be determined both by experimental and theoretical methods. But some of these aerodynamic forces and moments, such as Magnus effect, can not be estimated good enough even by experimental methods. Besides, experimental methods are expensive and require a long lead time for the resultant data. Semi-empirical formulae can be used for the estimation of such unknowns but these methods are limited to the subsonic flow region.

A reasonable prediction of such force and moment coefficients can be performed by computational fluid dynamics methods (CFD). Advances in both computer technology and computational algorithms have allowed CFD to play an important role in the aerospace design process significantly reducing the costs required for the testing of physical models [3]. CFD offers quality aerodynamic information that can be routinely produced for many engineering applications [4] and it has been frequently used as an important design tool for the determination of missile aerodynamics [5].

From a computational point of view, much of the research effort has been focused on determining the static aerodynamics such as drag and pitching moment calculations [6]. On the other hand, roll characteristics are especially important for the stability requirements of the projectiles. Roll behavior of a flight vehicle can be characterized by the roll producing moment coefficient, C_{l_0} and the roll damping moment coefficient, C_{l_p} .

$$C_{l_p} = \frac{\partial C_l}{\partial p} \quad (1.1)$$

where “p” stands for the rotation rate.

Although rotation is not necessary for the prediction of C_{l_0} , since the moment is produced in the absence of spin, prediction of C_{l_p} , which is very important for the stability of the projectile, requires computations including rotational effects.

Also, for the determination of the Magnus forces and moments, which are developing as a result of roll at angle of attack or at angle of sideslip, rotation must be taken into account. Magnus moment coefficient definition is given in Equation (1.2). Magnus moment coefficient can be defined as the change of pitching moment coefficient with the angle of side slip in the presence of rotation.

$$C_{m_{\beta p}} = \frac{\partial^2 C_m}{\partial \beta \partial p} \quad (1.2)$$

The main motivation in this study is, therefore, the simulation of the rotational effects using CFD techniques.

Another aim of this study is to develop a parallelized three dimensional Navier Stokes / Euler solver for external flows around rotating bodies in order to determine the aerodynamic coefficients explained above.

In Chapter 2, brief information about missile aerodynamics has been given in order to see the importance of evaluation of dynamic stability derivatives. Then detailed literature survey on the use of different CFD techniques for modeling flow around a rotating projectile has been reviewed.

In Chapter 3, numerical method that has been utilized is introduced and the solver system that has been built is described. Governing flow equations are given, details about the numerical formulations are stated, implementation of Spalart-Allmaras turbulence model is described, flight mechanics calculations for the rotation of the projectile is explained, parallelization of the solver is presented and the flowchart of the solver system is described in detail.

Chapter 4 deals with 2-D and 3-D verification and validation studies. Viscous and inviscid, laminar and turbulent steady-state and unsteady moving body solutions for different test cases are given in this chapter.

In Chapter 5, 3-D turbulent flow over a projectile spinning across its axis is investigated. Computational results are compared with available experimental data. Discussion on the results of computations and the performance of the developed solver are also stated in this chapter.

Chapter 6 represents an overall conclusion about the study. In this chapter suggestions on the future work are also stated.

CHAPTER 2

LITERATURE SURVEY

2.1. Background on Missile Aerodynamics

Developments in defense industry, and rapid grow up of combat aircraft capabilities have brought a need for high capacity missiles. The missile aerodynamics differs from other flight vehicles in some respects. The basic aerodynamic properties of missiles will be presented in this chapter.

A typical unguided missile is composed of an aerodynamic nose, a slender aft body for engine and warhead, fins for stability purposes and canards as control surfaces, Figure 2.

In general, all missiles have symmetric geometries with respect to their centerline that passes through the nose and the base, which results in some simplifications for aerodynamic calculations. As a result of this symmetry property all of the in plane aerodynamic coefficients can be used also as out of plane aerodynamic coefficients. Namely, one can use the variation of pitch moment coefficient with angle of attack as the yaw moment variation with respect to sideslip angle.

Different type of mission profiles arise the needs for various missile configurations. Most generally, missiles can be categorized according to their firing and target positions. In this respect, they can be grouped as air-to-air, air to ground,

ground-to-ground and ground-to-air missiles. The variation of mission profiles requires determination of the aerodynamic forces and moments over a wide range of Mach numbers including supersonic, transonic, subsonic velocities and high angle of attacks.

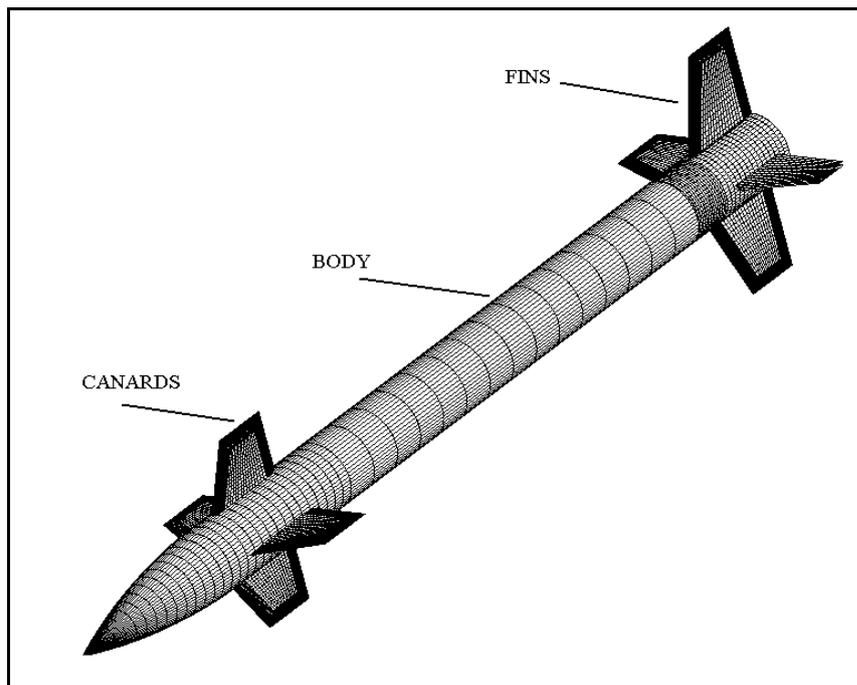


Figure 2. Standard Missile Configuration

The flow field around a missile shows different patterns for small, moderate and high angle of attacks. As the maneuverability of combat aircrafts increase, a need is felt for high angle of attack missiles. However, increasing angle of attack brings nonlinear effects on the aerodynamics of the missile. While the aerodynamics of a missile below 5° angle of attack can be taken as linear, above that angle nonlinear characteristics are observed that is caused from the interaction of fins with the vortex dominant flow field of the canard. Above 20° angle of attack,

large separations and horn vortices occurs on the missile body and control surfaces causing a high nonlinearity on aerodynamic forces and moments. Moreover the effect of side slip and control surface deflections has to be determined accurately. Consequently, the aerodynamics of a missile for all regimes stated above has to be determined precisely for a successful design [7].

Moreover, missiles can also be categorized as guided and unguided (dumb) ones. Each group has their own special aerodynamic characteristics. An unguided missile is generally fired from a tube which introduces also roll motion for stability purposes. However, missiles launched from tubes do not have enough space for their fins. Hence, wrapped around fins configurations are used to fit the missile into the launch tube. A sketch of wrapped around finned missile (WAF) is given in Figure 3.

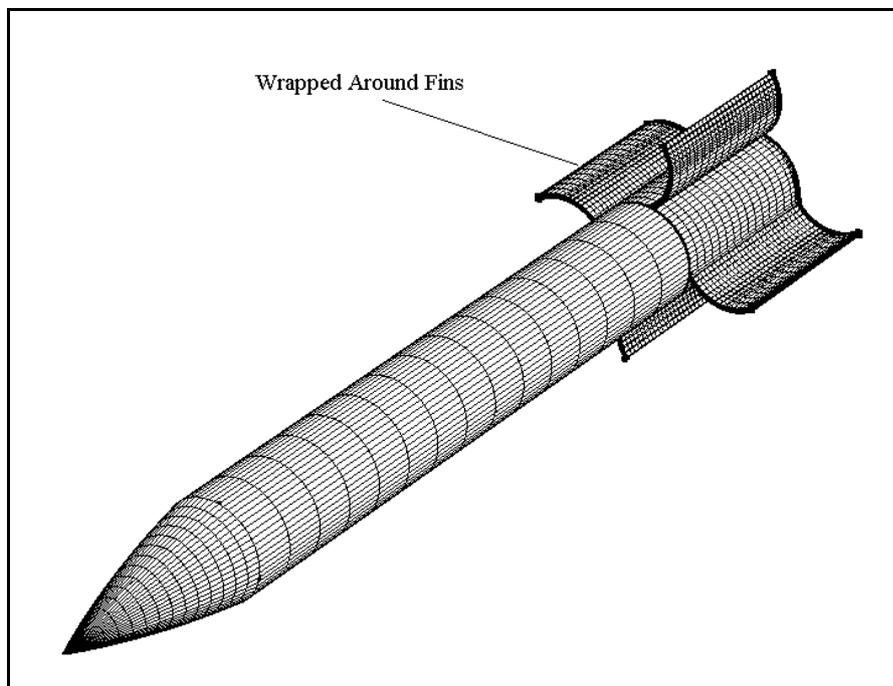


Figure 3. Wrapped Around Fin Configuration

Literature survey on the aerodynamics of wrapped around fins showed that, there are some distinct characteristics of wrapped around tail fins. As a result of the change in the symmetry of missile caused from the wrapped around fin configuration, there is always a roll tendency during the flight path which means that the roll moment of an unguided missile has to be calculated precisely. Moreover, out of plane static stability moment derivative, $C_{n\beta}$ and roll damping stability moment derivative, C_{l_p} must be calculated precisely for a correct trajectory prediction of WAF missile configuration.

Furthermore, one of the techniques that is used to minimize dispersion of the unguided missiles is giving rotational motion to it around its axis. This rotational motion introduces some important dynamic derivatives. Some of these derivatives are Magnus force, Magnus moment, pitch damping and roll damping coefficients which are designated as C_{Y_p} , $C_{m_{\dot{\beta}}}$, C_{m_q} and C_{l_p} respectively.

Consequently, aerodynamics of missiles differ from the other flight vehicles. There are some special aerodynamic coefficients and derivatives resulting from the shape and flight properties of them. For a successful design, geometric effects and resulting aerodynamic derivatives that are mentioned above have to be determined accurately.

2.1.1. Stability & Control

Controlled missile flight requires the continuous and precise balance of aerodynamic forces and inertial forces over a variety of conditions. The forces and moments experienced by a missile during its flight depend significantly on both the design details and the intended flight conditions of the vehicle. Maneuvering forces and moments on a missile can be significantly different from the static forces and

moments experienced during steady-static flight situations. Especially, forces and moments on a missile may exhibit nonlinear time and frequency-dependent behaviors, damping and lag effects during a maneuver. They may also involve the consideration of large angles of attack and sideslip, and moderate to massive flow separation. Missile stability and control derivatives quantify the changes in the aerodynamic forces or moments with respect to changes in the flow conditions. Moreover, some stability and control derivatives also quantify the changes in the aerodynamic forces or moments with respect to time. Stability and control derivatives are used to calculate, for example, the longitudinal short period, lateral pure roll, spin behaviors. These stability and control derivatives can be investigated in two main groups called as static and dynamic stability derivatives.

2.1.1.1. Static Stability

A missile is said to be statically stable, if the aerodynamic forces and moments on the body caused by a disturbance or given control deflection, forces to return the body to its equilibrium position, Figure 4.

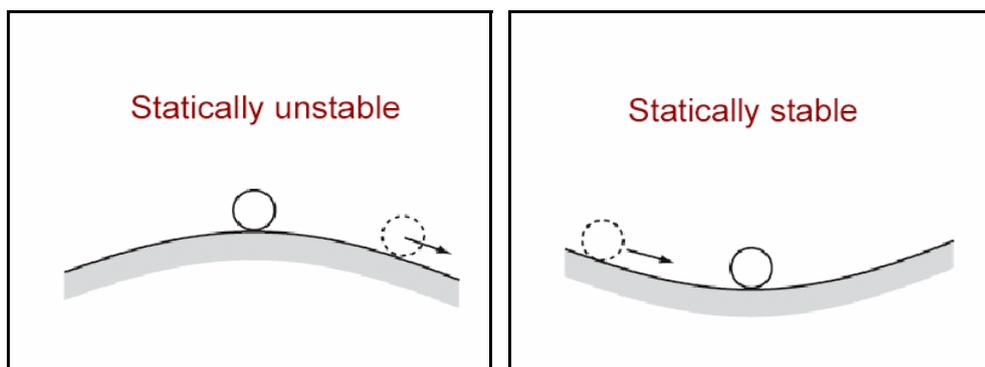
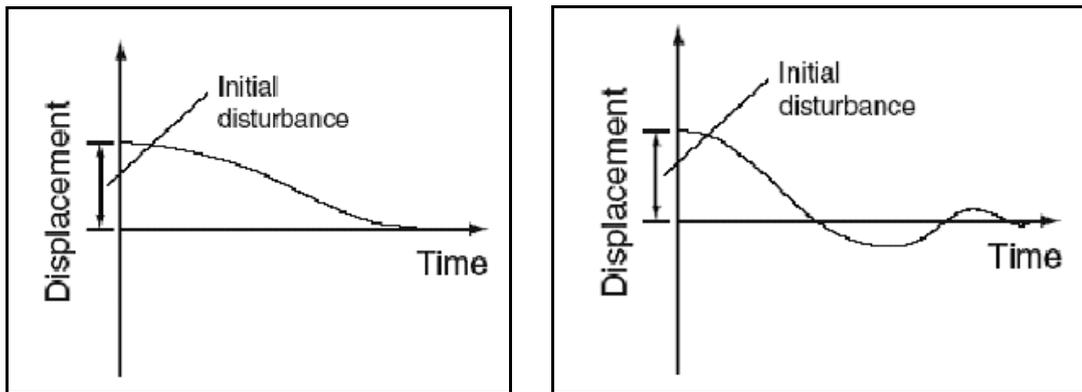


Figure 4. Illustration of Static Stability [8]

A statically stable missile must have an aerodynamic center behind the center of gravity for all flight regimes that will be faced [8]. The most important static stability derivatives for missiles are; static stability moment derivative, C_{m_α} , out of plane static stability moment derivative, C_{n_α} . These derivatives determine the recoverability of the control of an unguided missile, having an angle of attack, caused from a strong wing or control of a guided missile after a maneuver. Similarly the derivatives related to the control surface deflections (C_{m_δ} , C_{n_δ} , C_{l_δ}) are the other important static stability and control derivatives used to control a guided missile.

2.1.1.2. Dynamic Stability

In addition to the static stability derivatives, there exist some other important derivatives which also reflect the time history of the motion and affect the controllability of a missile. In other words, dynamic stability derivatives show the time history of the motion after the static stability characteristic of a missile. Hence, a missile is said to be dynamically stable if it eventually returns to its equilibrium position after a period of time. Dynamically stable missiles may show two types of motion character. In the first type, missile can recover its equilibrium position monotonically Figure 5-(a). In this case no oscillations around the equilibrium occur and displacement recovered smoothly. In the other type of dynamic stability, the equilibrium position is overshoot first and equilibrium point is obtained after some oscillations around the equilibrium point over a period of time Figure 5-(b). This type of motion is called damped oscillation [8]. On the other hand for a dynamically unstable missile the amplitude of the oscillations becomes larger and larger although it is statically stable, Figure 6.



(a)

(b)

Figure 5. Monotonic & Oscillatory Dynamic Stability [8]

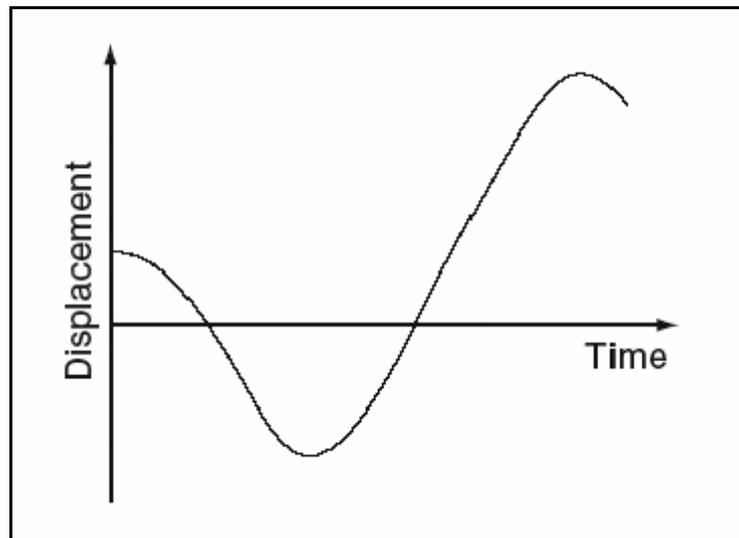


Figure 6. Dynamically Unstable Behaviour [8]

The most important dynamic stability coefficients in missile aerodynamics are Magnus moment, roll damping coefficient and pitch damping coefficient.

Magnus moment is the most important dynamic derivative to decrease the dispersion errors caused from the rotational motion of an unguided spinning missile. A force of the same magnitude and direction, which applies at the center of gravity, plus a moment, which is said to be Magnus moment, can substitute the Magnus force, applied at center of pressure. This moment tries to rotate the missile about an axis, perpendicular to its longitudinal axis. However, the gyroscopic effect also applies for the Magnus moment and the missile's axis will be shifted into the direction of the moment. Thus, the Magnus moment will have a stabilizing effect as it tends to decrease the yaw angle. It can be easily shown that this is only true, if the center of pressure of the Magnus force is located behind the center of gravity of the missile. The Magnus force destabilizes the missile and increases the yaw angle, if its center of pressure is located ahead of the center of gravity, which may come true in a specific velocity regime.

The roll damping is also another important parameter for an unguided missile or a controlled missile. One of the problems frequently encountered in missile design is that of providing adequate roll damping. This problem is primarily a consequence of the predominance of low-aspect-ratio surfaces on missile configurations. Very often, this problem is solved by a servomechanism which senses roll rate and actuates a control surface to give the necessary damping. Unfortunately, however, these servomechanisms require missile space and their complexity tends to decrease the overall reliability of the missile.

Finally pitch damping coefficient is important to predict the recovery of the angle of attack or side slip of the missile after a maneuver during a period of time.

As a conclusion, in addition to the static aerodynamic derivatives, to predict the motion and flight path of a missile the dynamical derivatives which are stated above have to be predicted precisely also.

2.2. Numerical Discretization Techniques

There are basically three discretization techniques for the numerical solution of the conservation laws, which have their own advantages and disadvantages regarding to the area of application:

1. Finite Difference Method
2. Finite Element Method
3. Finite Volume Method

2.2.1. Finite Difference Method (FDM)

FDM is probably the most commonly used technique for discretization of conservation equations because of its simplicity in application. It is based on Taylor series expansion of the derivatives of flow variables. FDM is mostly suitable for structured meshes with high degree of regularity.

2.2.2. Finite Element Method (FEM)

FEM is originally developed for structural analysis. FEM is more mathematically based and uses a weak variational form of the governing equations, along with polynomial shape functions, for discretization. FEM is best suited to irregular boundaries where local refinement is mostly needed.

2.2.3. *Finite Volume Method (FVM)*

FVM is based on the physical concept of using macroscopic control volumes to numerically solve the conservation laws of fluid motion [9, 10]. The use of integral form of the governing equations is the basis of FVM. The direct discretization of the conservation laws in integral form ensures that the mass, momentum and energy is conserved at discrete control volumes.

Jameson, *et al.* [11] reported one of the earliest successful implementations of this approach for solving the Euler equations on tetrahedral grids. FVM takes full advantage of an arbitrary mesh, where a large number of alternatives are available for the definition of the control volumes for conservation laws [12].

The freedom in the determination of the function representation of the flow field in FVM is much larger than in both FDM and FEM. The combination of the formulation of a flow problem on control volumes, which is the most physical way to obtain a discretization, with the geometric flexibility in the choice of the grid and the flexibility in defining the discrete flow variables, makes FVM extremely popular at engineering applications. In other words, FVM tries to combine the best from FEM, i.e. the geometric flexibility, with the best of FDM, i.e. the flexibility in defining the discrete flow field [13].

FVM comprise the most successful class of discretization techniques for the conservation laws of compressible fluid mechanics. Its success is based not only on its relative simplicity as compared to FDM and FEM approximations, but also on its flexibility and ability to unite ideas from FEM with those from FDM.

A common feature of most upwind codes is their finite volume representation. In each cell, the flow variables are supposed to be distributed in a specified way; piecewise constant; piecewise linear; or some higher-order representation.

FVM has been frequently preferred for the computational simulation of different aerodynamics problems. Frink *et al.* [4, 9, 10, 14, 15, 16, 17, 18, 19, 20] used finite volume formulation with upwind differencing for the solution of different aerodynamic problems. Batina *et al.* [21, 22] also used FVM formulation for complex aircraft aerodynamic analysis introducing the dynamic mesh algorithms into it. There are other researchers who used FVM especially for moving boundary problems [2, 23, 24].

As a result, it can be said that selection of FVM discretization technique is almost standard for numerical aerodynamic simulations.

Besides its outstanding advantages compared to other discretization techniques, FVM has an important drawback in the definition of the derivatives which is quite necessary for viscous flux calculations. Since the computational grid may not be orthogonal or equally spaced, which is the nature of the unstructured meshes that are commonly used with FVM, the definition of the derivatives of flow variables based on Taylor series expansion is impossible. Also, there is no mechanism like a weak formulation, as in FEM, to convert higher order derivatives into lower ones. This fact limits the use of FVM especially for viscous flow calculations. Some approximations have to be made and the effects of these approximations have to be watched carefully.

There are mainly two approaches for the approximation of mass, momentum, energy fluxes over the surface of control volumes in computational domain:

1. Cell vertex schemes
2. Cell centered schemes

2.2.3.1. Cell Vertex Scheme

In the cell vertex scheme, the flow properties are assigned at the vertices of the mesh cell. Flow properties are directly calculated at the cell nodes which gives the advantage of not distributing the flow variables from cell center values to cell nodes after numerical calculations are ended. Cell-vertex discretization offer advantages in accuracy especially on non-regular grids.

But cell vertex formulation is difficult to be introduced into the discretization algorithm when compared to cell centered scheme since it requires a complex numerical algorithm for generation of the control volumes. Representations of median dual cells generation for 2-D and 3-D elements for cell vertex formulation are shown in Figure 7.

Another disadvantage of cell vertex approach is that for the determination of viscous flow velocity gradients, this method may lead to undesirable results.

Cell vertex formulation has been applied by Barth [25], Venkatakrishnan [26] and Mavripilis *et al.* [27] to several applications of steady and unsteady computations.

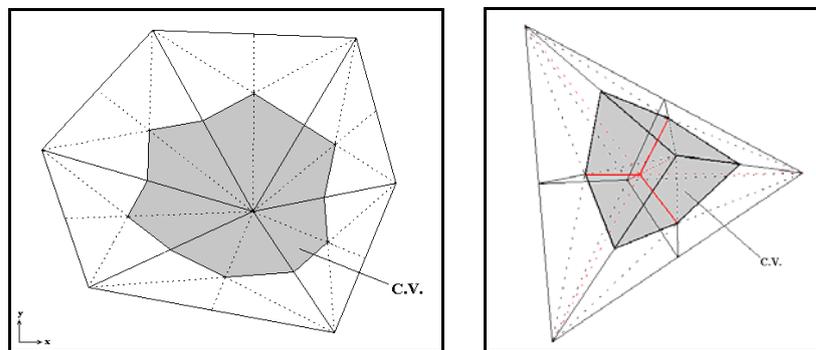


Figure 7. 2-D and 3-D Cell Vertex Median Dual Cells

2.2.3.2. Cell Centered Scheme

In the cell centered formulation, the flow properties are directly calculated at the center of the computational cell which itself is the control volume for finite volume discretization, Figure 8. This eliminates the need for the control volume generation affords. But cell centered formulation brings the disadvantage of needing some finite element approximations for distributing the variables to the nodes which may bring additional numerical errors to the results.

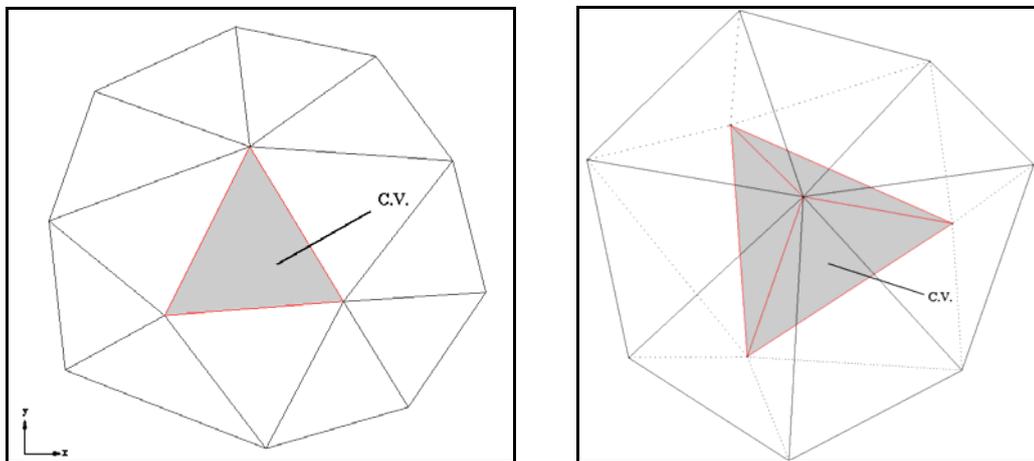


Figure 8. 2-D and 3-D Cell Centered Median Dual Cells

Commercially available grid partitioning programs that are used for parallel processing applications are generally ready to use with cell-centered approach which is an outstanding advantage. This is another important point that makes the use of cell centered formulation profitable. All of these facts have guided the author to use cell centered approximation.

Frink *et al.* [4, 9, 10, 14, 15, 16, 17, 18, 19, 20] , Batina *et al.* [21, 22] and Oktay *et al.* [28, 29, 30], who were investigating moving body aerodynamic problems or steady state aerodynamics preferred cell centered approach in their computations.

2.3. Numerical Schemes

2.3.1. Spatial Discretization

Most of the algorithms used for the solution of governing flow equations were based on either finite element or central differencing which need implementation of artificial dissipation for acceptable results. Schemes which are based on central differencing can not distinguish upstream from downstream influences because the physical propagation of flow information along the characteristics is not considered. For the flows where there exist no big discontinuities central schemes based on Taylor series expansion can be used with an acceptable accuracy. However, when discontinuities appear significantly, information from the upstream of the flow is required as an input in order to resolve the resulting non-linear behavior.

The capability of achieving high efficiency over a wide range of problems without need of any artificial input and obtaining physically meaningful results have made upwind schemes popular especially since 1980s. Nowadays, upwind schemes have become the main spatial discretization techniques and they are used nearly by all of the main research and commercial codes [14, 31, 32, 33, 34, 35, 36].

High order upwind schemes are used in order to take the physical propagation of perturbations along characteristics into account. Upwind differencing utilizes the propagation of information within a mesh in accordance

with the theory of characteristics in constructing type-dependent differencing for components of the information traveling in opposite directions in a separate and stable manner [14].

There is no need of scalar artificial dissipation formulas in upwind methods which are necessary for second order central schemes to damp odd-even oscillations generated especially in the vicinity of discontinuities. By using high order upwind methods, shocks or expansion waves that are observed at high speed compressible flows can be detected in a very sensitive and high accurate manner. Although this approach is more difficult than central differencing in computational sense, it brings the advantages of being more robust, having high convergence speed and requiring less user interaction. [12]

Another important advantage of upwind schemes was stated by Anderson and Bonhaus [34] that with the flux-difference splitting scheme of Roe [37, 38] the resolution of boundary layer details typically requires only half as many points as with a central differencing code.

Especially for hyperbolic partial differential equations in one space dimension; there is no question of the superiority of upwind methods but characteristic theory loses much of its clarity and simplicity in multidimensional problems. For 2-D or 3-D flow problems, choosing the upwinding direction being normal to the face of the computational cell across which the fluxes are computed is a commonly used way. [31]

Upwind schemes may be roughly divided into two categories: [12, 39]

1. Flux-Vector Splitting type schemes
2. Flux-Difference Splitting (Godunov) type schemes

2.3.1.1. Flux Vector Splitting Type Schemes

Upwind discretization is obtained by splitting the flux vector into two parts based on information coming from upwind and downwind of the cell face in flux-vector splitting algorithms [40, 41]. In other words, the flux terms are split according to the sign of associated propagation speeds.

The main drawback of flux vector splitting methods arises in the vicinity of sonic conditions since the splitting of the flux vectors is performed only with respect to the sign of the Mach number.

2.3.1.2. Flux Difference Splitting (Godunov) Type Schemes

In the flux-difference splitting schemes, Riemann problem (shock tube problem) on the cell faces are solved locally. The conservative variables are taken as piecewise constant over the mesh cells at each time step and time evolution is obtained by the solution of Riemann problem at the cell faces. By this way, exact contributions of local Euler equations are introduced to the numerical schemes which make sense in physical point of view. In the original Godunov [42] scheme, the local Riemann problem is solved exactly. Since this approach is computationally cumbersome some other approximate Riemann solvers have been built by Roe [37, 38], Osher and Solomon [43], Toro [44] and Harten [45].

2.3.2. Temporal Discretization

There exists basically two types of time stepping algorithms used both for integrating governing flow equations in time to obtain steady state solution and for unsteady applications.

1. Explicit Time Stepping Algorithms

2. Implicit Time Stepping Algorithms

Both explicit and implicit time stepping procedures are presently in widespread use. Due to the simplicity both in physical interpretation and numerical implementation, explicit time stepping methods generally find a wide range of application for the solution the Euler or Navier Stokes Equations [11, 14, 15, 22, 23]. These methods are simple, easily vectorizable and allow a good deal of flexibility in the treatment of boundary conditions.

The explicit time-marching schemes are computationally efficient when applied to meshes that are coarse but it is sure that for the solution of Navier-Stokes equations there is a need of high quality, fine meshes especially near the turbulent boundary layer. When fine meshes are used, the rate of convergence deteriorates significantly for explicit time marching methods. Batina [21] stated that for cases where finer meshes are used, an implicit temporal discretization which allows large time steps is required to obtain steady-state solutions in a computationally efficient manner. Because of this advantage implicit time stepping algorithms has found a widespread area of application especially for steady-state solutions [2, 10, 21, 34, 46].

Although implicit algorithms offer more stable and faster results, they have the shortcoming of large amount of memory usage. Also, the implementation of implicit time stepping algorithms especially for viscous flows is quite complicated since 3-D complicated viscous Jacobian matrices have to be derived and coded.

Because of above reasons, use of an explicit algorithm which is accelerated with implicit residual smoothing, with parallel processing is thought to be a good alternative for steady calculations. For unsteady calculations, special treatment must be given to implicit residual smoothing in order not to loose the time accuracy.

Most commonly used method of time discretization technique which is explicit in nature and of a high order of accuracy is the Runge-Kutta Method [11]. It achieves the accuracy of a Taylor series approach without any need of evaluation of higher derivatives. Explicit Runge-Kutta method is among the oldest and best-understood schemes in the numerical analysis methods. The simplicity of Explicit Runge-Kutta formula lies in its self-contained, one-step nature. However, because of its explicit nature, it needs careful stability analysis. A stability analysis for Runge-Kutta explicit time stepping algorithm has been performed by Hirsch [12] and Mavriplis [47] and stability regions for this scheme have been presented by them.

2.4. Computational Grid

As for sure, the very first step of the Computational Fluid Dynamics (CFD) applications is the construction of a discrete approximation for the region over which governing equations are desired to be solved.

Grid generation has very big importance in numerical solutions. A grid which is not well organized can lead to an unsatisfactory numerical result. Sometimes, improper choice of grid point locations can lead to numerical instability or lack of convergence.

There are mainly three types of grids evolved due to the relationship between the grid points. They are:

1. Structured Grids
2. Unstructured Grids
3. Hybrid Grids

2.4.1. Structured Grids

In structured grids, neighboring grid points in the physical space are the same as the neighboring points in the computational plane. In other words, structured grids have an implicit connectivity that allows specifying the grid locations by its computational coordinates.

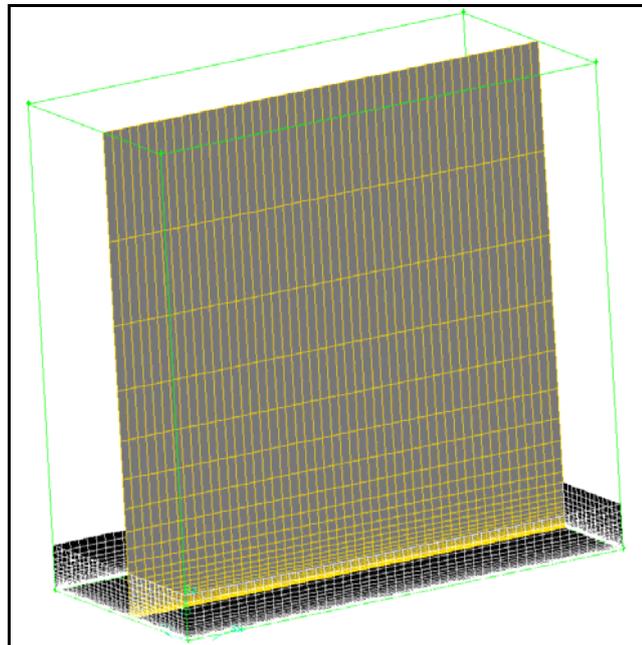


Figure 9. Sample 3-D Structured Grid

Structured grids have the advantage of simple coding when numerical algorithms is concerned but it is difficult and sometimes impossible to produce structured grids for complex domains. Multi-block grids are used in order to overcome this difficulty but still it takes too much time to generate a grid around a complex geometry, if it is possible, when compared with unstructured grid generation methods. Zhoa and Tai [48] stated that generating structured grids over a

complex geometry can take several months whereas running the solver takes just a small fraction of that time. This comment is totally true and has been experienced by the author also.

Also, the dense mesh which is required especially at wall boundaries for viscous calculations has to be carried all away into the far field for structured grids which obviously increases the memory requirements of the numerical code. In Figure 9 the unnecessary grid resolution at far field because of the highly dense mesh near the wall boundary can be observed.

2.4.2. Unstructured Grids

Solution of flow fields around complex geometries directly addresses the use of unstructured grids, since unstructured grids do not need any implicit connectivity like structured grids. Unstructured grids composed of triangular and tetrahedral elements in 2-D and 3-D, respectively, offer the designer the ability to model flows around complex bodies and to incorporate adaptive procedures to the solution. The flexibility in geometric modeling gives the advantage of using fewer cells to adequately model a given geometry when compared to the structured grids.

Unstructured meshes are also computationally feasible when the grid generation time is concerned. It can be said that the size of the mesh, which directly affects the memory requirements of the numerical algorithm, can only be minimized with the usage of unstructured grids.

For applications with complex geometry or requiring rapid turnaround time, the unstructured formulation appears to be the method of choice. [49]

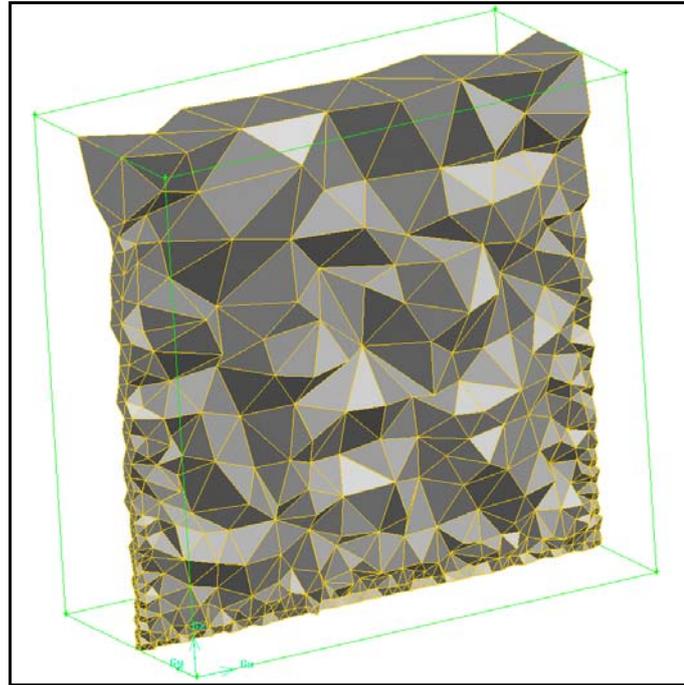


Figure 10. Sample 3-D Unstructured Grid

Maybe the only and the most important drawback of unstructured meshes is the issue of weakness associated with resolving boundary layer and viscous wakes [3]. The skewness of the tetrahedral meshes near the boundary surfaces creates seriously high amount of numerical diffusion. Figure 10 shows the inadequate quality of unstructured mesh near the wall boundary. But improvements in unstructured grid methods show that it will not take so much time for unstructured grids to achieve the capacity of structured grids about this issue.

In spite of this disadvantage, unstructured grids are widely used in aerodynamics society for the solution of flows, especially inviscid flows, around complex geometries such as airplanes and missiles [4, 15, 18, 34, 50, 51, 52, 53].

2.4.3. Hybrid Grids

In order to eliminate the difficulty in generating high quality unstructured viscous meshes within the boundary layer with the available grid generation tools, hybrid grids can be used. Hybrid grids offer usage of structured high quality grids in the vicinity of boundaries and usage of unstructured grids where dense mesh is not needed. By so, it is possible to obtain a computational mesh which is dense enough to observe the boundary layer and which is small in terms of number of elements, i.e. less memory usage and higher computational convergence rate. In Figure 11, high quality viscous structured mesh near the wall boundary with smoothly growing unstructured mesh up to the far field is presented.

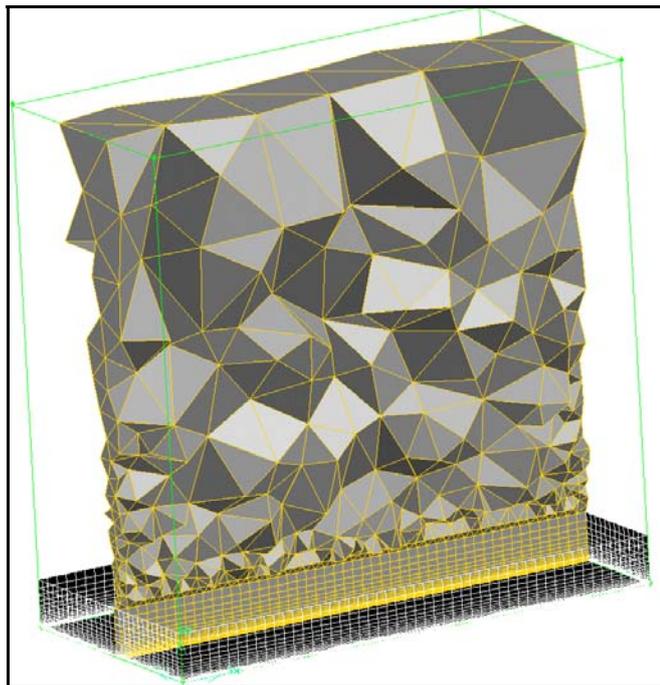


Figure 11. Sample 3-D Structured - Unstructured Hybrid Grid

Most commonly used approach is to employ layers of hexahedral cells around solid bodies and cover the rest of the domain with tetrahedral cells of rapidly increasing size for accurate and economic computations of viscous flows [54]. The interface between two different types of elements is filled with pyramid elements. Most of the commercial grid generation tools have this ability recently [ANSYS, FLUENT-GAMBIT, CFD-GEOM].

2.5. Parallel Processing

Recent activities on parallel processing have become a challenging subject in CFD. Main reason for this is the need of constantly increasing computational power in most CFD applications. Bruner [49] states that due to the finite speed of light and other physical limitations, there is an absolute speed limit for sequential computers, and we are rapidly approaching that limit. So parallel computing has become almost the only choice to offer that kind of computational power from the beginning of 1990's [49, 55, 56, 57, 58, 59, 60, 61, 62, 26].

Solution of full Navier-Stokes equations around complex geometries require massive computational resources. In this study rotational effects of missile configurations will also be included. It seems that only parallel architecture computers offer the promise of providing orders of magnitude greater computational power for such problems.

Parallel computing can be simply defined as distributing the computational load to a number of processors for simultaneous operations. It is obvious that a parallel computing environment is necessary for parallel computing.

One type of parallel computers is the one with shared memory. In such architecture, all of the processors share the same physical memory. Since all the

processors share the same memory there is almost no need of communication network between the nodes. These types of machines (ex. CRAY) have high performance when compared with other systems with same number of processors especially because of the high speed of data transfer between computing nodes. On the other hand, these computers are limited to certain number of processors. In other words, expansion of such systems is not quite possible. Also, when the cost of these computers is compared with their performance, it can be said that they are not feasible from economical point of view.

Another way of building parallel computers is the so called “Beawulf” clustering. These systems are composed of separate processors having their own local memories connected to each other by the means of a network connection, so called “distributed-memory systems”. This type of computers offer the advantage of having more processors that means higher computational power. Cost of such systems is quite reasonable when compared to shared memory systems. Luke [60] states that distributed memory models offer an alternative to shared memory models that is a more accurate representation of scalable computing architectures. All of the so-called “massively parallel” computers in existence today are distributed-memory machines [49].

The partitioning of the computational mesh among multiple processors is called “domain decomposition”. Domain decomposition algorithms make it possible to solve problems of large size by dividing them into smaller ones which can be treated by several computers with lower memory requirements. The divided sub-domains are interconnected to others by means of interfaces which are treated as interface boundary conditions.

There exist programs that can be obtained from open resources which performs domain decomposition issue. In this work, the grid partitioning program “METIS” [63] is used. Details about the parallel processing algorithm used in this work and the hardware used will be given in Chapter 3.

2.6. Turbulence

The viscous phenomena that occur around missile configurations, such as boundary layer separations, wakes and vortices are quite important for missile aerodynamic characteristics. Also it is well known that the viscous phenomena are of primary importance in the study of unsteady flows [24]. Shearing effects and possible instabilities result in small and large scale turbulent effects.

Up to certain Mach number and angle of attack values, inviscid calculations, i.e. Euler computations, give acceptable results. However, these computations can not be used to predict all of the aerodynamic characteristics. For example axial force and roll moment coefficients can not be accurately predicted by Euler computations [64].

All of the above reasons indicate that it is necessary to represent these viscous phenomena accurately in order to predict the missile aerodynamic coefficients in an acceptable manner. This fact brings the necessity of effective numerical solution of viscous flow equations.

One of the most important phenomena for complex viscous flows is turbulence, which is difficult to simulate due to the existence of a wide range of scales [65]. Its irregular nature, diffusive and dissipative effect and fully 3-D behavior make turbulence a difficult phenomenon to simulate. Unfortunately, almost all of the flows in real life are turbulent. Especially to compute flows at large Reynolds numbers, turbulence has to be numerically modeled. Moreover, the accuracy of the numerical solution directly depends on the turbulence model selected. Therefore, the development of accurate turbulence models and their implementation in conjunction with efficient solvers is of primary importance in the CFD analysis of viscous flows [24].

Turbulence models range from simplest algebraic model to the most accurate direct numerical simulation. Most commonly used turbulence models because of their computationally ease of use depend on work of Reynolds on turbulence. They are so called “Time-Averaged” or “Reynolds-Averaged” formulations. In this formulation, the Reynolds stress is assumed to be related to the mean strain rate by the eddy viscosity. The effect of unsteady behavior of turbulence is introduced in flow equations as a time averaged value. “Time-Averaged” turbulence models may be divided into three parts depending on the number of transport equations need to be solved to obtain the eddy viscosity [66]:

1. Algebraic (Zero-Equation) Models
2. One-Equation Models
3. Two-Equation Models

2.6.1. Algebraic (Zero-Equation) Models

Algebraic models are based on mixing length (an analog of the mean free path of a gas) hypothesis which was introduced by Prandtl (1925). This hypothesis is closely related to eddy viscosity concept and formed the basis of all time-averaged turbulence models. Among several models available, the one applied by Baldwin and Lomax [67], which is a modified version of Cebeci-Smith model [68], has widespread area of application.

2.6.2. One-Equation Models

Again work of Prandtl (1945) introduced the effect of turbulent fluctuations, k , into the evaluation of eddy viscosity. In this formulation, a partial-differential equation is modeled in order to approximate the exact equation for turbulent

fluctuations. By this way, turbulent stresses, thus the eddy viscosity, are affected by the flow history. Having an eddy viscosity that depends upon flow history provides a more physically realistic model.

One-equation models has found widespread of application area [Baldwin and Barth (1990), Goldberg (1991) and Spalart and Allmaras (1992)]. This interest to one-equation models is mainly because of the ease with which such model equations can be solved numerically, relative to two-equation models and stress-transport models. Among these recent one-equation models, model of Spalart and Allmaras appears to be the most accurate for practical turbulent-flow applications.

2.6.3. Two-Equation Models

Kolmogorov (1942) introduced a second parameter, ω ; "*the rate of dissipation of energy in unit volume and time*", in addition to turbulence fluctuations parameter, k . In this model, known as a k - ω model, ω ; satisfies another differential equation similar to the one for k . The model is thus called a two-equation model of turbulence. This model introduces turbulence time scale, mixing length and dissipation rate into the governing flow equations. The non-linear equations introduced with this model brought additional complicity to the Navier-Stokes equations and did not find much application area until computational power became available for such problems.

By this far, the most extensive work on two-equation models has been carried out by Launder and Spalding (1972). Launder's k - ϵ model is as well known as the mixing-length model and is the most widely used two-equation model. Although this model is used frequently and is very popular, it has some inadequacy problems for flows with adverse pressure gradient.

2.6.4. Other Turbulence Models

There exist some other turbulence models which are totally different from time-averaged turbulence models. One of them is the Reynolds stress model. Reynolds stress model does not use the concept of eddy viscosity. Instead, a transport equation for each component of the Reynolds stress tensor is solved directly. While time-averaged models, as their name imply, only solve the mean flow, large eddy simulation (LES) solve the large scale fluctuations in addition to the mean flow and only the effect of small scales (i.e., subgrid scales) are modeled.

Another and maybe the most accurate method is direct numerical simulation (DNS), where both mean flow and all the fluctuations are solved directly.

It is for sure that the Reynolds stress model, LES and DNS methods are much more accurate than the eddy viscosity based – time-averaged methods. But today's level of computer technologies does not offer a profitable choice for the use of these methods in numerical simulation of complex geometries yet because these methods require enormous amounts of CPU time and memory. Therefore, the most widely used turbulence models in industry are still those based on the concept of eddy viscosity. This is especially true for CFD codes using unstructured grids [19].

2.7. Unsteady Aerodynamics – Implementation of Rotation

One method for the introduction of rotation into governing flow equations is to select a non-inertial body-fixed rotating coordinate system [1, 23, 69, 70, 71, 72, 73].

Weinacht, Sturek *et al.* [1, 69, 72] analyzed the damping characteristics of coning motion of projectiles at supersonic conditions using this formulation. Park &

Green [70, 73] modified a currently developed CFD solver to compute stability derivatives. Yaniv [23] solved Reynolds Averaged Navier-Stokes equations which are formulated in Cartesian reference frame rotating with constant angular velocity in order to determine roll damping coefficient of a projectile. Kandil & Chuang [71] studied the flow about an oscillating delta wing in moving frame of reference.

Body-fixed frame can be utilized to predict the flow field provided that the governing equations are modified to incorporate the centrifugal and Coriolis body forces resulting from the non-inertial coordinate frame. This method corresponds to a fully Eulerian system in which the coordinate system is fixed on the missile and computational grid is treated as a fixed reference frame through which the fluid moves. The rotational effects are introduced by the governing equations and the boundary conditions. Implication of the inertial terms stated above is very complicated especially when full Navier-Stokes equations are thought. Also, implementation of the boundary conditions at far field is not so clear. So some other methods are investigated.

Another approach is selecting an inertial frame of reference and utilizing a moving computational grid. This method is so called “Arbitrary Lagrangian Eulerian Formulation” (ALE).

Batina [21, 22] has applied this method to aircraft aerodynamic analysis with unstructured dynamic mesh algorithm. However, he had to satisfy the geometric conservation law numerically since he was trying to simulate the deformations of complex aircraft configurations. These deformations are so small, for example flutter of the wings, and the motion of the domain is not periodical so moving the mesh results in highly distorted computational grid. For the movement of the computational distorted grid, he had developed a dynamic mesh algorithm. A similar approach has been followed by Uzun & Oktay [50, 28, 29, 30].

Farhat *et al.* [53, 74, 75, 76, 77, 78] stated the importance of the geometric conservation laws for Arbitrary Lagrangian Eulerian formulation especially when the moving mesh deforms. Trepanier *et al.* [79, 80, 81] also gave importance to the conservation of geometric quantities and suggested some formulations for this problem.

Pechier *et al.* [2] applied the Arbitrary Lagrangian Eulerian formulation to missile configurations to evaluate the Magnus effect over finned projectiles. In their work the mesh rotates without any distortion. Barakos *et al.* [24] applied the same method with a different numerical algorithm. They all suggest that there is no need to apply any geometric conservation law since the computational grid is rotated at each time step without deforming the computational cells. In this study, rotation of a projectile is modeled using this approach, which means that the entire computational domain is rotated.

Deconinck *et al.* [82], Demirdzic & Peric [83], Thomas & Lombard. [84], Smith [85], Khelil *et al.* [86], Moreau [87], Donea *et al.* [88, 89], Van Haaren *et al.* [90] have applied Arbitrary Lagrangian Eulerian formulation with different numerical algorithms to different areas of interest from missile aerodynamics to fluid-solid interaction problems.

In ALE formulation, the computational grid moves in any direction with a velocity different from the fluid velocity. The main advantage of this method is the simplicity of the governing flow equations. There are no centrifugal or Coriolis body forces to be introduced into the equations which is not so easy. Only grid velocities are added to the governing flow equations in a very simple manner.

The governing equations for such an algorithm that will be stated in the later chapters in detail are capable of solving any kind of motion if the necessary grid velocities are supplied.

CHAPTER 3

NUMERICAL METHOD

3.1. Coordinate System Convention

The Cartesian coordinate system convention for the flow over a missile configuration used in the flow solver developed is given in Figure 12, Figure 13 and Figure 14. The directions of velocities, forces and moments shown in these figures are taken as positive according to the right hand rule for moments. The coordinates:

x-axis: Longitudinal body axis which is positive in downstream direction

y-axis: Spanwise body axis

z-axis: Vertical direction of body which is positive upward

Angle of attack α is in x-z (pitch) plane and it is positive for the nose up condition. Angle of side slip β is in y-z (yaw) plane and positive for yaw to the left.

3.2. Governing Equations

For a three-dimensional flow through a finite volume Ω moving with a speed $\vec{V}_{c.v.}$ which is enclosed by the boundary surface S and an exterior normal \vec{n} ,

integral form of the conservation equations in non-inertial frame of reference are given as : [2, 21, 22]

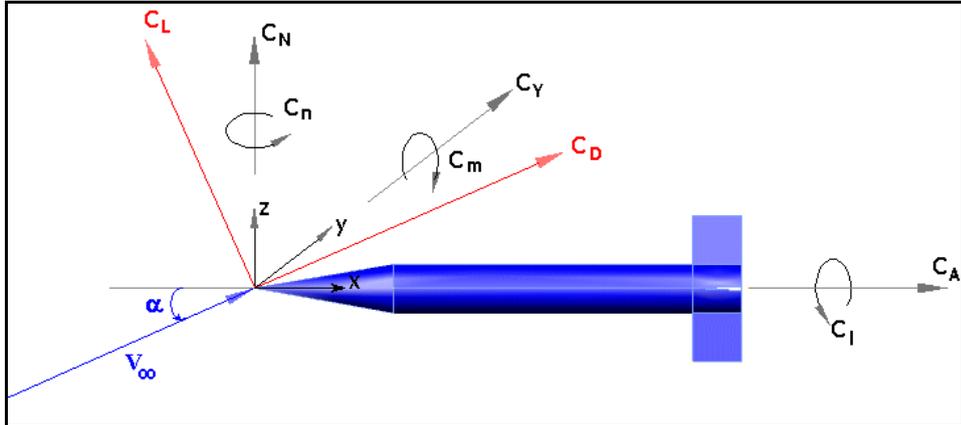


Figure 12. Coordinate System Convention in x-z Plane

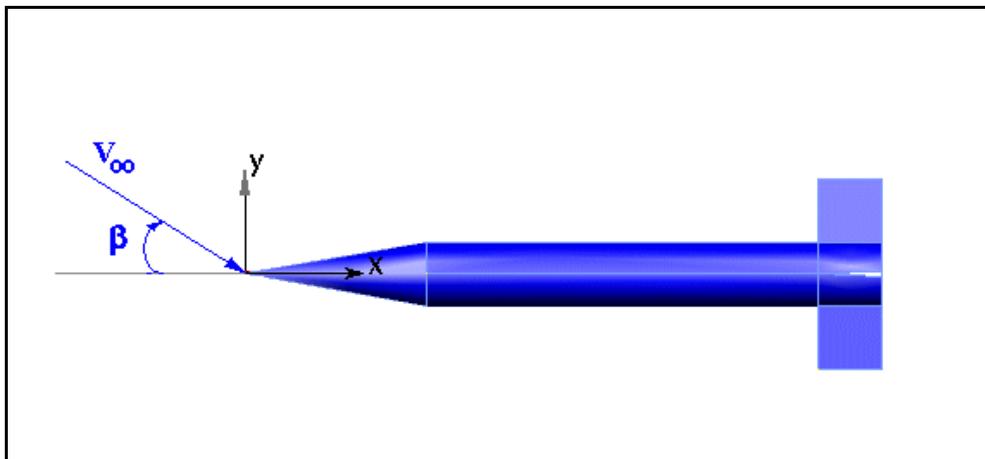


Figure 13. Coordinate System Convention in x-y Plane

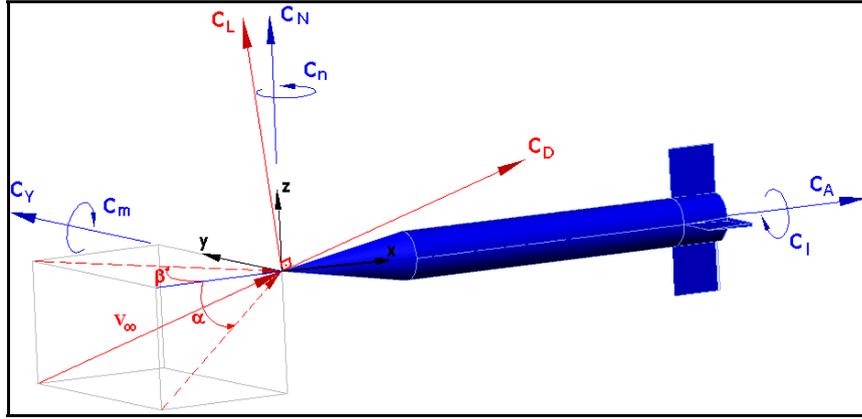


Figure 14. Coordinate System Convention – Three Dimensional View

3.2.1. Conservation of Mass

$$\frac{\partial}{\partial t} \int_{\Omega} \rho \, d\Omega + \oint_S \rho (\vec{V} - \vec{V}_{c.v.}) \cdot d\vec{S} = 0 \quad (3.1)$$

3.2.2. Conservation of Momentum

$$\frac{\partial}{\partial t} \int_{\Omega} \rho \vec{V} \, d\Omega + \oint_S (\rho \vec{V} \otimes \vec{V} + p \vec{I} - \rho \vec{V} \otimes \vec{V}_{c.v.}) \cdot d\vec{S} = \oint_S \tau_{ij} \cdot d\vec{S} \quad (3.2)$$

where $\vec{V} \otimes \vec{V}$ is the tensor product.

3.2.3. Conservation of Energy

$$\frac{\partial}{\partial t} \int_{\Omega} \rho E \, d\Omega + \oint_S (\rho E \vec{V} + P \vec{V} - \rho E \vec{V}_{c.v.}) \cdot d\vec{S} = \oint_S (\tau_{ij} \cdot \vec{V} - q) \cdot d\vec{S} \quad (3.3)$$

where viscous shear stress tensor is defined as ;

$$\tau_{ij} = \mu \left[(\partial_j V_i + \partial_i V_j) - \frac{2}{3} (\vec{\nabla} \cdot \vec{V}) \delta_{ij} \right] \quad (3.4)$$

3.2.4. Integral Compact Form of Governing Equations

The integral compact form of the Navier-Stokes equations, which is suitable for numerical calculations is given by;

$$\frac{\partial}{\partial t} \int_{\Omega} \vec{U} d\Omega + \oint_S \vec{F} \cdot d\vec{S} = \oint_S \vec{Q} \cdot d\vec{S} \quad (3.5)$$

where the column vector \vec{U} represents the conservative variables, column vector \vec{F} represents the convective flux vector, and column vector \vec{Q} represents the viscous diffusive flux vector.

$$\vec{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{bmatrix} \quad \vec{F} = \begin{bmatrix} \rho(\vec{V} - \vec{V}_{c.v.}) \\ (\rho \vec{V} \otimes \vec{V} + P \vec{I} - \rho \vec{V} \otimes \vec{V}_{c.v.}) \\ (\rho E (\vec{V} - \vec{V}_{c.v.}) + P \vec{V}) \end{bmatrix} \quad \vec{Q} = \begin{bmatrix} 0 \\ \tau_{ij} \\ (\tau_{ij} \cdot \vec{V} - q) \end{bmatrix} \quad (3.6)$$

where the fluid velocity is : $\vec{V} = u\vec{i} + v\vec{j} + w\vec{k}$ and

grid velocity is : $\vec{V}_{c.v.} = u_{c.v.}\vec{i} + v_{c.v.}\vec{j} + w_{c.v.}\vec{k}$

If the convective flux vector; $\vec{F} = F\vec{i} + G\vec{j} + H\vec{k}$ is expressed explicitly;

$$F = \begin{bmatrix} \rho(u - u_{c.v.}) \\ \rho u(u - u_{c.v.}) + p \\ \rho v(u - u_{c.v.}) \\ \rho w(u - u_{c.v.}) \\ (\rho E + p)(u - u_{c.v.}) + u_{c.v.} p \end{bmatrix} \quad (3.7)$$

$$G = \begin{bmatrix} \rho(v - v_{c.v.}) \\ \rho u(v - v_{c.v.}) \\ \rho v(v - v_{c.v.}) + p \\ \rho w(v - v_{c.v.}) \\ (\rho E + p)(v - v_{c.v.}) + v_{c.v.} p \end{bmatrix} \quad (3.8)$$

$$H = \begin{bmatrix} \rho(w - w_{c.v.}) \\ \rho u(w - w_{c.v.}) \\ \rho v(w - w_{c.v.}) \\ \rho w(w - w_{c.v.}) + p \\ (\rho E + p)(w - w_{c.v.}) + w_{c.v.} p \end{bmatrix} \quad (3.9)$$

If viscous diffusive flux vector; $\vec{Q} = \frac{M_\infty}{Re_L} \cdot (Q_x \vec{i} + Q_y \vec{j} + Q_z \vec{k})$ is expressed explicitly in the Cartesian coordinate system;

$$Q_x = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ \tau_{xx} u + \tau_{xy} v + \tau_{xz} w - \dot{q}_x \end{bmatrix} \quad (3.10)$$

$$Q_y = \begin{bmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ \tau_{yz} \\ \tau_{yx}u + \tau_{yy}v + \tau_{yz}w - \dot{q}_y \end{bmatrix} \quad (3.11)$$

$$Q_z = \begin{bmatrix} 0 \\ \tau_{zx} \\ \tau_{zy} \\ \tau_{zz} \\ \tau_{zx}u + \tau_{zy}v + \tau_{zz}w - \dot{q}_z \end{bmatrix} \quad (3.12)$$

where viscous shear stress tensors are :

$$\tau_{xx} = 2(\mu + \mu_t) \frac{\partial u}{\partial x} - \frac{2}{3}(\mu + \mu_t) \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) \quad (3.13)$$

$$\tau_{yy} = 2(\mu + \mu_t) \frac{\partial v}{\partial y} - \frac{2}{3}(\mu + \mu_t) \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) \quad (3.14)$$

$$\tau_{zz} = 2(\mu + \mu_t) \frac{\partial w}{\partial z} - \frac{2}{3}(\mu + \mu_t) \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) \quad (3.15)$$

$$\tau_{xy} = \tau_{yx} = (\mu + \mu_t) \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \quad (3.16)$$

$$\tau_{xz} = \tau_{zx} = (\mu + \mu_t) \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right) \quad (3.17)$$

$$\tau_{yz} = \tau_{zy} = (\mu + \mu_t) \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right) \quad (3.18)$$

and the heat conduction terms are:

$$\dot{q}_x = -\frac{1}{(\gamma-1)} \left(\frac{\mu}{Pr} + \frac{\mu_t}{Pr_t} \right) \frac{\partial T}{\partial x} \quad (3.19)$$

$$\dot{q}_y = -\frac{1}{(\gamma-1)} \left(\frac{\mu}{Pr} + \frac{\mu_t}{Pr_t} \right) \frac{\partial T}{\partial y} \quad (3.20)$$

$$\dot{q}_z = -\frac{1}{(\gamma-1)} \left(\frac{\mu}{Pr} + \frac{\mu_t}{Pr_t} \right) \frac{\partial T}{\partial z} \quad (3.21)$$

At last, the pressure is given by the equation of state for a perfect gas:

$$P = (\gamma-1) \left[\rho E - \frac{1}{2} \rho (u^2 + v^2 + w^2) \right] \quad (3.22)$$

Pechier *et al.* [2] stated that grid movement only modifies the convection flux at finite volume surface so there is no contribution of grid movement on diffusive flux terms.

3.2.5. Non-Dimensionalization of Governing Equations

The governing flow equations have been nondimensionalized by the free stream density ρ_∞ , free stream speed of sound c_∞ , free stream temperature T_∞ , free stream viscosity μ_∞ and reference length L. The nondimensionalized terms are given in Table 1.

Equations (3.6) to (3.22) are in nondimensional form with (*) sign being dropped. Nondimensional form of free stream variables, without (*) sign, is given in Table 2.

Table 1. Nondimensional Variables

$u^* = \frac{u}{c_\infty}$	$v^* = \frac{v}{c_\infty}$	$w^* = \frac{w}{c_\infty}$	$c^* = \frac{c}{c_\infty}$
$P^* = \frac{P}{\rho_\infty c_\infty^2}$	$T^* = \frac{T}{T_\infty}$	$\rho^* = \frac{\rho}{\rho_\infty}$	$E^* = \frac{E}{c_\infty^2}$
$\mu^* = \frac{\mu}{\mu_\infty}$	$\mu_t^* = \frac{\mu_t}{\mu_\infty}$	$x, y, z^* = \frac{x, y, z}{L_{\text{ref}}}$	$t^* = \frac{t}{L_{\text{ref}}/c_\infty}$

Table 2. Nondimensional Form of Free Stream Variables

$c_\infty = 1$	$\rho_\infty = 1$	$P_\infty = 1/\gamma$
$T_\infty = 1$	$\mu_\infty = 1$	$E_\infty = \frac{1}{\gamma \cdot (\gamma - 1)} + \frac{M_\infty^2}{2}$

3.3. Spatial Discretization

A cell centered finite volume discretization is applied to Equation (3.5) which is in integral form. Time rate of change of conservative variable vector \bar{U} within a computational domain Ω moving with a speed $\vec{V}_{c.v.}$ is balanced by the net convective and diffusive fluxes across the boundary surface S . For this purpose the

computational domain is divided into finite number of unstructured – tetrahedral elements. Each tetrahedral element serves as a computational cell for cell centered approach. The effect of moving boundary to the numerical formulation is introduced in Section 3.7.

In the finite-volume formulation, for a constant control volume of tetrahedron, Equation (3.5) becomes:

$$\Omega \frac{\partial}{\partial t} \bar{U}_i + \sum_{j=1}^4 [(\vec{F} \cdot \vec{n}S)_j]_i - \sum_{j=1}^4 [(\vec{Q} \cdot \vec{n}S)_j]_i = 0 \quad (3.23)$$

where $i = 1, 2, \dots, \# \text{ of cells}$

If Equation (3.23) is written explicitly across the four faces of a tetrahedron;

$$\Omega \frac{\partial}{\partial t} \bar{U}_i + \sum_{j=1}^4 ((F - Q_x)n_x + (G - Q_y)n_y + (H - Q_z)n_z) \cdot \Delta S_i = 0 \quad (3.24)$$

3.3.1. *Inviscid – Convective Fluxes*

Inviscid flux quantities $\vec{F}(\vec{U})$ (Equations 3.23 and 3.24) are computed using Roe's [37, 38] flux-difference splitting scheme across each cell face of cell centered control volumes. If downstream of the flow, i.e. the left hand side of the cell face is denoted by the subscript “L” and upstream of the flow, i.e. the right hand side of the cell face is denoted by subscript “R”, the flux across each cell face according to Roe's formulation is expressed as:

$$F_{\text{face}} = \frac{1}{2}(F(U_L) + F(U_R)) - \frac{1}{2}|A|(U_R - U_L), \text{ or} \quad (3.25)$$

$$F_{\text{face}} = F(U_L) + A^-(U_R - U_L), \text{ or} \quad (3.26)$$

$$F_{\text{face}} = F(U_R) - A^+(U_R - U_L) \quad (3.27)$$

where $F(U_L)$ and $F(U_R)$ are the conservative variables at the left and right of the cell face and “A” is the flux Jacobian matrix which is evaluated using the so called Roe averaged flow variables. The definition of flux Jacobian and Roe averaged flow variables are given below [12, 37, 38]:

$$A \equiv \frac{\partial \vec{F}}{\partial \vec{U}} \quad (3.28)$$

$$\rho_{\text{face}} = \sqrt{\rho_L \rho_R} \quad (3.29)$$

$$u_{\text{face}} = \frac{\sqrt{\rho_L} u_L + \sqrt{\rho_R} u_R}{\sqrt{\rho_L} + \sqrt{\rho_R}} \quad (3.30)$$

$$v_{\text{face}} = \frac{\sqrt{\rho_L} v_L + \sqrt{\rho_R} v_R}{\sqrt{\rho_L} + \sqrt{\rho_R}} \quad (3.31)$$

$$w_{\text{face}} = \frac{\sqrt{\rho_L} w_L + \sqrt{\rho_R} w_R}{\sqrt{\rho_L} + \sqrt{\rho_R}} \quad (3.32)$$

$$H_{\text{face}} = \frac{\sqrt{\rho_L} H_L + \sqrt{\rho_R} H_R}{\sqrt{\rho_L} + \sqrt{\rho_R}} \quad (3.33)$$

$$a_{\text{face}}^2 = (\gamma - 1) \left(H_{\text{face}} - \frac{1}{2} (u_{\text{face}}^2 + v_{\text{face}}^2 + w_{\text{face}}^2) \right) \quad (3.34)$$

From Equation (3.28);

$$F(U_R) - F(U_L) = |A|(U_R - U_L) \quad (3.35)$$

Introducing the diagonalization matrices – left eigenvectors L and L^{-1} and the diagonal matrix of eigenvalues Λ , the flux Jacobian A is defined as [12]:

$$|A| = L |\Lambda| L^{-1} \quad (3.36)$$

where;

$$|\Lambda| = \begin{vmatrix} \bar{V} - c \\ \bar{V} \\ \bar{V} \\ \bar{V} \\ \bar{V} + c \end{vmatrix} \quad (3.37)$$

So Equations (3.25) to (3.27) become;

$$F_{\text{face}} = \frac{1}{2}(F(U_L) + F(U_R)) - \frac{1}{2} L |\Lambda| L^{-1} \Delta U \quad (3.38)$$

$$F_{\text{face}} = F(U_L) + L |\Lambda| L^{-1} \Delta U \quad (3.39)$$

$$F_{\text{face}} = F(U_R) - L |\Lambda| L^{-1} \Delta U \quad (3.40)$$

There exist three ΔF flux difference components, each of which is associated with a distinct eigenvalue $\Lambda_1, \Lambda_2 = \Lambda_3 = \Lambda_4, \Lambda_5$.

$$L|\Lambda|L^{-1}\Delta U = |\Delta F_1| + |\Delta F_4| + |\Delta F_5| \quad (3.41)$$

By introducing these flux differences, which are calculated using Roe's averaged variables, to one of the Equations (3.25), (3.26) or (3.27), one can get the average flux at the each cell face [15]. These flux vectors at each four faces of tetrahedral computational cell is calculated and then summed up over the computational domain to satisfy Equation (3.23).

For a first order scheme, the state of the primitive variables at each cell face is set to cell-centered averages on either side of the face. For the higher order scheme, state variables at the cell faces are obtained from an extrapolation between the neighboring cell averages.

An important point in a tetrahedral cell-centered finite-volume scheme is the accurate distribution of cell-averaged data to the triangular faces for flux computation as illustrated in Figure 15. Frink *et al.* [9] derived a novel cell reconstruction process which is based on an analytical formulation for computing the gradient term of a Taylor series expansion within tetrahedral cells. The scheme consists of simple, universal formula for tetrahedral cells. For the face which is composed of nodes 1, 2 and 3:

$$\tilde{U}_{\text{face}(1,2,3)} = \tilde{U}_{\text{center}} + \frac{1}{4} \left[\frac{1}{3} (\tilde{U}_{\text{node1}} + \tilde{U}_{\text{node2}} + \tilde{U}_{\text{node3}}) - \tilde{U}_{\text{node4}} \right] \quad (3.42)$$

where $\tilde{U} = [\rho \quad u \quad v \quad w \quad p]^T$ represents the primitive flow variables.

Frink [9] stated that use of such a reconstruction scheme gives acceptable results around flow discontinuities without requiring the introduction of higher order spatial discretizations with limiters. This is quite advantageous when it is

thought that higher order spatial upwind discretizations cause oscillations like central schemes around discontinuities if appropriate limiters are not introduced.

There is also a need for distributing the cell centered data to a node which is common with the surrounding cells. This is performed by a weighted averaging procedure:

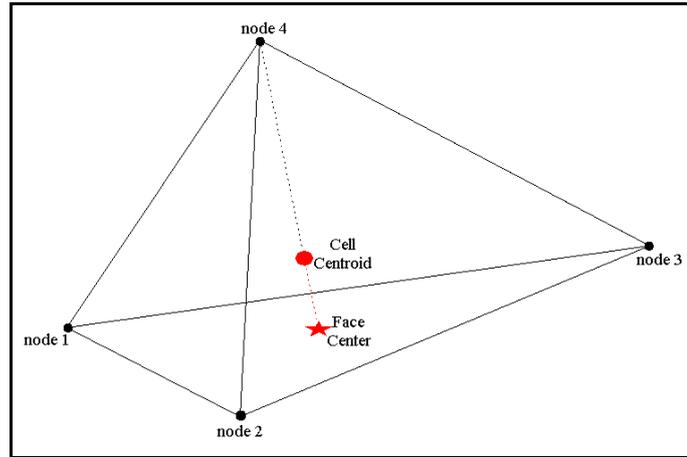


Figure 15. Reconstruction Scheme for Cell-Centered Approach [9]

$$\tilde{U}_{\text{node}} = \frac{\left(\sum_{i=1}^{\text{\# of cells}} \frac{\tilde{U}_{\text{center},i}}{r_i} \right)}{\left(\sum_{i=1}^{\text{\# of cells}} \frac{1}{r_i} \right)} \quad (3.43)$$

where $r_i = \left((x_{\text{center},i} - x_{\text{node}})^2 + (y_{\text{center},i} - y_{\text{node}})^2 + (z_{\text{center},i} - z_{\text{node}})^2 \right)^{1/2}$

It is well known that upwind algorithms are originated from 1-D gas dynamics equations. In order to eliminate the problem in multi dimensional problems rotational invariance property of the Euler equations is used. This

property does not only allow the proof of hyperbolicity of Euler equations in time but also can be used for computational purposes to deal with domains that are not aligned with Cartesian directions [44].

By this way the flux vector is rotated such that it is coincident with the normal of the cell face. Equations (3.7), (3.8) and (3.9) are converted to

$$\cos\theta^{(y)} \cos\theta^{(z)}F(U) + \cos\theta^{(y)} \sin\theta^{(z)}G(U) + \sin\theta^{(y)}H(U) = T^{-1}F(TU) \quad (3.44)$$

where $\theta^{(y)}$ and $\theta^{(z)}$ are the direction cosine angles and T is the simple transformation matrix which is the product of $T^{(y)}$ and $T^{(z)}$.

3.3.2. *Viscous – Diffusive Fluxes*

The viscous fluxes $\vec{Q}(\vec{U})$, (Equations. 3.23, 3.24), are approximated at the cell face centers by first computing the velocity gradients at cell centroids, then averaging the values with neighbor cell centroids at the shared cell face. The velocity gradients at the cell centroids are calculated using the divergence theorem. This theorem can be considered as defining the average of the gradient of a scalar U as a function of its values at the boundaries of the finite volume under consideration. For an arbitrary volume Ω [12]:

$$\int_{\Omega} \vec{\nabla} \vec{U} \cdot d\Omega = \oint_S \vec{U} \cdot d\vec{S} \quad (3.45)$$

where S is the closed boundary surface. The averaged gradients can be defined as :

$$\left(\overline{\frac{\partial U}{\partial x}}\right)_\Omega \equiv \frac{1}{\Omega} \int_\Omega \frac{\partial \bar{U}}{\partial x} \cdot d\Omega = \frac{1}{\Omega} \oint_S \bar{U} \cdot \bar{n}_x \cdot d\bar{S} \quad (3.46)$$

$$\left(\overline{\frac{\partial U}{\partial y}}\right)_\Omega \equiv \frac{1}{\Omega} \int_\Omega \frac{\partial \bar{U}}{\partial y} \cdot d\Omega = \frac{1}{\Omega} \oint_S \bar{U} \cdot \bar{n}_y \cdot d\bar{S} \quad (3.47)$$

$$\left(\overline{\frac{\partial U}{\partial z}}\right)_\Omega \equiv \frac{1}{\Omega} \int_\Omega \frac{\partial \bar{U}}{\partial z} \cdot d\Omega = \frac{1}{\Omega} \oint_S \bar{U} \cdot \bar{n}_z \cdot d\bar{S} \quad (3.48)$$

For 3-D, Figure 16, control cell velocity gradients and temperature gradients at cell centroids came out to be:

$$\frac{\partial \bar{V}}{\partial x} \equiv \frac{1}{\Omega} \int_\Omega \frac{\partial \bar{V}}{\partial x} \cdot d\Omega = \frac{1}{\Omega} \sum_{k=1}^4 \bar{V}_k \cdot S_k \cdot \bar{n}_{x,k} \quad (3.49)$$

$$\frac{\partial \bar{V}}{\partial y} \equiv \frac{1}{\Omega} \int_\Omega \frac{\partial \bar{V}}{\partial y} \cdot d\Omega = \frac{1}{\Omega} \sum_{k=1}^4 \bar{V}_k \cdot S_k \cdot \bar{n}_{y,k} \quad (3.50)$$

$$\frac{\partial \bar{V}}{\partial z} \equiv \frac{1}{\Omega} \int_\Omega \frac{\partial \bar{V}}{\partial z} \cdot d\Omega = \frac{1}{\Omega} \sum_{k=1}^4 \bar{V}_k \cdot S_k \cdot \bar{n}_{z,k} \quad (3.51)$$

$$\frac{\partial T}{\partial x} \equiv \frac{1}{\Omega} \int_\Omega \frac{\partial T}{\partial x} \cdot d\Omega = \frac{1}{\Omega} \sum_{k=1}^4 T_k \cdot S_k \cdot \bar{n}_{x,k} \quad (3.52)$$

$$\frac{\partial T}{\partial y} \equiv \frac{1}{\Omega} \int_\Omega \frac{\partial T}{\partial y} \cdot d\Omega = \frac{1}{\Omega} \sum_{k=1}^4 T_k \cdot S_k \cdot \bar{n}_{y,k} \quad (3.53)$$

$$\frac{\partial T}{\partial z} \equiv \frac{1}{\Omega} \int_\Omega \frac{\partial T}{\partial z} \cdot d\Omega = \frac{1}{\Omega} \sum_{k=1}^4 T_k \cdot S_k \cdot \bar{n}_{z,k} \quad (3.54)$$

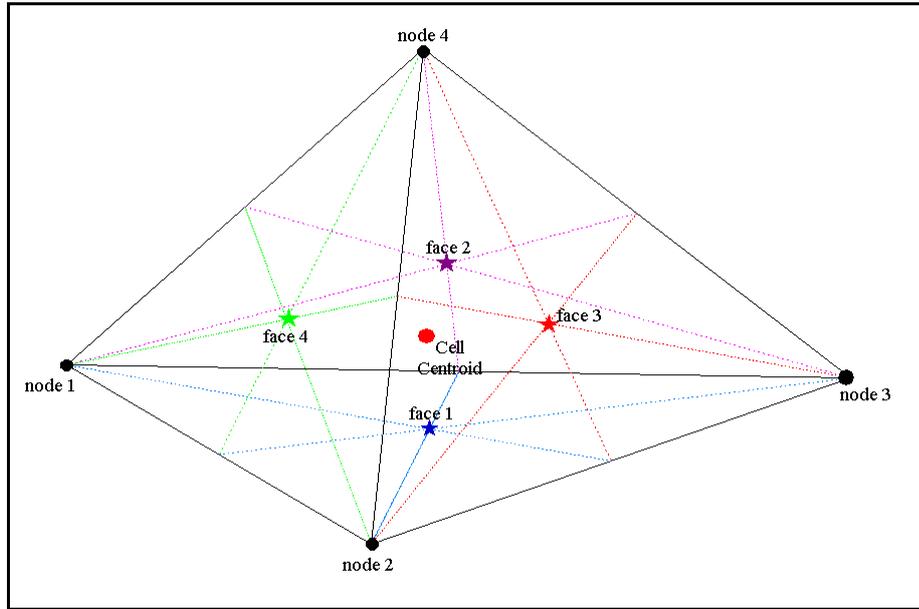


Figure 16. 3-D Velocity Gradient Computation Stencil at Cell Centroid

After the velocity gradients are calculated at cell centers, viscous stresses at the cell faces obtained by averaging the cell center values with the neighbor cell values at the shared face, Figure 17. For boundaries, since no velocity gradient have been calculated at ghost cells, an approximation has been made so that the gradient values calculated at the cell centroids that are near a boundary are projected to the boundary face. In other words, the viscous terms at the boundary faces took the value of the nearby cell centroid's.

3.4. Temporal Discretization

Equation (3.24) can be written as:

$$\Omega \frac{\partial}{\partial t} \bar{U}_i + R_i = 0 \quad \text{for} \quad i = 1, 2, \dots, \# \text{ of cells} \quad (3.55)$$

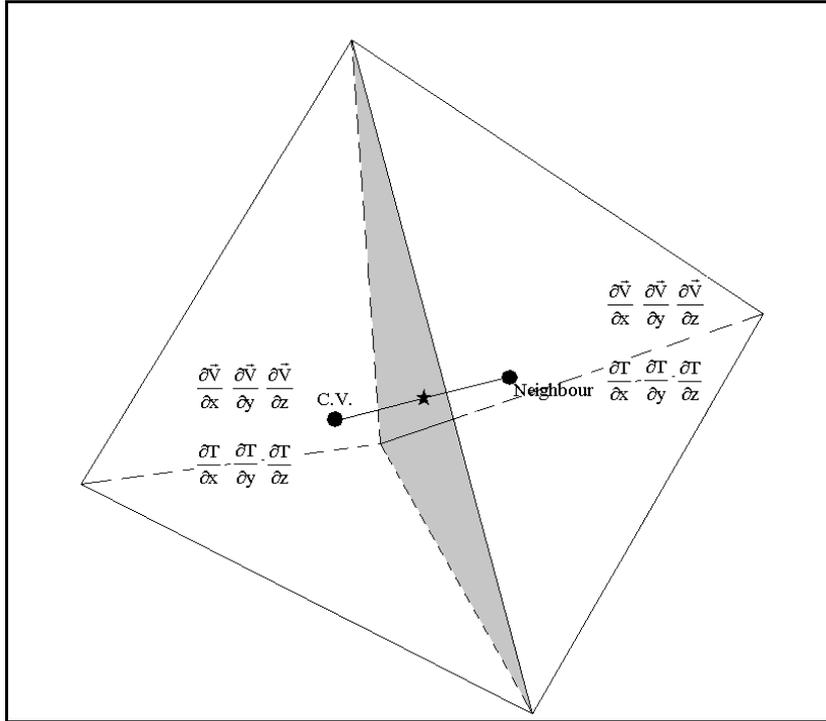


Figure 17. 3-D Face Viscous Stress Calculation Stencil

where

$$R_i = \sum_{j=1}^4 \left((F - Q_x) n_x + (G - Q_y) n_y + (H - Q_z) n_z \right) \cdot \Delta S_j \quad (3.56)$$

R_i is the residual which is the summation of the fluxes through the four faces of the tetrahedral computational cell. These set of equations are integrated in time using a fully explicit third order Runge-Kutta scheme developed by Jameson [11].

$$U_i^{(0)} = U_i^{(n)}$$

$$U_i^{(1)} = U_i^{(n)} - \alpha_1 \Delta t R(U_i^{(0)})$$

$$\begin{aligned}
U_i^{(2)} &= U_i^{(n)} - \alpha_2 \Delta t R(U_i^{(1)}) \\
U_i^{(3)} &= U_i^{(n)} - \alpha_3 \Delta t R(U_i^{(2)}) \\
U_i^{(n+1)} &= U_i^{(3)}
\end{aligned} \tag{3.57}$$

This scheme is used for both steady state and unsteady calculations. For unsteady calculations, time step is chosen to be global constant value which is the minimum of all computational cells. On the other hand for steady state calculations, spatially variable time stepping is used for fast steady state convergence. Time step is calculated for each computational cell for spatially variable time stepping by :

$$\Delta t_i = \frac{\text{CFL} \cdot \Delta x_{\min,i}}{\bar{V}_i + c_i} \tag{3.58}$$

There are some techniques applied to accelerate the convergence to steady state. First, as explained in previous paragraph, spatially variable time step is used in order to eliminate the computational speed loss due to usage of big time steps where not necessary. The other important accelerator is the usage of implicit residual smoothing.

Implicit residual smoothing means introducing an additional implicit term, which comes from implicit averaging of the residuals with their neighbors, to increase maximum allowable Courant number and convergence rate. By this method, larger time steps can be used which will accelerate the convergence.

Simply, the code is run at a large, most probably unstable Courant number, where the stability is obtained by smoothing the residual using an implicit character. If the solution converges, implicit residual smoothing does not change the result.

In order to prevent the loss of time accuracy in unsteady solutions, CFL number has been taken to be less than 1.0 in moving body calculations and implicit residual smoothing has not been used.

3.5. Turbulence Model

One-equation turbulence model of Spalart-Allmaras has been introduced in this study. The Spalart-Allmaras [91] turbulence model is a relatively simple one-equation model that solves a modeled transport equation for the kinematic eddy (turbulent) viscosity. It was designed specifically for aerospace applications involving wall-bounded flows and has been shown to give good results for boundary layers subjected to adverse pressure gradients.

The differential equation is derived by “using empiricism and arguments of dimensional analysis, Galilean invariance and selected dependence on the molecular viscosity” [91]. Guillen , *et al.* [92] stated that Spalart Allmaras one-equation turbulence model does not require finer grid resolution than the one required to capture the velocity field gradients with algebraic models.

The transported variable in the Spalart-Allmaras model, $\tilde{\nu}$, is identical to the turbulent kinematic viscosity except in the near-wall (viscous-affected) region. The transport equation for $\tilde{\nu}$ is

$$\frac{\partial \tilde{\nu}}{\partial t} + \vec{V} \cdot \nabla \tilde{\nu} = C_{b1}(1 - f_{t2})\tilde{S}\tilde{\nu} + \frac{1}{\sigma}[\nabla \cdot \{(\nu + \tilde{\nu})\nabla \tilde{\nu}\} + C_{b2}\nabla \tilde{\nu} \nabla \tilde{\nu}] - \left(C_{w1}f_w - \frac{C_{b1}}{\kappa^2}f_{t2} \right) \left(\frac{\tilde{\nu}}{d} \right)^2 \quad (3.59)$$

where the right-hand-side terms represent turbulence eddy viscosity production, diffusion and near-wall turbulence destruction terms, respectively. $\sigma = \frac{2}{3}$, $C_{b1} = 0.1355$, $C_{b2} = 0.622$, $\kappa = 0.4187$ and $C_{w1} = \frac{C_{b1}}{\kappa^2} + \frac{1 + C_{b2}}{\sigma} = 3.2059$ are constants, d is the minimum distance from the wall and ν is the molecular kinematic viscosity.

Turbulent viscosity is defined as:

$$\mu_{\text{turb}} = \rho \tilde{\nu} f_{v1} \quad (3.60)$$

where $f_{v1} = \frac{\chi^3}{\chi^3 + C_{v1}^3}$ is the viscous damping function, $\chi \equiv \frac{\tilde{\nu}}{\nu}$ and $C_{v1} = 7.1$ is a constant term.

The vorticity magnitude S which appears in the turbulence production term is modified such that \tilde{S} maintains its log-layer behavior [92]:

$$\tilde{S} = S f_{v3}(\chi) + \frac{\tilde{\nu}}{\kappa^2 d^2} f_{v2}(\chi) \quad (3.61)$$

$$\text{where } f_{v2}(\chi) = 1 - \frac{\chi}{1 + \chi f_{v1}} \quad (3.62)$$

$$\text{and } f_{v3} = 1 \quad (3.63)$$

Modified versions of the functions $f_{v2}(\chi)$ and $f_{v3}(\chi)$ are introduced by Spalart in order to eliminate the poor convergence of the residual turbulence especially near reattachment:

$$f_{v2}(\chi) = \left(1 + \frac{\chi}{C_{v2}}\right)^{-3} \quad (3.64)$$

$$\text{and } f_{v3}(\chi) = \frac{(1 + \chi f_{v1})(1 - f_{v2}(\chi))}{\chi} \quad (3.65)$$

Guillen , *et al.* [92] stated that these forms of $f_{v2}(\chi)$ and $f_{v3}(\chi)$ functions result in a modification of the natural laminar-turbulent transition of the Spalart-Allmaras turbulence model.

The function f_{t2} is introduced into the production and destruction terms in order to make $\tilde{\nu} = 0$ a stable solution to the linearized problem [93]. This term does not allow eddy viscosity to increase in regions where it has the value corresponding to half of the laminar viscosity [94].

$$f_{t2} = C_{t3} \exp(-C_{t4} \cdot \chi^2) \quad (3.66)$$

where $C_{t3} = 1.3$ and $C_{t4} = 0.5$ are constants.

The eddy viscosity production is related to the vorticity [94]. S is a scalar measure of the deformation tensor which is based on the magnitude of vorticity.

$$S = |\vec{S}| = |2 \cdot \Omega_{ij}| \quad \text{and} \quad (3.67)$$

$$\Omega_{ij} = \frac{1}{2} \left\{ \left(\frac{\partial w}{\partial y} - \frac{\partial v}{\partial z} \right) \vec{i} + \left(\frac{\partial u}{\partial z} - \frac{\partial w}{\partial x} \right) \vec{j} + \left(\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \right) \vec{k} \right\} \quad (3.68)$$

$$S = \sqrt{\left(\frac{\partial w}{\partial y} - \frac{\partial v}{\partial z} \right)^2 + \left(\frac{\partial u}{\partial z} - \frac{\partial w}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \right)^2} \quad (3.69)$$

Equation (3.69) takes the following form in two-dimensional space;

$$S = \text{abs} \left(\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \right) \quad (3.70)$$

Saxena and Nair [94] stated that in a boundary layer the blocking effect of the wall is felt at a distance through the pressure term, which acts as the main destruction term for the Reynolds shear stress. In order to obtain a faster decaying behavior of destruction in the outer region of the boundary layer, a function f_w is used:

$$f_w = g \left\{ \frac{1 + C_{w3}^6}{g^6 + C_{w3}^6} \right\}^{\frac{1}{6}} \quad (3.71)$$

$$\text{where } g = r + C_{w2} (r^6 - r) \quad (3.72)$$

$$\text{and } r = \frac{\tilde{v}}{\tilde{S} \kappa^2 d^2} \text{ is the characteristic length} \quad (3.73)$$

and $C_{w2} = 0.3$, $C_{w3} = 2.0$ are constants.

In order to adapt Equation (3.59) into the integral form, divergence theorem is applied to conservative and diffusive parts of the equation. Production term, destruction term and a part of the diffusion term that is excluded from the area integral are left as source terms in the integral form of the equation.

$$\begin{aligned}
& \frac{\partial}{\partial t} \int_{\Omega} \tilde{v} \, d\Omega + \oint_S \tilde{v} \, \vec{V} \cdot \vec{n} \, dS - \oint_S \frac{1}{\sigma} (\nu + \tilde{\nu}) \nabla \tilde{v} \cdot \vec{n} \, dS \\
&= \int_{\Omega} \left\{ C_{b1}(1-f_{t2}) \tilde{S} \tilde{v} + \frac{C_{b2}}{\sigma} \nabla \tilde{v} \nabla \tilde{v} - \left(C_{w1} f_w - \frac{C_{b1}}{\kappa^2} f_{t2} \right) \left(\frac{\tilde{v}}{d} \right)^2 \right\} \cdot d\Omega
\end{aligned} \tag{3.74}$$

Non-dimensional form of the equation including the ALE formulation for moving bodies came out to be:

$$\begin{aligned}
& \frac{\partial}{\partial t} \int_{\Omega} \tilde{v} \, d\Omega + \oint_S \tilde{v} (\vec{V} - \vec{V}_{c.v.}) \cdot \vec{n} \, dS - \oint_S \frac{M_{\infty}}{Re_{\infty}} \frac{1}{\sigma} (\nu + \tilde{\nu}) \nabla \tilde{v} \cdot \vec{n} \, dS \\
&= \int_{\Omega} C_{b1}(1-f_{t2}) \left\{ S f_{v3}(\chi) + \frac{M_{\infty}}{Re_{\infty}} \frac{\tilde{v}}{\kappa^2 d^2} f_{v2}(\chi) \right\} \tilde{v} \, d\Omega \\
&\quad + \int_{\Omega} \frac{M_{\infty}}{Re_{\infty}} \left\{ \frac{C_{b2}}{\sigma} \nabla \tilde{v} \nabla \tilde{v} - \left(C_{w1} f_w - \frac{C_{b1}}{\kappa^2} f_{t2} \right) \left(\frac{\tilde{v}}{d} \right)^2 \right\} \cdot d\Omega
\end{aligned} \tag{3.75}$$

Note that as it was for the conservation equations, grid velocity is only introduced in the convective term.

Special attention has been given to the f_w term which has a dimensional term r inside. This term is non-dimensionalized as follows:

$$r = \frac{M_{\infty}}{Re_{\infty}} \frac{\tilde{v}}{\tilde{S} \kappa^2 d^2} \tag{3.76}$$

where all terms in this equation are previously non-dimensionalized.

The integral compact form of Equation (3.75), which is suitable for numerical calculations, is given by;

$$\frac{\partial}{\partial t} \int_{\Omega} \vec{U}_{\text{turb}} d\Omega + \oint_S (\vec{F}_{\text{turb,convective}} - \vec{F}_{\text{turb,diffusive}}) \cdot d\vec{S} = \int_{\Omega} \vec{Q}_{\text{turb}} d\Omega \quad (3.77)$$

where the column vector \vec{U}_{turb} represents the passive scalar vector for working variable \tilde{v} , column vector \vec{F}_{turb} represents the convective and diffusive flux terms, and column vector \vec{Q}_{turb} represents the source term;

$$\vec{U}_{\text{turb}} = |\tilde{v}| \quad (3.78)$$

$$\vec{F}_{\text{turb,convective}} = [\tilde{v}(\vec{V} - \vec{V}_{\text{c.v.}})] \quad (3.79)$$

$$\vec{F}_{\text{turb,diffusive}} = \left[\frac{M_{\infty}}{\text{Re}_{\infty}} \frac{1}{\sigma} (\nu + \tilde{v}) \nabla \tilde{v} \right] \quad (3.80)$$

$$\vec{Q}_{\text{turb}} = \left[\begin{array}{l} C_{b1}(1-f_{t2}) \left\{ S f_{v3}(\chi) + \frac{M_{\infty}}{\text{Re}_{\infty}} \frac{\tilde{v}}{\kappa^2 d^2} f_{v2}(\chi) \right\} \cdot \tilde{v} \\ + \frac{M_{\infty}}{\text{Re}_{\infty}} \left\{ \frac{C_{b2}}{\sigma} \nabla \tilde{v} \nabla \tilde{v} - \left(C_{w1} f_w - \frac{C_{b1}}{\kappa^2} f_{t2} \right) \cdot \left(\frac{\tilde{v}}{d} \right)^2 \right\} \end{array} \right] \quad (3.81)$$

Non-dimensionalized integral form of this transport Equation (3.75) has been solved separately from the conservation flow equations. For the spatial discretization of the conservative part of the equation, Equation (3.79), HLLC approximate Riemann solver has been used by implementing the turbulent working variable \tilde{v} as a passive scalar in the formulation, [44]. Face averaged values of the conservative flow variables are calculated using the same formulation described for Roe's upwind scheme, Equations 3.29 to 3.34. Face averaged value for the passive scalar turbulent working variable \tilde{v} at left and right states are calculated by :

$$U_{\text{turb,face,L/R}} = \rho_{L/R} \left(\frac{S_{L/R} - u_{L/R}}{S_{L/R} - S_{\text{face}}} \right) \tilde{v}_{L/R} \quad (3.81)$$

The HLLC flux term at the face of the cell interface is defined as:

$$F_{\text{turb,convective,face}} = \begin{cases} F_{\text{turb,L}} & \text{if } 0 \leq S_L \\ F_{\text{turb,L}} + S_L (U_{\text{turb,face,L}} - \tilde{v}_L) & \text{if } S_L \leq 0 \leq S_{\text{face}} \\ F_{\text{turb,R}} + S_R (U_{\text{turb,face,R}} - \tilde{v}_R) & \text{if } S_{\text{face}} \leq 0 \leq S_R \\ F_{\text{turb,R}} & \text{if } 0 \geq S_R \end{cases} \quad (3.82)$$

where S is the wave speed.

The diffusive part of the equation has been solved by using the same methodology described in Section 3.3.2 which was used for the computation of viscous fluxes.

A third order Runge-Kutta explicit scheme has been adapted for the temporal discretization.

Turbulence contribution is reduced to the viscous shear stress tensors and heat conduction terms as stated at Equations (3.13 – 3.18) and (3.19 – 3.21).

3.6. Boundary Conditions

The boundary conditions are enforced by using the idea of ghost cells at the boundaries. Types of the boundary conditions applied in the solver are explained below.

3.6.1. Far Field Boundary Condition

As the far field boundary condition for external flows, characteristic Riemann invariants (Equations 3.83 and 3.84) corresponding to the incoming and outgoing waves traveling in characteristic directions, which are defined as normal to the boundary, are used [15, 16].

$$R^+ = \bar{U}_{\text{int}} \cdot \bar{n} + \frac{2c_{\text{int}}}{\gamma - 1} \quad (3.83)$$

$$R^- = \bar{U}_{\infty} \cdot \bar{n} - \frac{2c_{\infty}}{\gamma - 1} \quad (3.84)$$

The invariants are used to determine the locally normal velocity component and speed of sound (Equations 3.85 and 3.86). By this way subsonic - supersonic inlet and subsonic – supersonic outlet boundary conditions are determined according to the characteristic directions automatically.

$$\bar{U} \cdot \bar{n} = \frac{1}{2} [R^+ + R^-] \quad (3.85)$$

$$c = \frac{\gamma - 1}{4} [R^+ - R^-] \quad (3.86)$$

Right going and left going waves and the Riemann invariants are illustrated in Figure 18 for subsonic inlet or outlet.

The density is computed from the entropy relation, and the pressure from the perfect gas law using the square of the speed of sound.

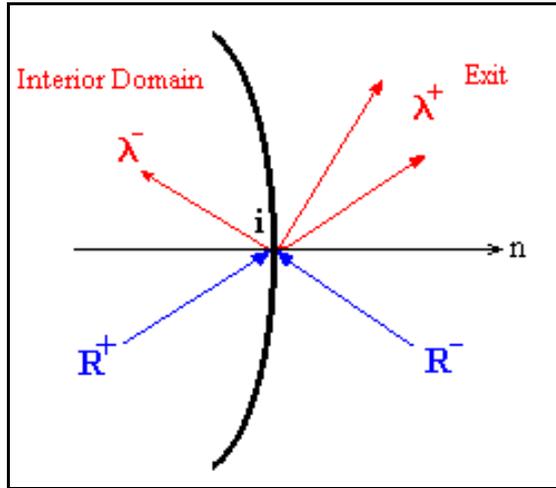


Figure 18. Illustration of Right & Left Going Waves and Riemann Invariants

Batina [22] stated that such method correctly accounts for wave propagation at far field which is important for convergence rate and serves as a non-reflecting boundary condition for unsteady applications.

For moving boundary problems, as it was stated before, the entire computational domain is rotated instead of deforming the mesh. Since far field boundary conditions are evaluated using the wave propagation in characteristic directions, movement of the mesh will not affect the results at far field. At each time step far field values are obtained at that specific condition. But it must be guaranteed that far field boundaries are enough far away from the moving body so that they are not influenced from the movement of the wall.

3.6.2. Wall Boundary Condition

The viscous wall boundary condition imposes a no-slip condition of the flow, a zero pressure gradient, and the appropriate heat transfer condition (adiabatic

or constant temperature) at the zone boundary (wall surface). The no-slip condition can involve a non-zero velocity if the wall is moving. In the application of this boundary condition, negative signed value of the cell face centered velocity components are assigned to the neighbour ghost cell. Also the velocity components are explicitly set to zero at the boundary faces for non moving walls. Pressure and density values at the ghost cells are taken as equal to the values in the adjacent interior cell.

The inviscid wall boundary condition imposes flow tangency at the zone boundary (wall surface) while maintaining the same total velocity as the point adjacent to the boundary. Only the velocity component of the ghost cell normal to the surface is assigned as the negative of the corresponding cell's value. By this way, flow tangency is enforced at the wall. Left and right states of wall boundary are set to be equal so that there exist no mass or energy transfer through these surfaces while only a pressure flux exists.

The moving wall boundary condition enables a tangential velocity to be applied at no-slip walls in order to model spinning motion of the body. For moving boundaries a corresponding treatment, that is explained for stationary walls, is applied to the relative velocity [79]. Hirsch [12] and Toro [44] proposed the following approach for evaluating normal velocity component at solid wall boundary moving with a speed $u_{C.V}$:

$$\mathbf{u}_{\text{ghost cell}} = -(\mathbf{u}_i - 2 \cdot \mathbf{u}_{C.V}) \quad \rho_{\text{ghost cell}} = \rho_i \quad p_{\text{ghost cell}} = p_i \quad (3.87)$$

3.6.3. Symmetry Boundary Condition

Symmetry boundary condition is the same as the condition defined for inviscid wall boundary condition. It is simply a non penetration boundary condition.

Flow tangency at the zone boundary (wall surface) is enforced by maintaining the same total velocity as the point adjacent to the boundary.

3.6.4. Turbulence Model Boundary Conditions

On the no-slip surfaces, the working variable $\tilde{\nu}$ is set to zero. For tangent-flow surfaces, in other words for the symmetry boundary condition, zero gradient of the working variable is applied. For far field boundary, it is checked whether it is inflow or outflow first and then working variable is set to $\tilde{\nu} = 1.0$ for the inflow boundaries which corresponds to a free stream turbulent kinematic viscosity of $\nu_t = 0.02786$. For outflow boundaries the value of $\tilde{\nu}$ is extrapolated from the interior mesh. Initial value of $\tilde{\nu}$ has been taken as the same with the free stream value.

3.7. Implementation of Rotation Into Governing Equations

It was stated that for the introduction of rotation into flow equations Arbitrary Lagrangian Eulerian (ALE) formulation would be used. In this formulation, the computational grid moves in any direction with a velocity different from the fluid velocity. The resulting nondimensionalized governing equations have been stated (Equations 3.1 – 3.22). Here, the physics of these equations will be stated.

Figure 19 shows the representative view of the first order state vector distribution in a control volume and discontinuous variation at the cell boundary. Now the fluid flow and boundary movement will be treated separately. First,

consider the fluid flow. Figure 20 shows the flow of fluid across the cell boundary between i^{th} and $i+1^{\text{th}}$ control volumes.

The Eulerian flux vector $F_{\text{eul}}(\mathbf{U})$ passing through the cell boundary for one-dimension is given by Roe [37, 38] as:

$$F_{\text{eul}}(\mathbf{U}) = F_{L,\text{eul}}(\mathbf{U}) + \sum_j \alpha_j \Lambda_{\text{eul}}^- \mathbf{R}^{(j)} \quad (3.88)$$

$$\text{where; } F_{L,\text{eul}} = \begin{bmatrix} \rho_L u_L \\ \rho_L u_L u_L + p \\ (\rho_L E_L + p)u_L \end{bmatrix} \quad (3.89)$$

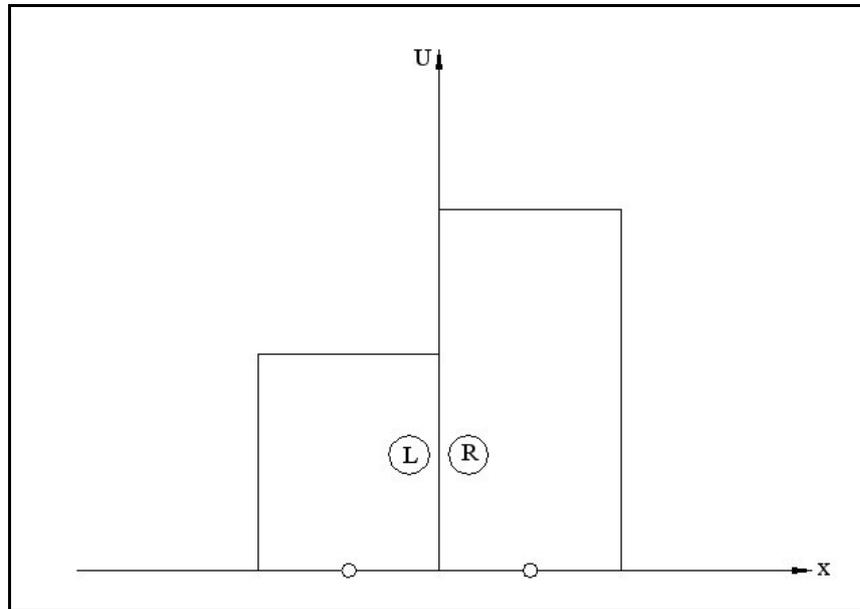


Figure 19. One Dimensional 1st Order Finite Volume Description

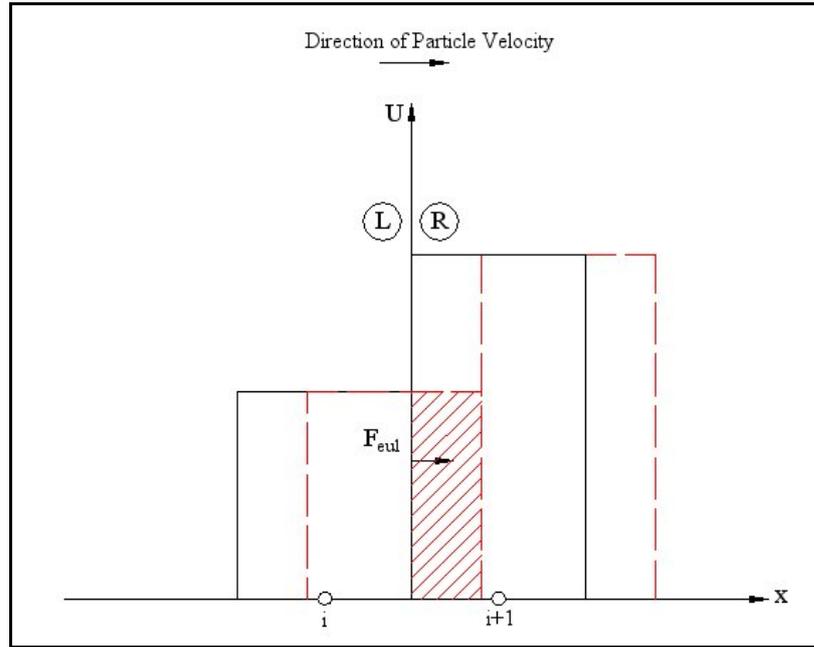


Figure 20. Motion of Fluid across the Cell Boundary

and eigenvalues , $\Lambda_{\text{eul}}^- = \begin{bmatrix} u - c \\ u \\ u + c \end{bmatrix}$ (3.90)

In Figure 21, this time the boundary is moved with the speed \dot{x} to the right and the fluid is thought to be stationary. The Lagrangian flux vector $F_{\text{lgr}}(U)$ passing through the cell boundary but in the opposite direction is:

$$F_{\text{lgr}}(U) = F_{\text{L,lgr}}(U) + F_{\text{lgr}}(\Delta U) \quad (3.91)$$

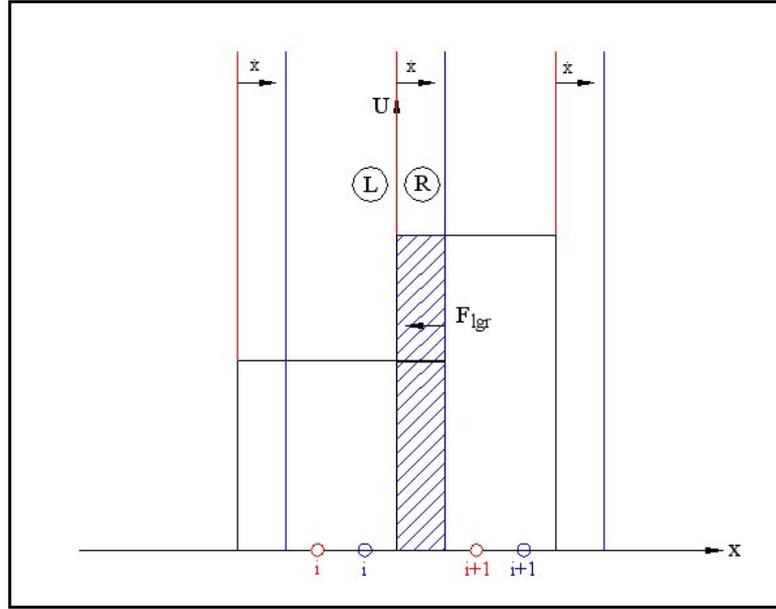


Figure 21. Motion of the Cell Boundary with respect to Stationary Fluid

$$\text{where; } F_{L,igr} = \begin{bmatrix} \rho_L \\ \rho_L u_L \\ \rho_L E_L \end{bmatrix} \cdot \dot{x}; \quad (3.92)$$

$$F_{igr}(\Delta U) = \begin{bmatrix} \Delta \rho \\ \Delta(\rho u) \\ \Delta(\rho E) \end{bmatrix} \cdot \dot{x}; \quad (3.93)$$

and;

$$\Delta \rho = \rho_R - \rho_L \quad (3.94)$$

$$\Delta(\rho u) = (\rho u)_R - (\rho u)_L \quad (3.95)$$

$$\Delta(\rho E) = (\rho E)_R - (\rho E)_L \quad (3.96)$$

After proper simplifications, the net flux passing through the cell boundary becomes:

$$F_{\text{ale}}(\mathbf{U}_{\text{rel}}) = F_{\text{eul}}(\mathbf{U}) - F_{\text{igr}}(\mathbf{U}) \quad (3.97)$$

$$F_{\text{ale}}(\mathbf{U}_{\text{rel}}) = F_{L,\text{ale}}(\mathbf{U}) + \sum_j \alpha_j \Lambda_{\text{ale}}^- \mathbf{R}^{(j)} \quad (3.98)$$

$$\text{where } \Lambda_{\text{ale}}^- = \begin{bmatrix} \mathbf{u} - \mathbf{c} - \dot{\mathbf{x}} \\ \mathbf{u} - \dot{\mathbf{x}} \\ \mathbf{u} + \mathbf{c} - \dot{\mathbf{x}} \end{bmatrix}; \quad (3.99)$$

$$\text{and } F_{L,\text{ale}} = \begin{bmatrix} \rho_L (\mathbf{u}_L - \dot{\mathbf{x}}) \\ \rho_L \mathbf{u}_L (\mathbf{u}_L - \dot{\mathbf{x}}) + p \\ (\rho_L E_L + p)(\mathbf{u}_L - \dot{\mathbf{x}}) + \dot{\mathbf{x}} p \end{bmatrix} \quad (3.100)$$

The above numerical discretization of Roe [37, 38] for moving boundaries is valid for the previously derived three dimensional governing equations.

From a physical point of view, the grid motion only affects the convective variables. As shown above, to calculate the new convective terms and eigenvalues, the velocity $\vec{V}_{\text{c.v.}}$ of the face of a control volume is required. Trepanier *et al.* [79] stated that for deforming meshes, the total volumetric increment is composed of elementary increments along each of its faces. Accordingly, the relevant facial velocity associated with this facial volume increment $\Delta\Omega$, during a time step Δt is defined by

$$\bar{V}_{c.v.} = \frac{\Delta\Omega}{S\Delta t} \quad (3.101)$$

Trepanier *et al.* ended up with a finite volume simple explicit approach noting that the grid is deforming.

$$U^{n+1} = \frac{\Omega^n}{\Omega^{n+1}} \left\{ U^n + \frac{\Delta t}{\Omega^n} \sum_k F_{ale,k}(U_{rel}) S_k \right\} \quad (3.102)$$

This is also the same as the present case although the grid is not deforming. When volume does not change, above equation reduces to:

$$(U^{n+1} - U^n)\Omega = \Delta t \sum_k F_{ale,k}(U_{rel}) S_k \quad (3.103)$$

In this study, the computational domain is rotated at each time step without deforming the computational cells. In other words, the grid remains undistorted and follows the motion of the body. For non-deforming control volumes, the facial volumetric increments sum up to zero. So, the use of the flux vectors and eigenvalues derived at the beginning of this chapter guarantees the satisfaction of geometric conservation law. Barakos *et al.* [24] also stated that there is no need to apply any geometric conservation law in this formulation.

Some further literature survey on geometric conservation laws have been performed in order to avoid error propagation in solutions due to omission of these laws for further applications of deforming meshes.

The geometric conservation laws are defined by Trepanier *et al.* [79, 80, 81] as follows: There are two additional equations besides the physical conservation laws which state, the balance between the relevant geometric parameters for static or moving meshes. First one states that the cell volumes must be closed by its

surfaces (Surface Conservation Law). Second one states that the volumetric increment of a moving cell must be equal to the sum of the changes along the surfaces that enclose the volume (Volume Conservation Law). A numerical scheme that does not satisfy these equations produces errors in flow fields. The numerical violation of surface conservation law will lead to a misrepresentation of the convective velocities while violation of volume conservation law will produce extra sources and sinks in the physically conservative media. Also, the violation of these laws brings sever restrictions to the capabilities of the numerical solvers. These geometric laws are stated as :

$$\frac{\partial \Omega}{\partial t} - \oint_B \vec{V}_{C.V.} \cdot d\vec{S} = 0 \quad \text{Volume Conservation Law} \quad (3.104)$$

$$\oint_B \vec{a} \cdot d\vec{S} = 0 \quad \text{Surface Conservation Law} \quad (3.105)$$

where Ω is the volume of the material volume; S is the vectorized area in the outward normal direction and \vec{a} is an arbitrary direction.

Volume conservation law states that the volume increase in a control cell during a time interval equals to the summation of the volumetric increases along its faces. Surface conservation law states that each control cell must be closed by its surfaces.

Trepanier *et al.* [79, 80, 81] suggest a method, so called Implicit Geometric Conservation Laws, to satisfy the above laws. According to them, once the initial position of the grid and its motion have been specified, the volumes and surfaces provide sufficient geometric information to maintain the geometric conservation laws without using any law other than geometric-based information. In other words, exact numerical computation of volumes and surfaces automatically satisfies the conservation laws. By this method, there is no need for the solution of geometric

conservation laws simultaneously with the flow equations. They are implicitly satisfied in the numerical schemes independent of the method used.

Surface conservation law is usually satisfied for finite volume schemes in two dimensions but special attention must be given in three dimensional domains. The way to maintain surface conservation law is to evaluate the surface vectors exactly.

Satisfying the volume conservation law requires the exact calculation of the facial volumetric increments. This is not a big problem in two dimensions but in three dimensions a special attention must be given to have planar surfaces. Use of unstructured tetrahedrons directly solves this problem since it is guaranteed that the resultant face generated because of grid motion will be planar. This comes from the evident fact that three non-collinear points define a plane. Also, the direct computation of volumes and surfaces is not computationally expensive in tetrahedrons.

Trepanier, *et al.*, after several manipulations, end up with an equation by which facial velocities can be approximated by a constant which is expressed in terms of the exact facial increment. (Equation 3.101) They have shown that the error in this approximation came out to be at least second order.

In this study, a finite volume solver has been developed knowing the facts that, from a physical point of view, the grid motion only affects the convective fluxes and from a mathematical point of view, the grid motion only modifies the eigenvalues of the Jacobian. Also, as it was stated before, non deforming computational mesh has been used in this work. So there is no need for an effort to satisfy the geometric conservation laws explicitly.

The velocities of the computational grids are simply evaluated by subtracting the grid locations after domain movement from their previous locations

and then dividing the resulting displacement to the time step used (Equations 3.106, 3.107, 3.108). The evaluation of the locations of the computational grid points after rotation is explained in detail in Section 3.8.

$$u_{C.V.,i} = \frac{x_i^{n+1} - x_i^n}{\Delta t} \quad (3.106)$$

$$v_{C.V.,i} = \frac{y_i^{n+1} - y_i^n}{\Delta t} \quad (3.107)$$

$$w_{C.V.,i} = \frac{z_i^{n+1} - z_i^n}{\Delta t} \quad (3.108)$$

3.8. Grid Movement

In steady coning motion of a missile, the longitudinal axis of the missile performs a rotation at a constant angular velocity about a line parallel to the free stream velocity vector and coincident with the projectile center of gravity, while oriented at a constant angle with respect to the free stream velocity vector [69]. In particular the projectile may rotate about its longitudinal axis also, Figure 22. So in order to simulate the motion of a missile realistically both rotations must be taken into account. Coning and spinning rates are given by aeroballistics analysis. With respect to the fixed coordinate system, the vertical and horizontal components of angle of attack α and β , vary in a periodic fashion as the projectile rotates about the free stream velocity vector. However, the total angle of attack, $\alpha_t \approx \sqrt{\alpha^2 + \beta^2}$ is constant [69].

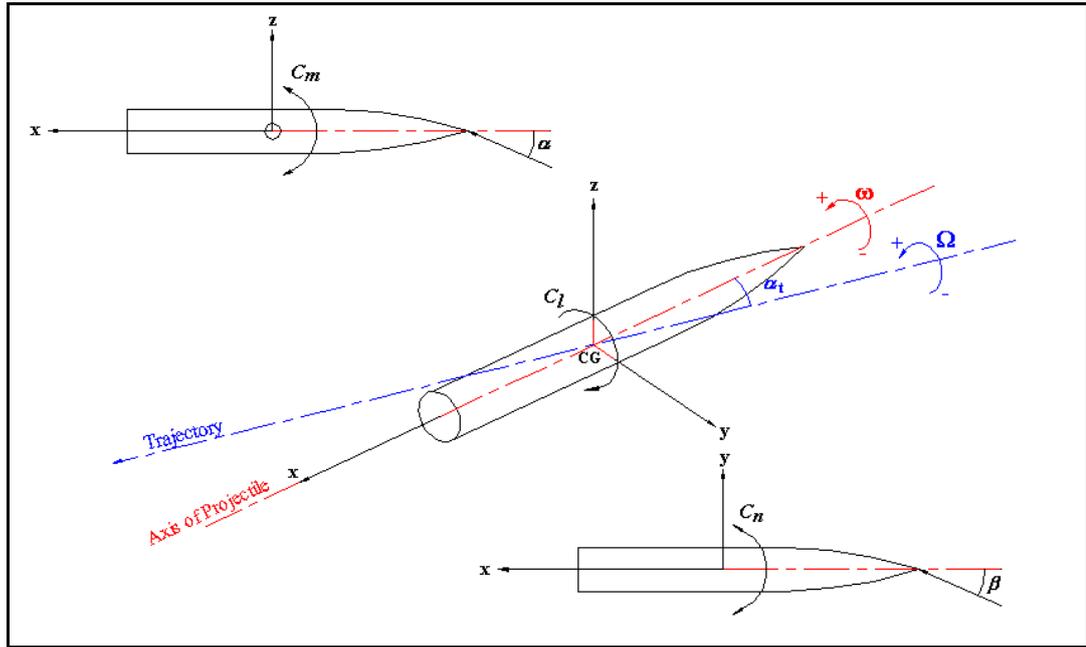


Figure 22. Schematic Figure of Rotational Motion of a Missile

The rotation of the computational grid around the missile about the axis of projectile and about the free stream velocity vector can be performed by multiplication of simple transformation matrices [95].

The solver that has been developed introduces angle of attack α and angle of side slip β by initially rotating the computational mesh first with angle α and then with angle β in order to make the global x-axis coincident with the axis of trajectory. By this way, it is possible to give the velocity vector only in x-direction which makes the numerical calculations simpler. Related transformation matrices are given by Equations (3.109) to (3.112).

$$[X'] = [X] \cdot [R_{z\beta}] \cdot [R_{y\alpha}] \quad (3.109)$$

where

$$[X] = \begin{bmatrix} x^n \\ y^n \\ z^n \end{bmatrix} \quad (3.110)$$

is the initial position of the computational grid.

$$[R_{z\beta}] = \begin{bmatrix} \cos \beta & \sin \beta & 0 \\ -\sin \beta & \cos \beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.111)$$

$$[R_{y\alpha}] = \begin{bmatrix} \cos \alpha & 0 & -\sin \alpha \\ 0 & 1 & 0 \\ \sin \alpha & 0 & \cos \alpha \end{bmatrix} \quad (3.112)$$

The matrix multiplication given below (Equation 3.113) describes spinning motion and coning motion of the projectile in the given order. These matrix calculations are performed at each time step of the numerical calculations in order to simulate the rotating motion of the projectile.

Note that initially the projectile has been rotated to the axis of trajectory due to the algorithm of the flow solver. Therefore, for the rotation of the projectile around its own axis first, the domain must be returned back to its origin. After rotation is performed around the original global x-axis, the x-axis of the projectile is again made coincident with the axis of trajectory for the coning motion around the axis of trajectory. Transformation matrices in Equation (3.113) represent this procedure.

$$[X''] = [X'] \cdot \{ [R_{y\alpha}]^{-1} \cdot [R_{z\beta}]^{-1} \} \cdot [R_{x\theta}] \cdot \{ [R_{z\beta}] \cdot [R_{y\alpha}] \} \cdot [R_{x\phi}] \quad (3.113)$$

where

$$[X''] = \begin{bmatrix} x^{n+1} \\ y^{n+1} \\ z^{n+1} \end{bmatrix} \quad (3.114)$$

is the final position of the computational grid after one time step of rotation and coning motion.

Note that $\{[R_{y\alpha}]^{-1} \cdot [R_{z\beta}]^{-1}\}$ in Equation (3.113) is the inverse rotation matrix in order to move the axis of the body back to its original position and the product of $[X'] \cdot \{[R_{y\alpha}]^{-1} \cdot [R_{z\beta}]^{-1}\}$ is equal to the original global coordinate system.

Rotation matrix required to rotate the body around the original global x-axis which is coincident with body axis with angle θ is:

$$[R_{x\theta}] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & \sin \theta \\ 0 & -\sin \theta & \cos \theta \end{bmatrix} \text{ where } \theta = \omega \cdot \Delta t \quad (3.115)$$

Rotation matrix to rotate the body around the final global x-axis which is coincident with its trajectory axis with angle ϕ is :

$$[R_{x\phi}] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix} \text{ where } \phi = \Omega \cdot \Delta t \quad (3.116)$$

3.9. Parallel Processing

It was stated in Chapter 2 that only parallel architecture computers offer the promise of providing orders of magnitude greater computational power for the present problem. It is possible to deal with more complicated problems with using parallel processing due to the reduction in computational time.

For this purpose, a parallel version of the serial solver has been developed. A “Beawulf” parallel computing center that has been constructed in TÜBİTAK-SAGE that is composed of personal computers has been used for computations.

Domain decomposition has been performed by the program called “METIS” [63] which is obtained from open sources. An illustration of domain decomposition and the concept of interface boundary between partitions are given in Figure 23. An example of domain decomposition performed by “METIS for a 2-D computational domain is presented in Figure 24. “PVM – Parallel Virtual Machine” message passing software libraries of Linux operating system has been employed for information exchange between computing processors and the main computer.

The parallel code that has been developed is composed of two separate executables. The executable that has been named as “Master” works on the main processor and performs the following jobs in sequence:

1. Reads the necessary input data (subroutine READIN) (Inflow conditions, computation flags, etc. – A sample input data is given in APPENDIX A)
2. Reads the size of the computational mesh and allocates the memory to the arrays (subroutine GETSIZE)

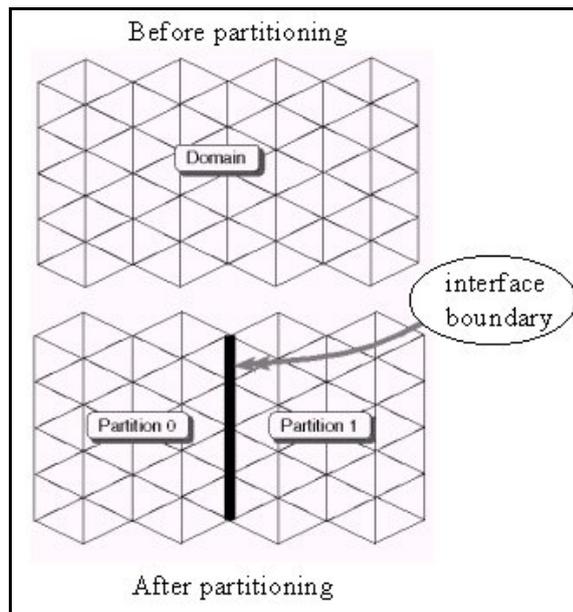


Figure 23. Domain Decomposition and Interface Boundary Concept

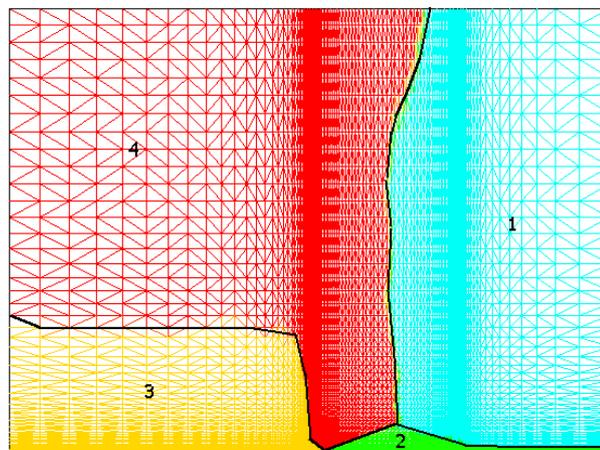


Figure 24. Sample Domain Decomposition for a 2-D Computational Domain

3. Reads the computational mesh (subroutine CONFIG) (Grid file format is given in APPENDIX B)

4. Rotates the computational mesh with side slip angle β and angle of attack α to make the global x-axis coincident with the axis of trajectory (subroutine CONFIG)
5. Calculates the neighbors of each of the computational cells according to the boundary condition flags read from computational mesh file (subroutine CONFIG)
6. Calls the domain decomposition program “METIS” to partition the computational mesh to the given number of processors (subroutine METIS)
7. Evaluates local index of cells owned by each partition and local index of interface ghosts cells for each partition (subroutine METIS)
8. Initializes the computational domain with free stream conditions or previous calculations’ results (subroutine INIT)
9. Spawns the computing nodes (subroutine PVMFSPAWN)
10. Sends the necessary input data to the computing nodes (subroutine SENDINIT)
11. Sends the local grid information of each partition to the computing nodes (subroutine SENDGRQ)
12. Waits the results from the computing nodes

13. Receives the conservative variables, turbulent viscosity variable, aerodynamic coefficients and additional information from each partition at prescribed iteration steps (subroutine RECVQ)
14. Calculates the overall residual and aerodynamic coefficients
15. Prints the residuals and aerodynamic coefficients both on screen and corresponding files
16. Prepares the output files for viewing the results on wall surface and inside the computational volume (subroutine RESULT)
17. Stops the program if prescribed convergence criteria is satisfied or prescribed maximum number of iterations is reached

The other executable that has been named as “Worker” works on each of the parallel computing processors and performs the following jobs in sequence:

1. Receives the necessary input data from the main program “Master” (subroutine RECVINIT)
2. Receives the local grid information from the main program “Master” (subroutine RECVGRQ)
3. Calculates the geometric properties of cells, that are surface normals, surface areas, cell volumes and direction cosines of cell faces (subroutine SETGEO)
4. Sets the interface ghost cells in each partition (subroutine SETIBC)

5. Sends the interface boundary conditions (conservative variables and velocity gradients) to the adjacent partition and receives interface boundary conditions from the adjacent partition (subroutine IBC)
6. Calculates the local time step at each iteration if the solution is for steady state. (For unsteady calculations minimum time step for each partition is calculated once and then sent to “Master”. At “Master” minimum time step among all partitions are calculated and the global time step is sent back to the partitions) (subroutine CALDTL)
7. Performs the grid movement, calculates the new locations of the computational grids and evaluates grid velocities (subroutine ALE)
8. Distributes the cell center values of the conservative variables to the cell nodes (subroutine QNODES)
9. Starts the third order Runge-Kutta time stepping algorithm (subroutine RK3)
10. For each of the three steps of this algorithm, for each computational cell:
 - a. Calculates the velocity gradients at cell center; (subroutine VGRAD)
 - b. Evaluates the values of conservative variables at cell faces, i.e. defines the left and right states at cell faces (subroutine QFACE)
 - c. Introduces the boundary conditions to right state of the cell faces (subroutine BC)

- d. Computes flux differences for each cell face and then computes the flux passing through that faces (subroutine ROE3D)
 - e. Computes viscous fluxes at each cell face (subroutine VISCOUS)
 - f. Sums up all fluxes passing through four faces of the tetrahedron
 - g. Calculates the pressure values from conservative variables (subroutine PRESSURE)
 - h. Applies implicit residual smoothing (subroutine SMOOTH)
11. Starts another third order Runge-Kutta time stepping algorithm for the calculation of the turbulent viscosity after one full step Runge-Kutta time stepping algorithm is finished for conservative variables. (subroutine SPALART)
12. Calculates the pressure values from conservative variables (subroutine PRESSURE)
13. Sends the interface boundary conditions (conservative variables and velocity gradients) to the adjacent partition and receives interface boundary conditions from the adjacent partition (subroutine IBC)
14. Calculates the residual for each partition (subroutine STEP)
15. Evaluates the aerodynamic coefficients by integrating the pressure values over wall surfaces (subroutine LOADS)

16. Sends the conservative variables, aerodynamic coefficients and additional information to the “Master” (subroutine SENDQ)

17. Repeats these steps until maximum number of iteration is reached or a “stop” command comes from the “Master”

Figure 25 illustrates the data flow between the executables “Master” and “Worker”s.

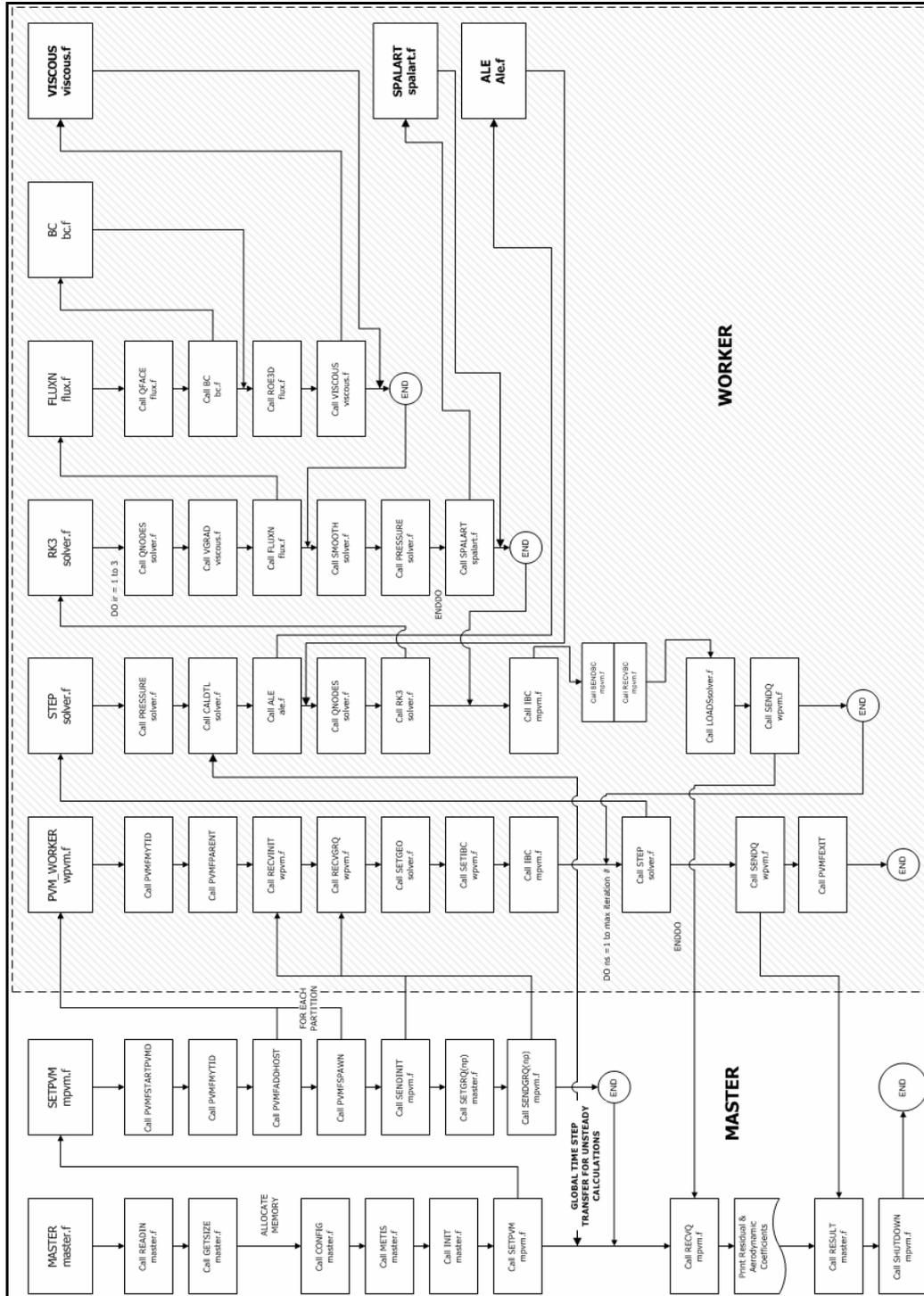


Figure 25. Data Flow between Executables “MASTER” and “WORKER”

CHAPTER 4

VERIFICATION AND VALIDATION STUDIES

4.1. 3-D Verification Studies for the Euler Solver

Verification studies for three-dimensional inviscid flows are performed on two different finned projectile configurations. The aim of this verification study is to observe the accuracy of the three-dimensional Euler solver developed before implementing the viscous flux computation algorithms and Spalart-Allmaras turbulence model into the code in two and three dimensional space. These studies will also serve as steady-state solution for the moving body calculations.

4.1.1. Inviscid Flow over Finned Projectile Configuration

Static and dynamic stability test results are available for a finned projectile configuration with aft-mounted fins arranged in cruciform pattern [96]. The projectile is a gun launched one with the fins folded within the body. After the projectile is launched, the fins fold out and damp out the rolling motion which is imparted to the projectile by the helical grooves in the gun barrel.

The model is a blunt nosed projectile having four control fins. Geometry of the projectile and the implemented coordinate axes are shown in Figure 26.

An unstructured mesh composed of 1,208,298 tetrahedral elements and 210,237 computational nodes is generated by grid generation tool of ANSYS. Far field is taken as approximately 10 times the length of the projectile away from the projectile body in order to correctly simulate the shock formations without an effect from far field boundary conditions as shown in Figure 27. High quality but not viscous mesh is generated especially at the nose of the projectile and the control fins where large pressure gradients are expected as shown in Figure 28.

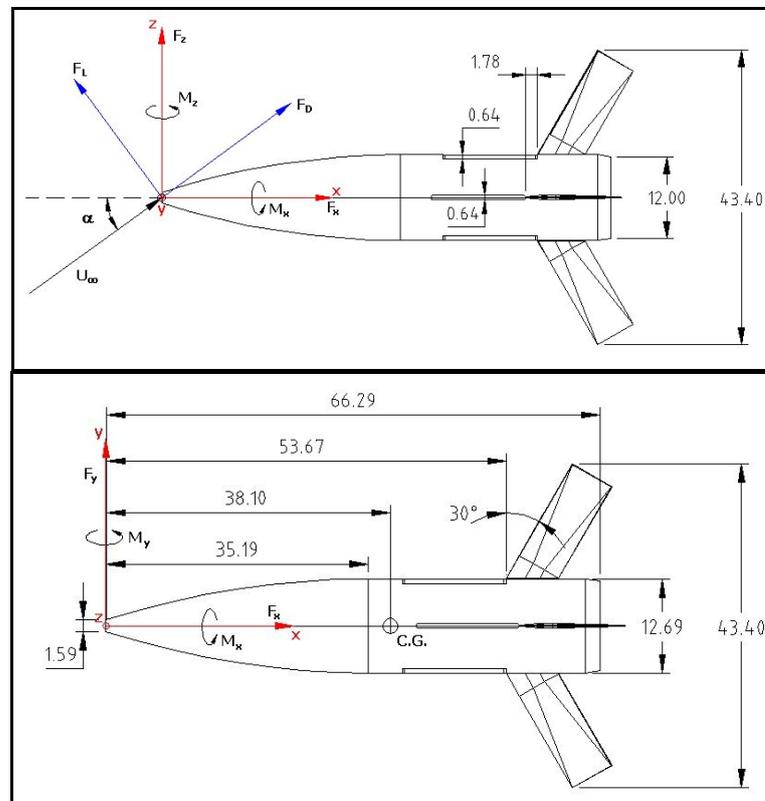


Figure 26. Detailed Geometry of the Finned Projectile and the Coordinate Axes
(all dimensions are in mm)

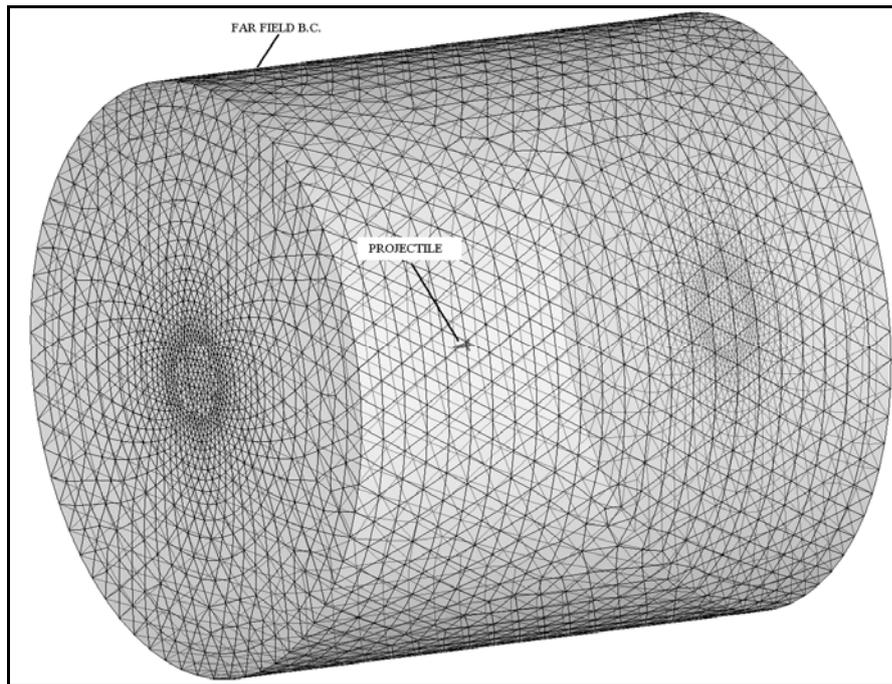


Figure 27. Unstructured Volume Mesh around the Finned Projectile

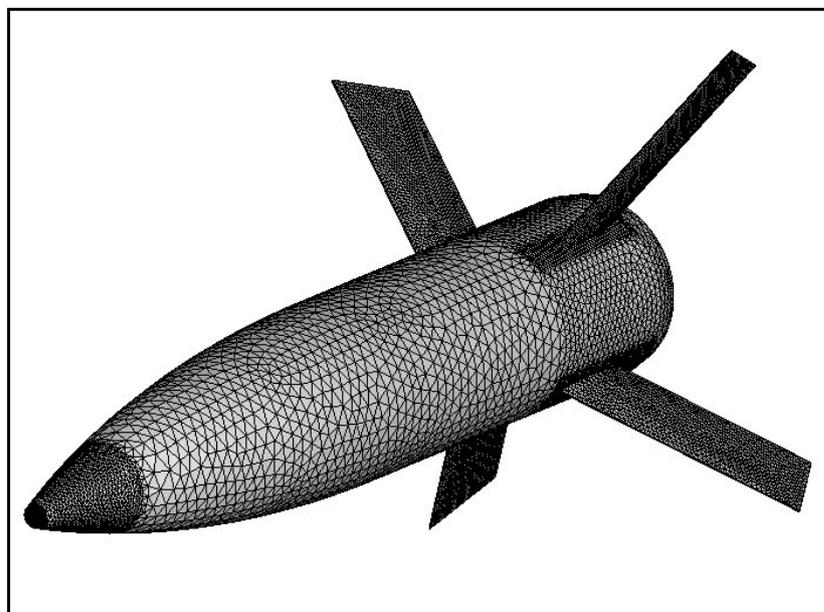


Figure 28. Unstructured Mesh Details on the Surface of the Finned Projectile

Steady-state inviscid flow calculations for this projectile are performed for subsonic, transonic and supersonic flow conditions at different angle of attack values up to 24°. Rotation of this projectile will be investigated in later chapters. Solution matrix is shown in Table 3.

For this model it is known that the experimental model is hold from the base for experimental measurements. Because of this, it was stated in the reference [96] that all drag measurements were corrected to a condition of free stream static pressure acting on the base of the model. In order to make a meaningful comparison of computational results with experimental measurements, base pressure is also changed to free stream conditions after full convergence has achieved.

Table 3. Solution Matrix for Finned Projectile

Mach # (M)	Angle of Attack (α)
0.7	0°, 5°, 10°, 17°, 24°
0.9	0°, 5°, 10°, 17°, 24°
1.2	0°, 5°, 10°, 17°, 24°

In Figure 29, Figure 30 and Figure 31 the variation of drag force coefficient, lift force coefficient and pitching moment coefficient with angle of attack at different Mach numbers in comparison with experimental results are shown respectively.

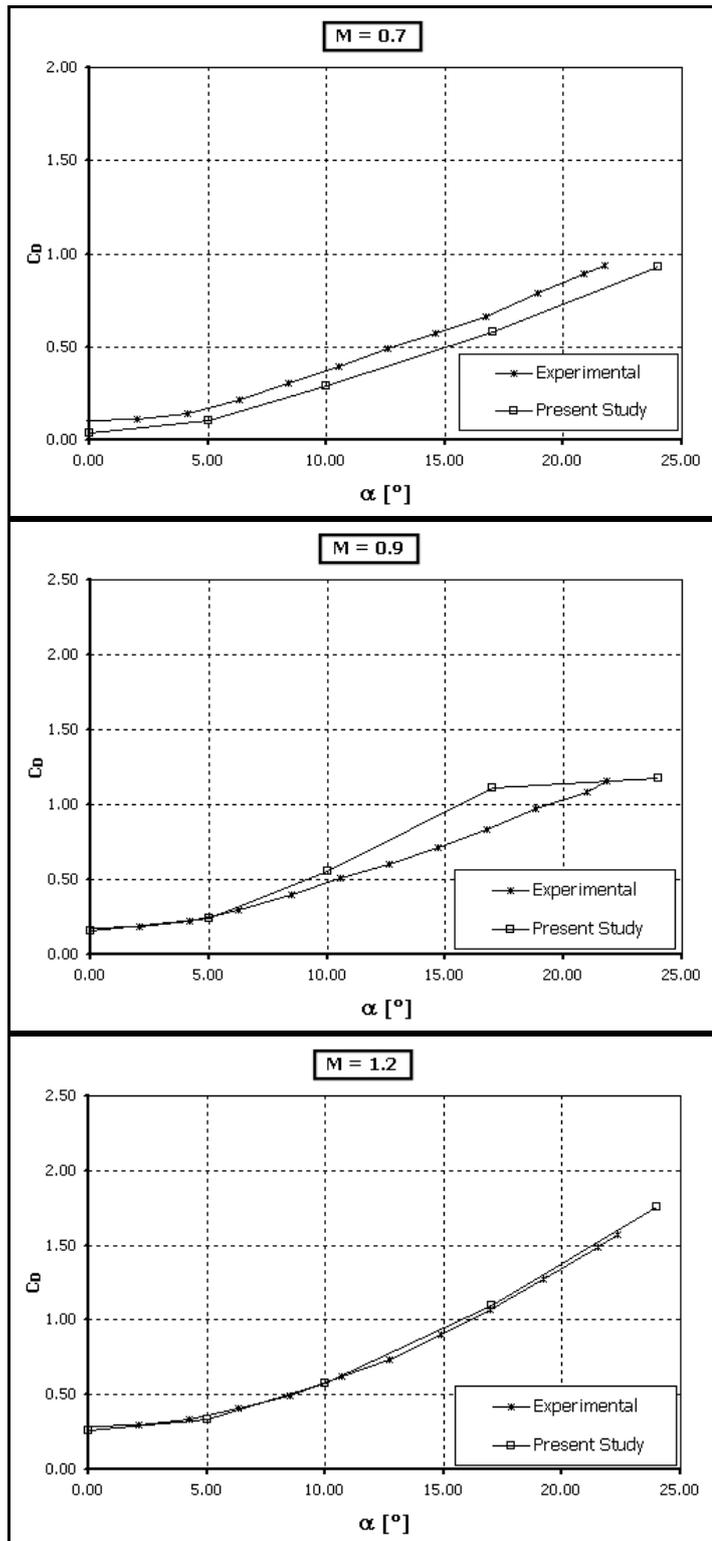


Figure 29. Finned Projectile-Variation of Drag Coefficient w.r.t. Angle of Attack

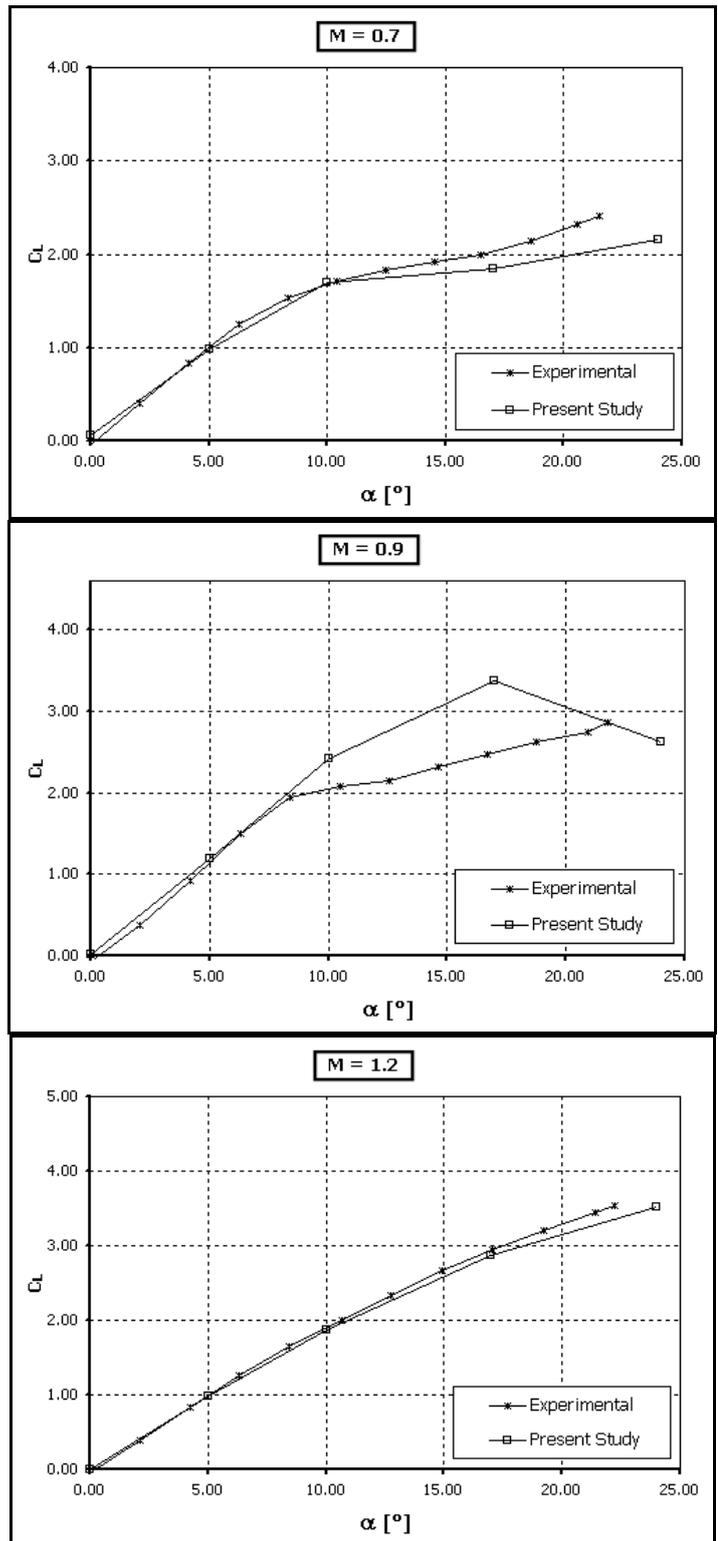


Figure 30. Finned Projectile-Variation of Lift Coefficient w.r.t. Angle of Attack

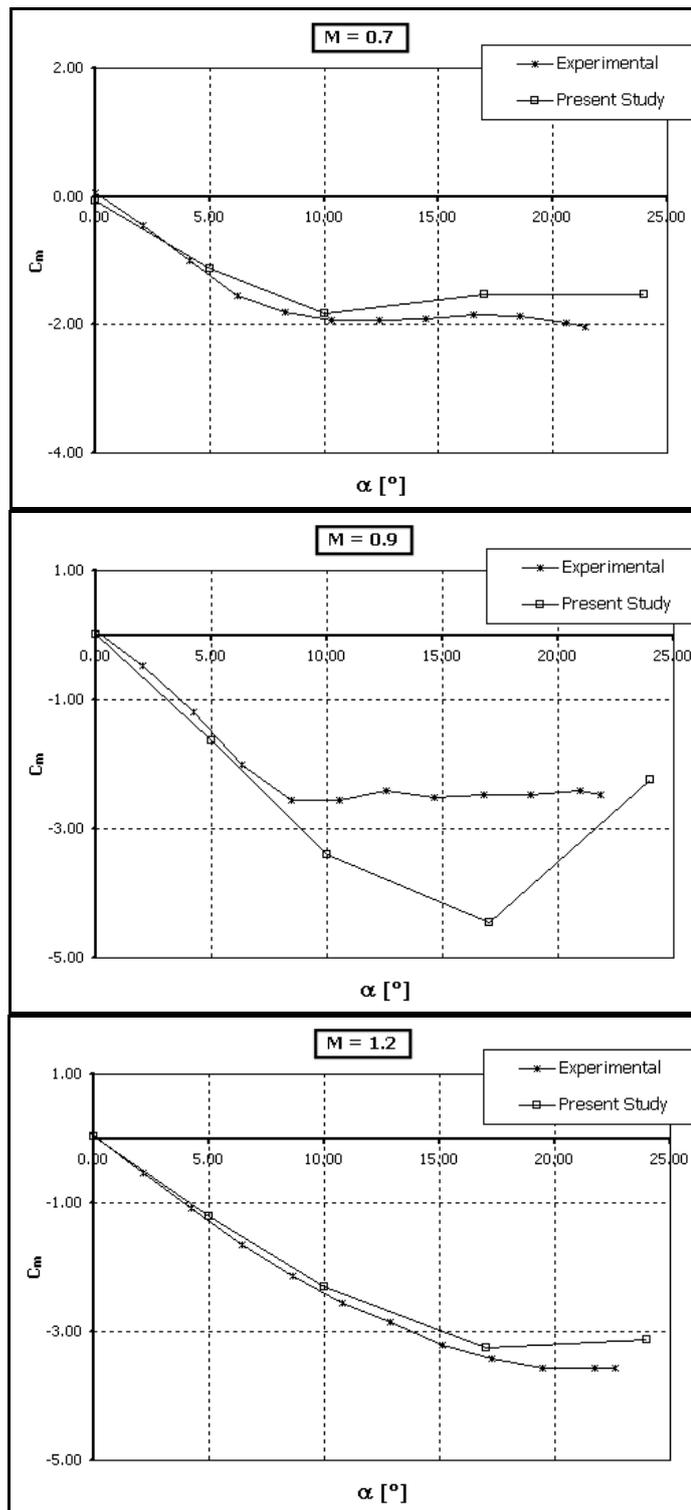


Figure 31. Finned Projectile-Variation of Pitching Moment Coefficient w.r.t. Angle of Attack

If the results are examined in general, the variation of aerodynamic coefficients with respect to angle of attack seems to be in qualitative correspondence with experimental measurements. Especially for low angle of attacks ($\alpha < 10^\circ$), the results came out to be exactly the same with experimental data as expected from an Euler solver. Results are quite satisfactory for all Mach numbers except for the sonic conditions where viscous effects are dominant. But especially for supersonic flows the computational results are acceptable both in numerical accuracy and behavior of the variations of the aerodynamic coefficients.

4.1.2. Inviscid Flow over Sparrow Missile

An experimental investigation was conducted on a model of a wing control version of Sparrow missile to determine the static aerodynamic characteristics over a wide range of angle of attack at supersonic speeds by Langley Research Center, NASA [97]. This missile is selected as a verification case in order to see the performance of the inviscid solver at high supersonic flow conditions.

The model of Sparrow consists of an ogive-cylinder body, cruciform wings and in-line tails. Geometry of the missile and the implemented coordinate axes are shown in Figure 32.

An unstructured mesh composed of 1,238,305 tetrahedral elements and 214,630 computational nodes is generated by the grid generation tool of ANSYS. Far field is taken as approximately 10 times the length of the projectile away from the missile as shown in Figure 33. High quality but not viscous mesh is generated especially at the nose of the projectile and at the wings and tails where large pressure gradients are expected as indicated in Figure 34.

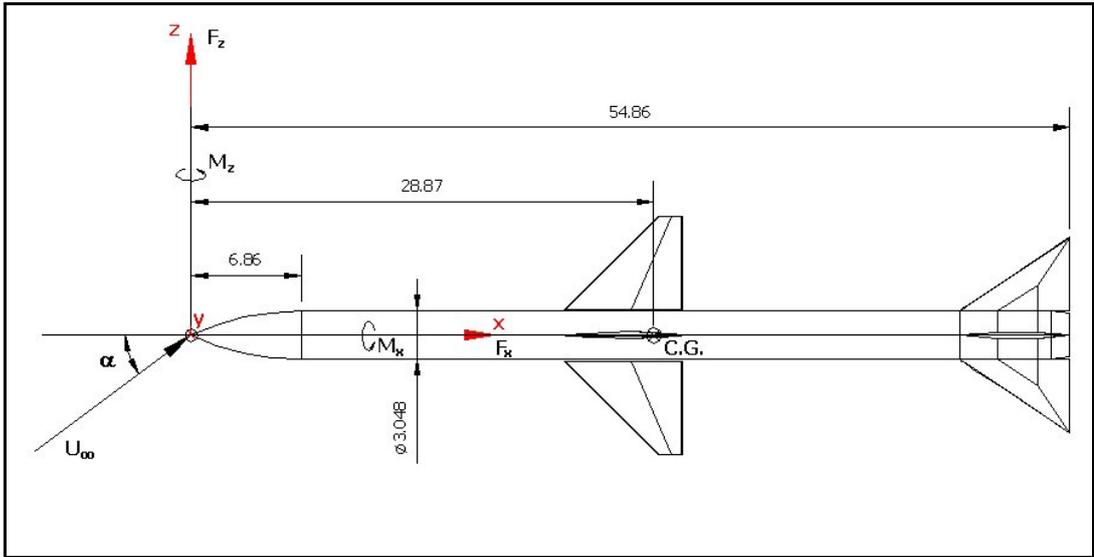


Figure 32. Detailed Geometry of Sparrow Missile and the Coordinate Axes
(all dimensions are in mm)

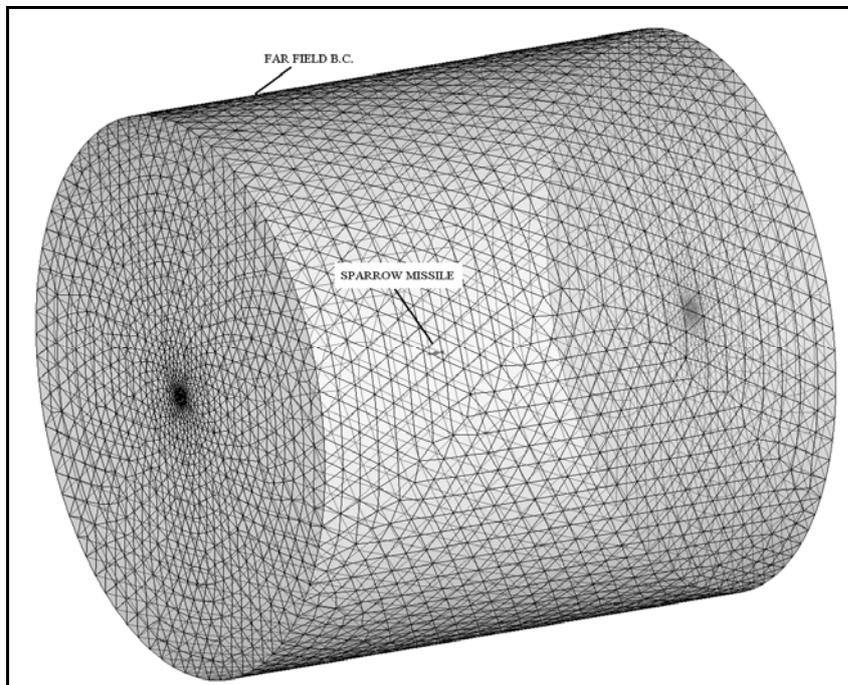


Figure 33. Unstructured Volume Mesh around Sparrow Missile

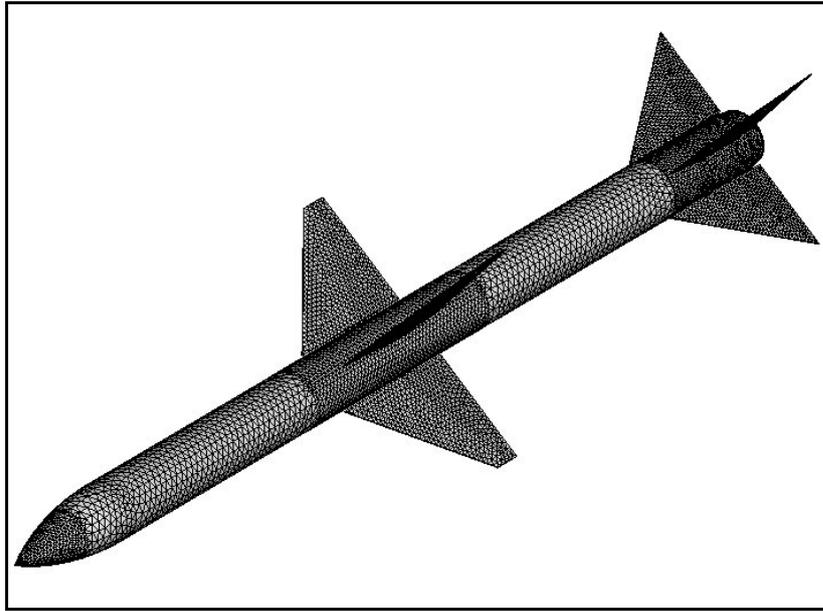


Figure 34. Unstructured Mesh Details on the Surface of Sparrow

Steady-state inviscid flow calculations for this missile are performed for high supersonic flow conditions at different angle of attack values up to 40° . Results show the accuracy of the three-dimensional inviscid solver at high angle of attack values. Solution matrix is shown in Table 4.

Table 4. Solution Matrix for Sparrow Missile

Mach # (M)	Angle of Attack (α)
1.5	$0^\circ, 5^\circ, 10^\circ, 15^\circ, 20^\circ, 30^\circ, 40^\circ$
2.35	$0^\circ, 5^\circ, 10^\circ, 15^\circ, 20^\circ, 30^\circ, 40^\circ$
3.95	$0^\circ, 5^\circ, 10^\circ, 15^\circ, 20^\circ, 30^\circ, 40^\circ$

Just like the finned projectile model that has been investigated in Section 4.1.1, during the measurements of Sparrow model, experimental data could not be obtained from the base of the model because of the physical limitations of the wind tunnel. In the related reference [97] it was stated that drag and axial force coefficients have been adjusted to correspond to free-stream static pressure acting over the base of the model. This correction is also applied in numerical computations in order to make a meaningful comparison with experimental measurements. The base pressure correction is performed after full convergence is achieved.

In Figure 35, Figure 36 and Figure 37 the variation of axial force coefficient, normal force coefficient and pitching moment coefficient with angle of attack at different Mach numbers in comparison with experimental results are presented, respectively.

It can be concluded from the results that the variation of aerodynamic coefficients with respect to angle of attack seems to be in qualitative correspondence with experimental measurements. Especially normal force and pitching moment coefficients show a full agreement with experimental data for low angle of attacks ($\alpha < 10^\circ$). This success is expected from the solver for high supersonic flow conditions because it is well known that especially for inviscid calculations for flows at low Mach number flows and transition regime create problem.

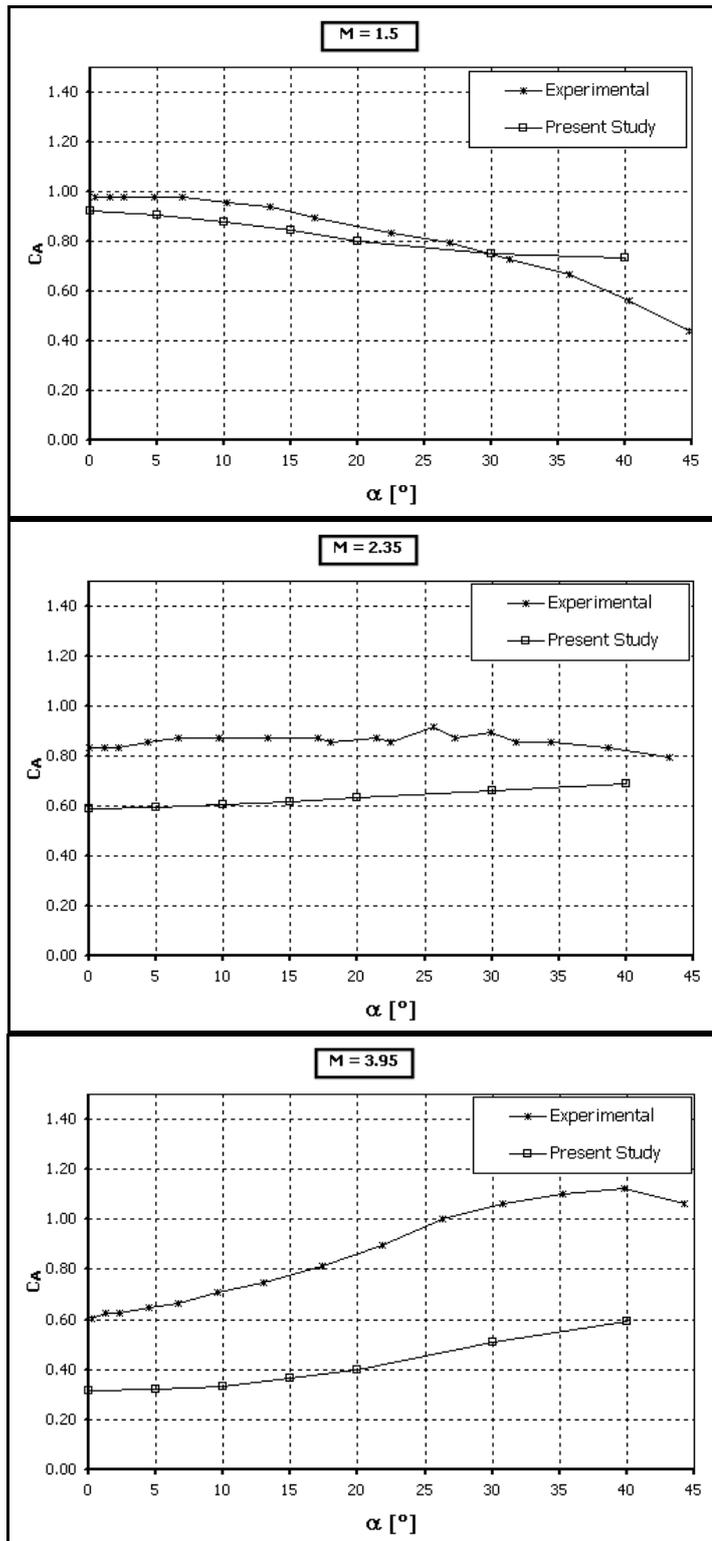


Figure 35. Sparrow-Variation of Axial Force Coefficient w.r.t. Angle of Attack

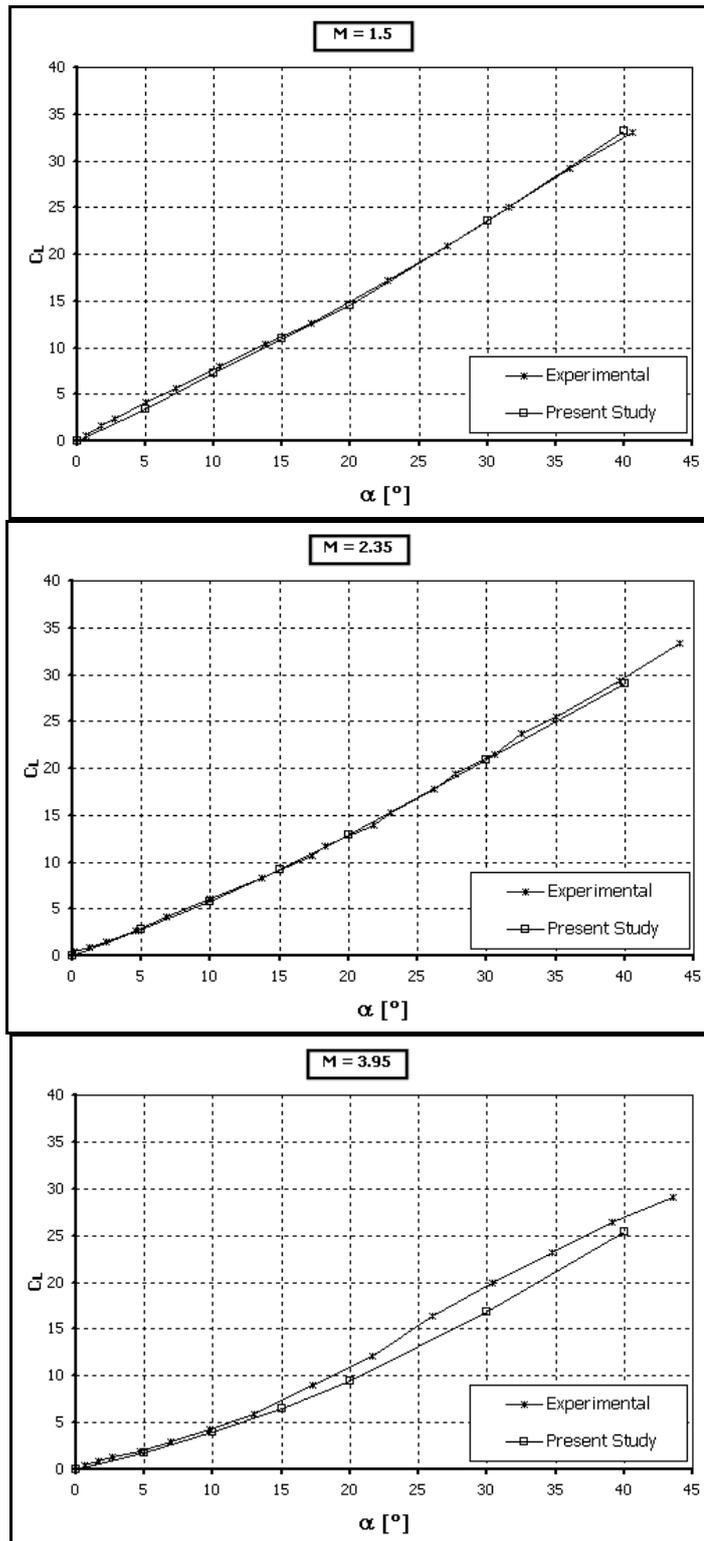


Figure 36. Sparrow-Variation of Normal Force Coefficient w.r.t. Angle of Attack

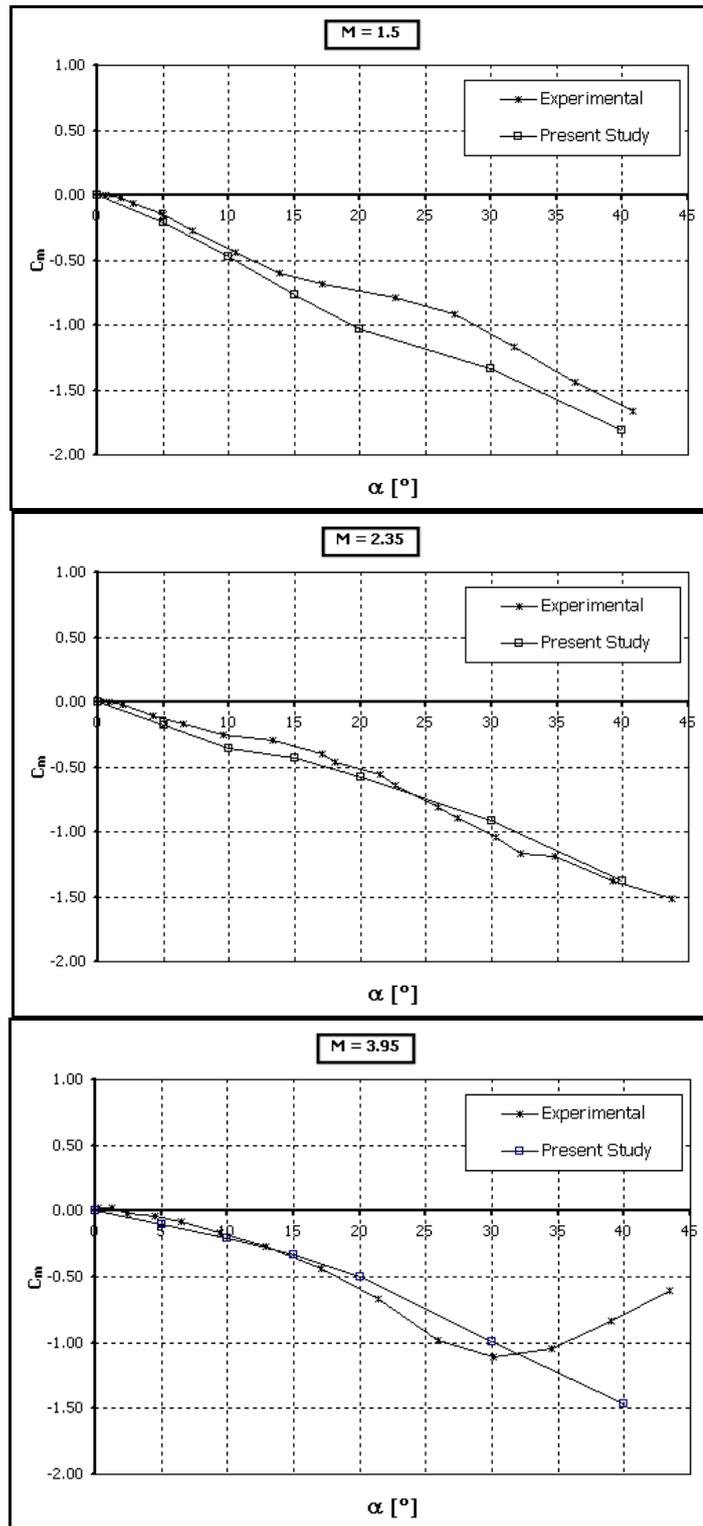


Figure 37. Sparrow-Variation of Pitching Moment Coefficient w.r.t. Angle of Attack

4.2. 3-D Validation Study for the Euler-ALE Solver

It was stated in Section 4.1.1 that the finned projectile has experimental dynamic stability derivative measurements [96]. For the validation of Arbitrary Lagrangian Eulerian formulation and the grid rotation algorithms in 3-D, inviscid flow over projectile rotating around its axis for a specific roll rate at which some experimental data is available, is investigated.

Results of the steady-state inviscid flow calculations were presented in Section 4.1.1 at different Mach numbers for a range of angle of attack values. Unsteady flow calculations for the spinning projectile are performed at a Mach number of 0.7 with zero angle of attack. Reduced frequency of 0.0093 is taken which corresponds to approximately rotational speed of 340 rpm. Equation of the reduced frequency is given as:

$$k = \frac{\alpha L_{\text{ref}}}{2V_{\infty}} \quad (4.1)$$

where α is the rotation rate in radians per second, V_{∞} is the free stream velocity and L_{ref} is the reference length which is defined as the diameter of the projectile in this case.

Roll damping coefficient measurements are available for different angle of attack values but it can be seen from the experimental results that roll damping coefficient is almost independent of angle of attack especially for low values, ($\alpha < 5^\circ$). In order to see the accuracy of the ALE formulation for inviscid flow calculations, which will be a base for 3-D turbulent flow conditions, computations are performed at zero angle of attack value. Roll damping coefficient is calculated by the linear approximation:

$$C_{lp} = \frac{\partial C_l}{\partial p} = \frac{C_l|_{k=k} - C_l|_{k=0}}{k} \quad (4.2)$$

Steady state calculation results give the roll producing moment coefficient C_{l_0} to be zero because the model is symmetric and there exists no angle of attack. The projectile is rotated with given roll rate starting from the steady-state situation. In Figure 38, the steady-state non-dimensional pressure contours on the surface of the projectile for zero angle of attack at Mach number 0.7 is shown.

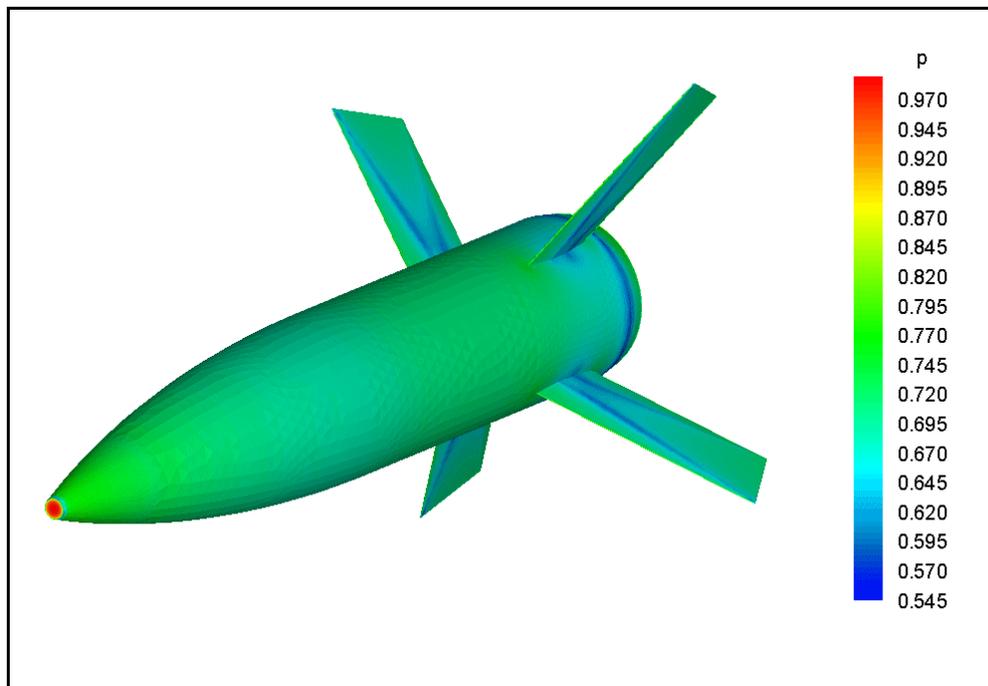


Figure 38. Steady-State Non-Dimensional Pressure Contours on the Projectile

In Figure 39, the variation of roll moment coefficient with iteration number and angle of rotation for $M=0.7$ is presented. It is observed from this figure that after steady state is achieved after 5,000 iteration steps, with the start of rotation,

roll moment coefficient has a sudden jump which is afterwards monotonically damped to a constant value of $C_l = -0.18$ after approximately 20° of rotation.

Experimental measurements gave a roll damping derivative of $C_{l_p} = -24$ where numerical computations gave approximately a roll damping derivative of $C_{l_p} = -20$. There exists approximately 15 % difference between computed results and experimental data. This result seems to be quite acceptable when it is thought that the flow has been taken as inviscid and a linear trapezoidal approximation has been assumed for the calculation of roll damping derivative. Computations showed that a rotation of approximately 15 to 20 degrees is enough for the evaluation of the roll damping coefficient for inviscid flow conditions.

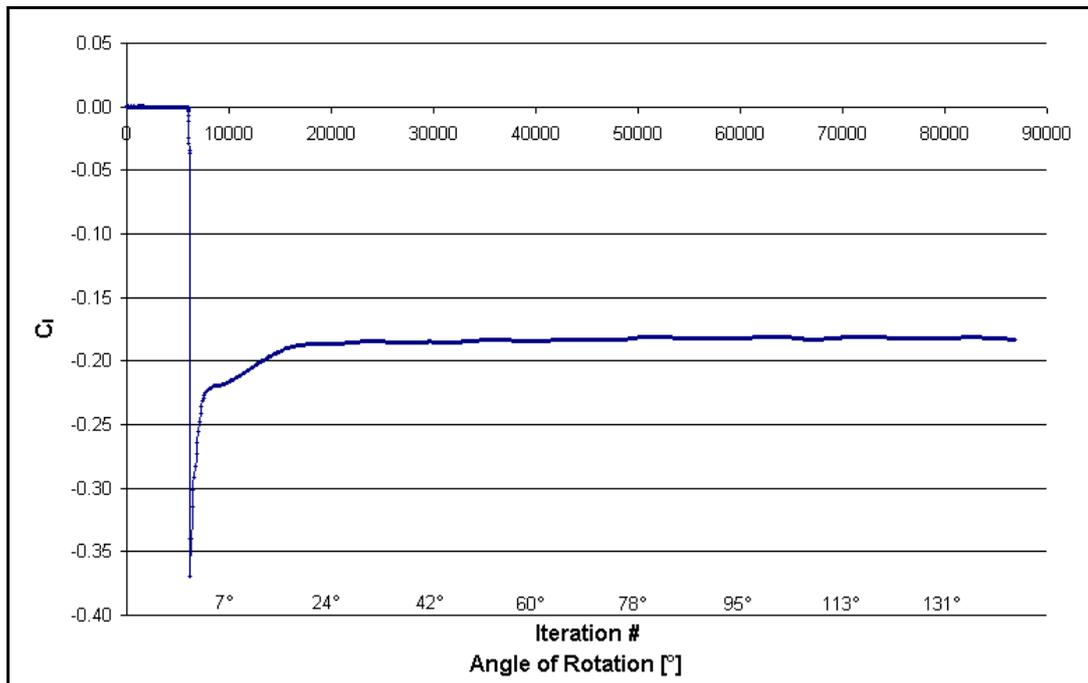


Figure 39. Variation of Roll Moment Coefficient w.r.t. Rotation of the Projectile

In Figure 40, Figure 41, Figure 42, Figure 43 and Figure 44 the non-dimensional pressure distribution near the fins after 3.5° , 10.6° , 21.3° , 39° and 60.3° rotation are given, respectively. It can be observed from these figures that sudden rotation of the projectile causes pressure development on the sides of the fins that are facing the rotation. Opposite sides of the fins face low pressure as expected. After certain rotation, approximately 20° , pressure contours remain the same. This situation was also observed in Figure 39 where roll moment coefficient, which was caused mainly by the motion of the fins, became constant after approximately 20° of rotation. Note that the cross-sections of the fins are not the same in these figures. This is because of the location of the fins on the projectile. Two pairs of fins are located at different axial locations as shown in Figure 26.

It is thought that viscous turbulent flow calculations will give better results but high resolution viscous computational mesh is very difficult to obtain for such finned projectiles. The grid generation tools (ANSYS, FLUENT-GAMBIT, CFD-GEOM) are capable of generating viscous boundary layer meshes with hexahedral elements. But generating unstructured boundary layer mesh composed of full tetrahedral elements seems to be impossible with available tools. Generating a hexahedral boundary layer mesh near the wall and filling the rest of the domain with tetrahedral elements and then dividing the hexahedral elements into tetrahedral ones is a way that has been tried to generate a viscous unstructured mesh. This method has failed because of the disordered structure of the hexahedral elements which disabled the division of these elements into tetrahedral elements without the generation of crossed faces.

Because of the reasons explained above another validation case is studied and presented in Chapter 5 at which turbulent flow over spinning projectile without fins is investigated. Viscous mesh generation for this geometry will be explained in the related chapter.

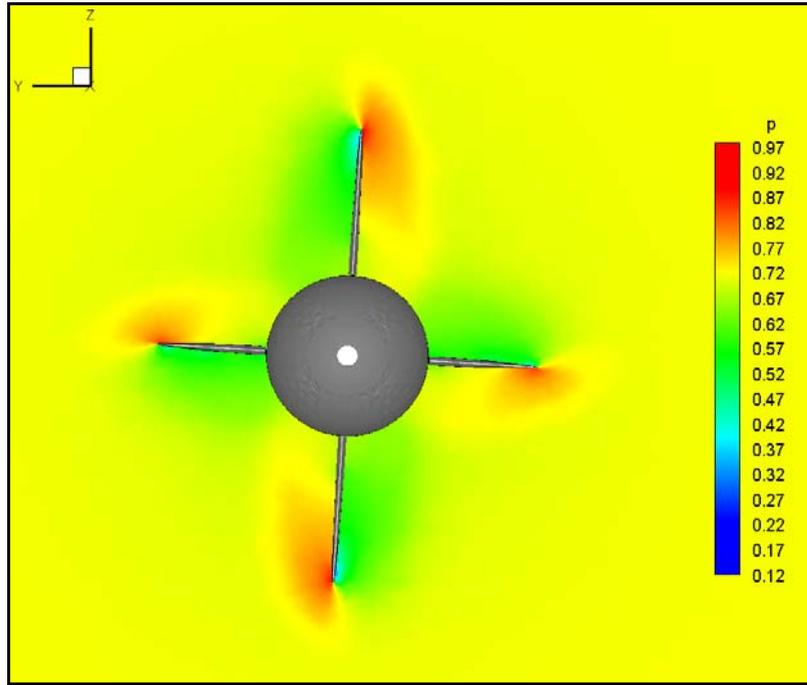


Figure 40. Non-Dimensional Pressure Distribution Near Fins ($\theta = 3.5^\circ$)

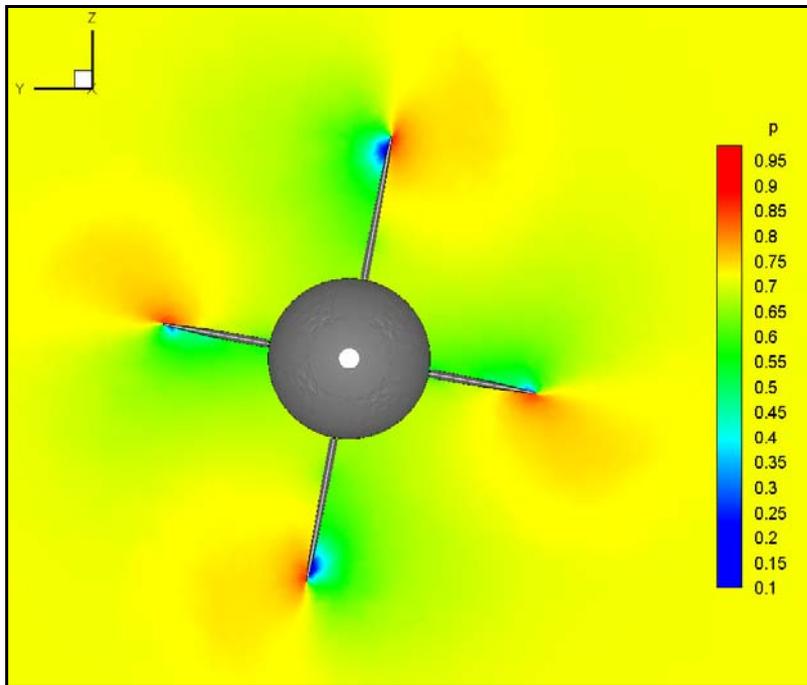


Figure 41. Non-Dimensional Pressure Distribution Near Fins ($\theta = 10.6^\circ$)

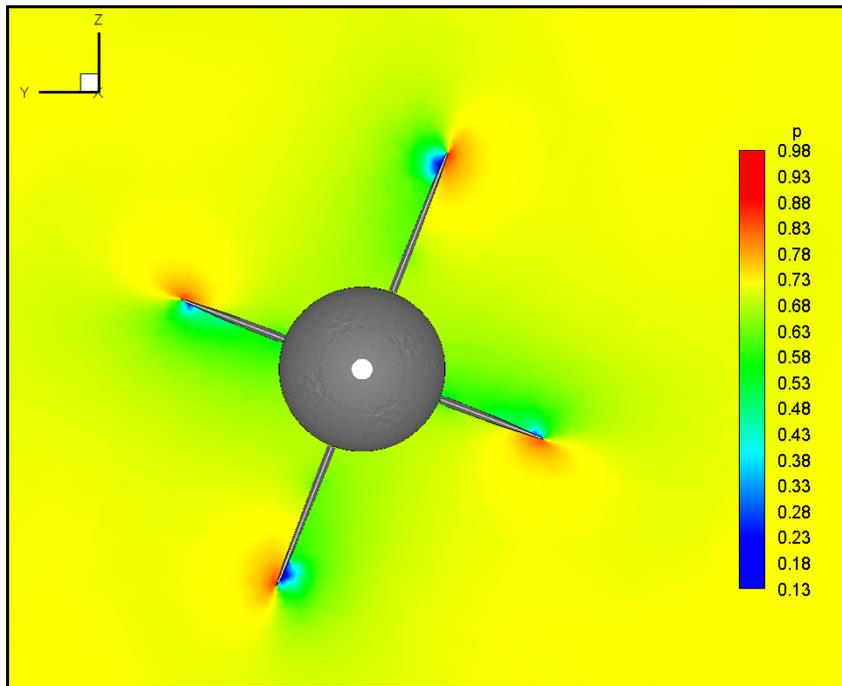


Figure 42. Non-Dimensional Pressure Distribution Near Fins ($\theta = 21.3^\circ$)

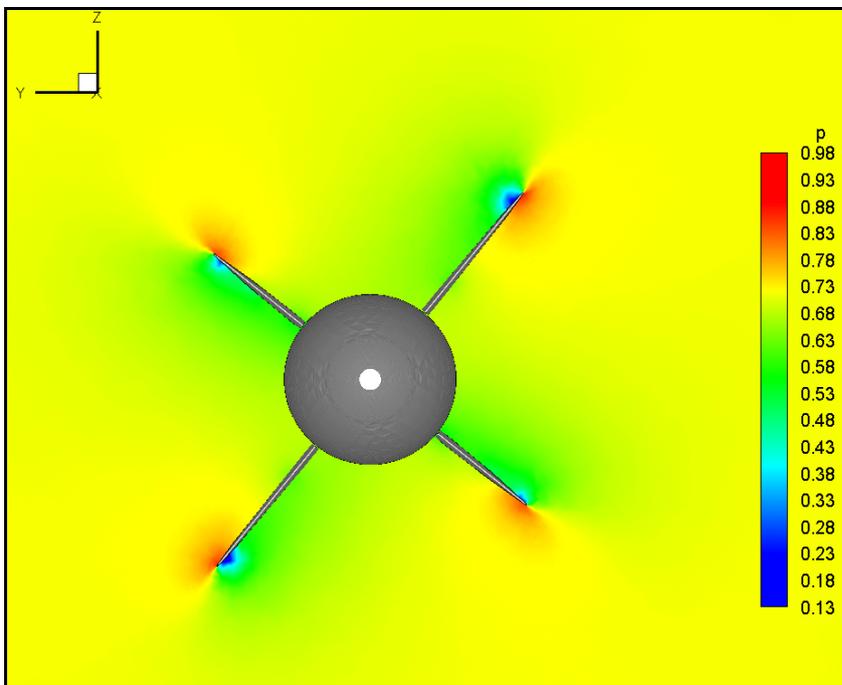


Figure 43. Non-Dimensional Pressure Distribution Near Fins ($\theta = 39^\circ$)

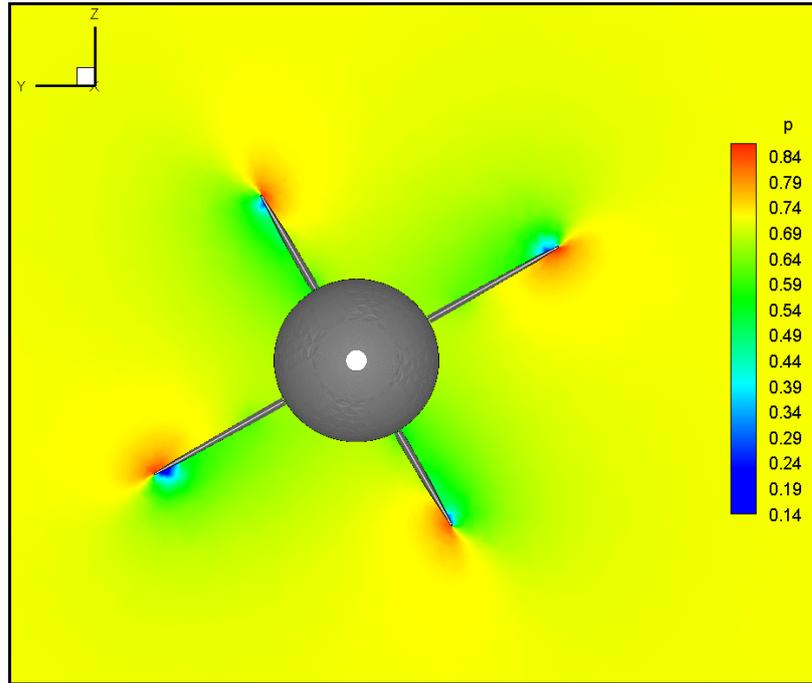


Figure 44. Non-Dimensional Pressure Distribution Near Fins ($\theta = 60.3^\circ$)

4.3. 2-D Validation Studies for the Navier-Stokes / ALE Solver

In order to validate the viscous terms added to the 2-D Euler solver, first a flat plate problem is solved in which the dominant forces are due to viscous forces. Then laminar flow over NLR 7301 airfoil and NACA0012 airfoil are investigated to validate the performance of the 2-D Navier-Stokes solver on airfoil like curved surfaces where effective pressure gradients occur.

For the validation of the ALE formulation implemented in the 2-D Navier-Stokes solver, flow over NACA 0012 airfoil which is rapidly pitched to a certain incidence angle is investigated.

Implementation of Spalart-Allmaras turbulence model to the 2-D Navier-Stokes solver is validated by solving turbulent flow over a flat plate and over NACA0012 airfoil.

4.3.1. Laminar Flow over 2-D Flat Plate

The accuracy of the viscous flux computation algorithms added to 2-D Euler solver is validated by examining the ability of the methods to reproduce the well known boundary layer solution over a thermally insulated flat plate. There exists analytical solution for flow over flat plate with which the computational results can be compared. For this purpose, laminar flow over a flat plate at Reynolds number 35,000 and Mach number 0.3 is investigated, which was also handled by Haliloğlu [98].

An unstructured mesh which has a grid clustering in normal direction near the wall is generated by a commercial grid generation program, CFD-GEOM. CFD-GEOM is capable of generating the unstructured mesh from a structured one in two dimensions. This capability of this mesh generator is used in order to control the size of the meshes normal the wall. By this way, the minimum Δy normal to the wall is defined to be 0.00025 times the plate length and this resulted in more than 10 grid nodes inside the boundary layer. There exist 129 grid points on the wall and 80 grid points up to the far field. The upstream boundary is located two plate lengths ahead of the leading edge, and the upper far-field boundary is located at a distance of three plate lengths. The mesh points are clustering in the streamwise direction near the leading edge of the plate in order to better resolve the stagnation point flow in this region. The mesh generated is composed of 26,386 triangular control cells and 13,440 nodes. The mesh and its details at the leading and trailing edges can be seen in Figure 45 and Figure 46, respectively.

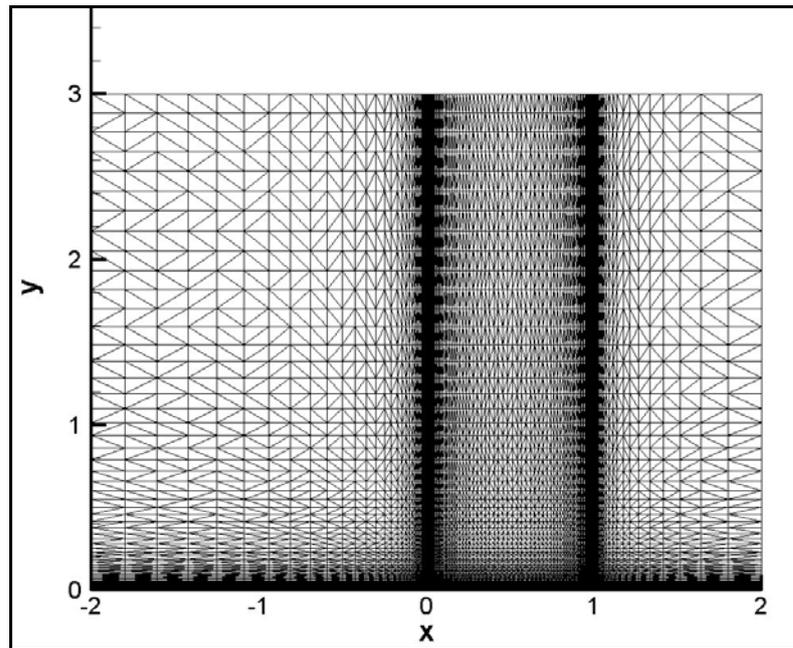


Figure 45. Unstructured Viscous Mesh for Laminar Flat Plate Problem

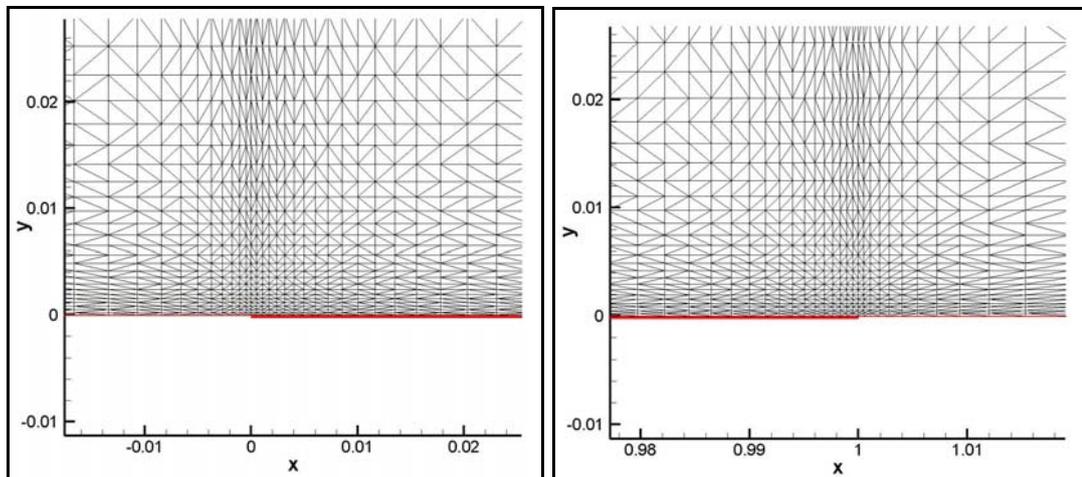


Figure 46. Mesh Details at the Leading and Trailing Edges of Flat Plate

In Figure 47, the residual history of the solution is presented. It is seen that the residual dropped approximately 2.5 – 3 orders of magnitude and the solution reached the steady state. The computational run time for 100,000 explicit iterations came out to be approximately half an hour with the use of 20 Pentium IV processors which are working in parallel.

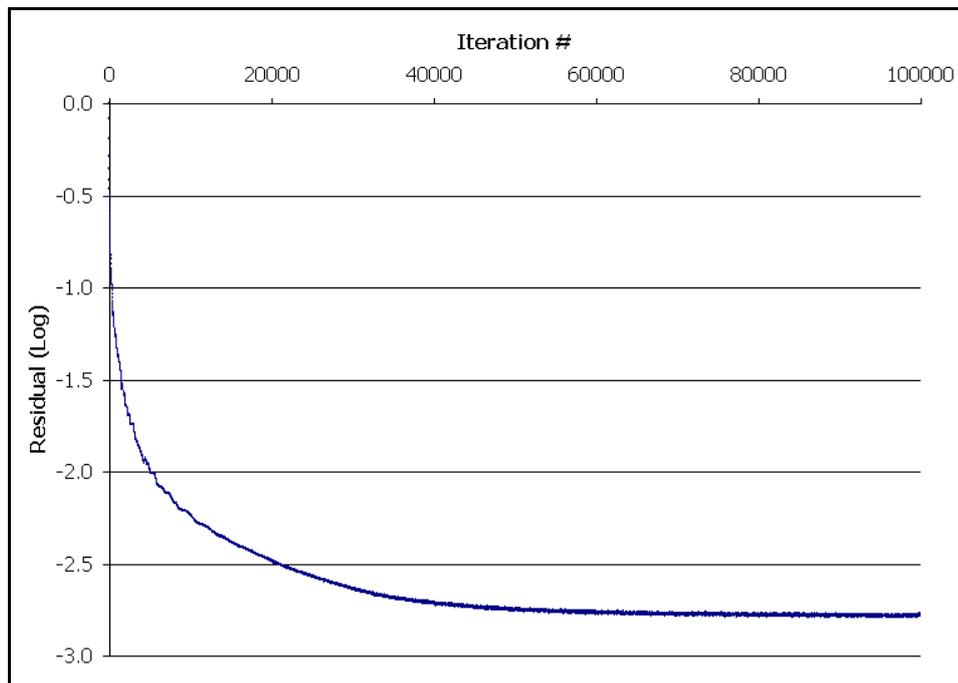


Figure 47. Residual History of Laminar Flow over a 2-D Flat Plate

The viscous flow solutions obtained are presented in Figure 48 and Figure 49. In these figures the boundary layer development at the leading edge of the flat plate can be observed. The boundary layer profile and flow streamlines seem to be as expected especially at the leading edge.

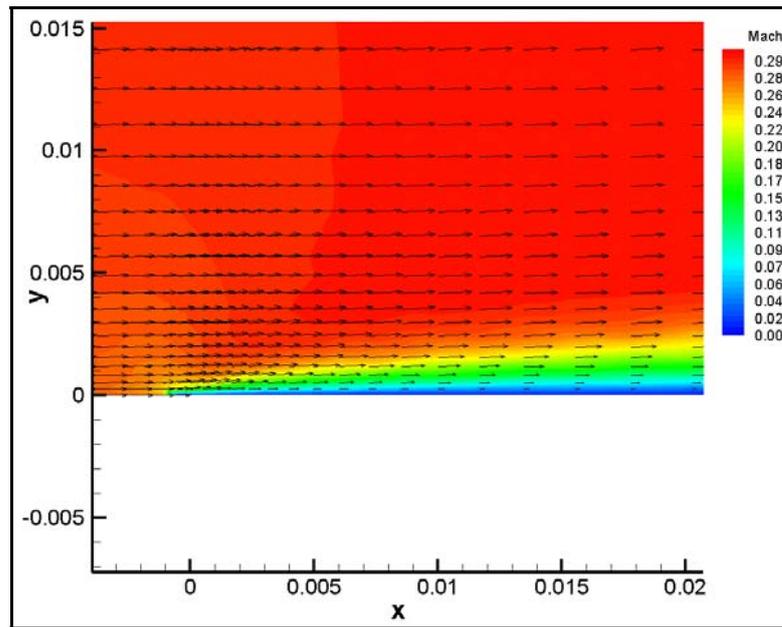


Figure 48. Laminar Boundary Layer Development at the Leading Edge of Flat Plate

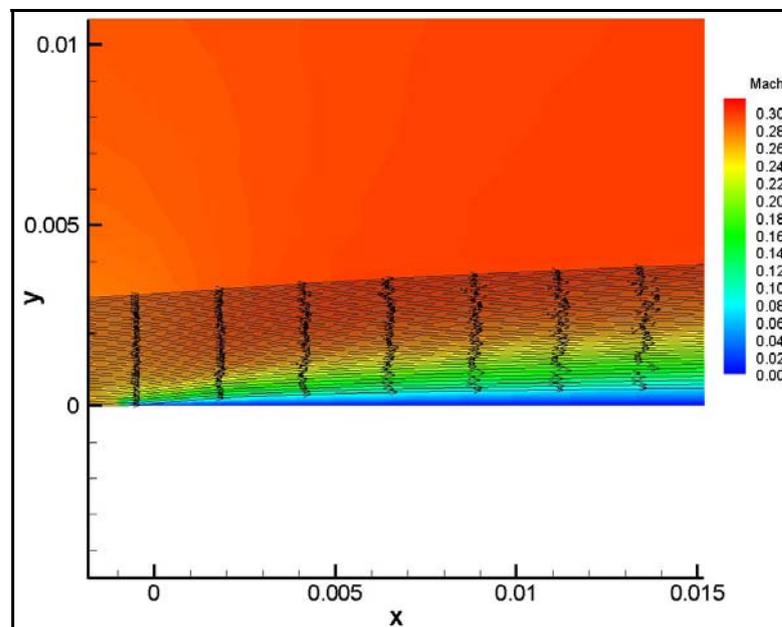


Figure 49. Laminar Boundary Layer Development at the Leading Edge of Flat Plate
(Streamlines)

In Figure 50 the continuous growth of the boundary layer till the trailing edge of the flat plate is observed. It can be seen that there is no deterioration of the boundary layer through the flat plate.

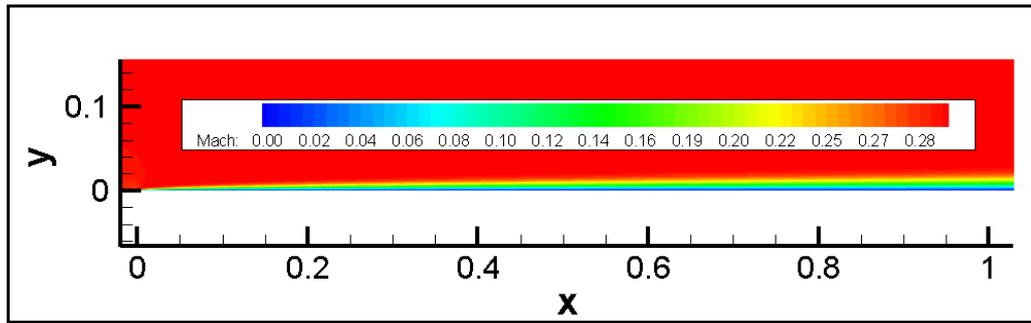


Figure 50. Overall Laminar Boundary Layer Development on the Flat Plate

Computed results are compared with the analytical solution of Blassius' [99] in order to see whether the velocity profiles are accurately estimated or not. In Figure 51, streamwise velocity profiles are presented respectively at 10%, 25%, 50%, 75% and 90% of the chord, which are compared with Blassius' analytical solution for incompressible laminar flow,

$$\text{where } Y = \frac{y}{x} \sqrt{\text{Re}_x} \quad \text{and} \quad \text{Re}_x = \frac{\mu \cdot u \cdot x}{\rho} \quad (4.3)$$

It is observed from this figure that computed streamwise velocity profiles are in excellent agreement with the analytical solution in quantitative and qualitative manner expect for the results at 10 % of flat plate. Such an overestimation near the leading edge of the flat plate is expected since it is known that as the distance form the leading edge approaches to zero wall shear stress calculated by Blassius' analytical solution becomes infinity. But it is an

experimental fact that the shear stress can not be infinite at the leading edge of the flat plate. This is due to the fact that the boundary layer theory ceases to apply near the leading edge of the plate. Therefore, Blasius' analytical solution is not valid in the vicinity of the leading edge [99].

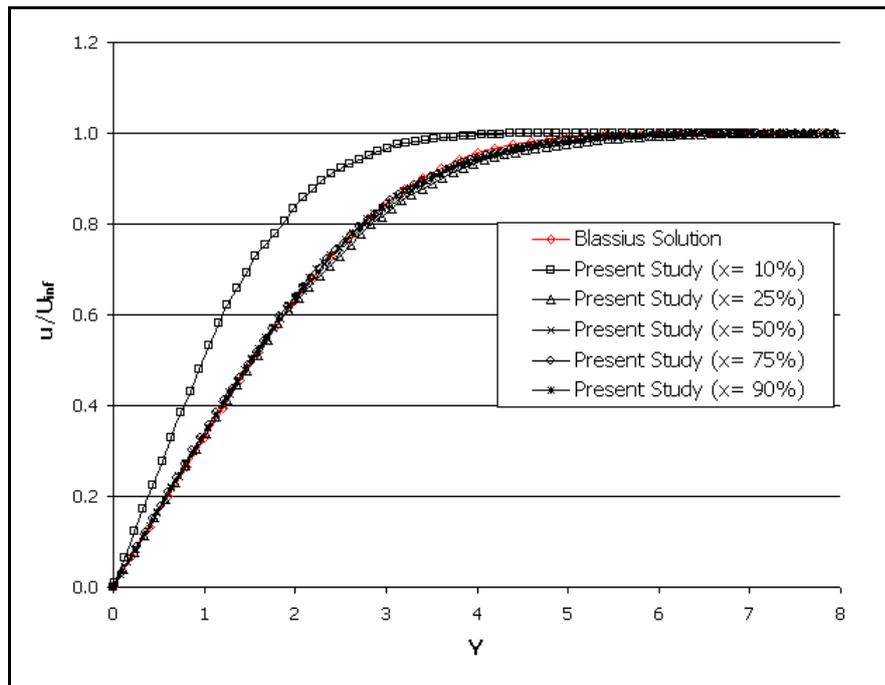


Figure 51. Axial Velocity Distribution at 10%, 25%, 50%, 75% and 90% of the Flat Plate in Comparison with Blasius Solution

The variations of the skin friction coefficient computed are compared with Blasius' analytical solution in Figure 52 and Figure 53. Skin friction coefficient is defined as:

$$C_{f,x} = \frac{\tau_w}{\frac{1}{2} \cdot \rho_\infty \cdot V_\infty^2} \quad \text{where } \tau_w = \mu \cdot \left. \frac{\partial u}{\partial y} \right|_{y=0} \quad (4.4)$$

Similarly, a very good agreement between analytical and computed results is observed in Figure 52 and Figure 53. Only at the region close to the leading edge, where effects of the stagnation point flow are still present, there exists a slight over prediction of the skin friction coefficient. It should be noted that skin friction directly involves velocity gradients, which are shown to be computed accurately with the 2-D Navier-Stokes solver developed.

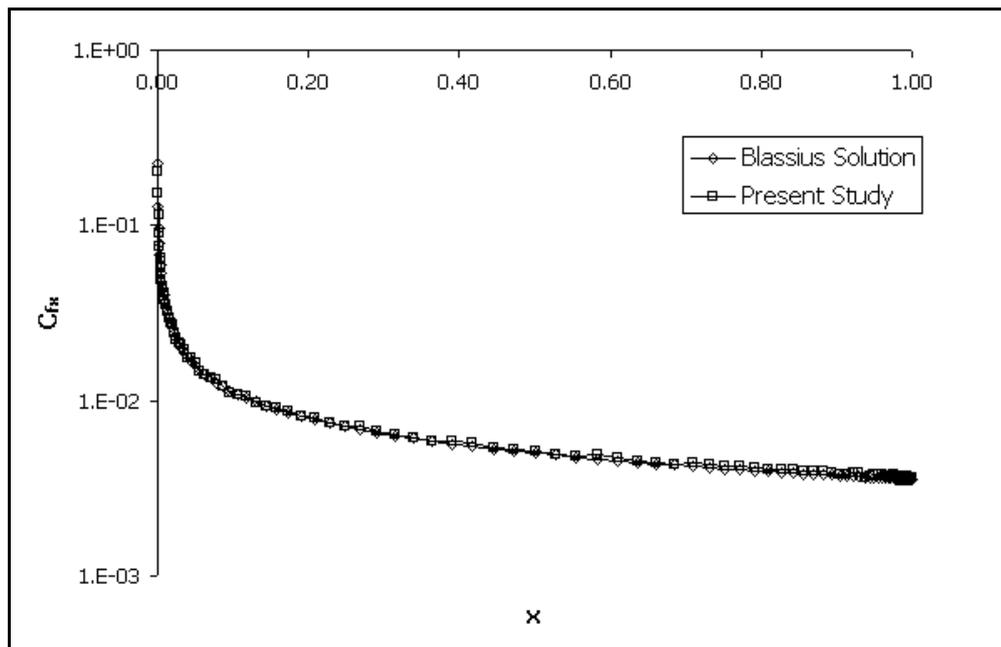


Figure 52. Skin Friction Coefficient vs. Axial Location

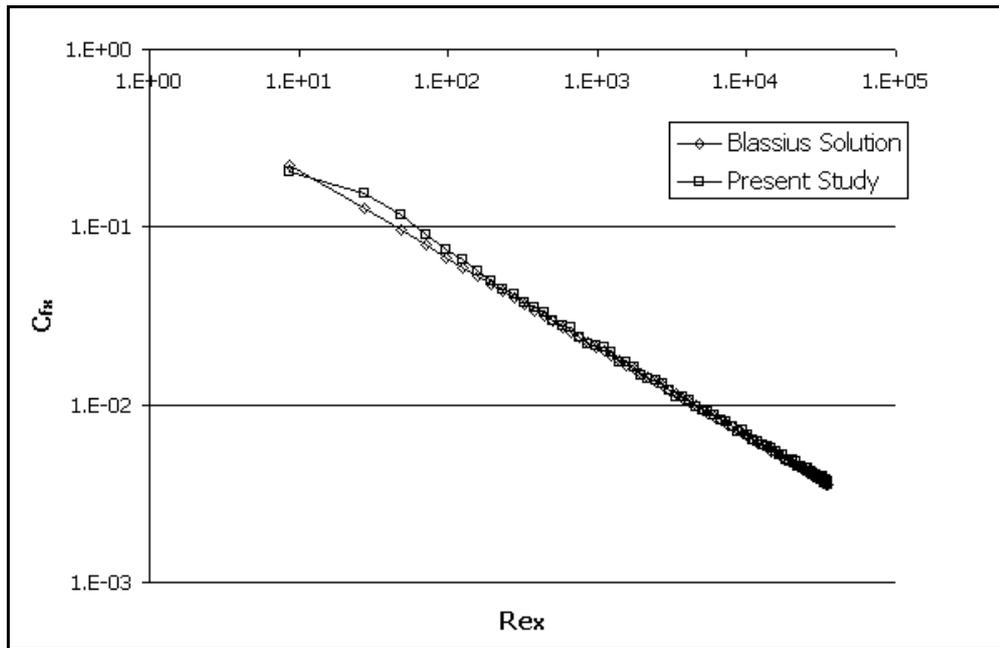


Figure 53. Skin Friction Coefficient vs. Reynolds Number

In general, it can be concluded from this validation study that except close to the leading edge, excellent agreement with analytical solution is obtained. Results validate the accuracy of the viscous flux computation scheme and velocity gradient calculation algorithm that are implemented into the Euler solver in two-dimensions.

4.3.2. Laminar Flow over NLR 7301 Airfoil With Trailing Edge Flap

Test data is available for a two-dimensional flap configuration which has been so designed that no flow separation occurs, apart from a small laminar separation bubble on the wing nose for zero angle of attack. Results of the solver developed is compared with the experimental data that has been obtained for a

Reynolds number of 2.51×10^6 and Mach number of 0.185 at angles of attack values 0° , 6° , 10.1° and 13.1° [100].

A hybrid like unstructured mesh is generated by CFD-GEOM. Nearby the wall, the unstructured mesh is obtained by dividing the rectangular structured mesh. Away from the wall coarser unstructured mesh is generated. The far-field boundary is located at a distance of 15 times the airfoil chord length. The mesh is composed of 78,067 triangular control cells and 39,405 nodes. The mesh and its details nearby the gap between airfoil and the flap are presented in Figure 54 and Figure 55 respectively.

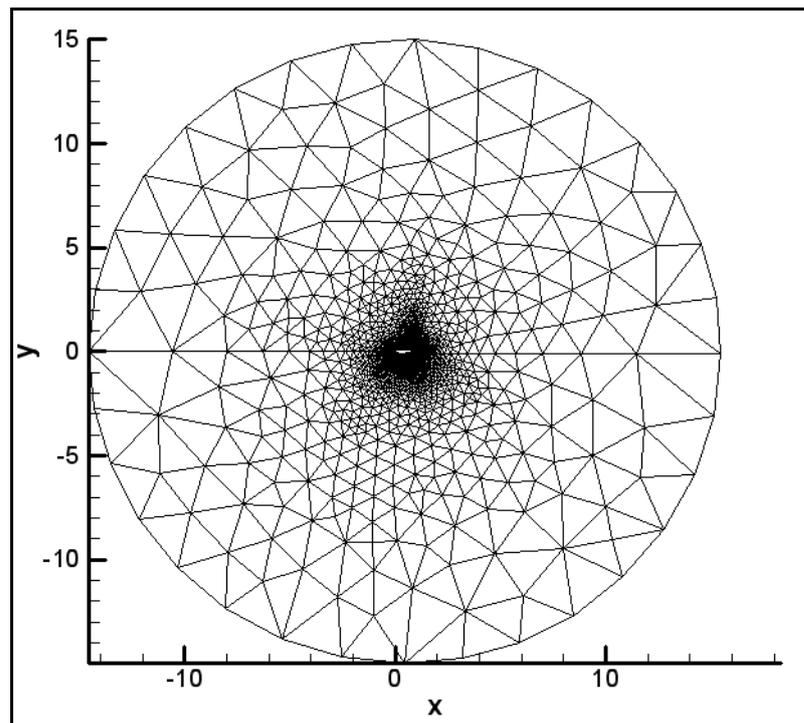


Figure 54. Unstructured Viscous Mesh for NLR Airfoil

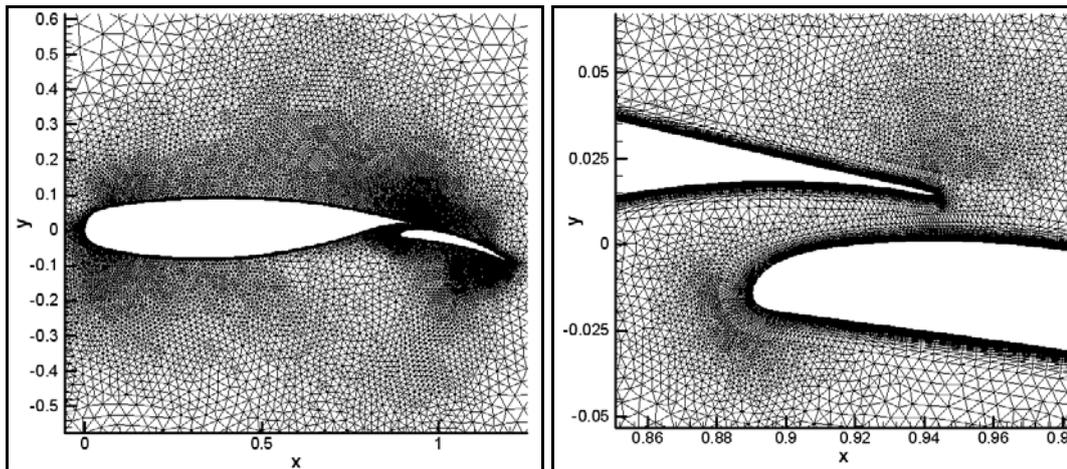


Figure 55. Mesh Details between NLR Airfoil and Trailing Edge Flap

In Figure 56, residual histories of the solutions for angle of attack values of 0° , 6° , 10.1° and 13.1° are presented. It is observed from this figure that for ($\alpha < 10.1^\circ$) approximately 3 orders of magnitude residual drop is obtained where for ($\alpha = 13.1^\circ$) only 1.5 orders of magnitude drop is obtained. It can also be seen that fluctuations in the residual is higher for ($\alpha = 13.1^\circ$) when compared with solutions at lower angle of attack values. This is possibly because of the separation of flow behind the airfoil after a certain value of angle of attack. Computational run time for this case is approximately 5 seconds for 100 iterations with the use of 20 Pentium IV processors in parallel.

In Figure 57, Figure 58, Figure 59 and Figure 60 the non-dimensional pressure contours on NLR airfoil and trailing flag with the streamlines for angles of attack 0° , 6° , 10.1° and 13.1° are shown, respectively. It is observed that up to 10.1° angle of attack there exists no separation. At 10.1° , recirculation starts to occur at the suction side of the NLR airfoil. But at an angle of attack of 13.1° two big separation bubbles one originating from the trailing flap occurred and passed away from the airfoil.

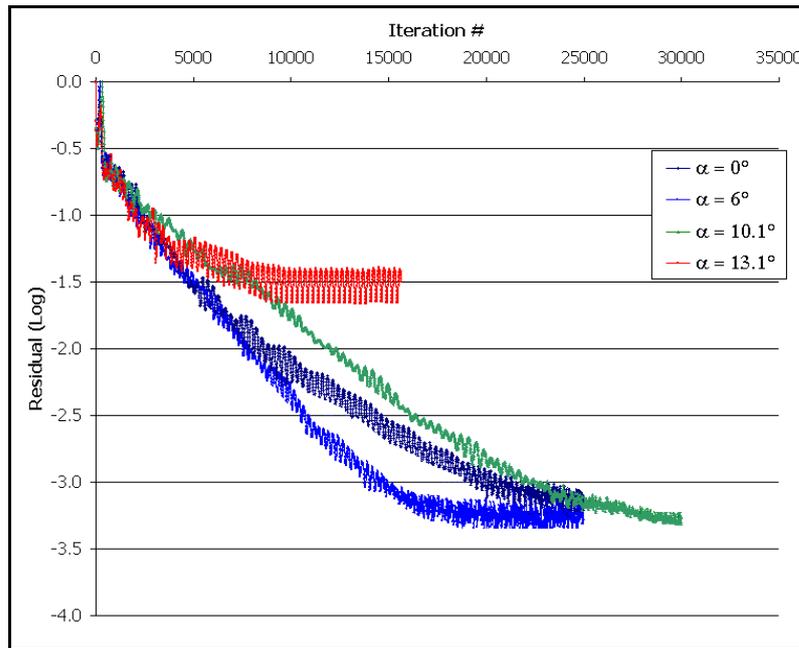


Figure 56. Residual Histories of Laminar Flow Over NLR Airfoil

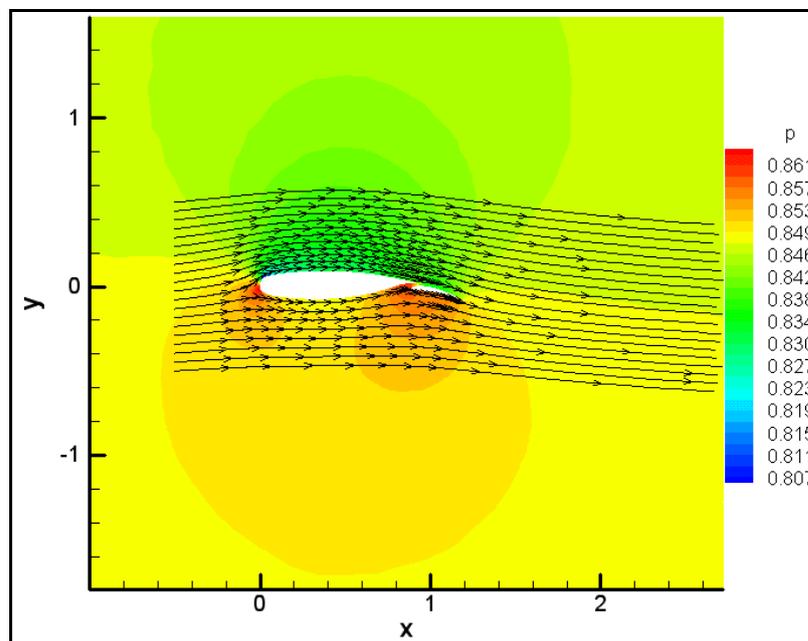


Figure 57. Non-Dimensional Pressure Contours and Streamlines on NLR Airfoil
($\alpha = 0^\circ$)

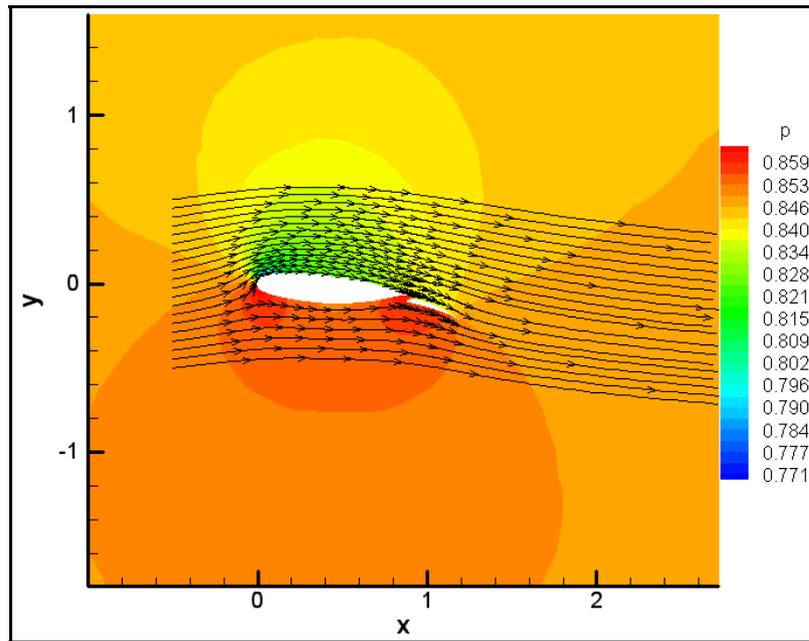


Figure 58. Non-Dimensional Pressure Contours and Streamlines on NLR Airfoil
 $(\alpha = 6^\circ)$

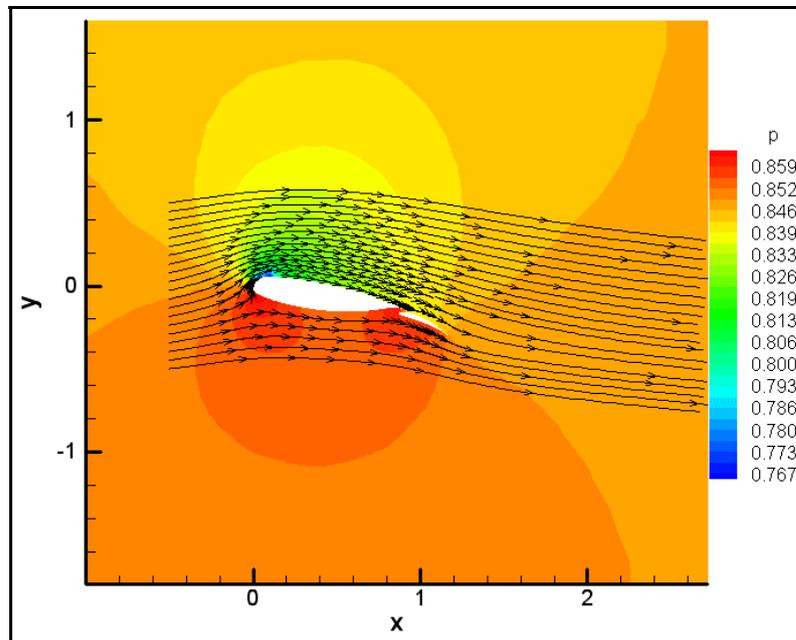


Figure 59. Non-Dimensional Pressure Contours and Streamlines on NLR Airfoil
 $(\alpha = 10.1^\circ)$

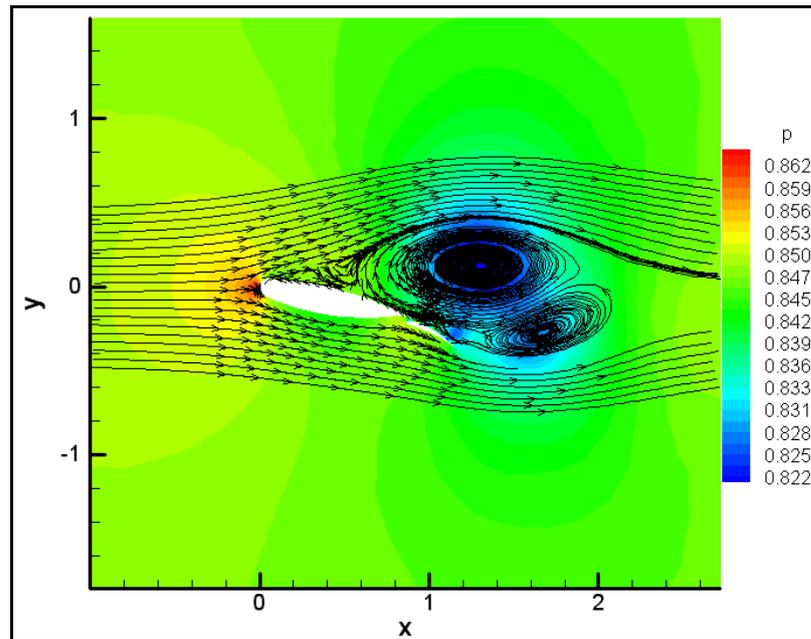


Figure 60. Non-Dimensional Pressure Contours and Streamlines on NLR Airfoil
 $(\alpha = 13.1^\circ)$

In Figure 61, Figure 62 and Figure 63 the comparison of calculated pressure coefficients with experimental results are presented for angles of attack of 6° , 10.1° and 13.1° , respectively. For each of these cases, computational results are in good agreement with experimental results. Only for $\alpha = 13.1^\circ$, at the suction side there exists a difference from experimental results because of the separation that has occurred.

The results presented here show that 2-D laminar solver gave acceptable results for airfoil type bodies also. Phenomena like vortex generation and separation of flow which are the characteristics of viscous flows are successfully simulated with the 2-D Navier-Stokes solver developed.

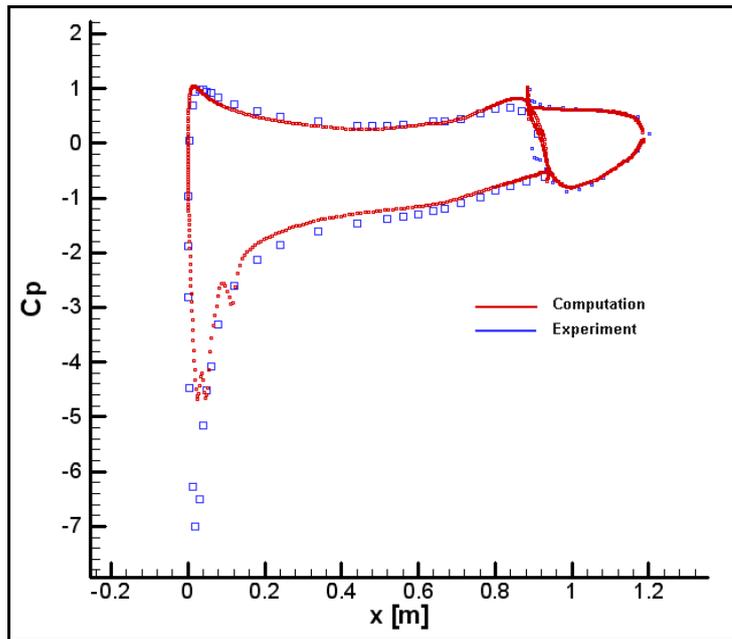


Figure 61. Pressure Coefficient Distribution on NLR Airfoil and Trailing Flap
 $(\alpha = 6^\circ)$

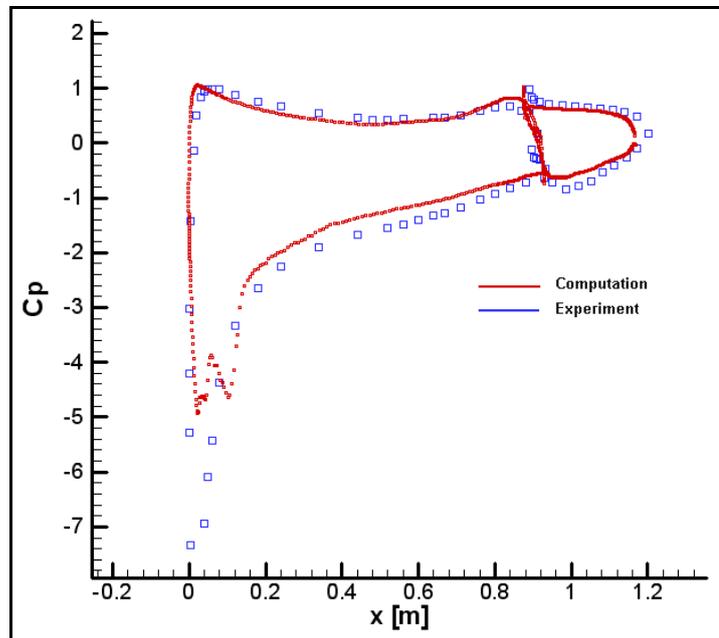


Figure 62. Pressure Coefficient Distribution on NLR Airfoil and Trailing Flap
 $(\alpha = 10.1^\circ)$

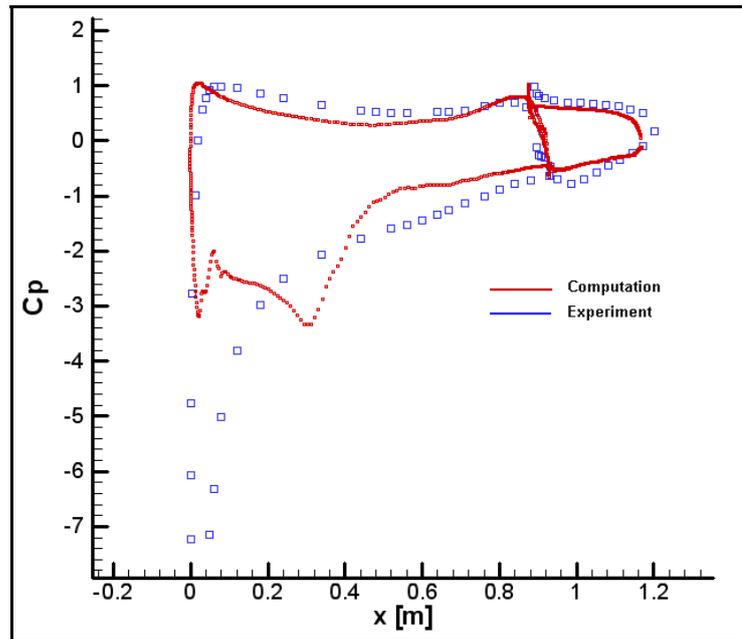


Figure 63. Pressure Coefficient Distribution on NLR Airfoil and Trailing Flap
 $(\alpha = 13.1^\circ)$

4.3.3. *Laminar Flow over NACA0012 Airfoil*

As another test case for the validation of 2-D Navier-Stokes solver, work of Mavripilis and Jameson [101] is investigated. In their work, they solved laminar flow over well known NACA0012 airfoil for the validation of their solver. Free stream conditions are taken as Mach number of 0.5 and Reynolds number of 5,000 at zero degrees of incidence.

In order to correctly simulate the viscous flow, a hybrid like unstructured mesh is generated by CFD-GEOM. Nearby the wall, the unstructured mesh is obtained from the structured mesh. Away from the wall, coarser unstructured mesh is generated. The far-field boundary is located at a distance of 15 times the airfoil chord length. The mesh is composed of 37,789 triangular control cells and 19,040

nodes. The mesh and its details nearby the wall, at the leading edge and at the trailing edge are presented in Figure 64, Figure 65 and Figure 66, respectively.

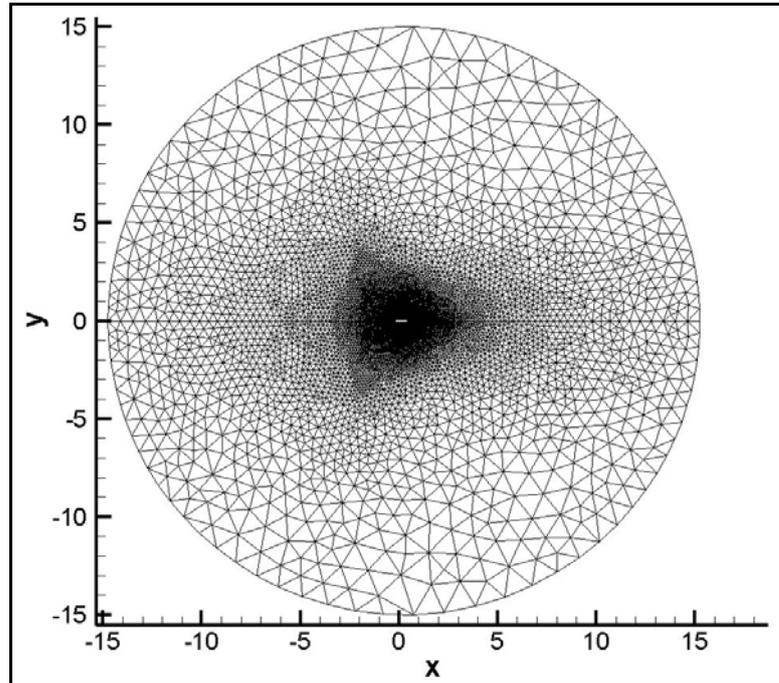


Figure 64. Unstructured Viscous Mesh for Laminar Flow over NACA0012 Airfoil

In Figure 67, the residual history of the solution is presented. It is seen that the residual drop of approximately 3.5 – 4 orders of magnitude is achieved in 50,000 time steps. Run time for this problem came out to be approximately 25 minutes with the use of 20 Pentium IV processors in parallel.

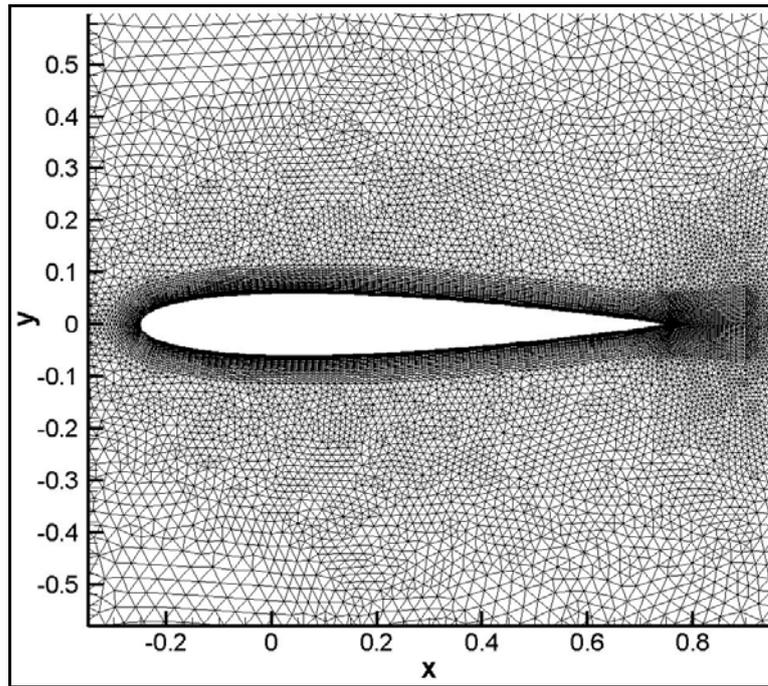


Figure 65. Viscous Mesh Detail nearby NACA0012 Airfoil

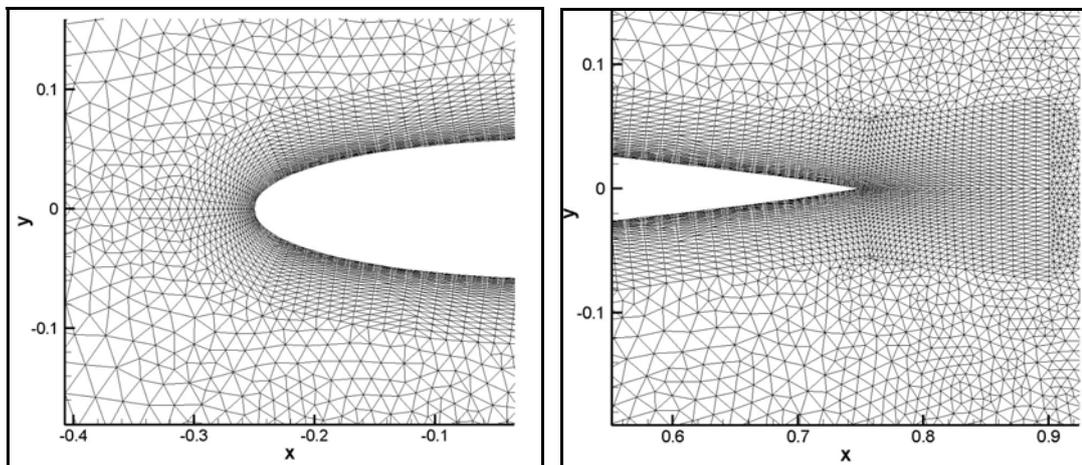


Figure 66. Mesh Details at the Leading and Trailing Edges of NACA0012 Airfoil

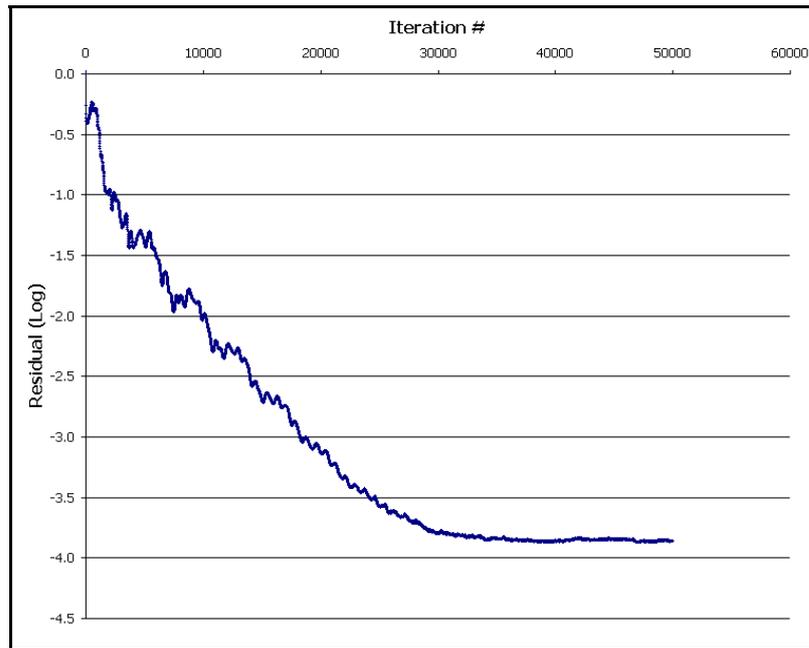


Figure 67. Residual History for Laminar Flow over NACA0012 Airfoil

Laminar boundary layer development on both surfaces of the airfoil is shown in Figure 68. In Figure 69 Mach number contours obtained by Mavripilis and Jameson [101] is presented. Mach number distribution obtained in this study is presented in Figure 70. When Figure 69 and Figure 70 are compared it is observed that Mach number contours came out to be very close to each other. Slight differences in the flow pattern may be explained by the usage of different spatial discretization techniques for the solvers. In both figures thin boundary layer development and wake regions are visible.

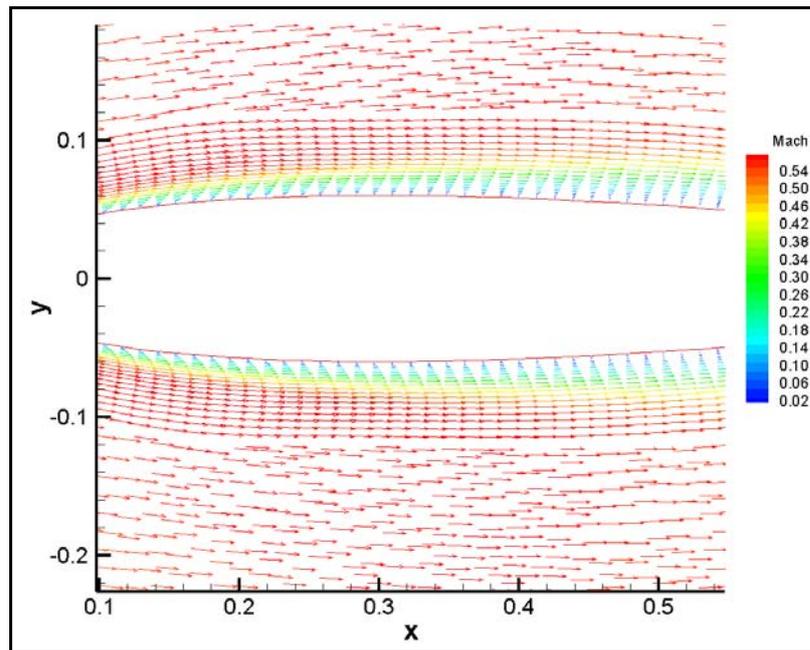


Figure 68. Laminar Boundary Layer Development on NACA0012 Airfoil, $M=0.5$

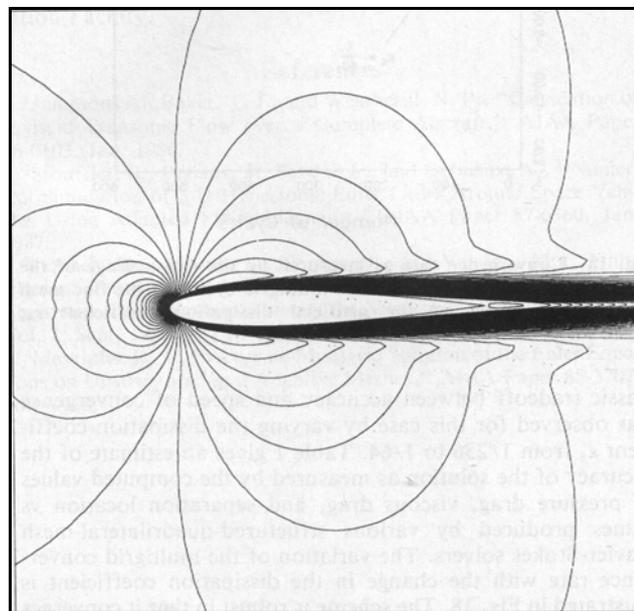


Figure 69. Mach Number Contours for Laminar Flow Over NACA0012 Airfoil,
Mavripilis and Jameson [101]

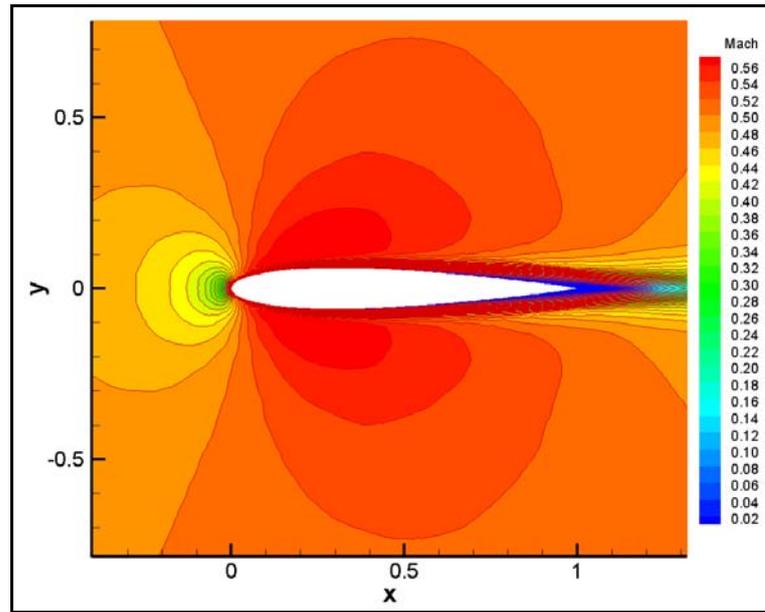


Figure 70. Mach Number Contours for Laminar Flow Over NACA0012 Airfoil,
Present Study

In Figure 71, surface pressure coefficients calculated are compared with the results of Mavripilis and Jameson [101]. It is observed from this figure that pressure coefficients at upper and lower surfaces of the airfoil came out to be very close to each other. Also, when compared to Mavripilis and Jameson's study, calculated pressure coefficients agree well with their results.

This test case study finalized the validation of 2-D Navier-Stokes solver. Results show that viscous flux calculation algorithms which are implemented to the 2-D Euler solver give reasonable and accurate results.

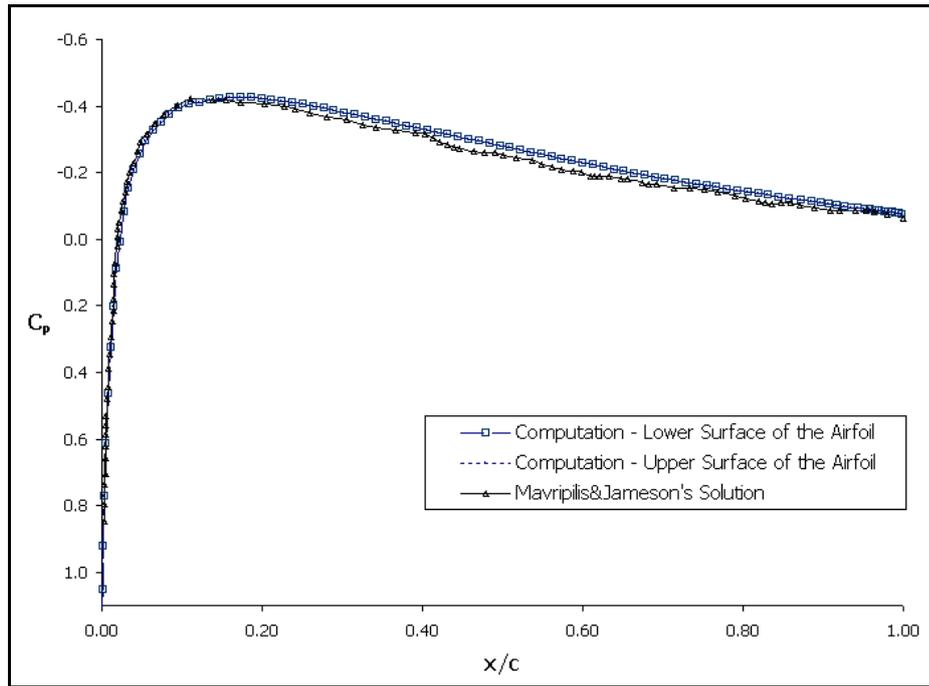


Figure 71. Pressure Coefficient Comparison for Laminar Flow over NACA0012 Airfoil, $Re = 5,000$

4.3.4. *Unsteady Aerodynamics Of Rapidly Pitched Airfoil*

In order to validate the ALE formulation and grid movement calculation algorithms added to 2-D Navier-Stokes solver, work of Wu, Wang and Tuncer is investigated [102]. They have worked on a problem of dynamic stall of an airfoil (NACA 0012) pitched rapidly at a constant rate up to a large angle of attack.

Wu, Wang and Tuncer's [102] work was for incompressible flow and the experimental data is also for incompressible flow. For that reason the problem is handled by solving compressible flow at very low Mach numbers in the present study which may represent incompressible flow (Mach number 0.2, Reynolds number 5,000). Initial calculations are performed for the evaluation of the steady-

state solution for 2-D laminar flow over the airfoil. The same computational mesh generated for the test case in Section 4.3.3 is used for these calculations (Figure 64).

In Figure 72, the residual history for the steady state solution is presented. It can be seen that a residual drop of approximately 4 orders of magnitude is obtained. Computational run time for 37,000 iterations came out to be approximately 20 minutes with 20 Pentium IV processors working in parallel.

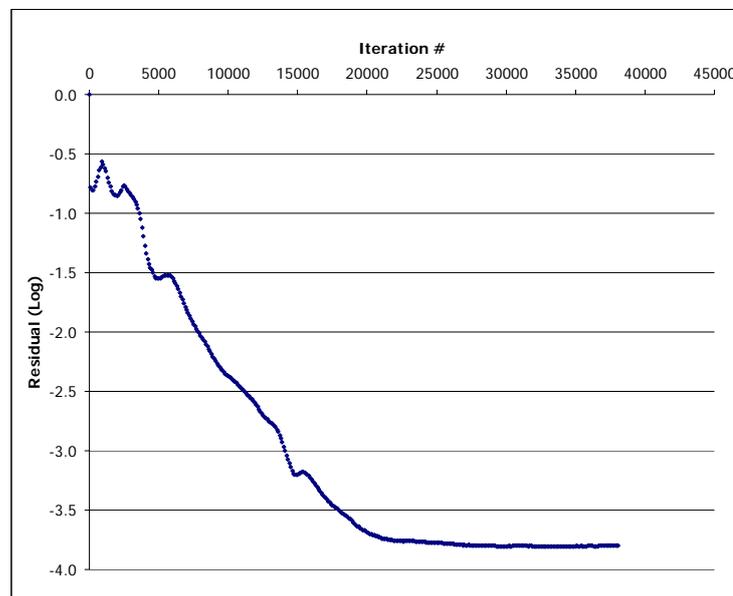


Figure 72. Residual History for Steady State NACA0012 Solution

In Figure 73, the Mach number distribution for the steady state solution of viscous flow around NACA 0012 Airfoil is presented. This solution shows all the behaviour of low Mach number viscous flow around an airfoil and the solution is quite symmetric when the upper and lower parts of the airfoil are investigated.

In Figure 74, the close up view of the solution shows that the boundary layer development on the airfoil surface is totally simulated.

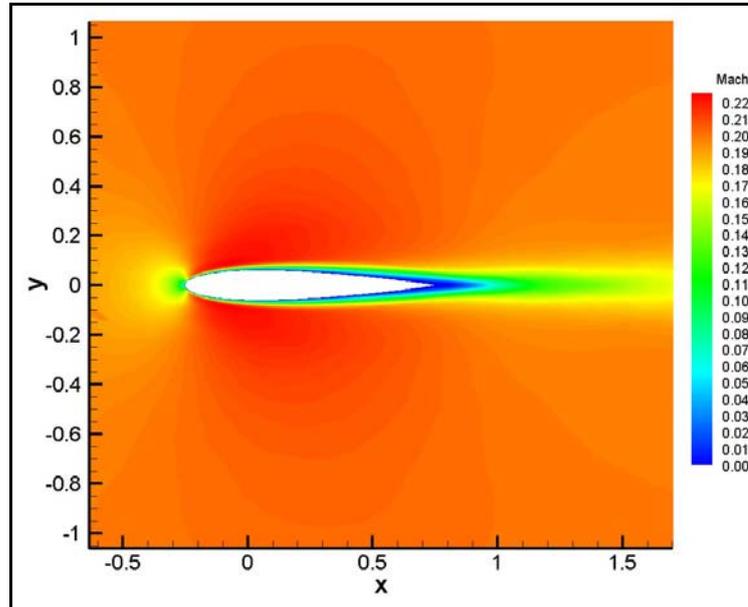


Figure 73. Steady State Mach Number Distribution Around NACA0012 Airfoil

Unsteady flow calculations for the rapidly pitching motion of the airfoil are performed at the flow conditions used for steady-state calculations ($M=0.2$, $Re=5000$). Reduced frequency value of 0.079 is investigated as it was proposed in Wu, Wang and Tuncer's [102] work. This value corresponds to approximately 300 degrees/sec pitching rate. Definition for reduced frequency was stated by Equation (4.1) where reference length value is taken as the chord length of the airfoil in this case.

After steady state solution is obtained, the airfoil is set to rotate around its center. The pitching rate α is kept constant until maximum incidence angle of

34.4° is reached. The airfoil is kept at this incidence angle for an additional period of time.

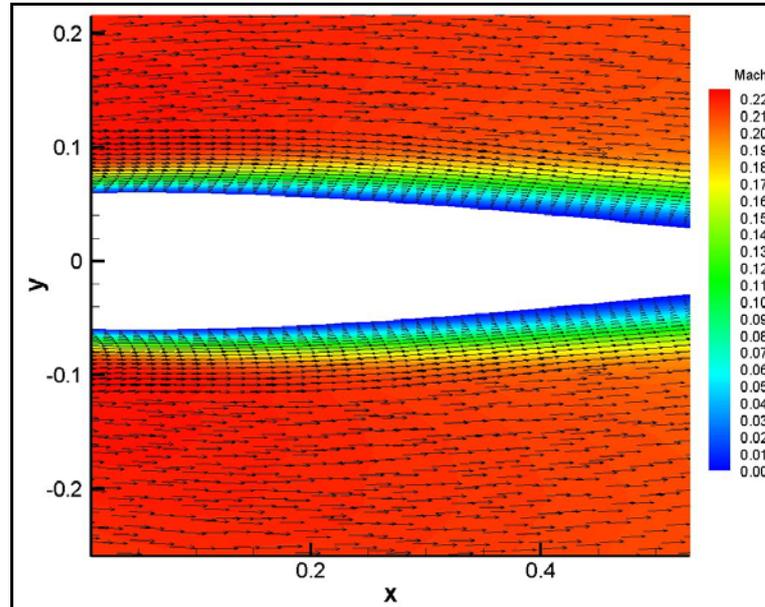


Figure 74. Laminar Boundary Layer Development on NACA0012 Airfoil, M=0.2

In Figure 75 and Figure 76 the computed unsteady lift and drag coefficients on the airfoil which are compared with the experimental results are presented respectively [102]. There seems an acceptable qualitative correspondence between computed and experimental results especially at mid-region of motion. At the beginning lift coefficient becomes suddenly large. This may be because of the large time step that has been chosen at the very beginning of the rotation. Both coefficients differ from the experimental data especially after maximum incidence angle is reached.

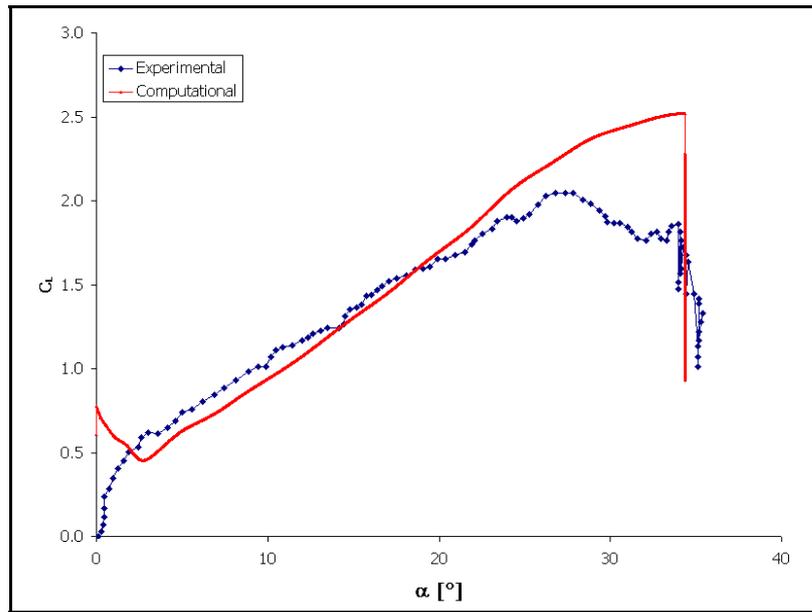


Figure 75. Lift Coefficient Variation w.r.t. Incidence Angle of Rapidly Pitched NACA0012 Airfoil

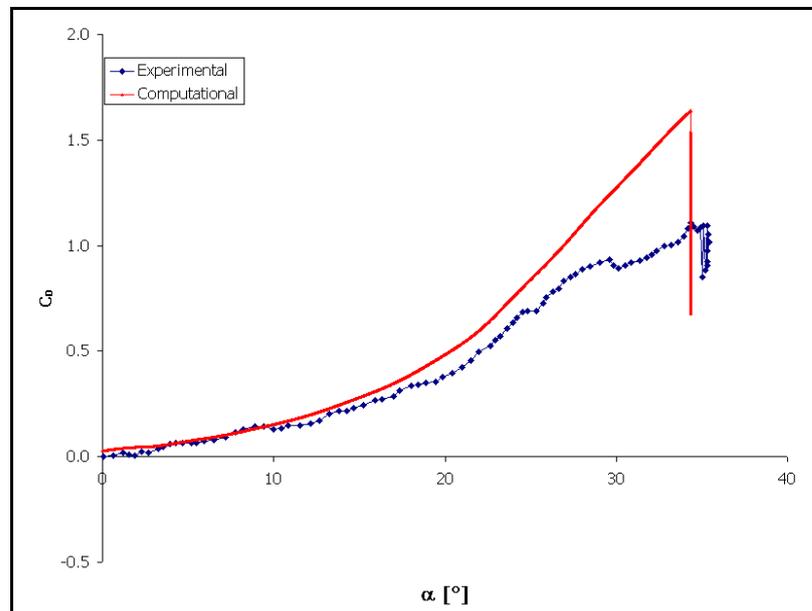


Figure 76. Drag Coefficient Variation w.r.t. Incidence Angle of Rapidly Pitched NACA0012 Airfoil

At high angle of attacks where vortex generation starts and separation occurs, turbulent flow gains importance. In this computation turbulence was not introduced. This may be the explanation of the difference at high incidence angles. It must also be noted that the experimental data is also obtained for incompressible medium. In Figure 77, Figure 78, Figure 79 and Figure 80, the streamlines computed by Wu, Wang and Tuncer [102] at several different time steps and solutions obtained in this study at the same time steps are presented. It can be said that, again, qualitatively the results are in good agreement with Wu, Wang and Tuncer's solutions. Separation formation starting at the leading edge can be observed easily. Since the Reynolds number is relatively low, formation of secondary vortices is negligible but still observable. At the trailing edge, the separation bubble formation can be observed. When the maximum rotation is reached, vortex shedding that occurs behind the airfoil can be observed in Figure 81.

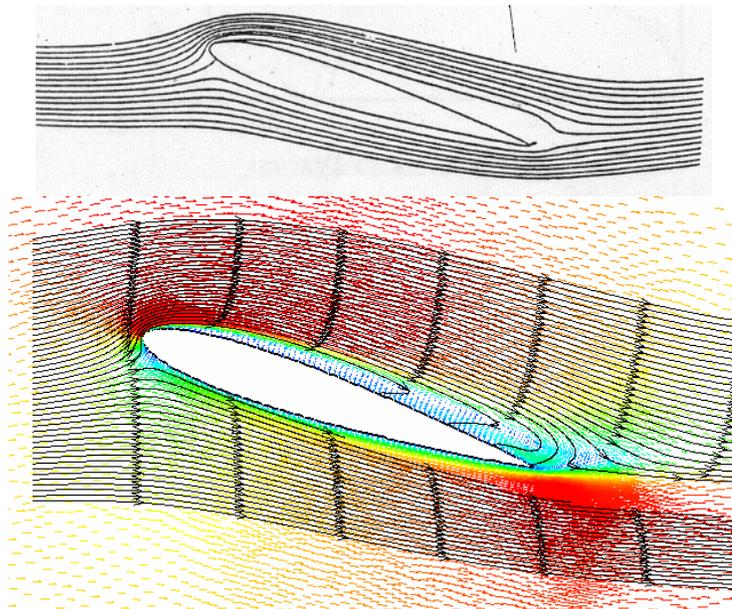


Figure 77. Streamlines of NACA0012 Airfoil at $\alpha = 17.7^\circ$

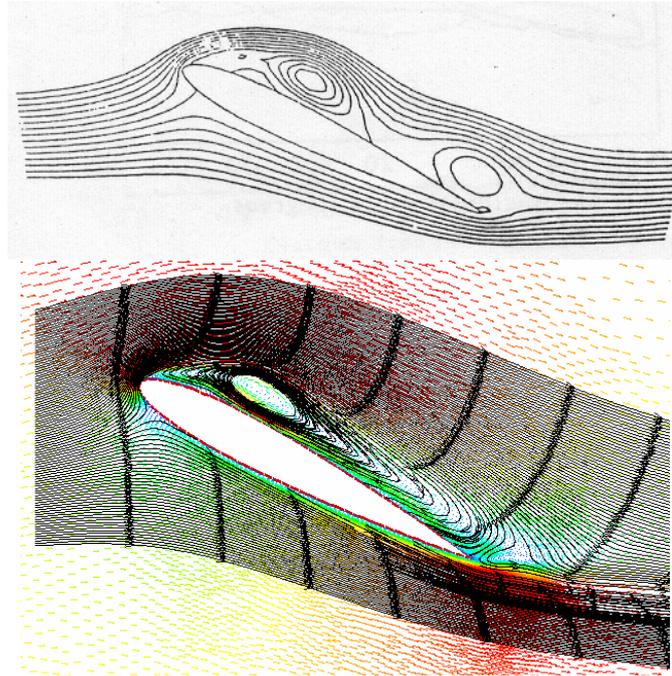


Figure 78. Streamlines of NACA0012 Airfoil at $\alpha = 26.7^\circ$

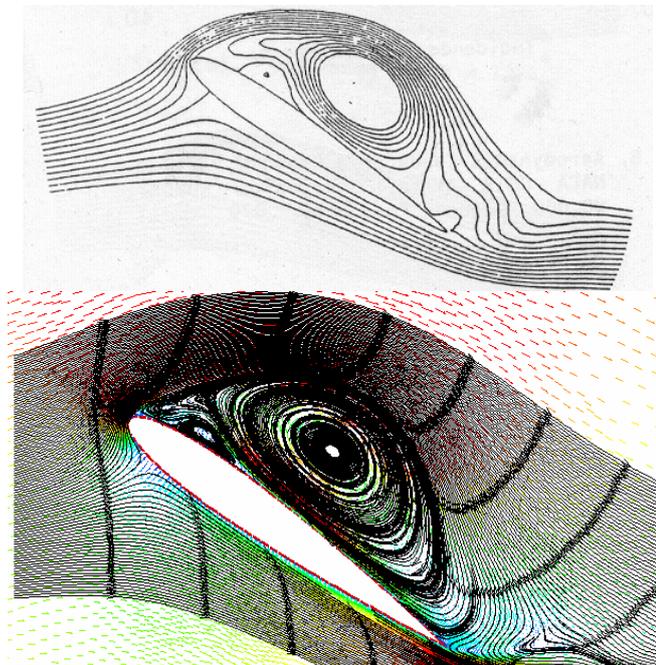


Figure 79. Streamlines of NACA0012 Airfoil at $\alpha = 34.4^\circ$

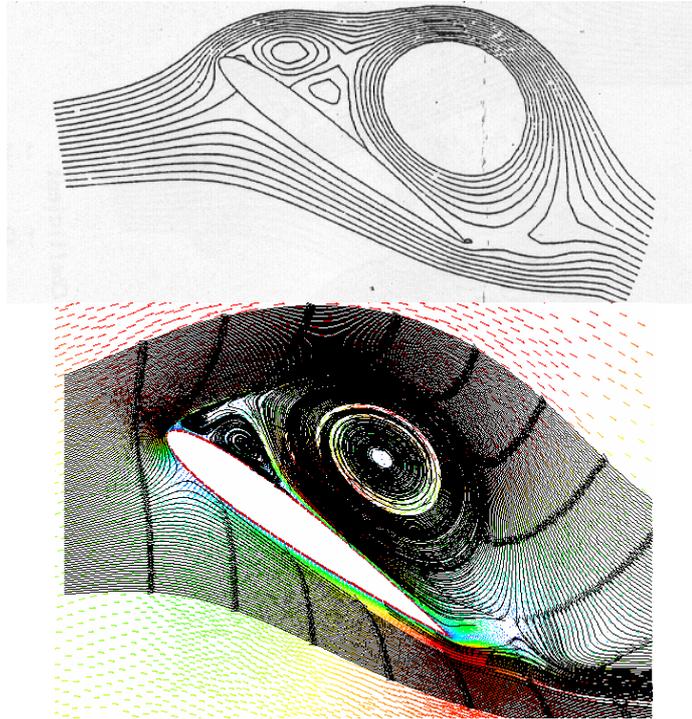


Figure 80. Streamlines of NACA0012 Airfoil at $\alpha = 34.4^\circ$ (After rotation stopped)

In general, it can be concluded from this validation case that ALE formulation and grid movement calculation algorithms give acceptable results for laminar flow conditions.

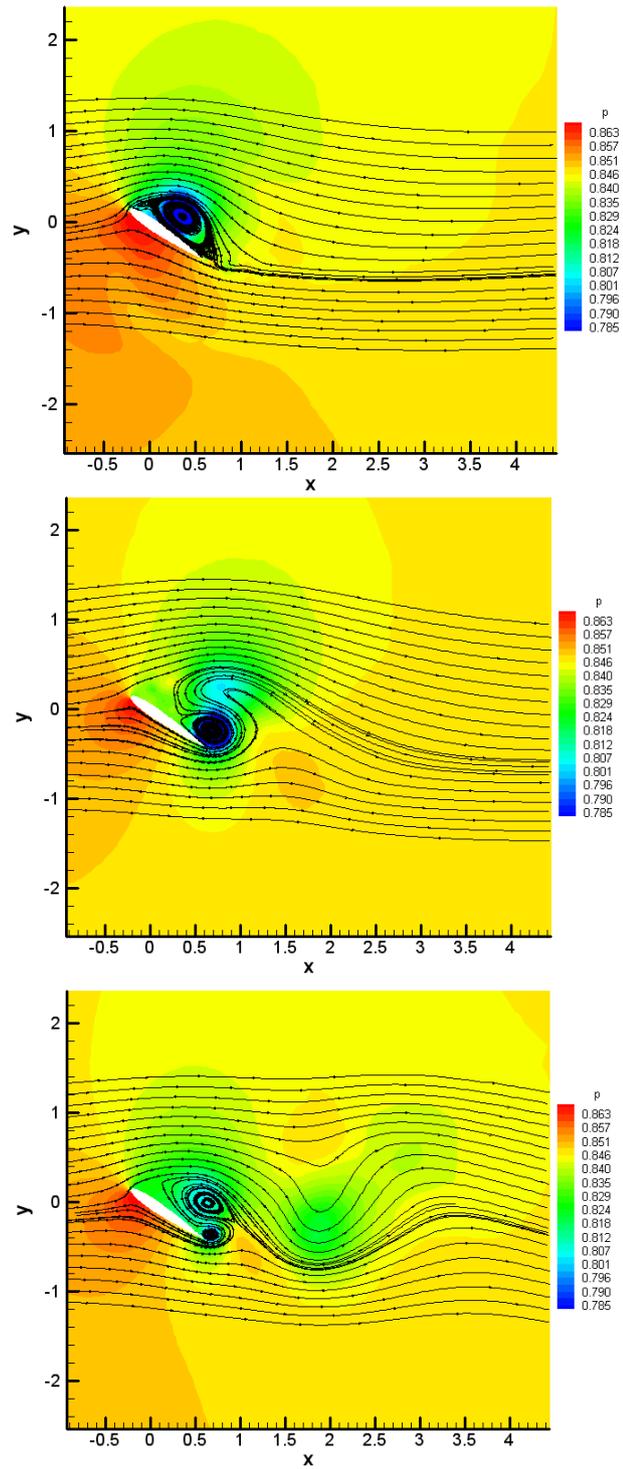


Figure 81. Vortex Shedding Behind Pitching NACA0012 Airfoil

4.3.5. Turbulent Flow over 2-D Flat Plate

After validating 2-D Navier-Stokes solver for laminar flows, again flat plate boundary layer solution is employed to assess the accuracy of the Spalart-Allmaras turbulence model implemented into the solver. The computations are performed on two-dimensional unstructured grid which was generated by FLUENT-GAMBIT mesh generation software, for a Reynolds number of 2×10^6 and Mach number of 0.5. This case was also studied by Frink for the validation of turbulence models [17].

Since Reynolds number is higher than the case that was studied for laminar flow conditions, a high resolution unstructured grid is generated in order to observe the turbulent boundary layer development. The minimum Δy near the wall is set to be to 1×10^{-5} times the plate length and this resulted in more than 40 grid nodes inside the turbulent boundary layer. There exist 100 grid points on the wall and 100 grid points up to the far field. Grid clustering is applied at the leading edge. The mesh generated is composed of 40,000 triangular control cells and 20,301 nodes. The mesh and its details at the leading edge can be seen in Figure 82 and Figure 83, respectively.

In Figure 84, the residual history of the solution both for the conservative variables and the turbulent viscosity variable is shown. It can be seen from this figure that first 100,000 iterations were performed for laminar flow. Turbulence was introduced after laminar boundary layer was developed in order to speed up the convergence for turbulent flow computations. It can be seen that the residual dropped approximately 3 orders of magnitude for conservative variables. For the turbulent viscosity variable, a residual drop of approximately 4 orders of magnitude was obtained. Total computational run time including the laminar flow calculation for the first 100,000 iterations lasted approximately 3 hours when using 20 Pentium IV processors in parallel.

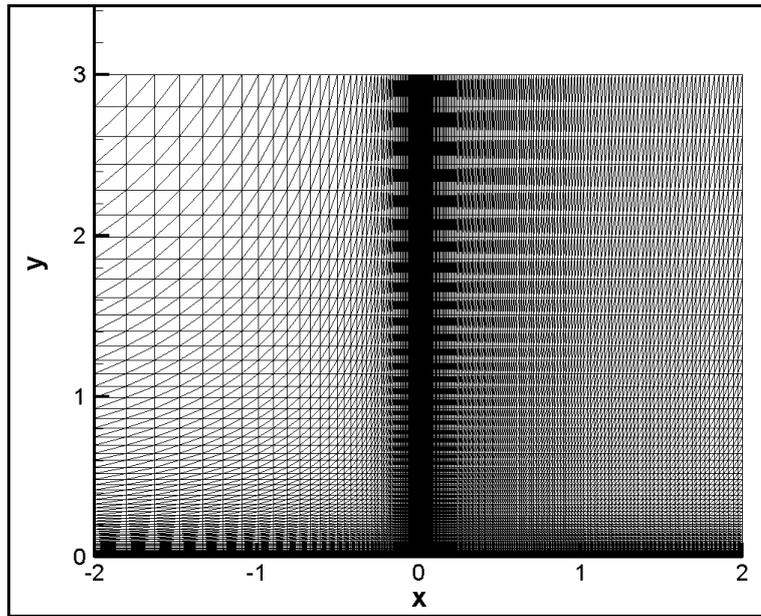


Figure 82. Unstructured Viscous Mesh for Turbulent Flat Plate Problem

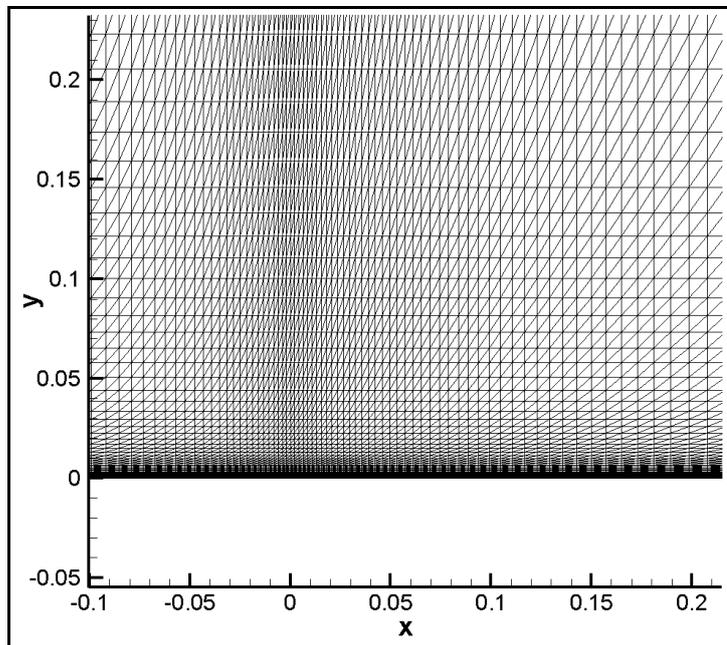


Figure 83. Mesh Details at the Leading Edge of Flat Plate

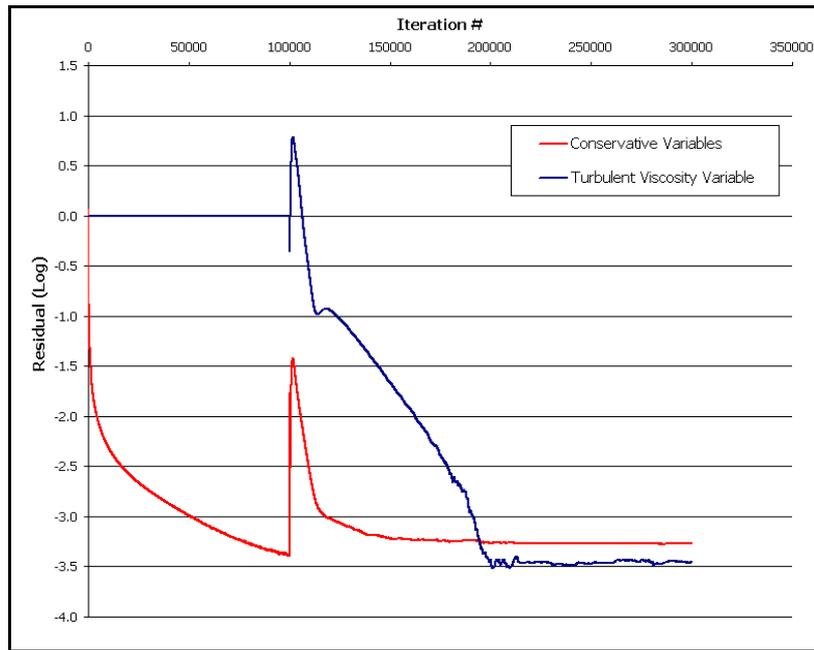


Figure 84. Residual History of Turbulent Flow over a 2-D Flat Plate

Figure 85 shows the turbulent boundary layer development at the middle of the flat plate. Figure 86 and Figure 87 show the variation of the turbulent viscosity along the flat plate. The turbulent boundary layer profile and the behavior of the variation of turbulent viscosity variable seem to be as expected. Near to the wall boundary, turbulent viscosity is zero. Afterwards, it faces a quick rise in the turbulent sub layer and finally reaches the free stream value. The effect of this turbulent viscosity can be observed in Figure 88 where laminar and turbulent boundary layers are compared. Difference between the laminar and turbulent boundary layer thicknesses can be observed from this figure.

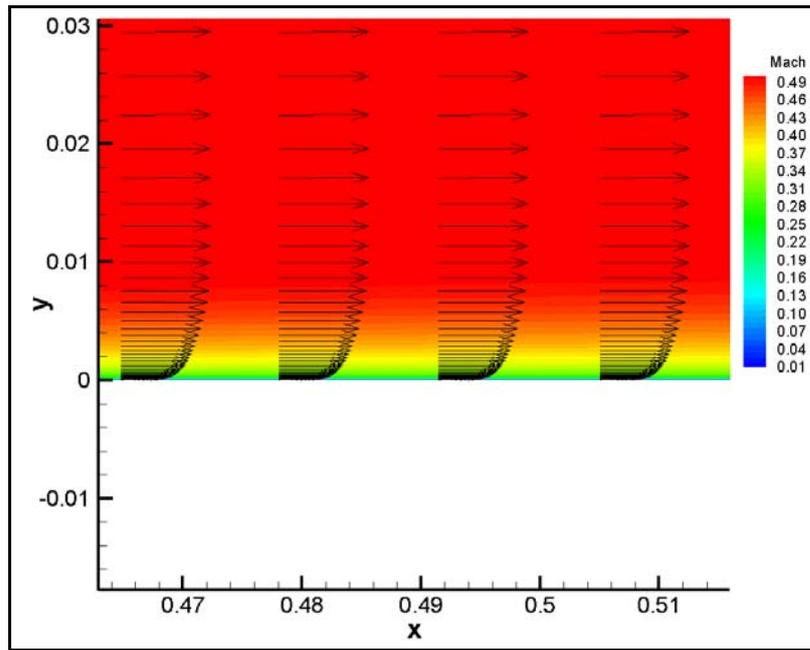


Figure 85. Turbulent Boundary Layer Development at the Middle of Flat Plate

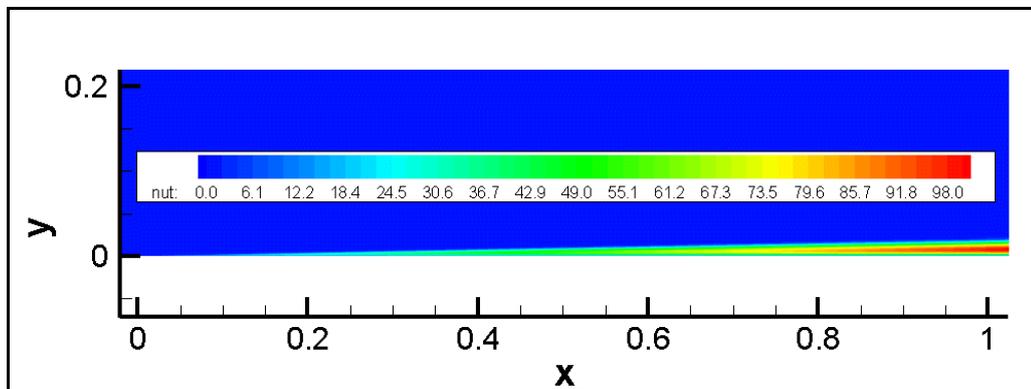


Figure 86. Variation of Turbulent Viscosity Variable over the Turbulent Flat Plate

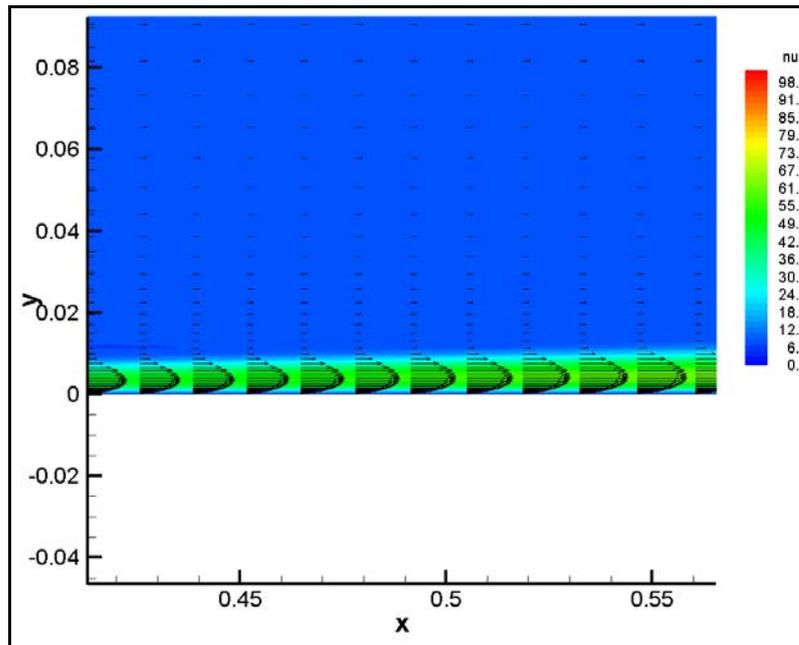


Figure 87. Variation of Turbulent Viscosity Variable at the Middle of Flat Plate
(Close-up View)

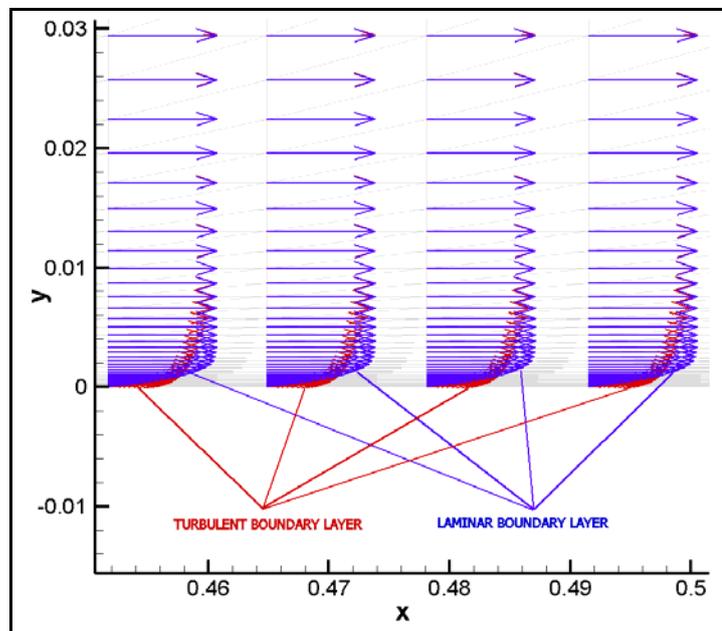


Figure 88. Comparison of Laminar and Turbulent Boundary Layers over Flat Plate

Figure 89 shows the axial velocity profiles at 10%, 25%, 50%, 75% and 90% of the chord, respectively, which are compared with the approximate solution of the turbulent boundary layer over a flat plate with zero pressure gradient;

$$\frac{u}{U_\infty} = \left(\frac{y}{\delta}\right)^{\frac{1}{n}} \quad n = \begin{cases} 7 & 5 \times 10^5 < Re_x < 1 \times 10^7 \\ 8 & 1 \times 10^7 < Re_x < 1 \times 10^8 \\ 9 & 1 \times 10^8 < Re_x < 1 \times 10^9 \end{cases} \quad (4.5)$$

$$\text{and} \quad \delta = 0.0575 \cdot \left(\frac{n^2 + 3n + 2}{n}\right)^{\frac{4}{5}} \cdot x \cdot Re_x^{-\frac{1}{5}} \quad (4.6)$$

where δ is the turbulent boundary layer thickness.

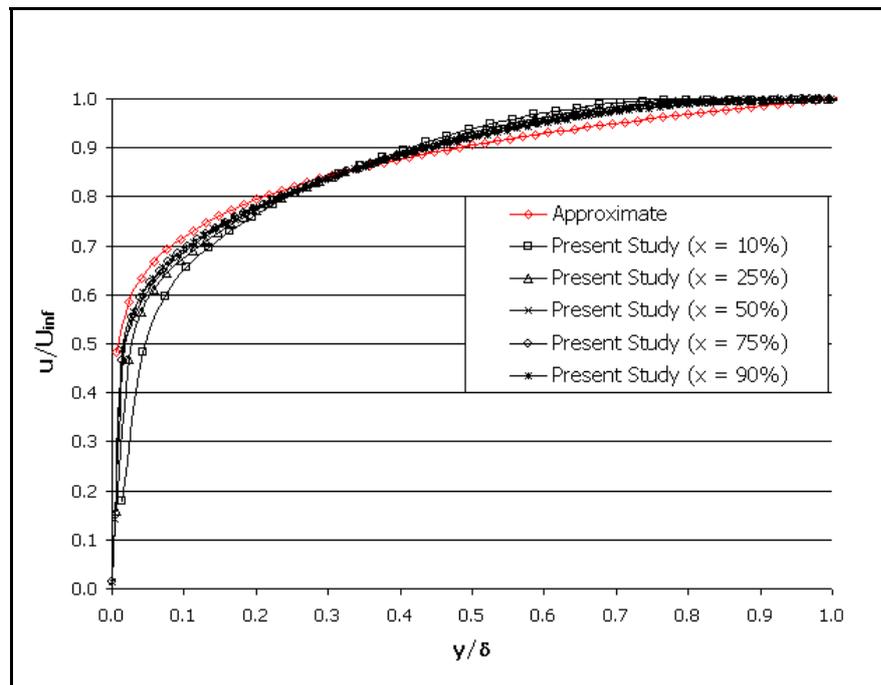


Figure 89. Turbulent Axial Velocity Distribution at 10%, 25%, 50%, 75% and 90% of the Flat Plate in Comparison with the Approximate Solution

It can be seen from this figure that computational results are in good agreement with the theoretical solution expect for the region near the leading edge of the plate. Highly stretched computational elements in normal direction to the wall may be the explanation of the slight difference between the computed results and the theoretical approximate solution. Discrepancy at the vicinity of the leading edge is because of the numerical invalidity of the approximate solution there.

In Figure 90 comparison of the computed velocity profile with respect to Universal Velocity Distribution is presented in terms of dimensionless normal distance (y^+) vs dimensionless velocity (U^+) at the middle (50%) of the flat plate.

$$y^+ = \frac{u_\tau y}{\nu} \text{ and } U^+ = \frac{u}{u_\tau} \quad (4.7)$$

where $u_\tau = \sqrt{\frac{\tau_w}{\rho}}$ is known as the friction velocity

Universal Velocity Distribution over a smooth flat plate is developed based on dimensional analysis and incorporation of experimental data to determine the constants appearing in the expression, [103].

It can be observed from Figure 90 that computational velocity profile is in full agreement with the Universal Velocity Distribution in almost every region of the turbulent boundary layer. Inside viscous sublayer, buffer zone and log layer zone the computed velocity profile perfectly exactly matches with the analytical and semi-empirical distribution. This figure shows that implementation of Spalart-Allmaras turbulence model successfully simulates the turbulent boundary layer development over a flat plate.

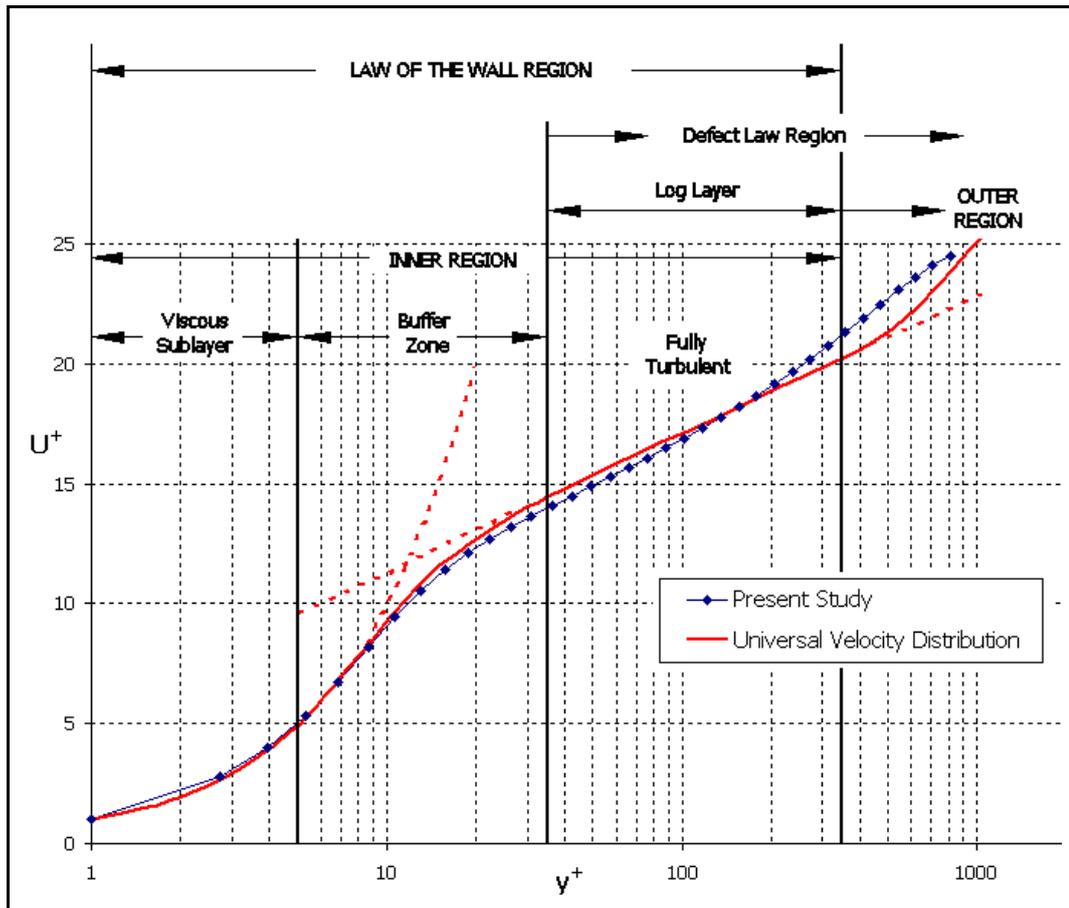


Figure 90. Non-Dimensional Velocity Profile for Turbulent Flow Over a Flat Plate and Identification of Different Regions within the Turbulent Boundary Layer [103]

Figure 91, compares the skin friction coefficient computed with the approximate $1/7^{\text{th}}$ power law solution for turbulent flow and Blassius' laminar flow solution. Approximate skin friction coefficient for turbulent flows is defined as:

$$C_{f,x} = 0.092 \cdot \left(\frac{n}{n^2 + 3n + 2} \right)^{\frac{1}{5}} \cdot \text{Re}_x^{-\frac{1}{5}} \quad (4.8)$$

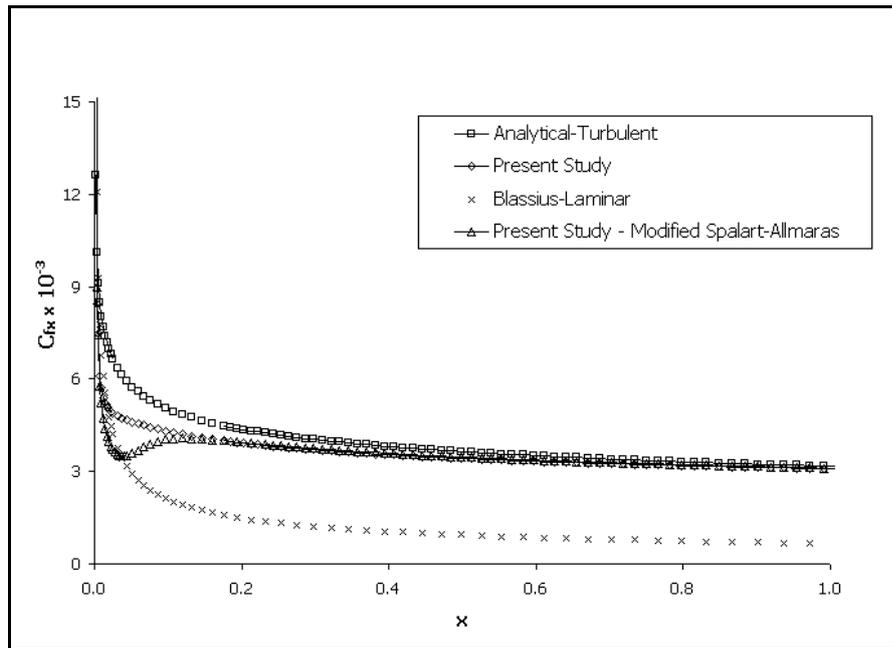


Figure 91. Turbulent Skin Friction Coefficient vs. Axial Location

Figure 91 shows that the computational results give a good correspondence with approximate analytical solution especially for $0.2 < x/L < 1.0$. Computed skin friction coefficient shows a laminar type behaviour near the plate leading edge ($0 < x/L < 0.2$). Use of modified functions $f_{v2}(\chi)$ and $f_{v3}(\chi)$ (Equations 3.62 and 3.63) in the Spalart-Allmaras turbulence model resulted in a laminar-turbulent transition behavior near the plate leading edge. It is also stated by Guillen, *et al.* [92] that use of these modified functions results in a modification of the natural laminar-turbulent transition of the Spalart-Allmaras turbulence model. Another reason for this discrepancy near the leading edge may be explained due to the numerical anomalies of the weighed averaging scheme at the stagnation point where an inviscid surface suddenly changes to a viscous surface as Frink [17] stated. Same situation near the leading edge is also observed by Pan and Cheng [104].

In general it can be concluded from Figure 91 that the turbulence model implemented works accurately since the results of the computation came out to be far away from Blassius' laminar flow solution and close to approximate analytical solution for turbulent flow.

This test case validated the Spalart-Allmaras turbulence model implementation into the 2-D Navier-Stokes solver. Results came out to be in good agreement with analytical approximations. The results of the solver will be compared with available experimental data for another test case in the coming section.

4.3.6. Turbulent Flow over NACA0012 Airfoil

As a last validation case for the 2-D turbulent Navier-Stokes solver, flow over NACA0012 airfoil geometry problem is investigated. There exists several experimental data in AGARD Advisory Report no. 138 [105] for this airfoil.

Free stream flow conditions are Mach number of 0.3 and Reynolds number of 1.86×10^6 at an incidence angle of 4.04 degrees. In order to observe the relatively thin boundary layer development when compared to laminar case on the airfoil surface a new denser mesh is generated for turbulent flow calculations. This time unlike for the laminar computational grid, Figure 64, entire domain unstructured mesh is generated from structured mesh obtained from CFD-GEOM grid generation software. Grid resolution on the airfoil surfaces is made denser which resulted in approximately 20 grid points inside the turbulent boundary layer all over the surface. There exist 158 grid points both on the suction and pressure sides of the airfoil surface. The mesh is composed of 30,720 triangular control cells and 15,803 nodes. The mesh and its details nearby the wall, at the leading edge and at the trailing edge are presented in Figure 92, Figure 93 and Figure 94 respectively.

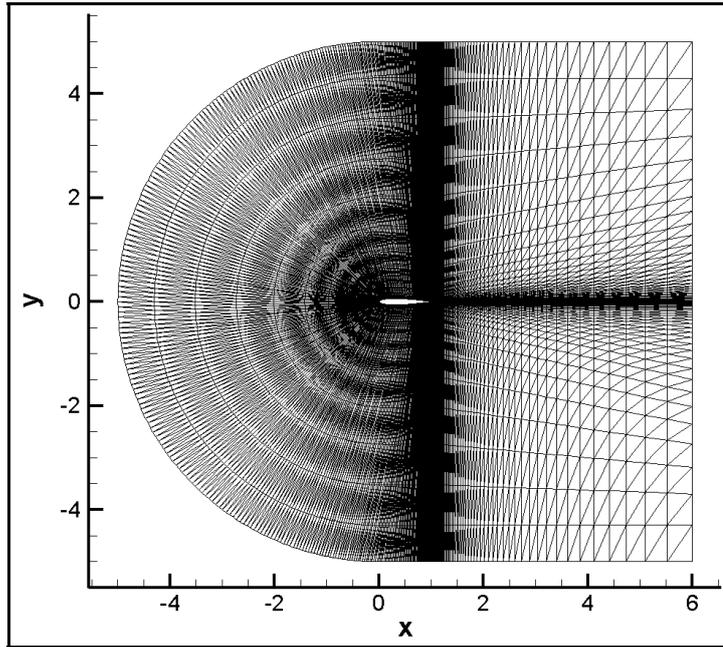


Figure 92. Unstructured Viscous Mesh for Turbulent Flow over NACA0012 Airfoil

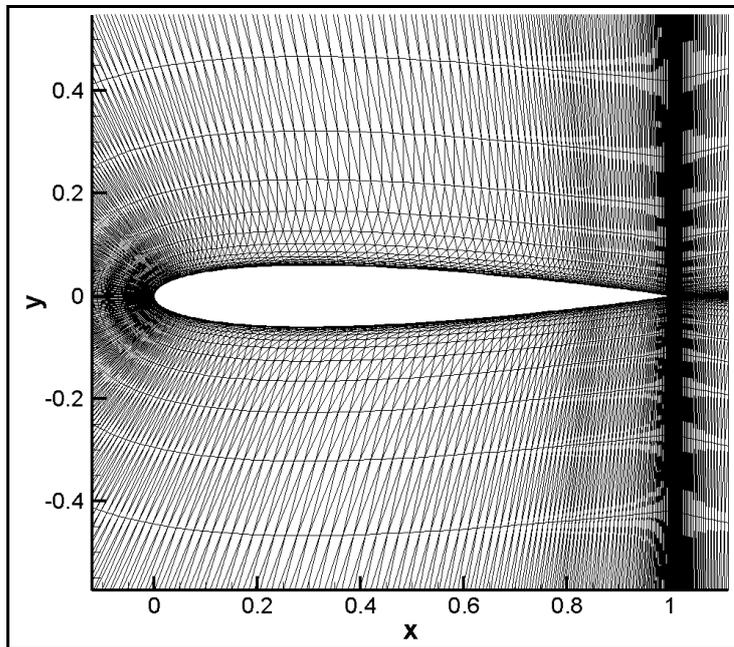


Figure 93. Viscous Mesh Detail nearby NACA0012 Airfoil

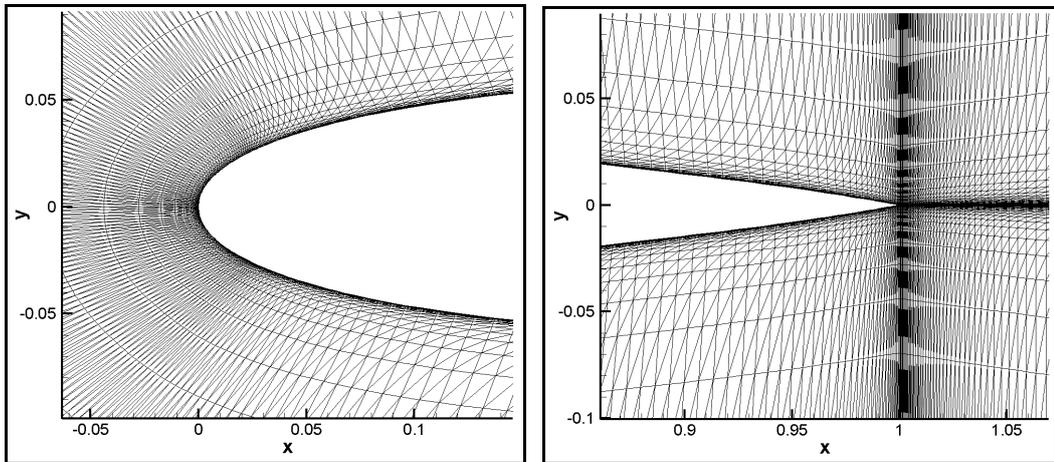


Figure 94. Mesh Details at the Leading and Trailing Edges of NACA0012 Airfoil

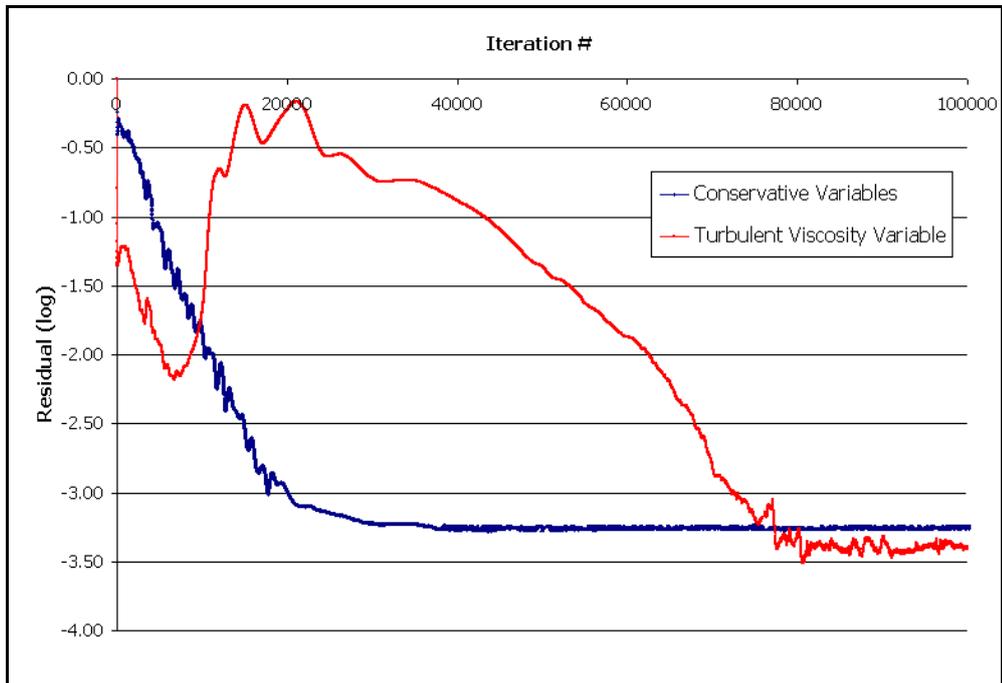


Figure 95. Residual History of Turbulent Flow over a NACA0012 Airfoil

In Figure 95, the residual history of the solution both for the conservative variables and the turbulent viscosity variable are shown. It can be seen from this figure that the residual dropped approximately 3 – 3.5 orders of magnitude for conservative variables. For the turbulent viscosity variable, a residual drop of approximately 3.5 orders of magnitude is obtained. Total computational run time for 100,000 iterations lasted approximately 1 hour with the use of 20 Pentium IV processors in parallel.

Same problem is solved by commercial CFD solver FLUENT in order to compare the boundary layer development on the airfoil surface. Same structured mesh generated by CFD-GEOM mesh generation program is used for FLUENT solver.

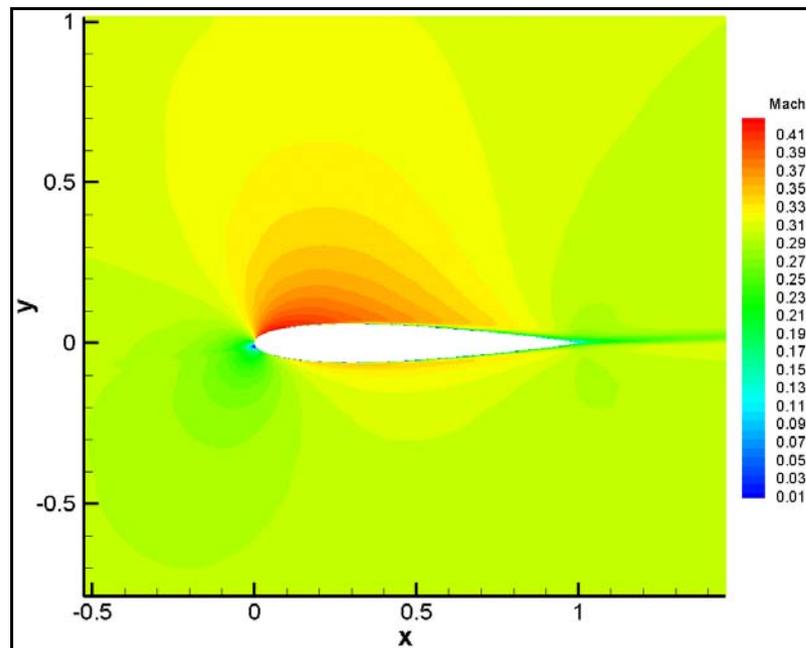


Figure 96. Mach Number Contours for Turbulent Flow Over NACA0012 Airfoil,
Present Study

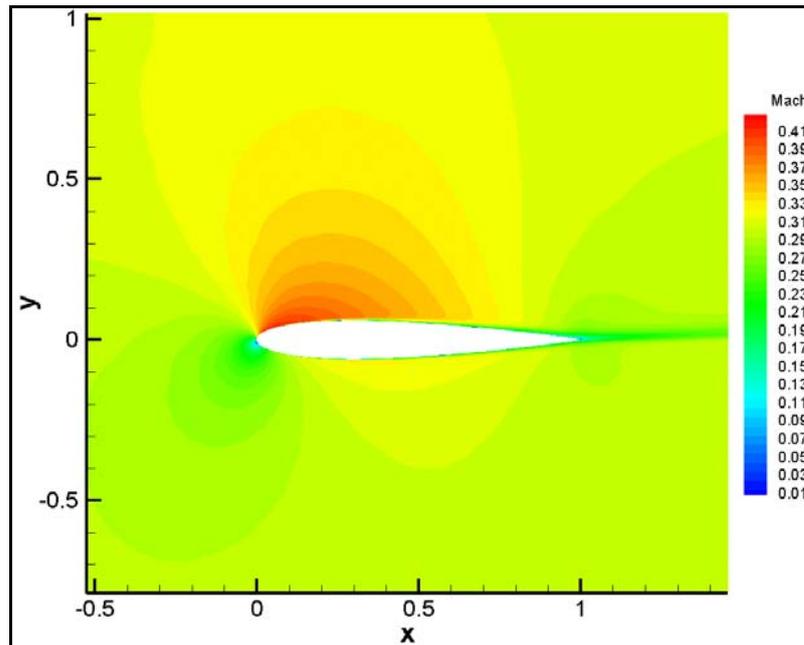


Figure 97. Mach Number Contours for Turbulent Flow Over NACA0012 Airfoil, FLUENT Solver

In Figure 96 and Figure 97, results of the present study and FLUENT solver are represented, respectively. It can be seen from these figures that mach number contours obtained over the airfoil by two solvers are in good agreement. In both figures wake flow at the trailing edge is visible. Suction and pressure side flow patterns are similar to each other.

Obtained turbulent viscosity variation contours over the airfoil by the solver developed and FLUENT solver are represented in Figure 98 and Figure 99, respectively. It can be observed from these figures that turbulence occurs especially at the wake of the airfoil due to large velocity gradients at the trailing edge. It can also be seen that generation of turbulence is more noticeable at the suction side of the airfoil in both solutions which results in thicker boundary layer formation. When results of two solvers are compared it can be said that both quantitatively and qualitatively they are in good agreement with each other.

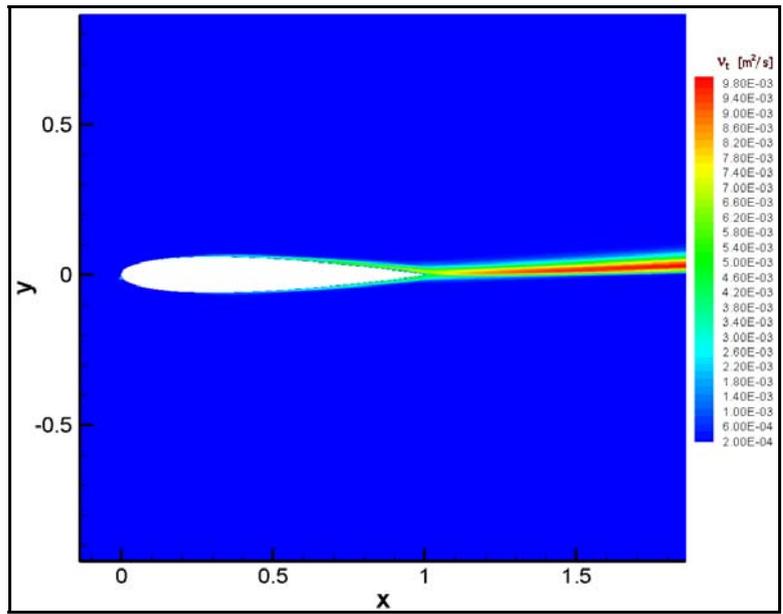


Figure 98. Turbulent Viscosity Contours for Flow over NACA0012 Airfoil, Present Study

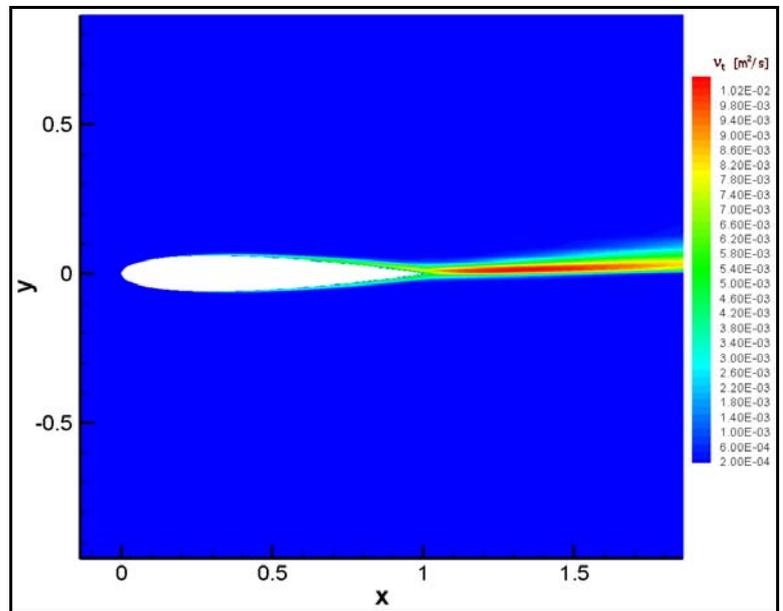


Figure 99. Turbulent Viscosity Contours for Flow over NACA0012 Airfoil, FLUENT Solver

In Figure 100, Figure 101, Figure 102 and Figure 103, computed velocity profiles on the upper and lower surfaces of airfoil at 20%, 50%, 80% chord length and at the trailing edge of the airfoil are presented, respectively. Results are compared with solution of FLUENT solver and laminar flow calculations. Development of turbulent boundary layer on the airfoil surface can be observed especially after 20% chord length. It must be noted here that modified versions of functions $f_{v_2}(\chi)$ and $f_{v_3}(\chi)$ for turbulence model are introduced in computations. Therefore, as it can be seen in Figure 100 near to the leading edge velocity profiles are closer to laminar boundary layer. After 20% chord length the difference between turbulent and laminar boundary layer profiles became noticeable. Especially at the trailing edge of the flat plate reversed flow which is observed for laminar flow calculations is disappeared for turbulent flow calculations. Computational results came out to be in good agreement with the results of FLUENT solver.

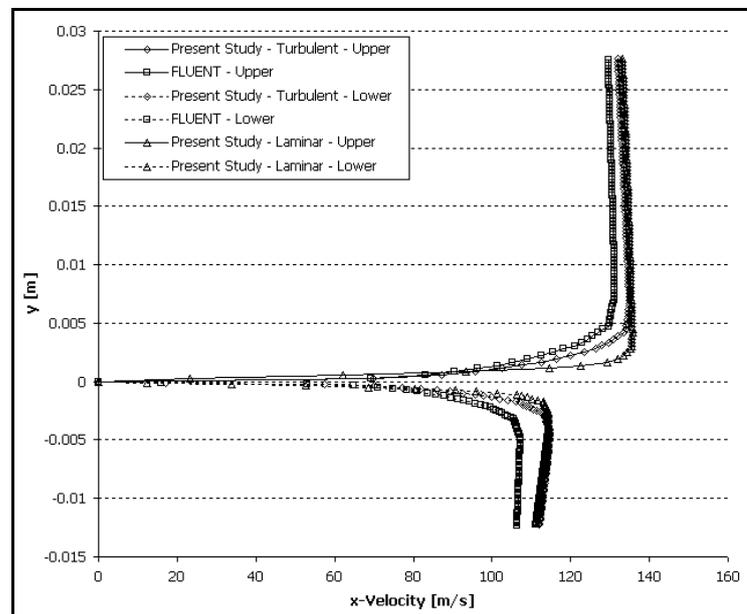


Figure 100. Velocity Profiles at the Upper and Lower Surfaces of NACA0012 Airfoil, $x/L = 0.2$

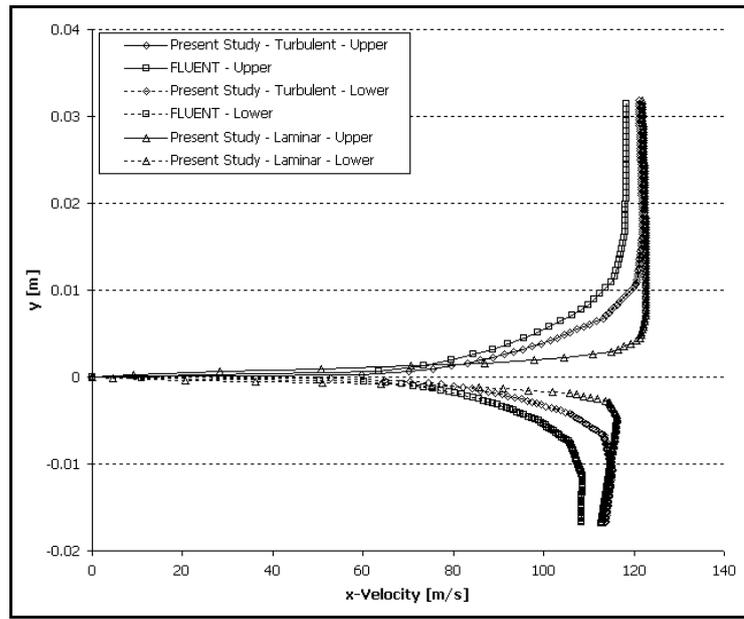


Figure 101. Velocity Profiles at the Upper and Lower Surfaces of NACA0012 Airfoil, $x/L = 0.5$

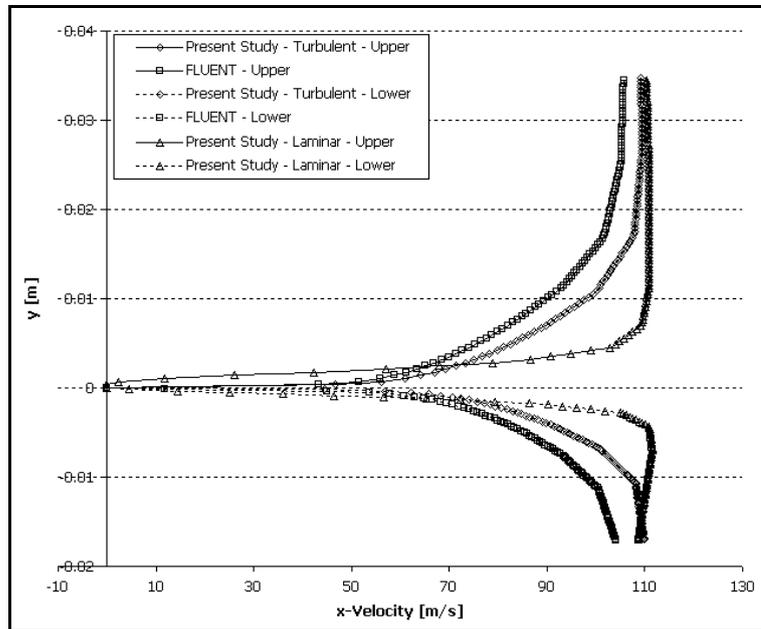


Figure 102. Velocity Profiles at the Upper and Lower Surfaces of NACA0012 Airfoil, $x/L = 0.8$

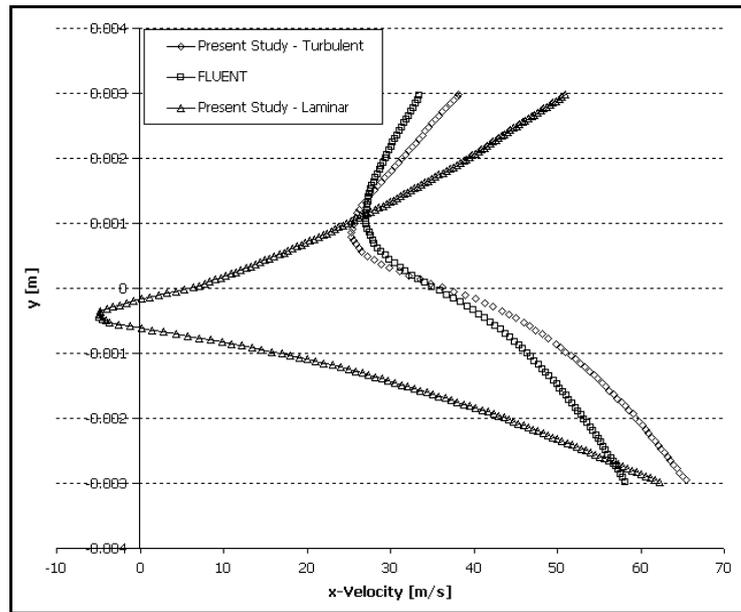


Figure 103. Velocity Profiles at the Trailing Edge of NACA0012 Airfoil

In Figure 104, Figure 105, Figure 106 and Figure 107, computed turbulent viscosity profiles on the upper and lower surfaces of airfoil at 20%, 50%, 80% chord length and at the trailing edge of the airfoil are presented, respectively. Results are compared with solution of FLUENT solver. It can be observed from these figures that turbulent viscosity value increases from leading edge to trailing edge. It can also be said that turbulent viscosity at the upper surface of the airfoil came out to be greater than the lower surface due to the incidence in the free stream flow. Increase in the velocity gradients on the upper surface of the airfoil because of incidence, results in stronger turbulent flow pattern at the suction side. In Figure 107 the turbulent viscosity profiles show the wake flow at the trailing edge of the airfoil. Qualitatively and quantitatively the computed results are in good agreement with results of FLUENT solver.

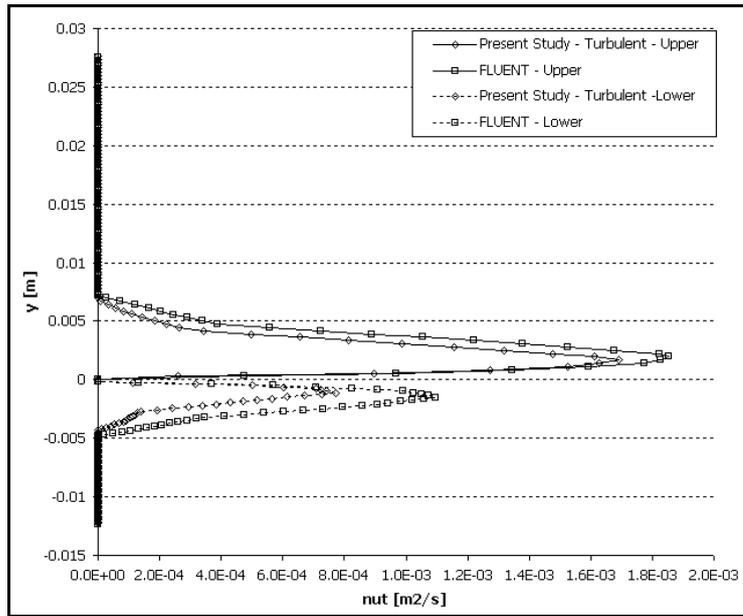


Figure 104. Turbulent Viscosity Profiles at the Upper and Lower Surfaces of NACA0012 Airfoil, $x/L = 0.2$

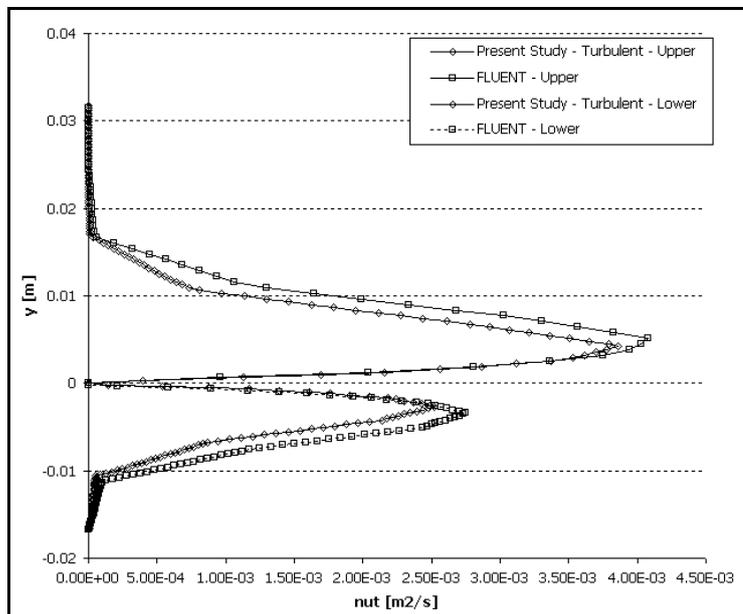


Figure 105. Turbulent Viscosity Profiles at the Upper and Lower Surfaces of NACA0012 Airfoil, $x/L = 0.5$

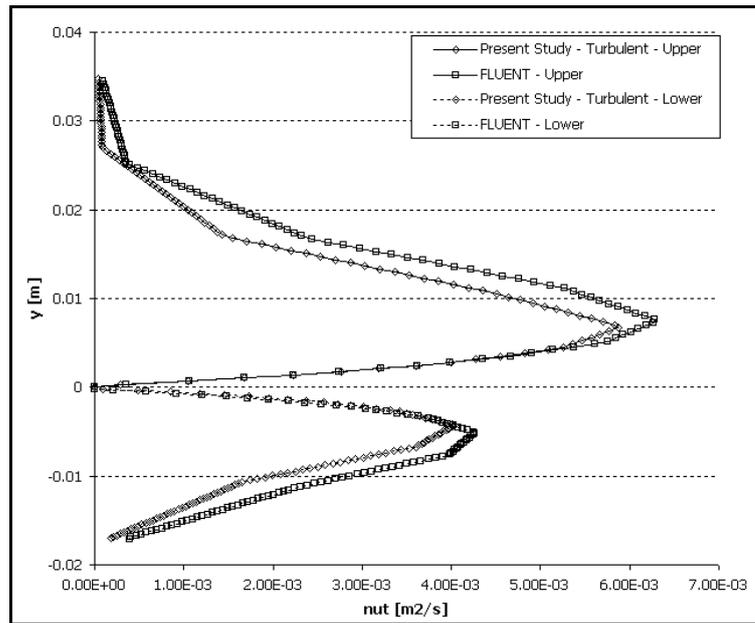


Figure 106. Turbulent Viscosity Profiles at the Upper and Lower Surfaces of NACA0012 Airfoil, $x/L = 0.8$

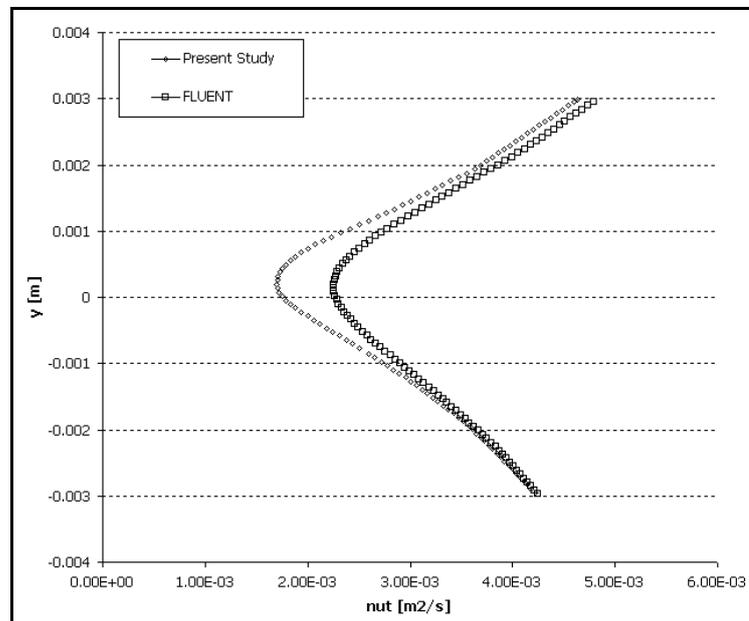


Figure 107. Turbulent Viscosity Profiles at the Trailing Edge of NACA0012 Airfoil

In Figure 108, the comparison of the calculated pressure coefficient with experimental data is presented. Computational results are in good agreement with the experimental data both for present study and FLUENT solver except for the leading edge. Laminar distribution differs from the turbulent flow solutions at the trailing edge where reverse flow is observed for laminar flow calculations, which is also observable in Figure 102 and Figure 103.

By this test case, validation of implementation of Spalart-Allmaras turbulence model into 2-D Navier-Stokes solver is completed. Results came out to be quite satisfactory and in good agreement with experimental results.

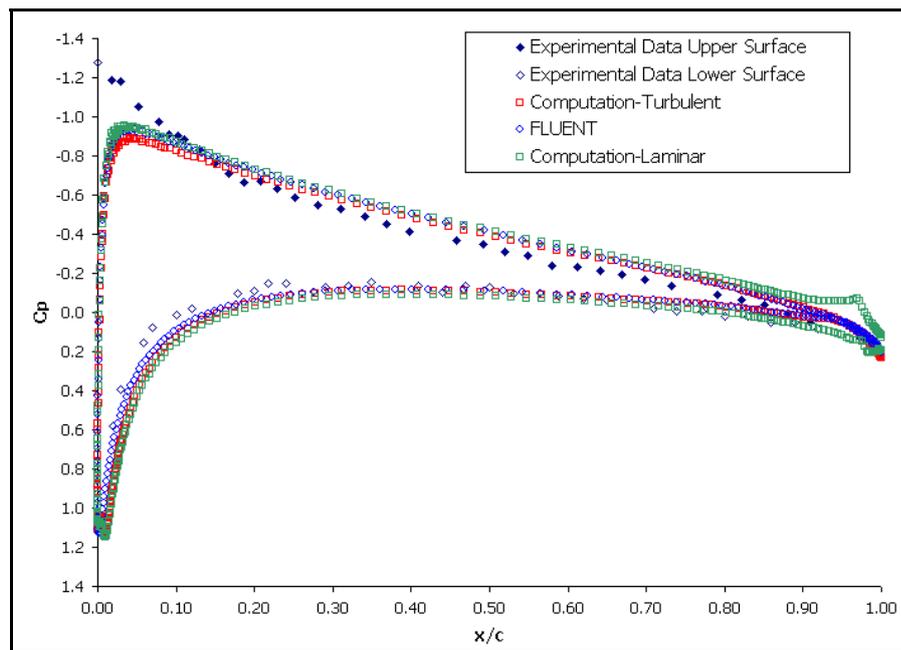


Figure 108. Pressure Coefficient Comparison for Turbulent Flow over NACA0012 Airfoil, $Re = 1.86 \times 10^6$

4.4. 3-D Validation Studies for the Navier-Stokes Solver

In order to validate the implementation of viscous term calculation algorithms and Spalart-Allmaras turbulence model into the 3-D Euler solver laminar and turbulent flow over a 3-D flat plate is investigated.

4.4.1. Laminar Flow Over 3-D Flat Plate

In order to verify the viscous algorithms in 3-D, the flat plate problem which was used for 2-D verification, is investigated. Same flow conditions with 2-D case are taken, Reynolds number of 35,000 and Mach number of 0.3.

A computational mesh composed of hexahedral elements have been generated by CFD-GEOM. Simply, the mesh generated for a 2-D case is extruded by 0.05 m in y-direction. The hexahedral, structured computational mesh is composed of 148x3x40 grid points. 80 of the grid points in x-direction are on the wall boundary. There exists 40 grid points up to the far field boundary in z-direction.

A post processing code has been written to obtain the unstructured mesh with tetrahedral elements from this structured mesh. Hexahedral elements are divided into 6 tetrahedral elements taking the normal direction of the faces into consideration. The mesh that has been generated is composed of 68,796 tetrahedral elements and 17,760 computational node points.

Figure 109 shows the unstructured computational mesh generated for 3-D flat plate. The detail of the computational mesh near the boundary layer at the leading edge is given in Figure 110. Cube shaped hexahedral elements have been tried to be generated in order to minimize the highly stretched tetrahedrons elements

especially near the wall boundaries. On the other hand, in order to observe the boundary layer on the flat plate, the minimum Δy near the wall is set to be 5×10^{-5} times the plate length which resulted in at least 20 grid nodes inside the boundary layer. Because of this grid clustering near the wall boundary, the formation of highly stretched tetrahedrons became unavoidable. If all of the hexahedral elements are generated as square cubes, which is the ideal situation for generating high quality unskewed tetrahedron elements, the mesh size will become enormously high.

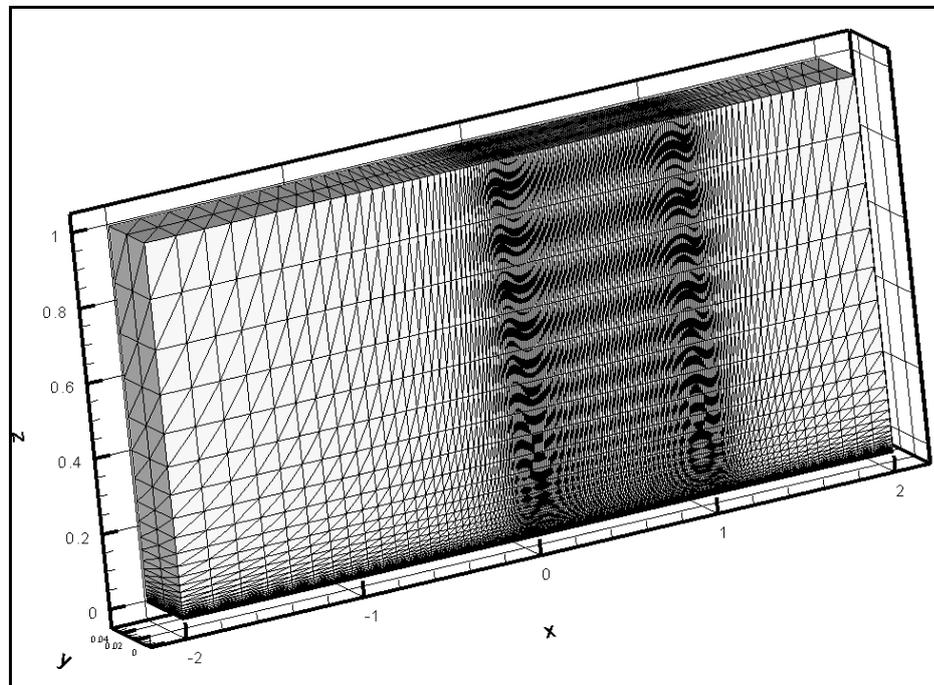


Figure 109. 3-D Unstructured Mesh for Flat Plate

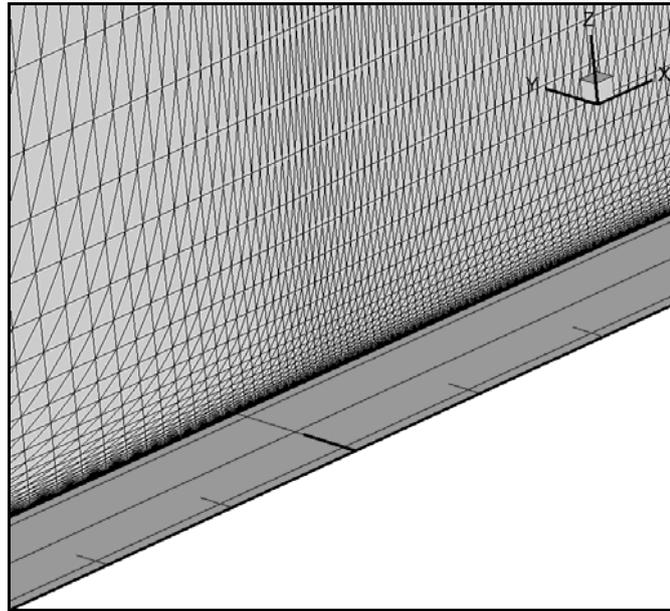


Figure 110. 3-D Unstructured Mesh Detail at the Leading Edge of Flat Plate

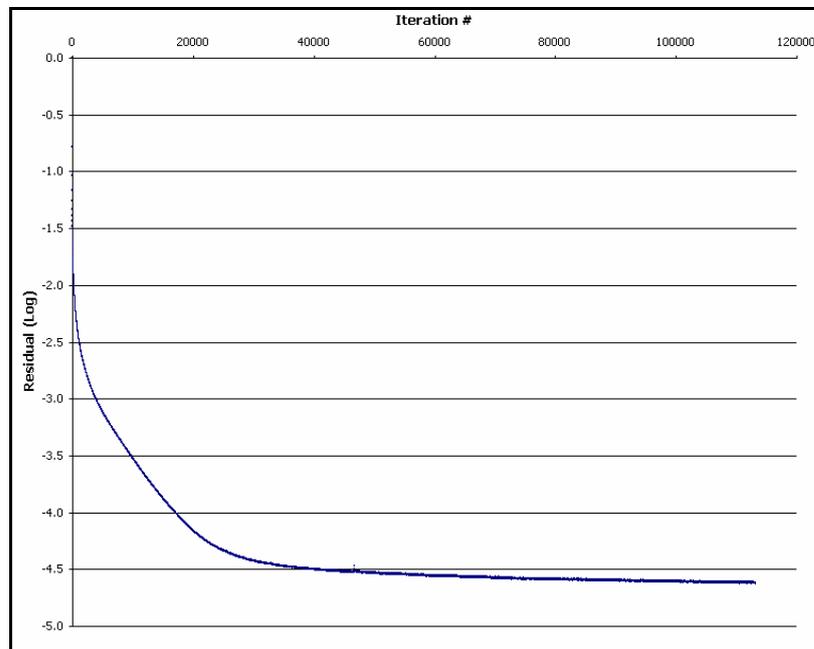


Figure 111. Residual History for Solution of 3-D Flat Plate Problem

In Figure 111, the residual history of the solution is presented. It can be seen that the residual dropped approximately 4.5 orders of magnitude in 110,000 iterations and the solution reached the steady state. Approximately 2 hours of computation time was required for the steady-state solution with parallel processing using 20 Pentium IV processors.

The boundary layer development at the mid-section of the computational domain ($y = 0.025$ m) can be observed in Figure 112. Laminar boundary layer development, which came out to be the same with the 2-D case (Figure 48), can be observed from this figure.

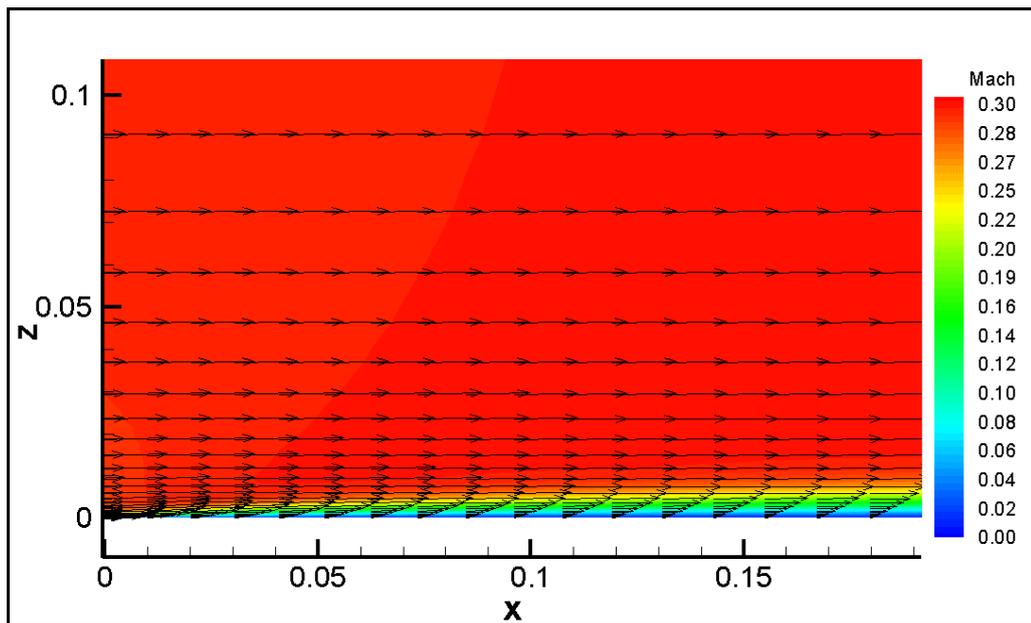


Figure 112. Laminar Boundary Layer Development at the Leading Edge of 3-D Flat Plate ($y = 0.025$ m)

The results have been compared with the analytical solution of Blassius' [99] as it was done for the two-dimensional case. Figure 113, Figure 114, Figure 115 and Figure 116 show the axial velocity profiles at 25%, 50%, 75% and 90% of the chord, respectively, and at different sections of the computational domain ($y = 0.01$, $y = 0.025$ and $y = 0.04$) which are compared with Blassius' analytical solution for incompressible laminar flow, Equation (4.1).

It can be seen from these figures that velocity profiles at all positions of chord and at all sections came out to be in good agreement with the analytical solution.

By this test case, the viscous flux computation algorithms implemented into the 3-D Euler solver are validated.

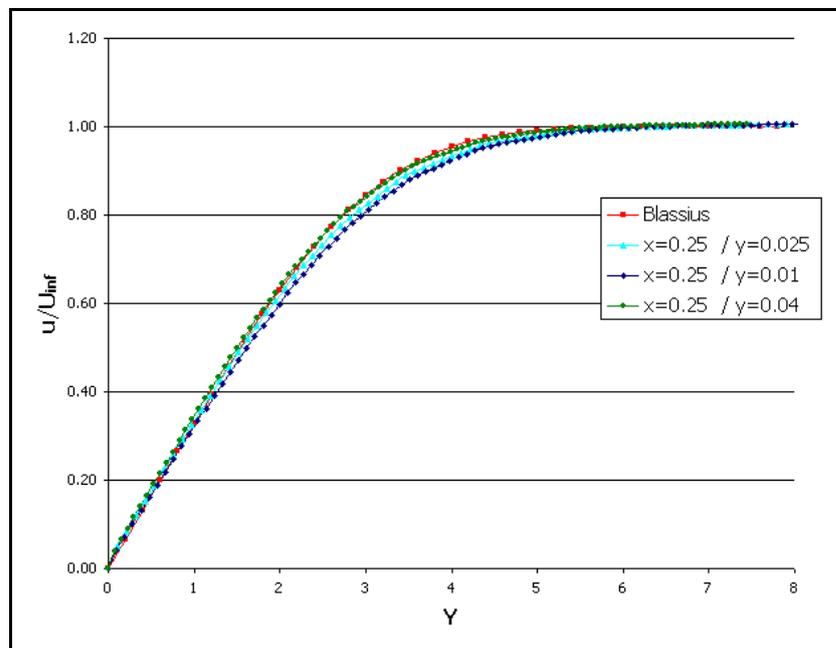


Figure 113. Axial Velocity Distribution at 25% of the 3-D Plate

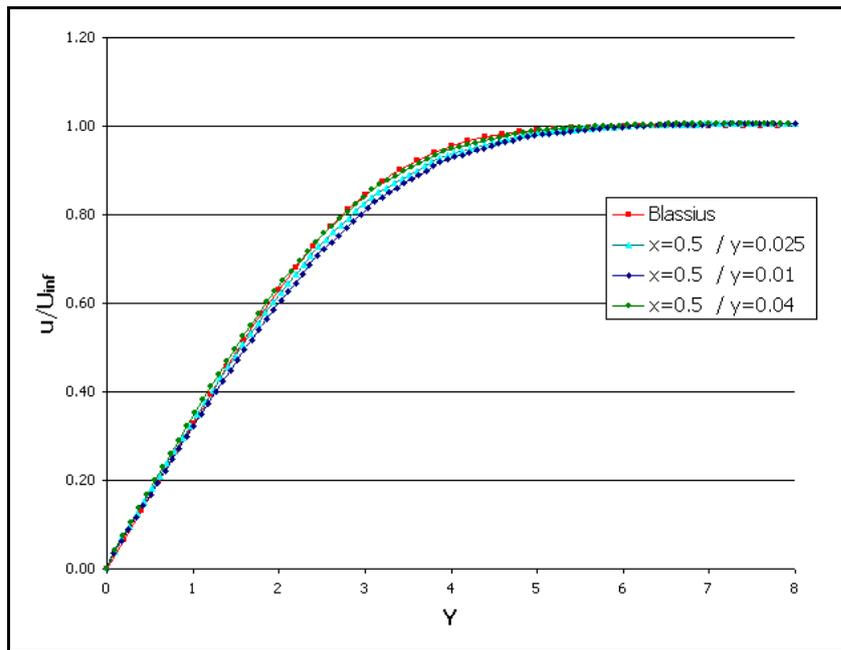


Figure 114. Axial Velocity Distribution at 50% of the 3-D Plate

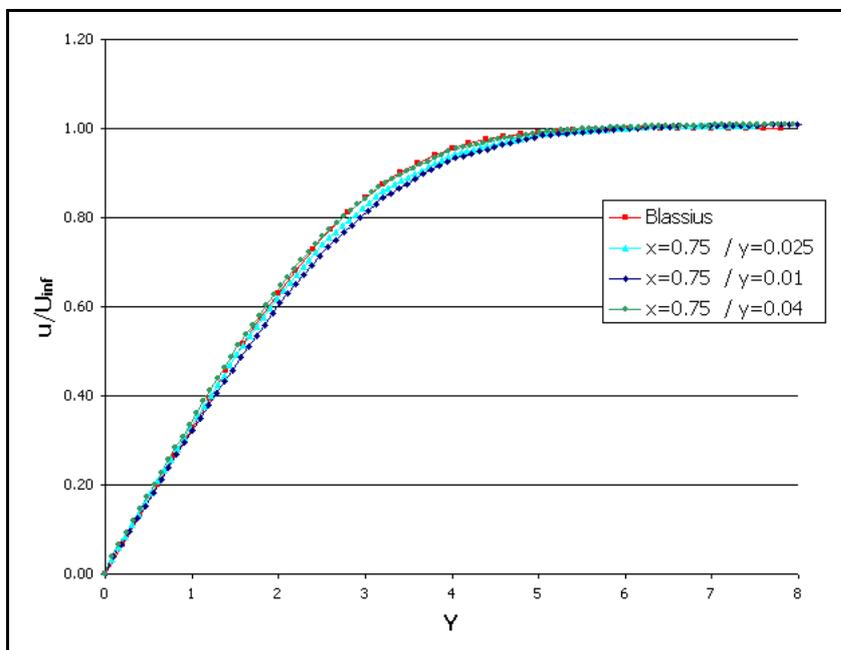


Figure 115. Axial Velocity Distribution at 75% of the 3-D Plate

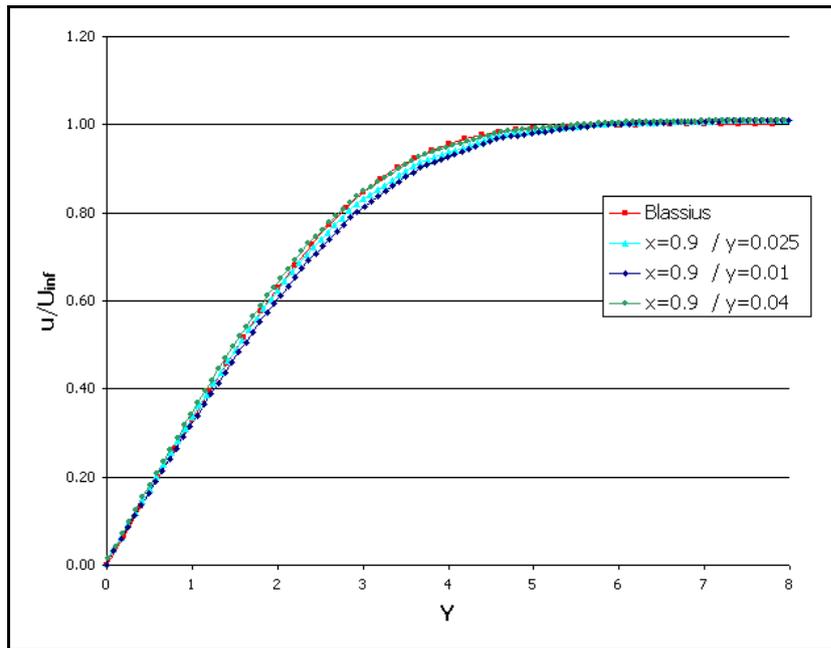


Figure 116. Axial Velocity Distribution at 90% of the 3-D Plate

4.4.2. Turbulent Flow over 3-D Flat Plate

In order to validate Spalart-Allmaras turbulence model implementation in 3-D Navier-Stokes solver, approximate solution for the turbulent flow over flat plate is used for the comparison of computed results.

In order to observe the turbulent boundary layer for large Reynolds numbers, a higher resolution unstructured grid than the one generated for the laminar case is generated by the CFD-GEOM. Mesh generation procedure is totally the same which is described in Section 4.4.1. As free stream flow conditions, Reynolds number of 2×10^6 and Mach number of 0.5 are taken in order to compare the results with 2-D solutions.

The hexahedral, structured computational mesh is composed of $198 \times 3 \times 100$ grid points. 100 of the grid points in x-direction are on the wall boundary. There

exists 100 grid points up to the far field boundary in z-direction. The unstructured mesh which is obtained from this structured mesh is composed of 234,036 tetrahedral elements and 59,400 computational nodes.

Figure 117 shows the unstructured computational mesh generated for 3-D flat plate. The detail of the computational mesh near the boundary layer at the leading edge is given Figure 118. Minimum Δy near the wall has been limited to 1×10^{-5} times the plate length which resulted in at least 40 grid nodes inside the boundary layer.

In Figure 119, the residual history of the solution both for the conservative variables and the turbulent viscosity variable is presented. As it was done for 2-D case, first 100,000 iterations were performed without turbulence. Turbulent flow calculations were performed using the laminar flow solution as an initial condition in order to accelerate the convergence. It can be seen that the residual dropped approximately 2.5 - 3 orders of magnitude for conservative variables. For the turbulent viscosity variable, a residual drop of approximately 1.5 orders of magnitude was obtained. All of the computations including the laminar flow calculations lasted approximately 12 hours with the use of 20 Pentium IV processors in parallel.

In Figure 120, the turbulent boundary layer development at the middle of the flat plate can be observed. Figure 121 shows the variation of the turbulent viscosity along the flat plate. The turbulent boundary layer profile and the behavior of the variation of turbulent variable seem to be as expected and quantitatively the same with 2-D case (Figure 85 and Figure 86).

Figure 122, Figure 123, Figure 124 and Figure 125 show the axial velocity profiles at 25%, 50%, 75% and 90% of the chord, respectively, which are compared with the approximate solution of the turbulent boundary layer (Equations 4.3 & 4.4) and 2-D solutions.

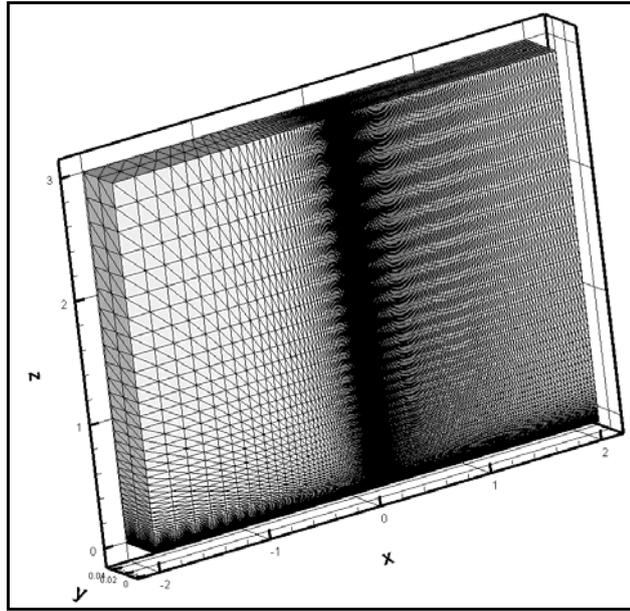


Figure 117. 3-D Unstructured Mesh for Flat Plate – Turbulent

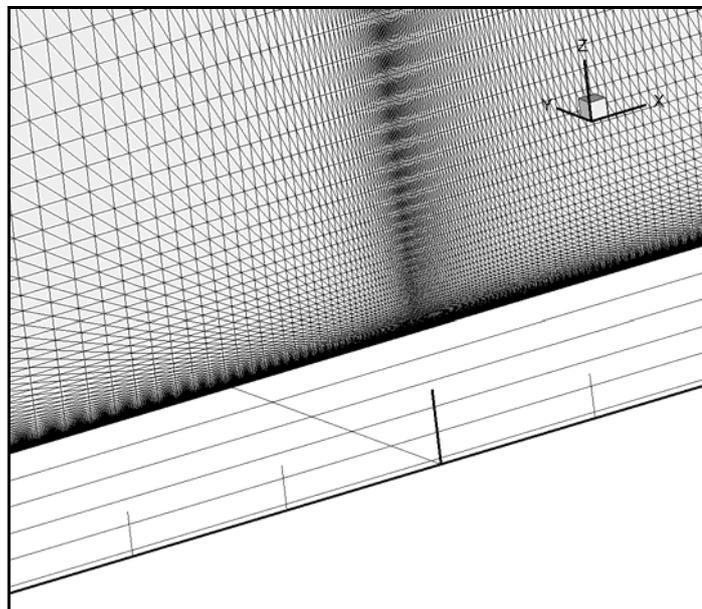


Figure 118. 3-D Unstructured Mesh Detail at the Leading Edge of Flat Plate –
Turbulent

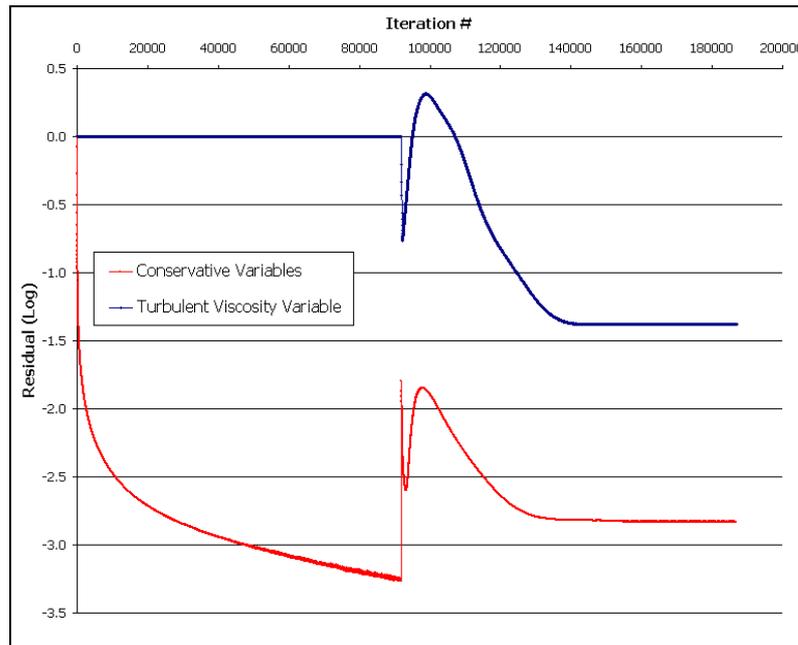


Figure 119. Residual History for Solution of Turbulent 3-D Flat Plate Problem

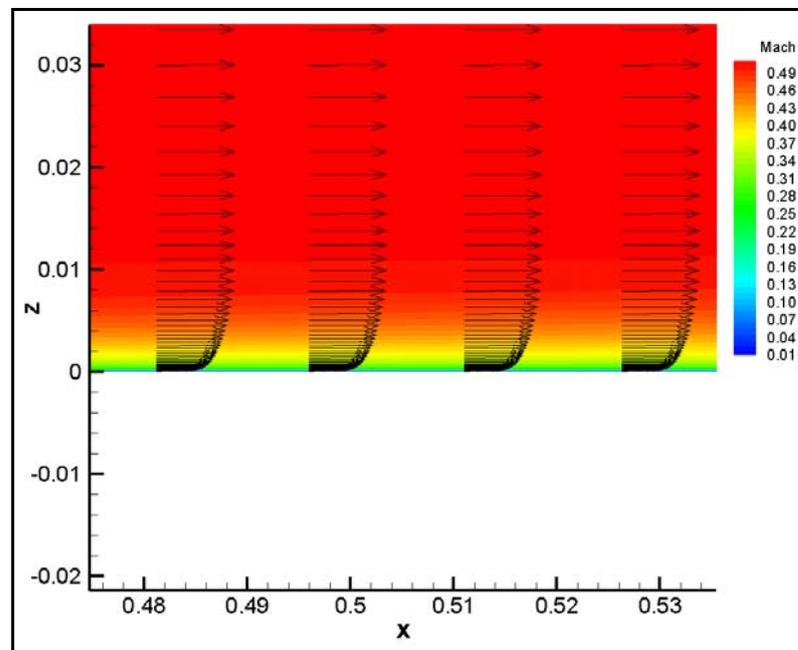


Figure 120. Turbulent Boundary Layer Development at the Middle of 3-D Flat Plate
($y = 0.025\text{m}$)

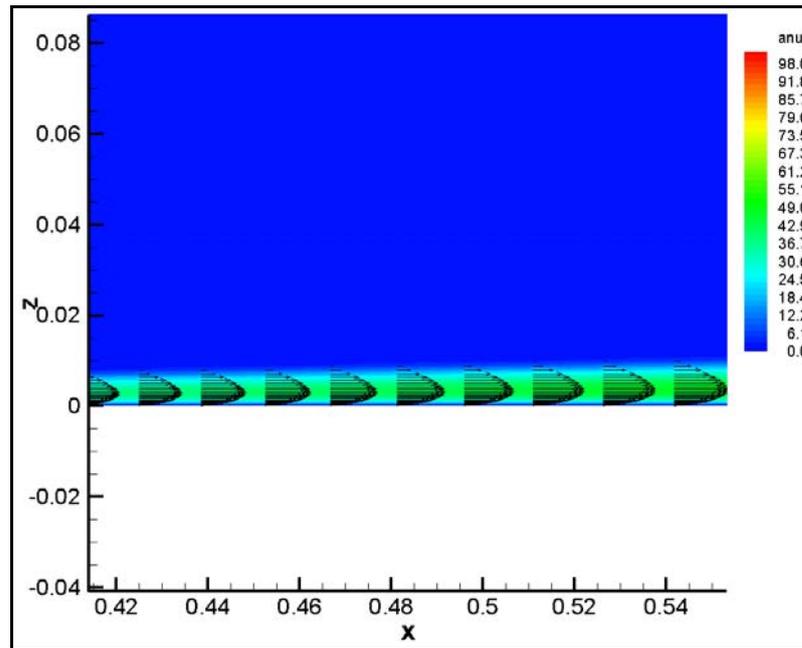


Figure 121. Variation of Turbulent Viscosity Variable at the Middle of 3-D Flat Plate ($y = 0.025\text{m}$)

In Figure 122, Figure 123, Figure 124 and Figure 125, it can be observed that the computational results give good correspondence with approximate analytical solution and 2-D solution. Results are a little bit far away from the approximate solution when compared with 2-D solution but the general behaviour of the velocity profiles is as expected.

This test case completed the validation studies performed on the 2-D and 3-D Navier-Stokes solvers. It can be concluded from the results that viscous flux calculations, velocity gradient calculation and grid velocity calculation algorithms are all validated with experimental and analytical data. Implementation of 2-D and 3-D ALE formulations and Spalart-Allmaras turbulence model are also validated. The 3-D solver system developed is ready to be tested for laminar and turbulent flow over a spinning projectile.

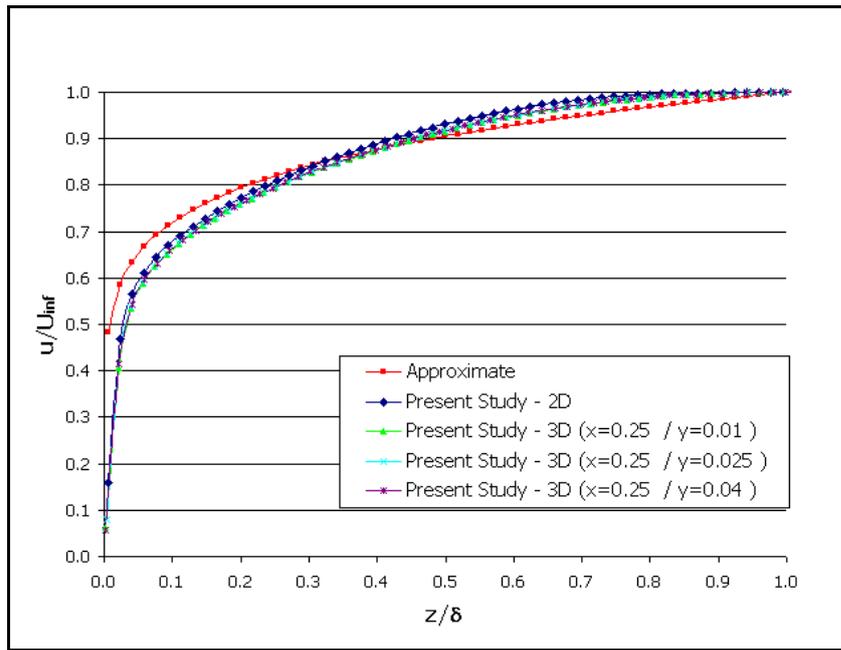


Figure 122. Axial Velocity Distribution at 25% of Turbulent 3-D Plate

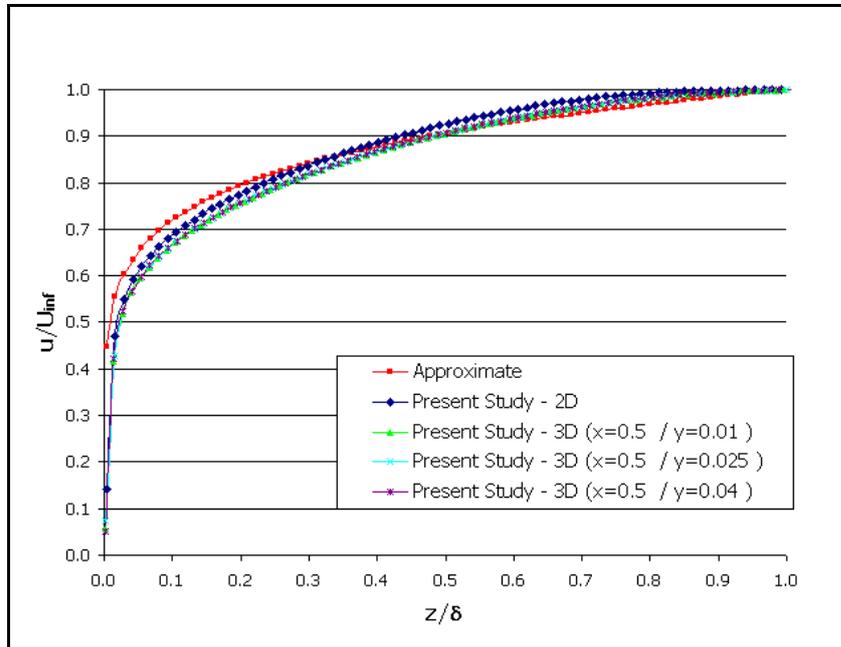


Figure 123. Axial Velocity Distribution at 50% of Turbulent 3-D Plate

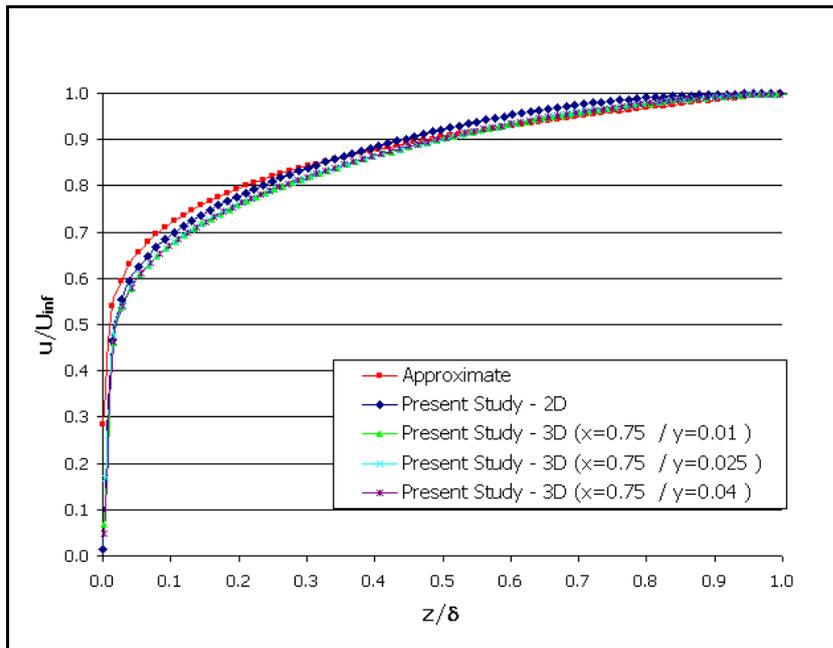


Figure 124. Axial Velocity Distribution at 75% of Turbulent 3-D Plate

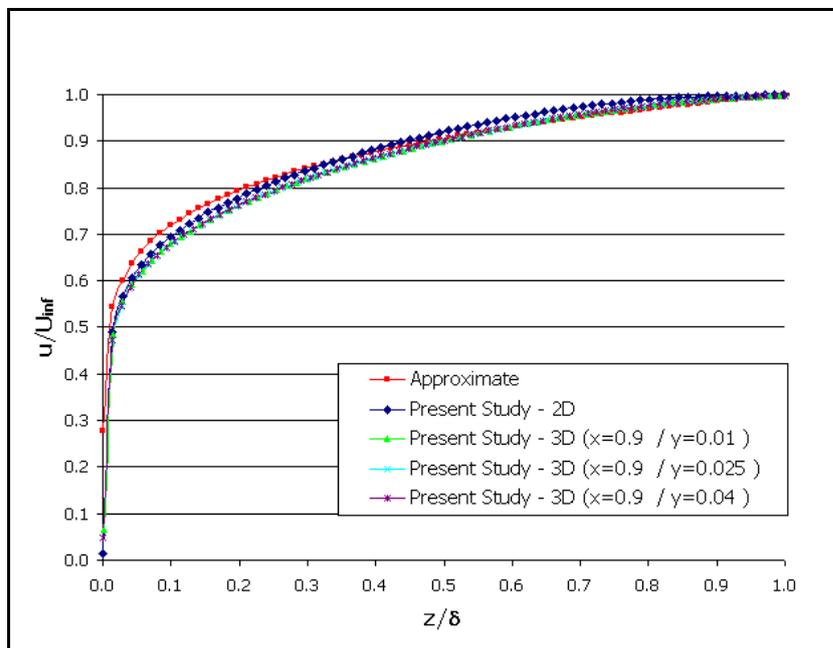


Figure 125. Axial Velocity Distribution at 90% of Turbulent 3-D Plate

CHAPTER 5

VISCOUS UNSTEADY FLOWS OVER A SPINNING PROJECTILE

After validating implementation of viscous flux computation and grid movement algorithms, ALE formulation and Spalart-Allmaras turbulence model in 2-D and 3-D solvers, a test case is studied in order to simulate the capabilities of the developed solver.

For this purpose, turbulent flow over M910 spin stabilized, 25 mm, target practice, discarding sabot-traced projectile is investigated. DeSprito and Heavey [106] studied this projectile for the CFD computation of Magnus moment and roll damping moment of a spinning projectile. There exists archival experimental data available for this projectile. The computational model of the M910 projectile is presented in Figure 126.

Most important reason why this projectile is selected is the fact that spin-stabilized projectiles usually provide some of the less demanding models to generate mesh. Absence of control fins simplifies especially the viscous mesh generation. In Section 4.2, the difficulties in viscous unstructured mesh generation especially for projectile geometries with fins was stated. Even for a projectile without fins available grid generation programs are not able to generate desired quality for fully unstructured viscous mesh especially near the wall boundaries. By the use of these grid generation programs, it is possible to obtain a high quality

hybrid viscous mesh on any geometry but full unstructured viscous mesh generation is still an outstanding problem even for a simple projectile geometry.

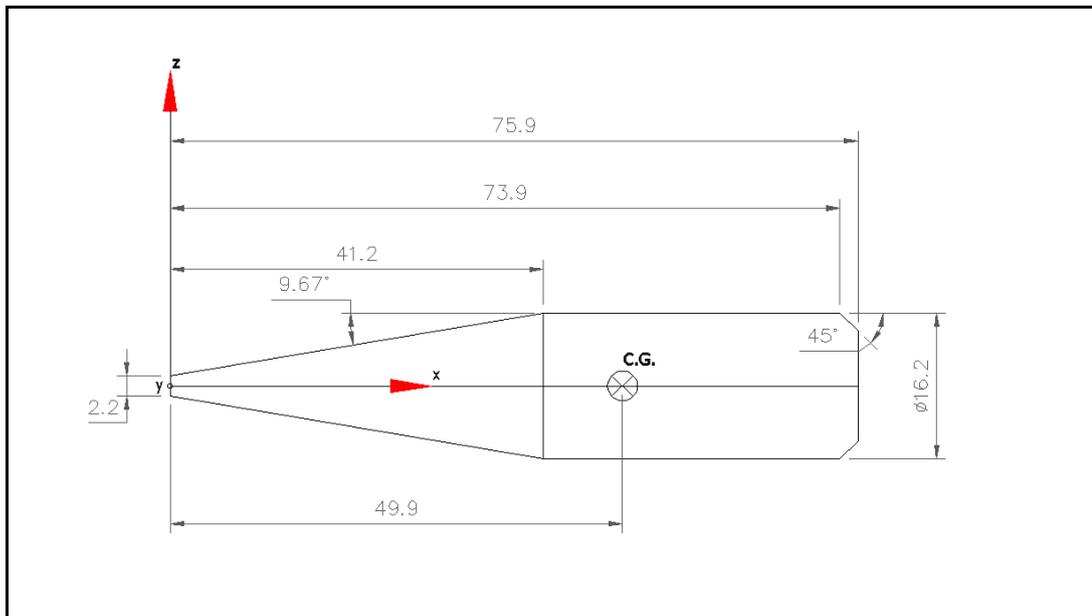


Figure 126. Detailed Geometry of M910 Projectile and the Coordinate Axes
(all dimensions are in mm)

Because of the problems stated above, a different mesh generation procedure is applied in this study. High quality viscous structured mesh is generated by the commercial grid generation program, CFD-GEOM. Obtained hexahedral elements is divided into 6 tetrahedral elements by a post processing program written in order to obtain fully unstructured computational mesh. Details of this grid generation procedure are given in APPENDIX C.

It is unavoidable to obtain unnecessary mesh resolution far away from the wall boundary by this grid generation procedure because of the nature of the

structured grids. This situation results in generation of high number of computational elements after the division of hexahedral elements into tetrahedral elements which prevents generation of dense mesh necessary especially at the base and the nose of the projectile.

Furthermore, when the hexahedral elements in the vicinity of the wall, that is the elements inside the boundary layer, are divided into tetrahedral elements, highly stretched elements are obtained. These highly stretched tetrahedral elements create problem especially in the calculation of minimum wall distance values for the Spalart-Allmaras turbulence model. The method used for the evaluation of the minimum wall distances is explained in APPENDIX D. Another important effect of these highly stretched computational elements is the deterioration of the numerical solutions after certain number of iterations.

Unstructured computational mesh for laminar and turbulent flow calculations which is generated by the procedure explained above are presented in Figure 127. Computational mesh is composed of 443,520 tetrahedral elements and 76,446 computational nodes. Far field is taken as approximately 15 times the length of the projectile away from the projectile body for both of the computational meshes. In Figure 128, mesh details on the surface of the M910 projectile are presented.

Grid clustering is performed along lateral body axis (y-axis) and vertical direction in body axis (z-axis) in order to observe the laminar and turbulent boundary layer development on the projectile surface due to the flow in longitudinal body axis (x-axis) correctly as shown in Figure 129. By this way, the minimum Δy and Δz on the wall are set to 5×10^{-5} times the projectile diameter and this resulted in more than 20 grid nodes inside the boundary layer.

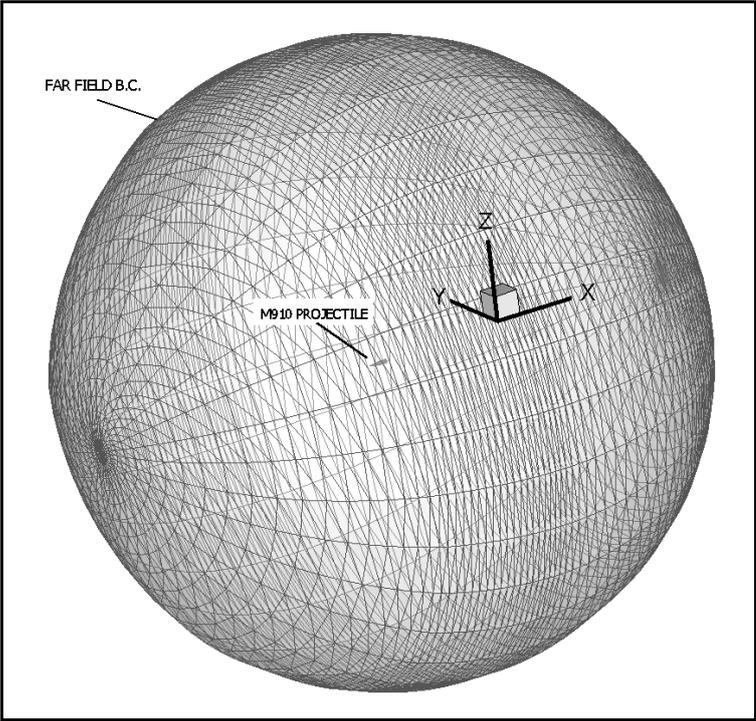


Figure 127. Unstructured Volume Mesh on M910 Projectile

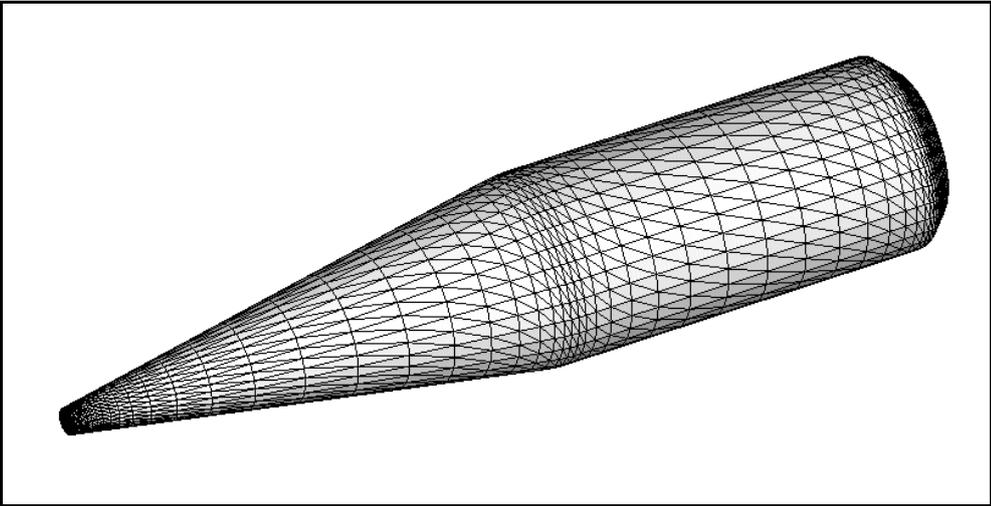


Figure 128. Unstructured Mesh Details on the Surface of M910 Projectile

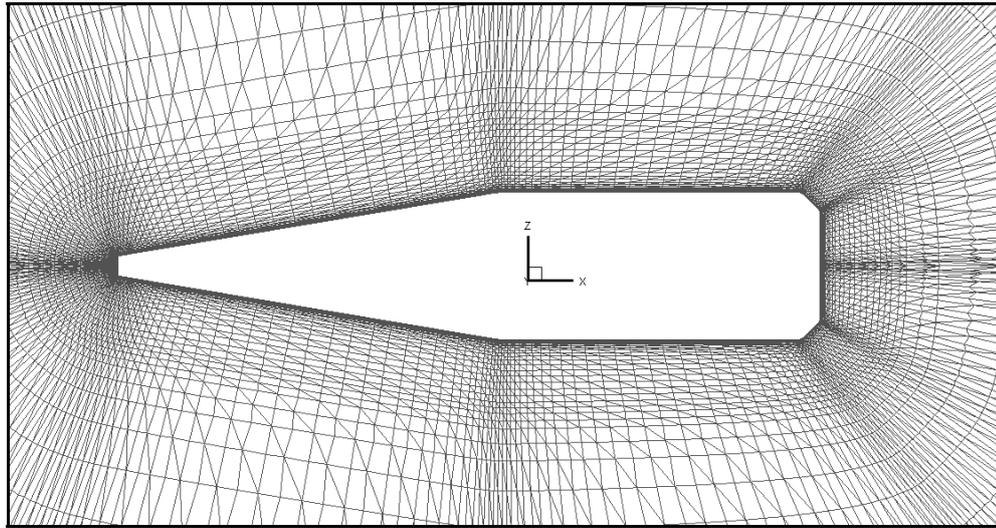


Figure 129. Unstructured Mesh Details on M910 Projectile in Vertical Direction to Body Axis

5.1. Laminar Flow Solutions over Non-Spinning M910 Projectile

Initial calculations are performed for laminar flow over non-spinning projectile. Actually, the flow conditions given by DeSprito and Heavey [106] are fully turbulent. But, in order to see the effect of the mesh generated to be flow solutions, computations are performed first without the turbulence model.

Three different flow conditions are examined for laminar flow computations. Solution matrix is given in Table 5.

In Figure 130, the convergence histories for the axial force coefficient is given for $M=0.4$, $M=0.6$ and $M=1.2$ laminar flow conditions. In Figure 131, the residual histories are presented. 10,000 iterations lasted approximately 1.5 hours on 20 Pentium IV processors working in parallel. It can be concluded from both of these figures that steady-state is reached after 10,000 iterations and axial force

coefficient became constant for all flow conditions. All of the other aerodynamic coefficients came out to be very close to zero as expected because of the symmetry of the projectile and flow with zero angle of attack.

Table 5. Solution Matrix for Laminar Flow Calculations

Mach # (M)	Re
0.4	$9.32 \times 10^{+6}$
0.6	$1.40 \times 10^{+7}$
1.2	$2.80 \times 10^{+7}$

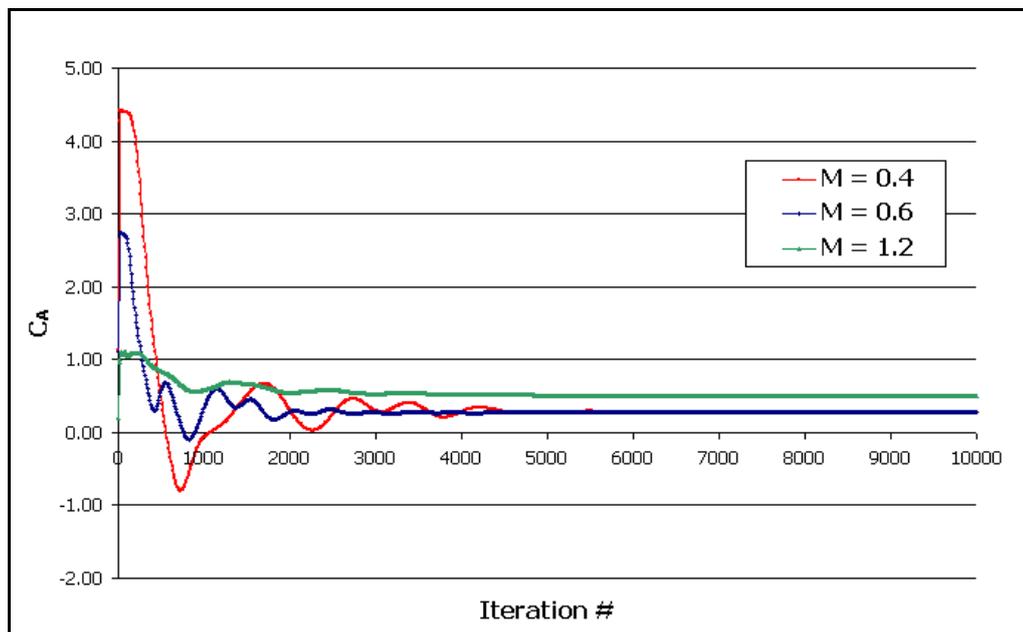


Figure 130. Convergence Histories of the Axial Force Coefficient for Non-Spinning Projectile, Laminar Flow Calculations

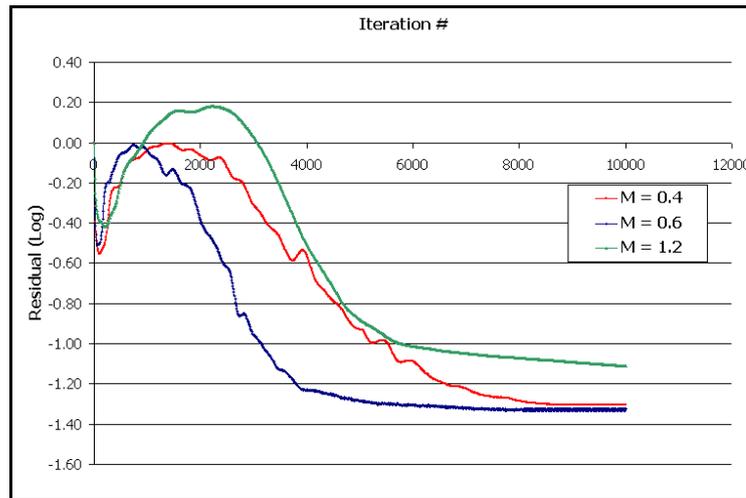


Figure 131. Residual Histories for Non-Spinning Projectile Laminar Flow Calculations

In Figure 132 and Figure 133, the Mach number contours over the projectile are presented in pitch and yaw planes, respectively for $M=0.4$. It can be seen from these figures that flow remains subsonic over the entire field and the computational results are quite symmetric.

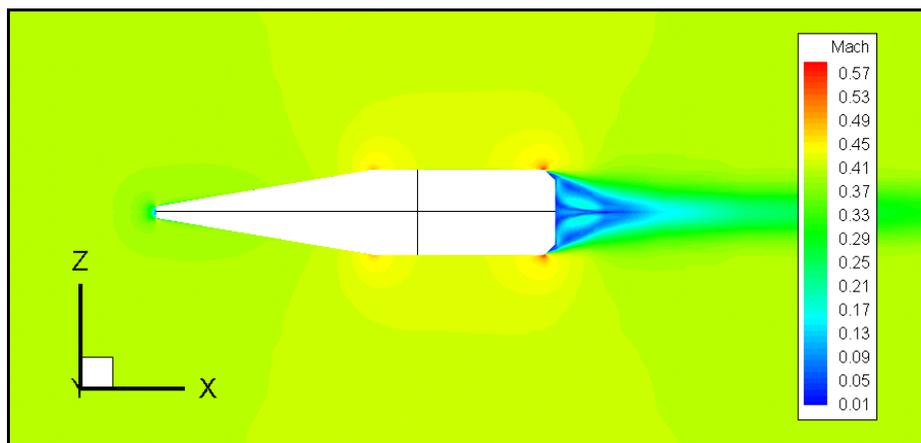


Figure 132. Mach Number Contours for Non-Spinning Projectile Laminar Flow Calculations, pitch plane, $M = 0.4$, $Re = 9.32 \times 10^6$

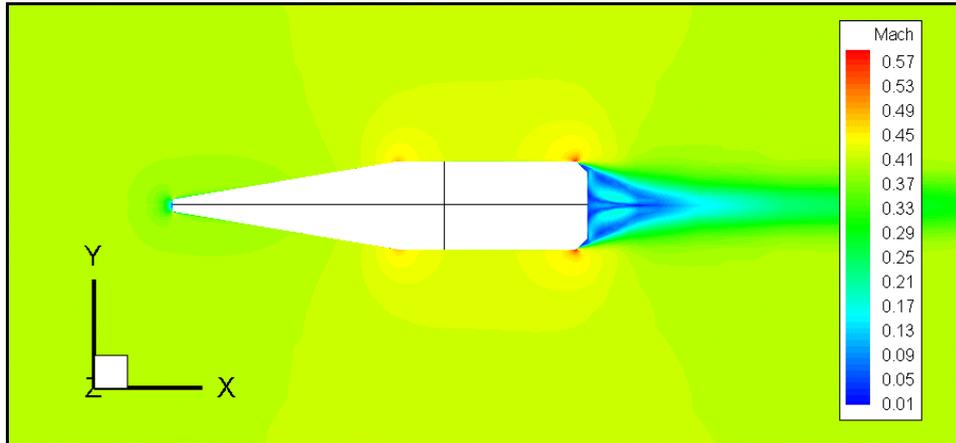


Figure 133. Mach Number Contours for Non-Spinning Projectile Laminar Flow Calculations, yaw plane, $M = 0.4$, $Re = 9.32 \times 10^6$

Recirculation development can be observed at the base the projectile as shown in Figure 134. In Figure 135, the laminar boundary layer development can be observed on the surface of the projectile.

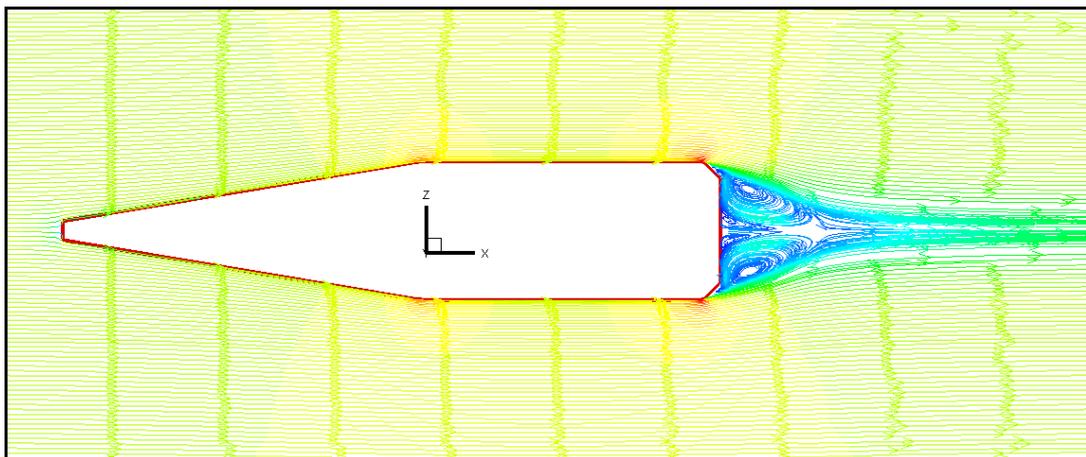


Figure 134. Streamlines Colored by Mach Number for Non-Spinning Projectile Laminar Flow Calculations, $M = 0.4$, $Re = 9.32 \times 10^6$

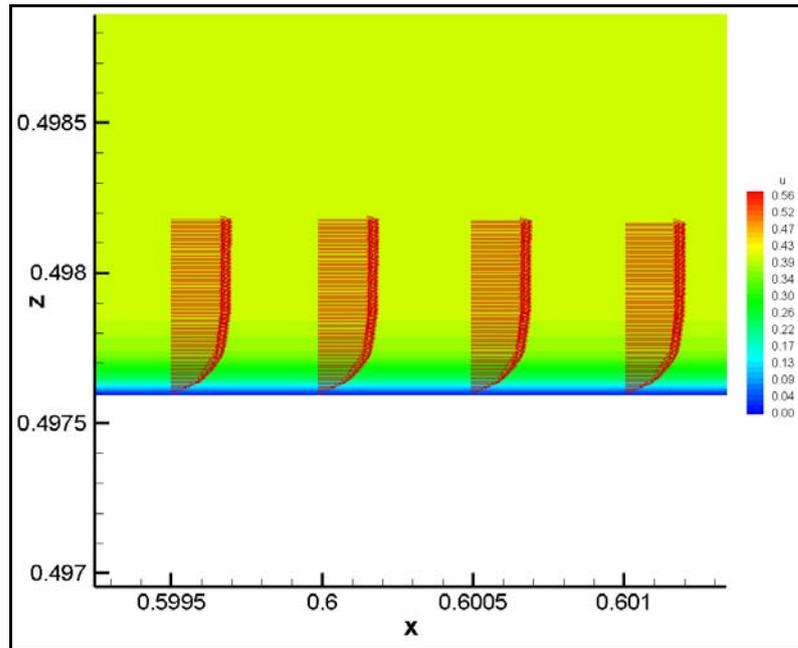


Figure 135. Laminar Boundary Layer Development on the Surface of M910 Projectile, $M = 0.4$, $Re = 9.32 \times 10^6$

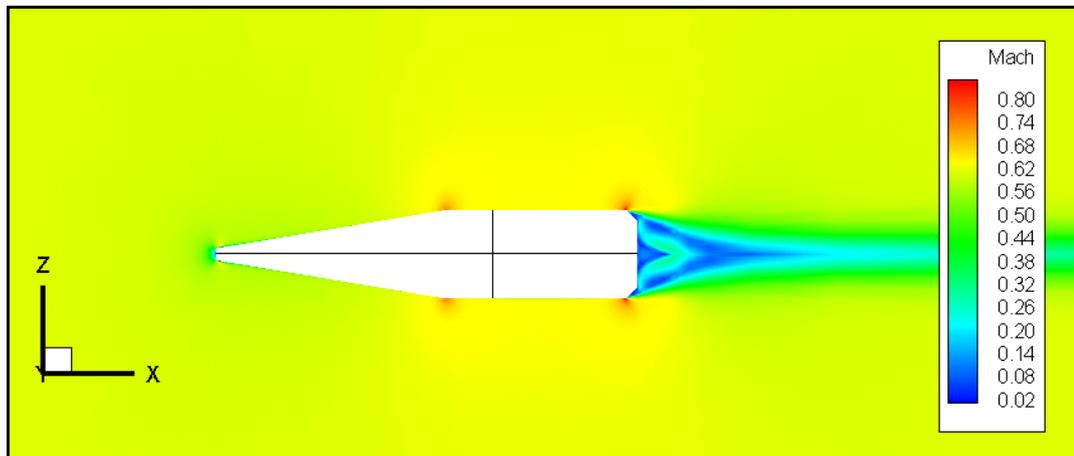


Figure 136. Mach Number Contours for Non-Spinning Projectile Laminar Flow Calculations, pitch plane, $M = 0.6$, $Re = 1.40 \times 10^7$

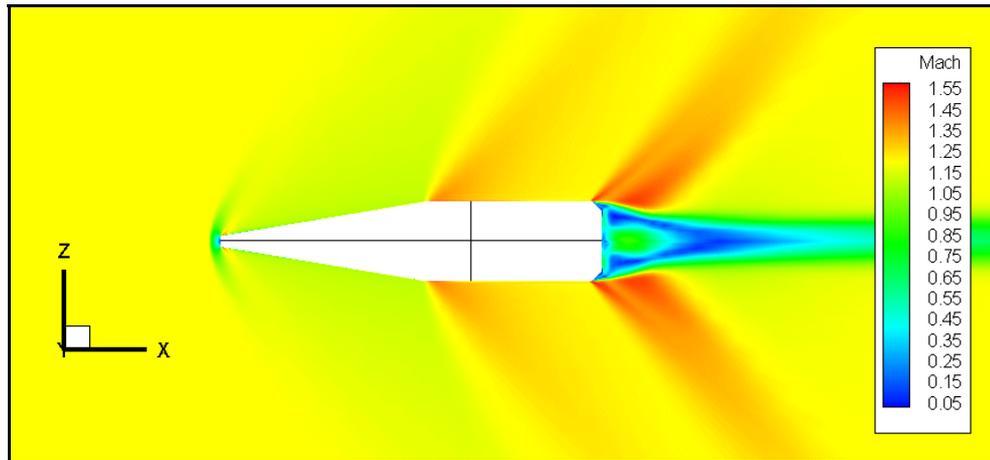


Figure 137. Mach Number Contours for Non-Spinning Projectile Laminar Flow Calculations, pitch plane, $M = 1.2$, $Re = 2.80 \times 10^7$

In Figure 136, Mach number contours for $M=0.6$ is shown at the pitch plane. Flow structure over the projectile is similar to $M=0.4$. Flow remains subsonic over the entire flow field. Recirculation zone at the base of the projectile became larger when compared to $M=0.4$ flow conditions.

For $M=1.2$, development of well defined expansion fans and recompression shocks aft of the ogive-body interface and projectile base can be observed in Figure 137. Furthermore, an oblique shock formation can be observed at the nose of the projectile.

In Figure 138, computed drag force coefficients at zero angle of attack are compared with the results of DeSprito and Heavey [106] and experimental data. Although entire solution domain investigated by DeSprito and Heavey [106] is not examined, obtained results show good agreement with both experimental data and DeSprito and Heavey's results. There is approximately 15% difference between the results of present study and computational results of DeSprito and Heavey that can be explained by the absence of the turbulence model in this level of calculations.

Another important reason may be the inadequate resolution of computational mesh in the boundary layer region. DeSprito and Heavey used computational meshes composed of 1 million to 2 million elements in their turbulent flow calculations where a computational mesh composed of approximately 450,000 elements have been used in this study for laminar flow calculations.

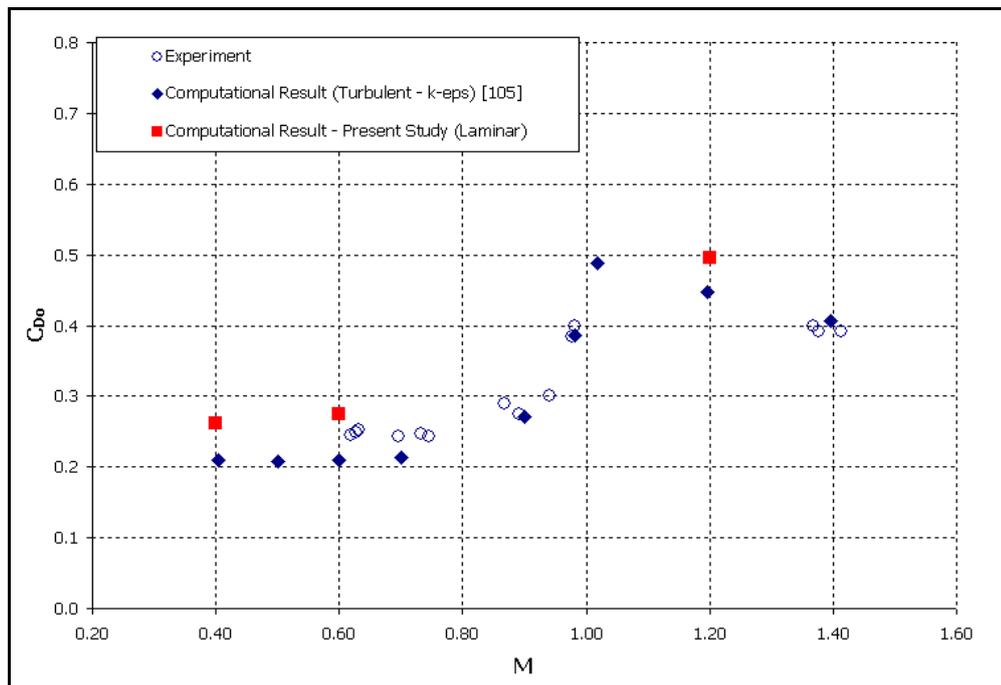


Figure 138. Zero angle of attack Drag Force Coefficient vs Mach Number

5.2. Laminar Flow Solutions over Spinning M910 Projectile

Results of steady-state calculations for laminar flow over non-spinning projectile are used as an initial starting point for the unsteady flow calculations over spinning projectile before introducing turbulence model. Unsteady flow calculations for the spinning projectile are performed at a Mach number of 0.4, a Reynolds number of 9.32×10^6 with zero angle of attack in order to see the accuracy of ALE formulation adapted. It was observed by DeSprito and Heavey [106] that roll moment is independent of angle of attack. Because of this, laminar flow calculations are performed for spinning projectile at zero angle of attack. Effect of incidence in flow will be investigated for turbulent flow calculations around spinning projectile.

Reduced frequency of 0.0852 is taken which is proposed by DeSprito and Heavey [106]. This value corresponds to 23.17 rad/sec rotational speed where DeSprito and Heavey [106] used 1,431 rad/sec for their computations. This difference comes from the non-dimensionalization of the flow equations in the present study. When the computational domain is non-dimensionalized, reference length, which is the diameter of the projectile, became unity. Equation (4.1) states that calculation of the reduced frequency requires the reference length. In order to provide the similitude between two studies, it is a must to rotate the projectile relatively slower than DeSprito and Heavey's [106] study. By this way the tangential velocities for two studies match with each other but there occurs a drastic increase in computational time required for the present study.

In Figure 139, the variation of roll moment coefficient with rotation is presented for laminar flow. Because of the reason stated above, only 6° of rotation can be obtained. But, it seems that roll moment coefficient reached its steady-state value after that amount of rotation. This is expected since there are no fins that will create extra roll moment other than the rotation itself, for this projectile. Oktay E.,

and Akay [30] also introduced 5° of rotation for the calculation of roll damping coefficient in their computations.

All of the computations including the steady-state calculations lasted approximately 25 hours with the use of 20 Pentium IV processors in parallel. It can be observed from this figure that a sudden drop in roll moment coefficient resulting from the beginning of rotation is followed by a monotonically damping behavior of the coefficient with respect to time.

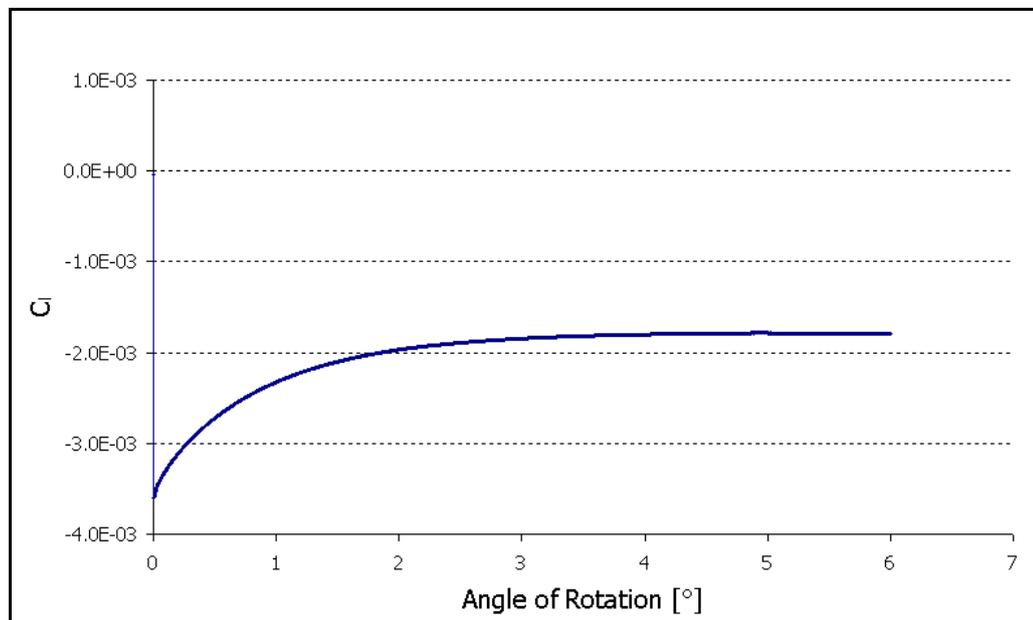


Figure 139. Roll Moment Coefficient Variation with Rotation for Laminar Flow,
 $M = 0.4$, $Re = 9.32 \times 10^6$

When roll damping coefficient is calculated by Equation (4.2) by taking the non-spinning roll moment coefficient as zero, a value of $C_{lp} = -0.021$ is obtained. DeSprito and Heavey [106] presented a value between ($C_{lp} = -0.038$ and

$C_{1p} = -0.029$) for different turbulence models where experimental value is given as approximately $C_{1p} = -0.035$. Main reason for the difference between present study and DeSprito and Heavey's results is the fact that turbulence model is not introduced in this case. It is known that turbulence is a dominant factor that affects the wake flow at the base of the projectile. Also, DeSprito and Heavey's results are for an angle of attack of 3° . Although they stated that for small values of angle of attack, roll damping coefficient is not much affected, this may be a source of the distinction of results of the present study from their results.

Another important factor is the computational mesh again. It was stated before that highly stretched skew elements in the boundary layer seriously deteriorates the flow pattern. Steady-state calculations seemed to be not much influenced from the mesh quality but complicated flow pattern resulting from the rotation of the projectile is directly affected from the inadequate resolution of the computational mesh especially at the wake region.

In Figure 140, Figure 141, Figure 142 and Figure 143, the stream traces on the projectile surface are presented after 0.1° , 1° , 3° and 6° of rotation, respectively. It can be seen from these figures that for 0.1° of rotation, effect of spinning motion is not felt much yet. After 1° of rotation, the streamlines became wrap on the projectile surface in the rotation direction and did not change their behavior much after 3° and 6° of rotation. In Figure 144, Figure 145, Figure 146 and Figure 147, the streamlines at the base of the projectile is presented again after 0.1° , 1° , 3° and 6° of rotation, respectively. It can be observed from these figures that the recirculation in the flow which was calculated in steady-state computations is carried outwards due to the effect of rotation. Again not much change in the flow pattern is observed after 1° of rotation.

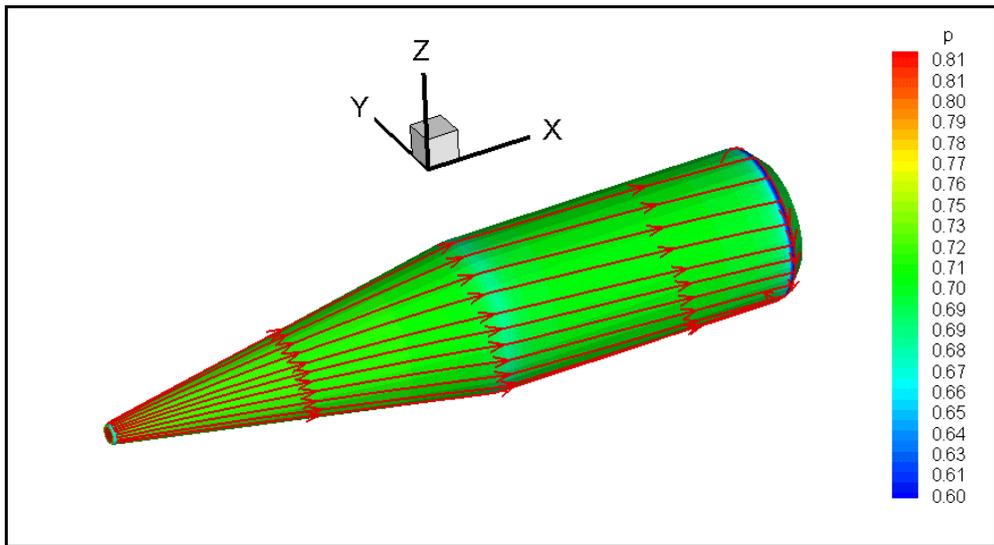


Figure 140. Laminar Flow Streamlines on M910 Projectile, $\theta = 0.1^\circ$

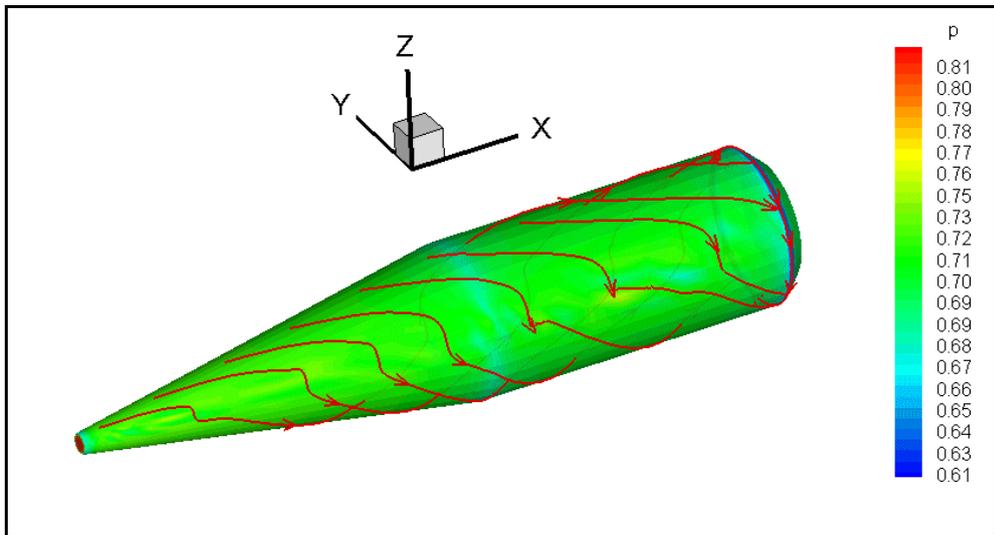


Figure 141. Laminar Flow Streamlines on M910 Projectile, $\theta = 1^\circ$

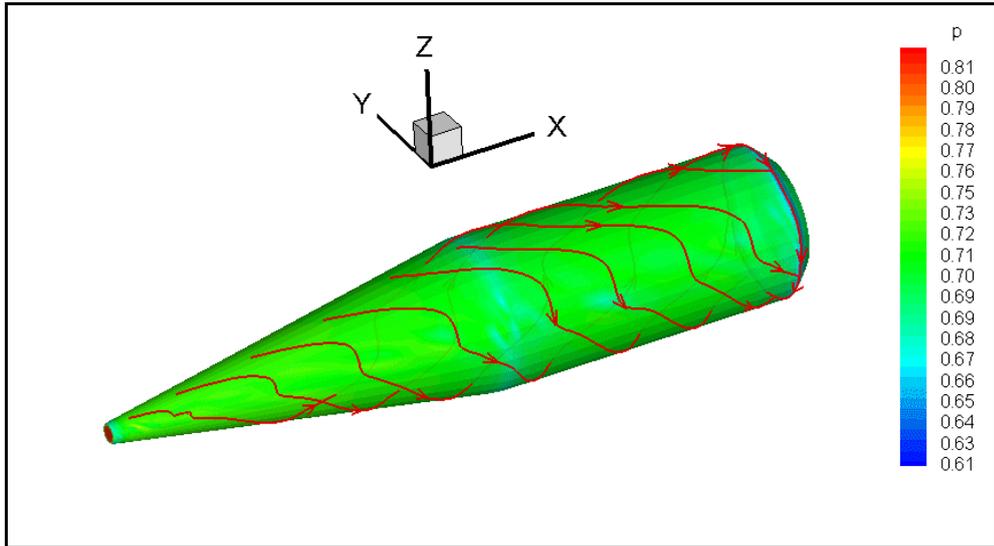


Figure 142. Laminar Flow Streamlines on M910 Projectile, $\theta = 3^\circ$

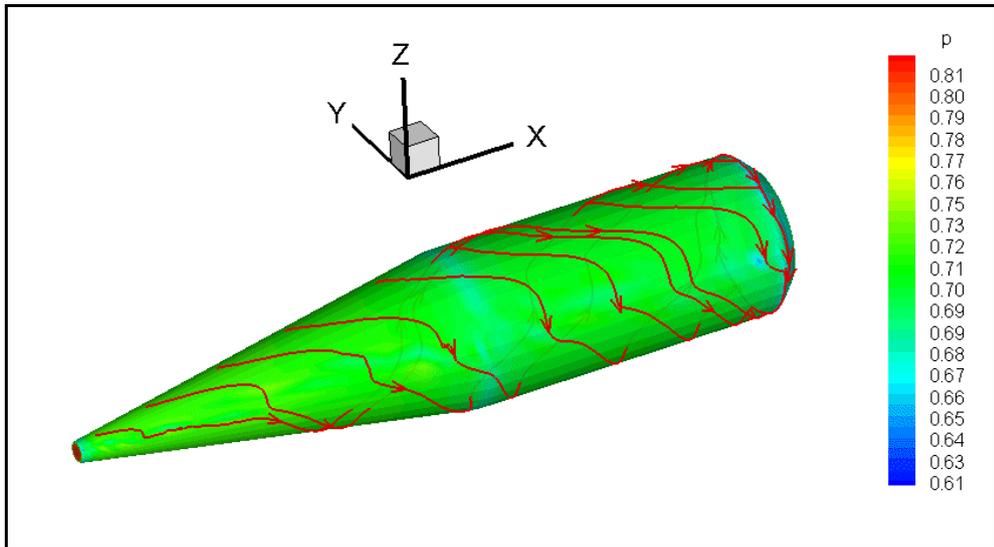


Figure 143. Laminar Flow Streamlines on M910 Projectile, $\theta = 6^\circ$

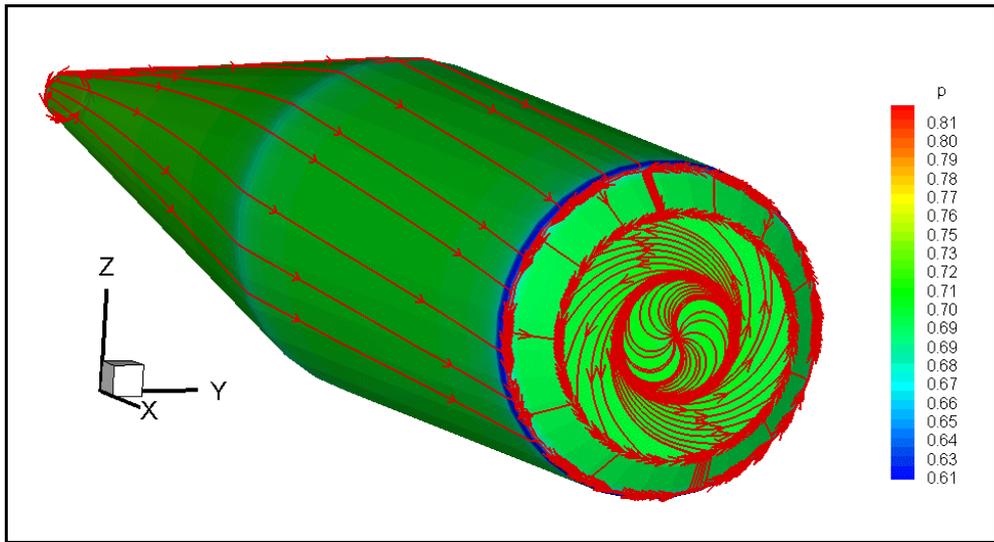


Figure 144. Laminar Flow Streamlines at the Base of M910 Projectile, $\theta = 0.1^\circ$

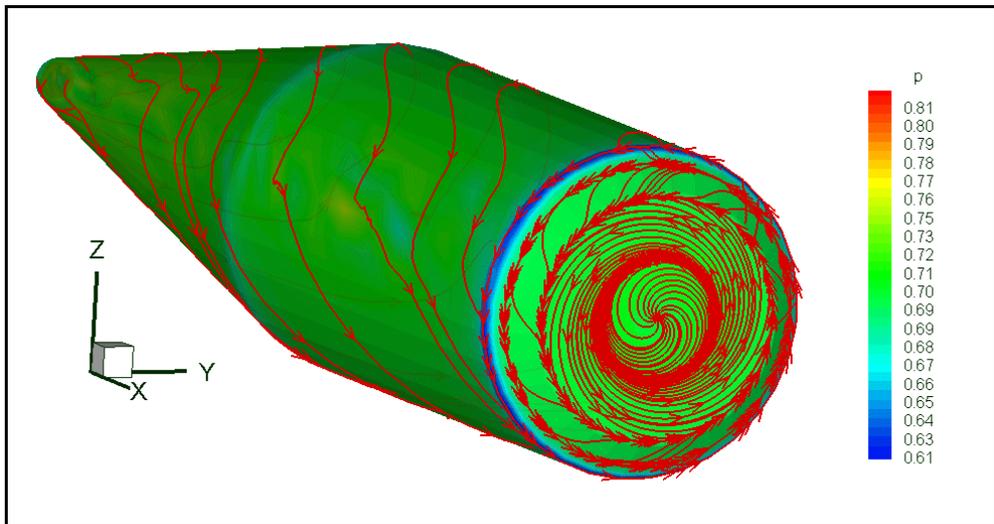


Figure 145. Laminar Flow Streamlines at the Base of M910 Projectile, $\theta = 1^\circ$

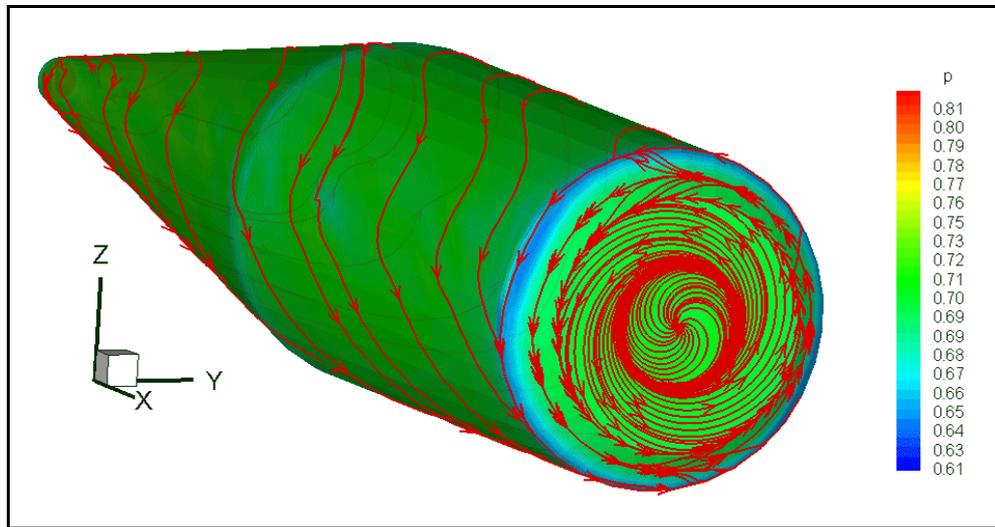


Figure 146. Laminar Flow Streamlines at the Base of M910 Projectile, $\theta = 3^\circ$

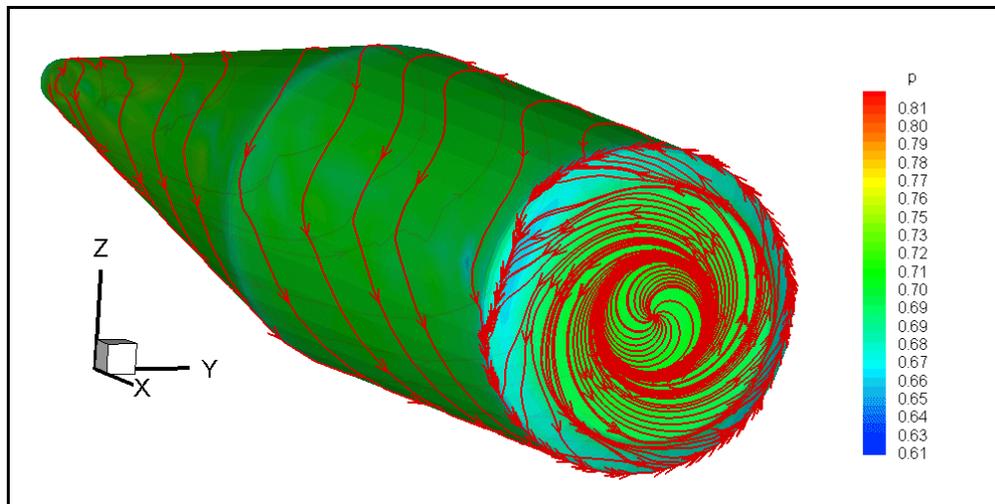


Figure 147. Laminar Flow Streamlines at the Base of M910 Projectile, $\theta = 6^\circ$

5.3. Turbulent Flow Solutions over Non-Spinning M910 Projectile

Steady-state turbulent flow calculations are performed for the flow conditions defined in Table 5 for an angle of attack value of 3° . Results of these computations are used as an initial starting point for unsteady flow computations. Steady-state solutions are also necessary for the calculation of dynamic stability coefficients.

In Figure 148, Figure 149 and Figure 150 the convergence histories for the drag force coefficient, roll moment coefficient and yaw moment coefficient for $M=0.4$, $M=0.6$ and $M=1.2$ turbulent flow conditions are presented, respectively. 10,000 iterations lasted approximately 5 hours for each case using 20 Pentium IV processors working in parallel.

Aerodynamic coefficients are the determining factor in convergence. Remembering this fact, it can be concluded from these figures that steady-state is reached after 10,000 iterations and drag force coefficient became constant for all flow conditions. Roll moment and yaw moment coefficients converged to values very close to zero which is expected for an un-finned projectile. Difference from zero can be explained by numerical dissipation due to the existence of highly stretched elements especially near the wall boundary and at the base of the projectile. The difficulty in unstructured viscous mesh generation for projectile geometries has been stated at the beginning of this chapter and in Section 4.2. Generation of unstructured mesh from the structured mesh causes highly stretched elements especially inside the boundary layer which results in generation of numerical dissipation in solutions.

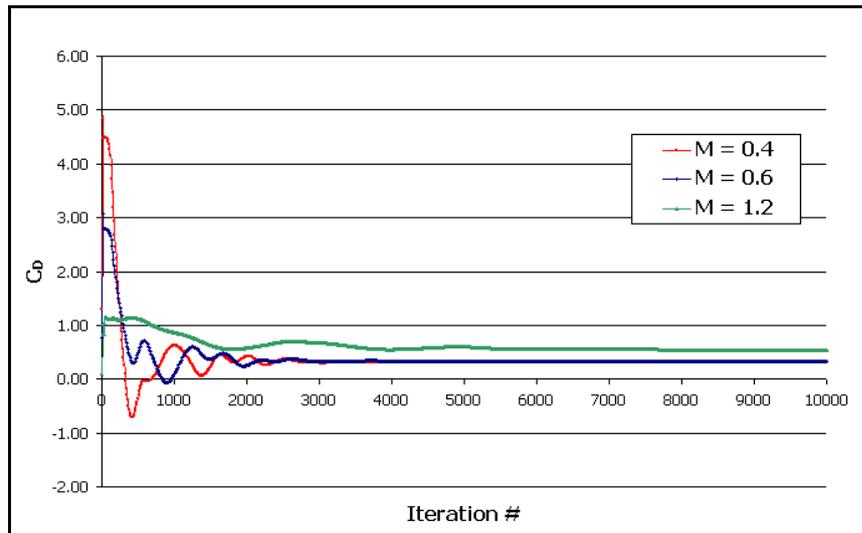


Figure 148. Convergence History of the Drag Force Coefficient for Non-Spinning Projectile, Turbulent Flow Calculations

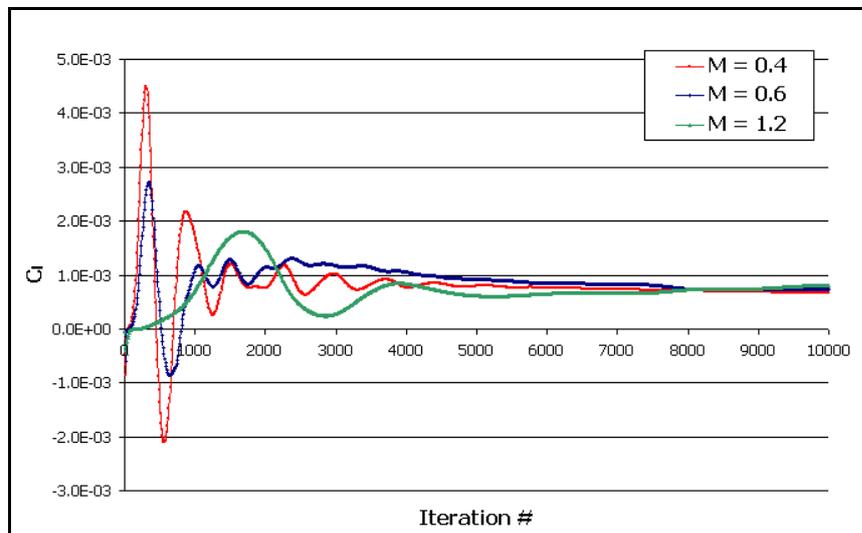


Figure 149. Convergence History of the Roll Moment Coefficient for Non-Spinning Projectile, Turbulent Flow Calculations

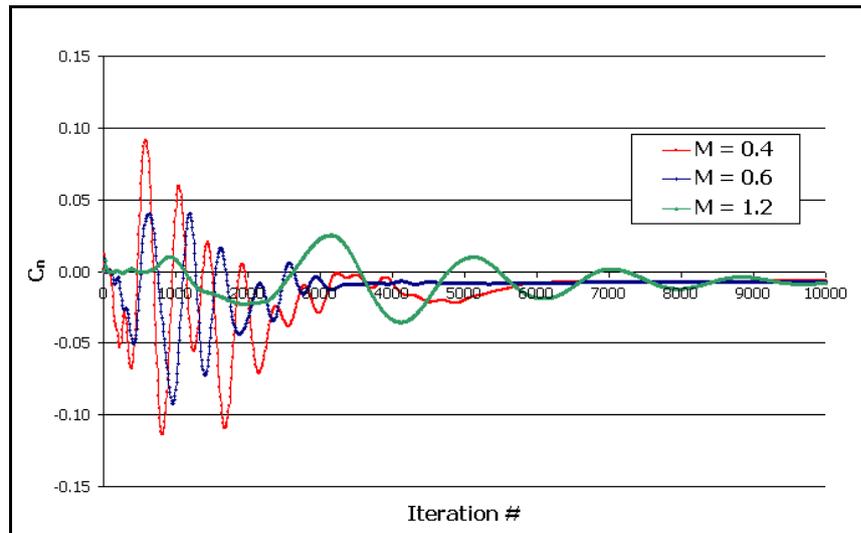


Figure 150. Convergence History of the Yaw Moment Coefficient for Non-Spinning Projectile, Turbulent Flow Calculations

In Figure 151, variation of Mach number contours over the projectile in pitch plane is presented for $M=0.4$. Flow pattern seems similar to laminar flow results except the recirculation zone at the base of the projectile. Recirculation zone became larger at the base due to the diffusive effect of turbulence. Effect of angle of attack is also observable especially at the base and the nose of the projectile. At the upper side of the projectile, which does not face the flow with incidence, the boundary layer came out to be thicker.

In Figure 152, variation of non-dimensional turbulent viscosity over the projectile in pitch plane is shown. Generation of high level of turbulence can be observed at the base of the projectile. Turbulent viscosity increases dramatically at the base due to the existence of large velocity gradients caused by recirculation. Formation of turbulent viscosity on the surface of the projectile especially near the nose where velocity gradients are high due to incidence in the flow is observable in

Figure 153 in which contour scales is reduced in order to see the turbulence generation on the projectile surface.

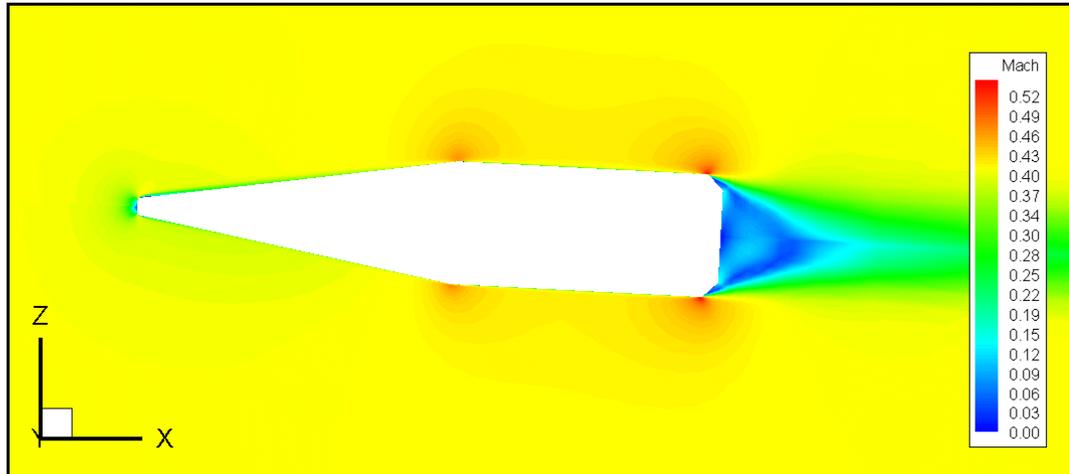


Figure 151. Mach Number Contours for Non-Spinning Projectile, Turbulent Flow Calculations, pitch plane, $M = 0.4$, $Re = 9.32 \times 10^6$, $\alpha = 3^\circ$

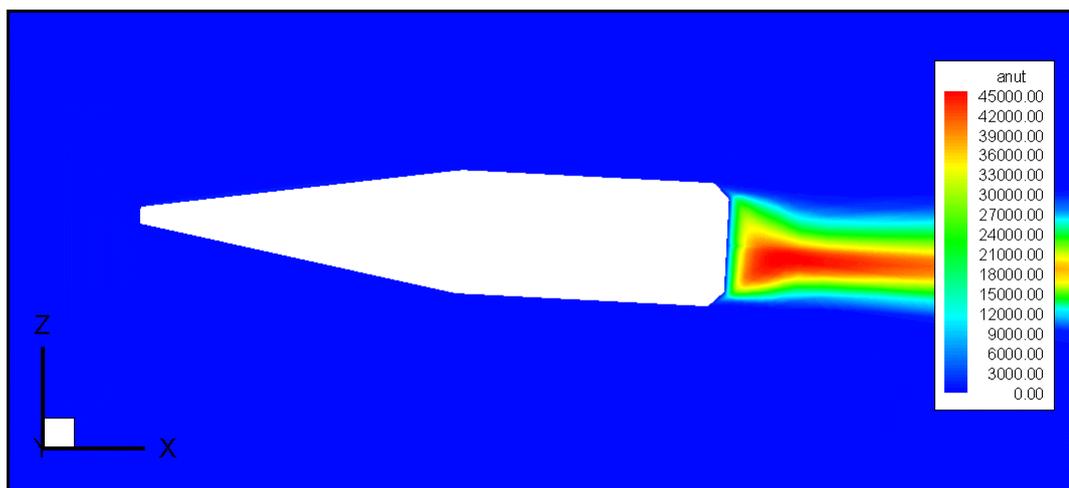


Figure 152. Turbulent Viscosity Contours for Non-Spinning Projectile, pitch plane, $M = 0.4$, $Re = 9.32 \times 10^6$

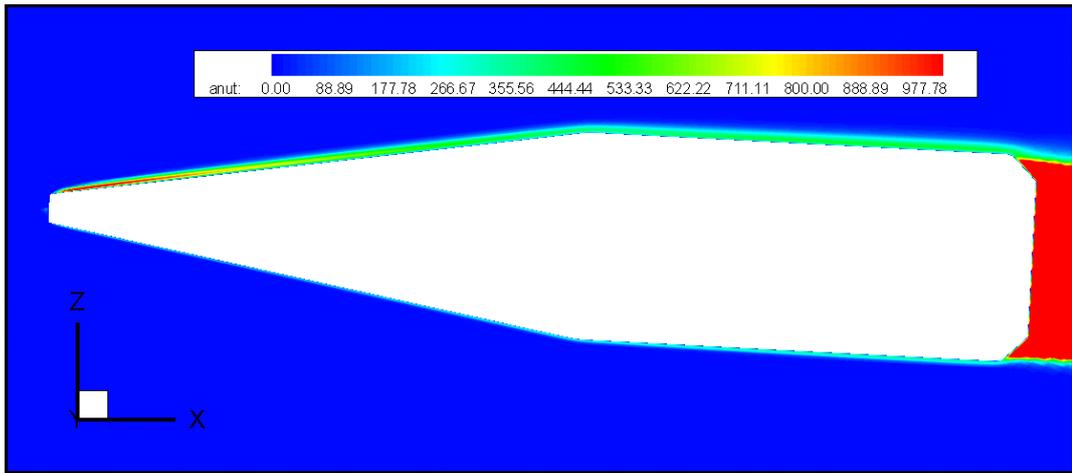


Figure 153. Turbulent Viscosity Contours for Non-Spinning Projectile, pitch plane, $M = 0.4$, $Re = 9.32 \times 10^6$, $\alpha = 3^\circ$, Reduced Contours

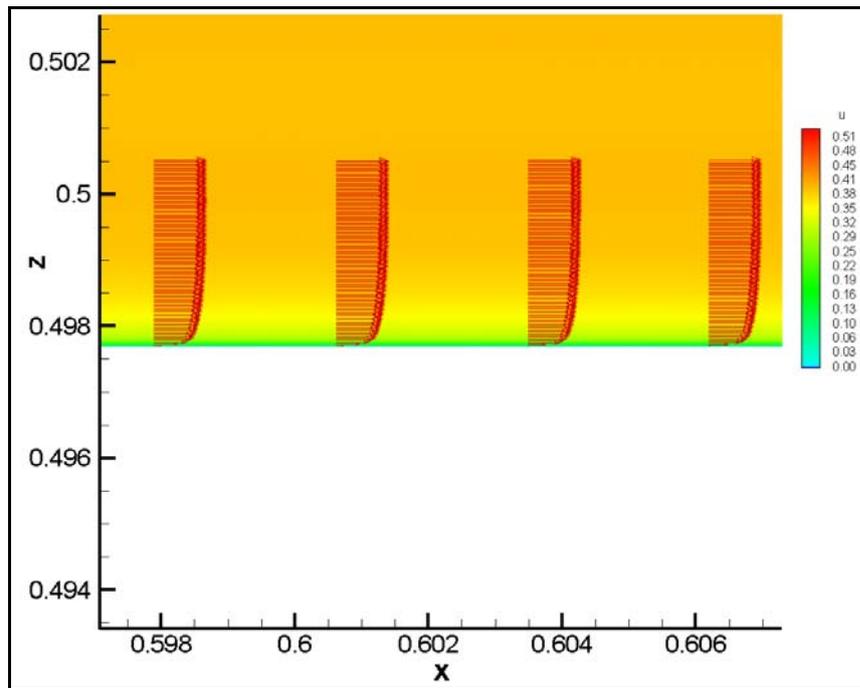


Figure 154. Turbulent Boundary Layer Development on the Surface of M910 Projectile, $M = 0.4$, $Re = 9.32 \times 10^6$

In Figure 154, formation of turbulent boundary layer on the surface of the projectile is presented. Figure 155 shows the variation of the turbulent viscosity in the boundary layer. Turbulent boundary layer profile and the behavior of the variation of turbulent viscosity seem to be as expected. Comparison of laminar and turbulent boundary layer developments is shown in Figure 156. The effect of the turbulent viscosity can be observed in this figure. Difference between the laminar and turbulent boundary layer thicknesses can be observed from this figure.

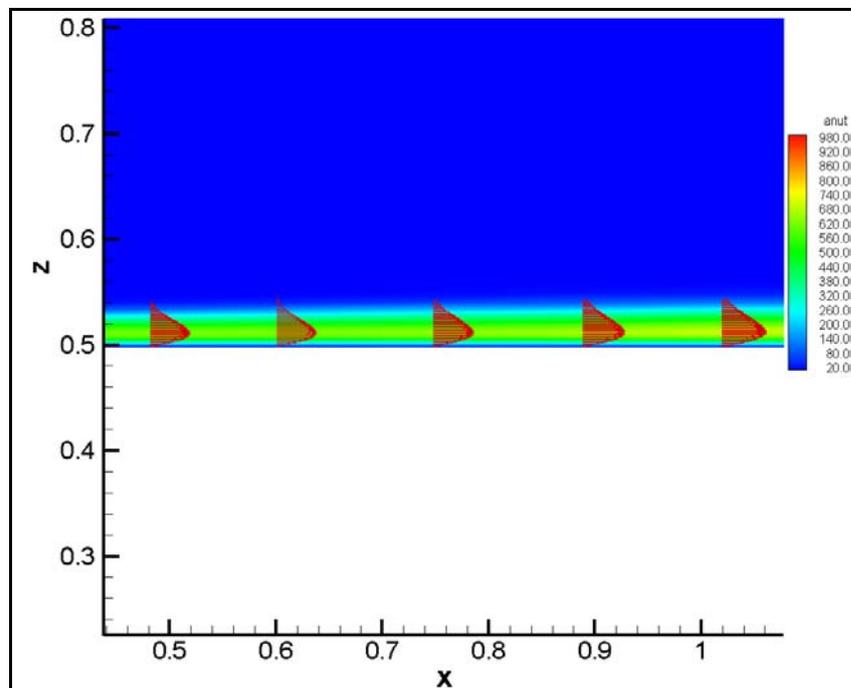


Figure 155. Variation of Turbulent Viscosity Variable on the Surface of M910 Projectile, $M = 0.4$, $Re = 9.32 \times 10^6$

In Figure 157, Mach number contours for $M=0.6$ is shown at the pitch plane. Flow structure over the projectile is similar to $M=0.4$. Effect of angle of attack is

also observable especially at the base and the nose of the projectile. Flow remains subsonic over the entire flow field. Development of the turbulent boundary layer on the projectile surface can be observed.

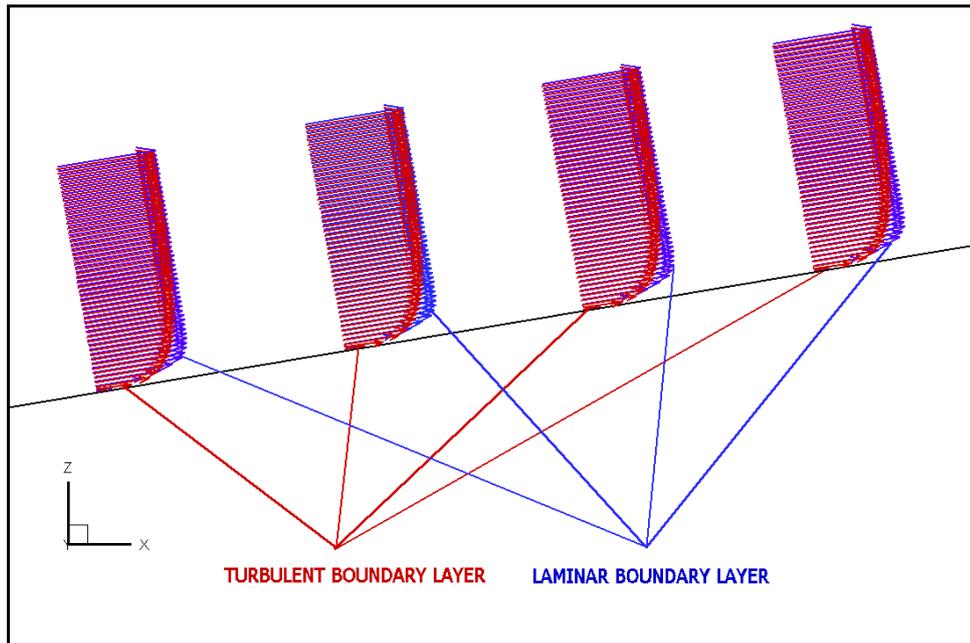


Figure 156. Comparison of Laminar and Turbulent Boundary Layers over M910 Projectile, $M = 0.4$, $Re = 9.32 \times 10^6$

In Figure 158, Mach number contours for $M=1.2$ is shown at the pitch plane. Flow structure over the projectile is similar to laminar flow computations. Development of expansion fans and recompression shocks aft of the ogive-body interface and projectile base can be observed. An oblique shock formation at the nose is also observable.

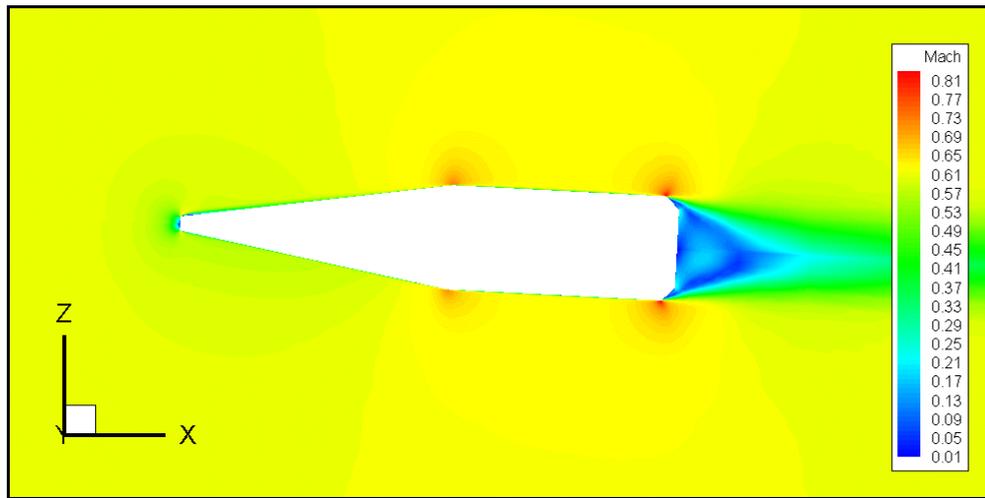


Figure 157. Mach Number Contours for Non-Spinning Projectile, Turbulent Flow Calculations, pitch plane, $M = 0.6$, $Re = 1.40 \times 10^7$

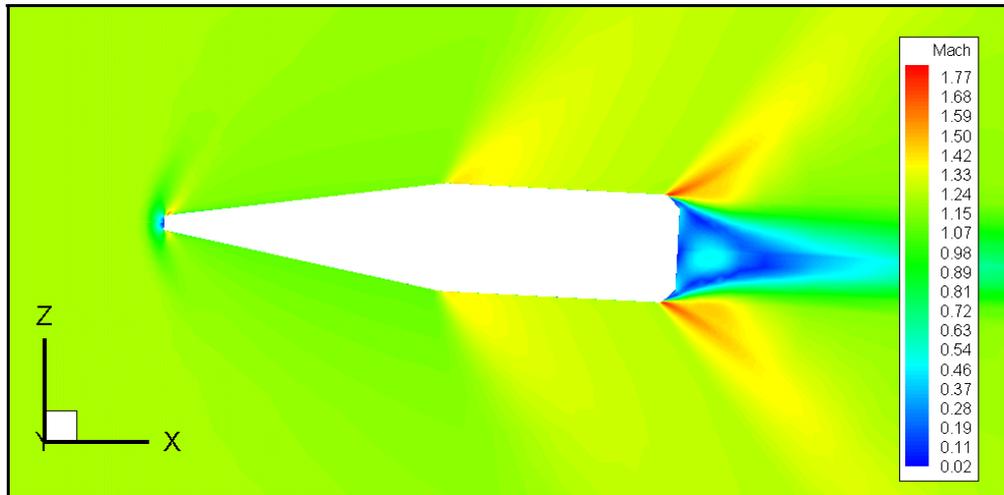


Figure 158. Mach Number Contours for Non-Spinning Projectile, Turbulent Flow Calculations, pitch plane, $M = 1.2$, $Re = 2.80 \times 10^7$

In Figure 159, computed drag force coefficients at angle of attack of 3° are compared with the results of DeSprito and Heavey [106] and experimental data. Obtained results show good agreement especially with experimental data and DeSprito and Heavey's results with detached-eddy simulation (DES) turbulence model. DeSprito and Heavey stated that DES calculations can be considered as the most accurate model of the turbulence models that they had investigated. DeSprito and Heavey stated that use of Reynolds Averaged Navier-Stokes (RANS) formulation for steady-state and unsteady flow calculations, is adequate to predict the aerodynamic coefficients except the Magnus moment. Differences from the experimental data can again be explained by the inadequate mesh stretching especially inside the boundary layer used in the present study. Resolving the viscous boundary layer is critical for predicting aerodynamic coefficients. It must also be noted here that the effect of the wall roughness, which will increase the skin friction, is not introduced to the turbulence model in this study.

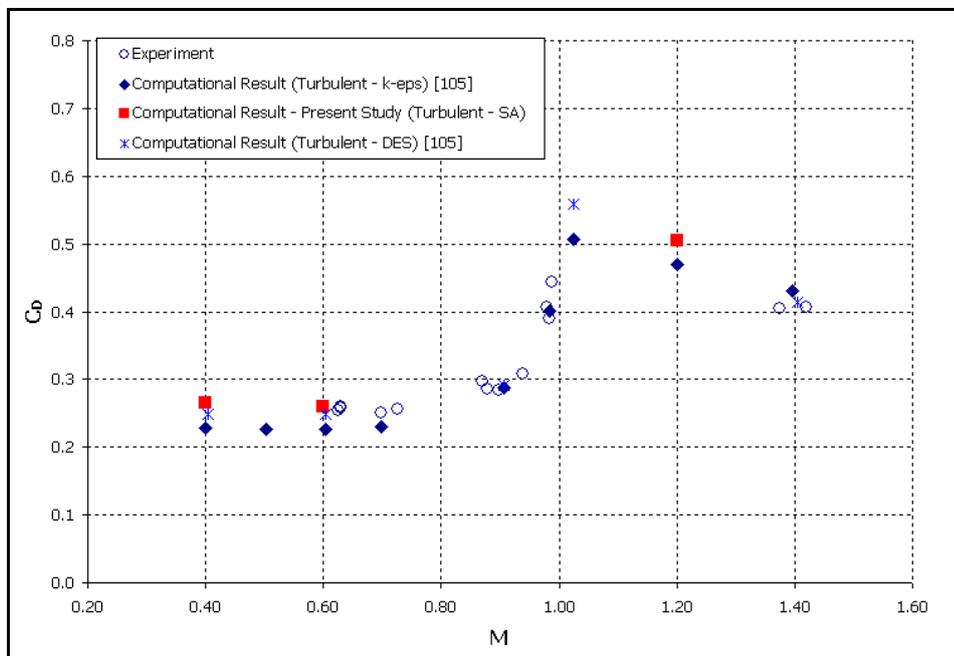


Figure 159. Drag Force Coefficient at 3° angle of attack vs Mach Number

5.4. Turbulent Flow Solutions over Spinning M910 Projectile

Results of steady-state calculations for turbulent flow over non-spinning projectile are used as an initial starting point for the unsteady flow calculations over spinning projectile. Unsteady flow calculations for the spinning projectile are performed for the flow conditions defined in Table 5 for an angle of attack value of 3° . Reduced frequency of 0.0852 is taken which is proposed by DeSprito and Heavey [106].

Roll damping coefficient is calculated using Equation (4.2) taking the value of roll moment coefficient for steady-state computations to be equal to zero. In Figure 160 the variation of roll damping moment coefficient with Mach number is presented. Results came out to be qualitatively in agreement with DeSprito and Heavey's results. But, both present results and DeSprito and Heavey's results are quantitatively apart from the experimental data. Especially for supersonic case result obtained from the present study came out to be quite apart from experimental data. For supersonic flow conditions interaction between shock formation and turbulent boundary layer presents a difficult phenomenon to predict especially for unsteady rotating computations. Inadequate quality of the computational mesh especially at the aft of the ogive-body interface and projectile base directly affects the computational results for supersonic flow conditions.

DeSprito and Heavey stated that RANS computations are adequate to calculate the roll damping coefficient within %15 of experimental data. For the present study, this difference from experimental data can be explained with high stretching ratio in computational mesh especially near the wall boundaries. It is known that resolving the viscous boundary layer is critical for predicting the Magnus and roll damping moments. Because of the mesh generation problem stated before it was impossible to obtain elements with low stretching ratio especially in the vicinity of the projectile surface.

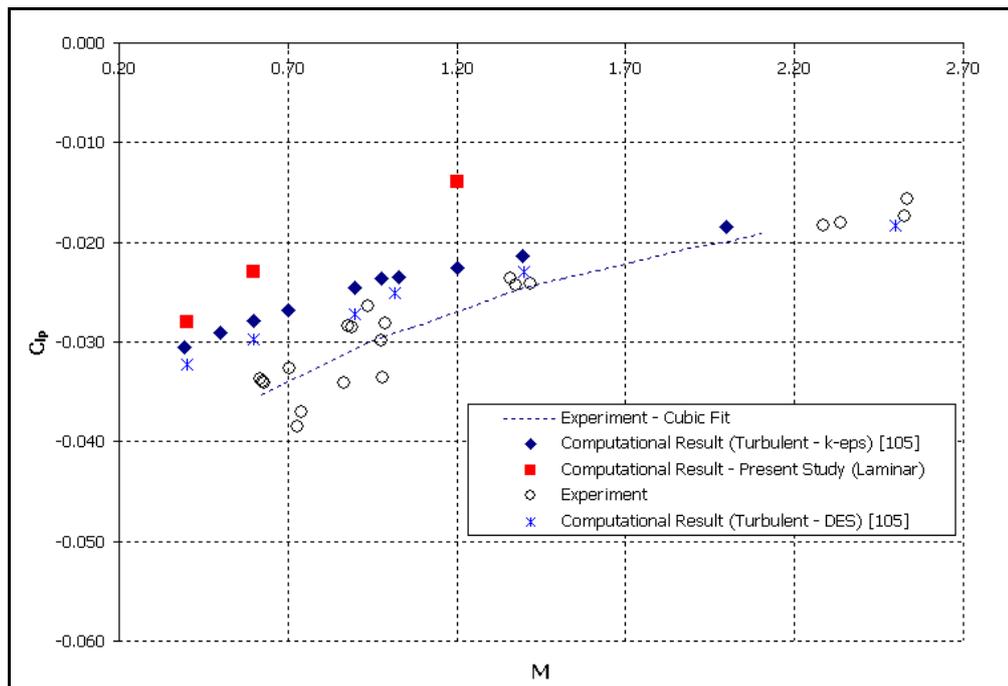


Figure 160. Roll Damping Coefficient vs Mach Number at $\alpha = 3^\circ$

In the developed code, preconditioning of the equations is not implemented. Although usually needed for low Mach number ($M < 0.2$) flows, preconditioning improves the convergence characteristics, due to the low speed flow in projectile wake region where all of the convergence problems observed in this study occurred at. This may be another reason for the quantitative difference of the computational results from the experimental data. But, in general it can be said that the trend of the computed roll damping coefficient is in agreement with experimental data.

As it was explained in Section 5.2, non-dimensionalization of flow and turbulence model equations and use of highly stretched elements near the wall boundary resulted in very small time step values which drastically increased the computational time required for the rotation of the projectile. For all of the cases studied, approximately 3° of rotation could be simulated. Computation time for this

amount of rotation came out to be approximately 15 hours for each flow condition. Roll moment and yaw moment coefficients seemed to reach a constant value after that amount of rotation but further rotation may improve the results.

Magnus moment coefficient derivative is calculated using the formula given below [106];

$$\begin{aligned}
 C_{np\alpha} &= \frac{\partial \left(\frac{\partial C_l}{\partial p} \right)}{\partial \alpha} = \frac{\frac{\partial C_l}{\partial p} \Big|_{\alpha=\alpha} - \frac{\partial C_l}{\partial p} \Big|_{\alpha=0^\circ}}{\sin(\alpha)} \\
 &= \frac{\left\{ \frac{(C_l|_{k=k} - C_l|_{k=0}) \Big|_{\alpha=\alpha}}{k} \right\} - \left\{ \frac{(C_l|_{k=k} - C_l|_{k=0}) \Big|_{\alpha=0^\circ}}{k} \right\}}{\sin(\alpha)}
 \end{aligned} \tag{5.1}$$

Equation (5.1) takes the form given below noting that yaw moment coefficient for a symmetric geometry must be equal to zero for steady-state computations and for unsteady computations with zero angle of attack [106];

$$C_{np\alpha} = \frac{(C_l|_{k=k, \alpha=\alpha})}{k \cdot \sin(\alpha)} \tag{5.2}$$

In Figure 161 the variation of Magnus moment coefficient derivative with Mach number is presented. Both present results and DeSprito and Heavey's results with turbulence models other than DES came out to be apart from the experimental data especially for low Mach numbers. DeSprito and Heavey stated that DES is necessary to predict the Magnus moment especially in the subsonic and transonic flow regimes. This situation can also be observed in the figure. None of the RANS turbulence models used by DeSprito and Heavey could predict the decreasing trend of Magnus moment below $M > 2.0$.

This drastic difference between results of the present study and DES results and experimental data indicates that the oscillatory wake flow has an impact on the upstream side forces. This effect is mainly observed in the subsonic and transonic flow regimes. Time accurate nature of DES can simulate fully unsteady oscillatory wake flow behind the projectile but as DeSprito and Heavey stated there is a drastic increased cost in computational time required for DES.

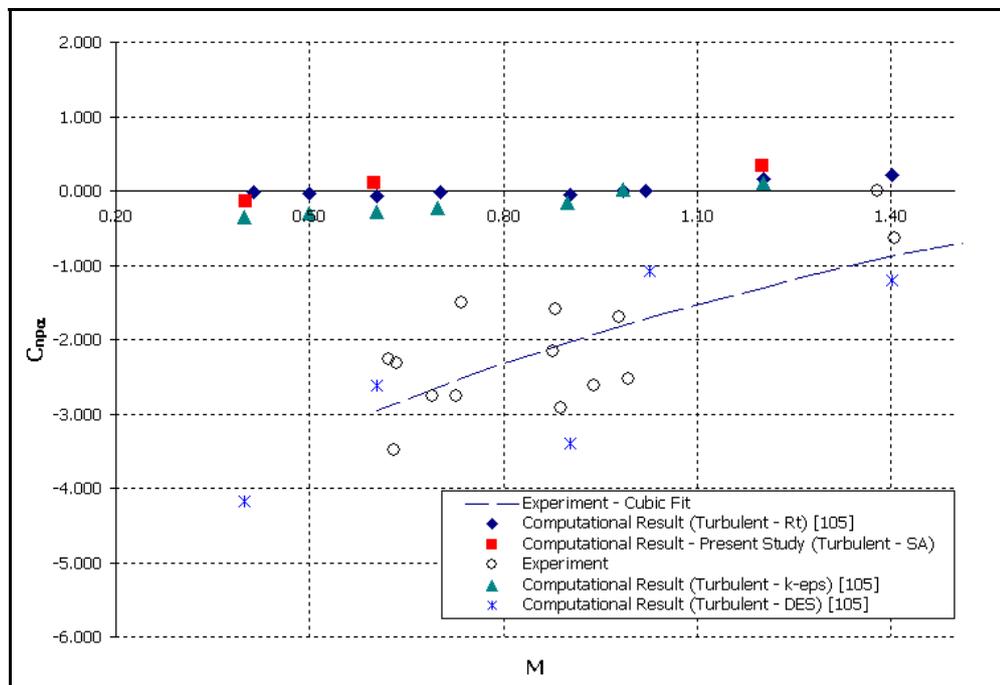


Figure 161. Magnus Moment Coefficient Derivative vs Mach Number at $\alpha = 3^\circ$

DeSprito and Heavey offers an increase in the mesh density in the projectile wake flow region in order to better predict the Magnus moment coefficient. Their work on mesh dependency showed that inadequate mesh density in the boundary layer region results in large changes in Magnus moment coefficient derivative. At the base of the projectile, denser mesh generation by the mesh generation method

described in Appendix C causes drastic increase in the mesh size and deteriorates the mesh quality inside the boundary layer. Use of structured hexahedral elements inside the boundary layer region and unstructured tetrahedral elements up to the far field boundary, that is use of hybrid grids, seems to eliminate this problem in computations.

Since Magnus force and moment coefficients are relatively small coefficients, especially at low angle of attack, they are affected by the transient flow in the projectile wake especially in the subsonic and transonic flow regimes which make them rather difficult to be predicted accurately.

In general, it can be said that the choice of turbulence model has the largest effect on the aerodynamic coefficients, primarily the Magnus moment and roll damping. Spalart-Allmaras turbulence model used in this study gave comparable results with other turbulence models studied by DeSprito and Heavey. But, especially for the prediction of Magnus moment coefficient derivative use of time accurate turbulence models with higher accuracy (DES, Large-Eddy Simulation, etc.) seems to be absolutely necessary.

In Figure 162, Figure 163, Figure 164, the stream traces on the projectile surface are presented after 0.1° , 1° and 2° of rotation for $M=0.6$ flow condition, respectively. It can be seen from these figures that for 0.1° of rotation, first impact of the rotation is felt on the ogive part of the projectile. There occurs a resistance of the turbulent boundary layer to the rotation at the very beginning. After 1° of rotation, the streamlines became to wrap on the projectile surface in the rotation direction and did not change their behavior much for 2° of rotation. This situation is also observed in the variation of aerodynamic coefficients which became almost constant approximately after 1° of rotation. Effect of angle of attack can be observed at the nose of the projectile where flow makes recirculation.

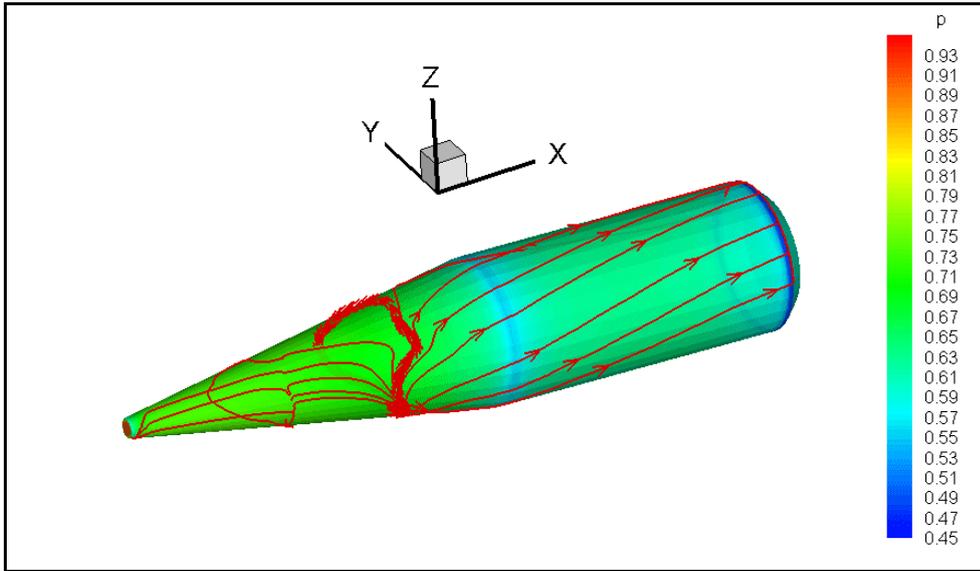


Figure 162. Turbulent Flow Streamlines on M910 Projectile, $\theta = 0.1^\circ$

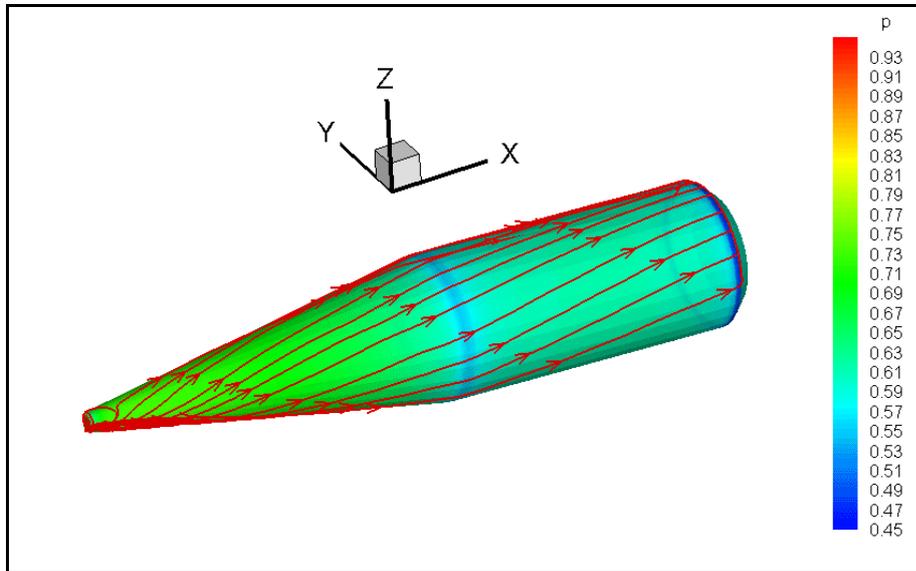


Figure 163. Turbulent Flow Streamlines on M910 Projectile, $\theta = 1^\circ$

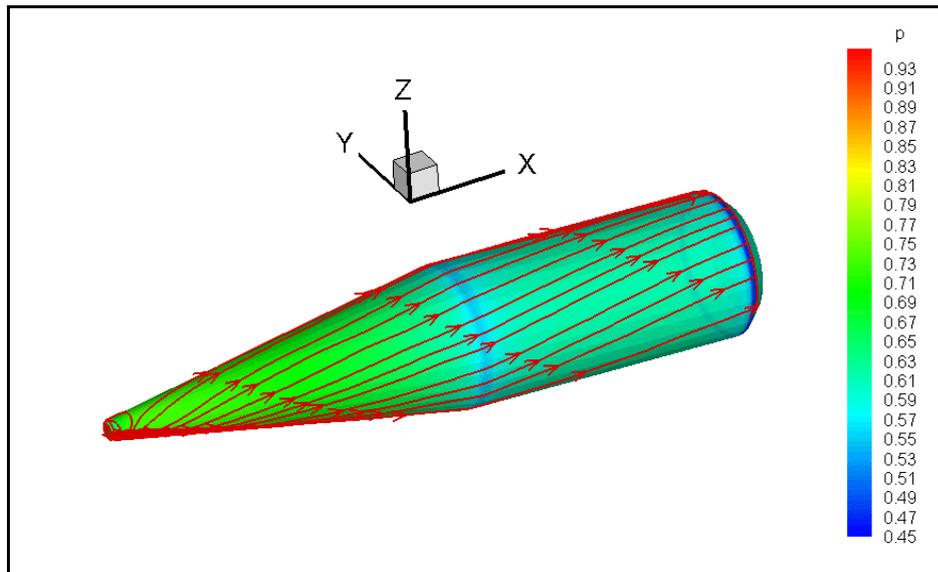


Figure 164. Turbulent Flow Streamlines on M910 Projectile, $\theta = 2^\circ$

CHAPTER 6

CONCLUSION & DISCUSSION

The main objective of this study is to develop a turbulent flow solver in order to evaluate the aerodynamic dynamic stability derivative coefficients for rotating missile configurations.

The study presented here involves the generation of a three-dimensional finite volume solver which uses Roe's upwind flux differencing scheme for spatial and Runge-Kutta explicit multistage time stepping scheme for temporal discretization on unstructured meshes for the unsteady solution of external viscous flow around rotating bodies. A parallel algorithm based on PVM message passing libraries has been adapted in order to run the solver on distributed memory computers.

Arbitrary Lagrangian Eulerian (ALE) formulation has been adapted to the solver for the simulation of the rotation of the body. A detailed literature survey on the implementation of rotation into governing flow equations has been performed and ALE formulation has been selected because of its simplicity in terms of application and its physically meaningful nature. Detailed derivation of the ALE formulation on Euler equations has been performed. Investigations and derivation studies showed that the grid motion only affects the convective variables from a physical point of view and the grid motion only modifies the eigenvalues of the Jacobian from a mathematical point of view. So implementation of ALE formulation into full Navier-Stokes equations came out to be the same as it is for Euler equations. Therefore eigenvalues of the Euler equations in ALE form has

been derived and introduced into the Roe's upwind flux differencing scheme directly.

Body rotation has been simply performed by rotating the entire computational domain including the body of the projectile by means of rotation matrices. Therefore the computational domain has not been deformed. This has eliminated the need of conserving geometric quantities. By ALE formulation it is possible to solve flow around a body having any kind of motion if necessary translation and rotation matrices are known.

Diffusive fluxes have been evaluated simply by using divergence theorem to calculate the velocity gradients. This is the simplest way to calculate the gradients but it works well as shown in the verification and validation studies.

The Spalart Allmaras turbulence model which is a widely used model is selected as the turbulence model. It is a one equation turbulence model and easy to implement into the solver when compared to two-equation models. As it was stated in related chapters 2-D and 3-D turbulent flow calculations predicts acceptable results. The main drawback of Spalart-Allmaras turbulence model came out to be the calculation of the minimum wall distances. As it will be explained later, especially in 3-D, generation of highly stretched tetrahedral elements is unavoidable with the mesh generation tools available. Since wall distance is more important near the wall boundaries, this situation created some problems. It is observed that attention has to be given for the calculation of minimum wall distance at near wall boundaries.

First, three-dimensional Euler solver is verified on two different finned missile configurations. Predictions are in good agreement with experimental data especially for low angle of attack values. The ALE formulation and the grid movement algorithms in 3-D are tested for inviscid flows over a rotating missile configuration. Results came out to be satisfactory for inviscid flow calculations.

Inviscid flow calculations have shown that spatial discretization scheme has the desired accuracy level. It could not be possible to perform enough verification studies on three-dimensional viscous calculations because of the mesh generation problem that was explained in the text.

The developed Navier-Stokes code has been validated in two-dimensions by performing steady state calculations for viscous (laminar and turbulent) flow over flat plate, NACA0012 airfoil and NLR 7301 airfoil with trailing edge flap. Steady-state calculations in two dimensions gave quite satisfactory results. The results showed that the spatial discretization scheme, viscous flux calculation algorithms and Spalart-Allmaras turbulence model implementation have the desired order of accuracy.

Then ALE formulation has been verified on a rapidly pitching NACA0012 airfoil for 2-D viscous flow. Comparison of the computational data for pitching airfoil with experimental data has showed that ALE formulation that has been adapted gave quite acceptable results. When the unsteady solutions are animated it has been seen that the solver is capable of observing the separation bubbles and vortices generated especially behind the airfoil. Even the vortex shedding behind the airfoil has been simulated.

Afterwards viscous flux computation algorithms and implementation of Spalart-Allmaras turbulence model in 3-D are validated by examining flow over flat plate problem. Formations of laminar and turbulent boundary layer profiles on the flat plate are accurately obtained. Comparison with analytical and approximate solutions showed that 3-D turbulent Navier-Stokes solver developed gives accurate and reasonable results.

Flow over a spinning projectile without fins is studied as a final test case. Selection of such a projectile was chosen in order to obtain high resolution boundary layer mesh around the rotating body. As it was explained before, mesh

generation has created significant problems especially for three dimensional viscous calculations. A post processing program has been written that takes the structured viscous mesh generated by a commercial mesh generation program, and converts it into unstructured one. By this way, it was possible to obtain required mesh density for a projectile without fins in order to observe the turbulent boundary layer on the projectile surface. But the main drawback of this procedure is that when high resolution, dense mesh is generated on the projectile surface, this mesh has to be extended up to far field because of the nature of the structured volume mesh generation. If it is noted that, the hexahedral elements of the structured elements are divided into six tetrahedral elements for unstructured mesh generation, it is unavoidable to obtain enormous size of computational mesh. Besides, there occur highly stretched tetrahedral elements especially near to the wall boundary which directly affects the numerical accuracy.

3-D viscous algorithms, ALE formulation and Spalart-Allmaras turbulence model implementation have been validated using the viscous computational mesh generated for M910 un-finned projectile. Results came out to be quite acceptable except for Magnus moment coefficient derivative. Turbulent boundary layer development on the projectile surface has been observed for steady-state calculations. Computations for flow over spinning projectile also gave qualitatively good results when compared with experimental data expect Magnus moment coefficient derivative. Inadequate quality of computational mesh in the vicinity of wall boundary because of the reasons explained above caused distinction from experimental data.

In literature it is observed that time accurate turbulence models such as Large-Eddy Simulation or Detached Eddy Simulation, came out to be the most suitable method to predict Magnus moment coefficient derivative especially for subsonic and transonic flow regime. Use of RANS turbulence models fails in prediction of this coefficient because of the fully unsteady oscillatory wake flow at the base of the projectile which has an impact on the upstream side forces. But, in

general trend of the computational results obtained in this study are in good agreement with both experimental data and results of previous studies.

Mesh generation problem has restricted the verification studies of three dimensional viscous algorithms. It seems that it will be unavoidable to arrange the solver developed so that it can handle any type of element. In other words, it seems it is necessary to make the solver handle hybrid meshes.

Especially for a better prediction of Magnus moment coefficient derivative, use of time accurate (Detached-Eddy Simulation, Large-Eddy Simulation) are strongly advised rather than the use of RANS turbulence models.

The technique applied in this study for the computation of minimum wall distance values for Spalart-Allmaras turbulence model, APPENDIX D, gives accurate and acceptable results but when the number of control volumes in the computational domain increase, computational time required for this method becomes enormously high. Use of fast marching level set methods to solve the Eikonal equation, which was developed by Sethian [107], to calculate the closest points to a surface may be utilized in order to minimize the computational time requirements.

As an overall conclusion, it can be said that the solver developed is able of predicting viscous external flows around rotating bodies. It is possible to evaluate the necessary dynamic stability characteristics with this solver. As a future work it can be stated that especially for viscous flow calculations capability of handling hybrid grids instead of full unstructured grids is absolutely necessary. In addition, use of implicit time stepping algorithms is also advised strongly. Stability limits of the explicit time stepping algorithms directly affects the computational time necessary to obtain satisfactory results. Use of implicit schemes will reduce the computational cost especially of unsteady moving body calculations.

REFERENCES

1. Weinacht, P., and Sturek, W. B., "*Computation of the Roll Characteristics of a Finned Projectile*", Journal of Spacecraft and Rockets, Vol.33, No.6, pp. 769-775, November-December, 1996
2. M. Pechier, P. Guillen, and R. Cayzac, "*Magnus Effect over Finned Projectiles*", Journal of Spacecraft and Rockets, Vol.38, pp. 542-549, July-August, 2001
3. Elliott, J., and Peraire, J., "*Practical 3-D Aerodynamic Design & Optimization Using Unstructured Meshes*", AIAA Journal, Vol.35, No.9, pp. 1479-1485, September, 1997
4. Pirzadeh, S.Z., and Frink, N.T., "*Assessment of the Unstructured Grid Software TetrUSS for Drag Prediction of the DLR-F4 Configuration*", AIAA 2002-0839, January, 2002
5. "Missile Aerodynamics", RTO-Meeting Proceeding 5, November, 1998
6. Sturek, W. B., Nietubicz, C. J., Sahu, J., and Weinacht, P., "*Applications of Computational Fluid Dynamics to the Aerodynamics of Army Projectiles*", Journal of Spacecraft and Rockets, Vol.31, No.2, pp. 186-199, March-April, 1994
7. Baçoğlu, Osman, "*Three Dimensional Aerodynamic Analysis Of Missiles By A Panel Method*", Ms. Thesis, METU, 2002
8. Anderson, J. D., *Introduction to Flight*, 3rd Ed., McGraw-Hill.,1989

9. Frink, N. T., and Pirzadeh, S. Z., "*Tetrahedral Finite-Volume Solutions to the Navier-Stokes Equations on Complex Configurations*", International Journal for Numerical Methods in Fluids, Vol.31, pp. 175-187, 1999
10. Frink, N. T., "*Assessment of an Unstructures-Grid Method for Predicting 3-D Turbulent Viscous Flows*", AIAA 96-0292, 1996
11. Jameson, A., Schmidt, W., and Turkel, E., "*Numerical Solution of the Euler Equations by Finite Volume Methods using Runge-Kutta Time Stepping Schemes*", AIAA 81-1259, 1981
12. Hirsch, C., *Numerical Computation of Internal and External Flows, Vol. 1&2*, John Wiley & Sons, 1989
13. Dick E., "*An Introduction to Computational Fluid Dynamics*", Von Karman Institute for Fluid Dynamics, LS 1989-02, January, 1989
14. Frink, N.T., "*Upwind Scheme for Solving the Euler Equations on Unstructured Tetrahedral Meshes*", AIAA Journal, Vol.30, No.1, pp. 70-77, January,1992
15. Frink, N.T., Parikh, P., and Pirzadeh, S., "*A Fast Upwind Solver for the Euler Equations on Three-Dimensional Unstructured Meshes*", AIAA 91-0102, 1991
16. Frink, N.T., "*Recent Progress Toward a Three-Dimensional Unstructured Navier-Stokes Flow Solver*", AIAA 94-0061, 1994
17. Frink, N. T. "*Tetrahedral Unstructured Navier-Stokes Method for Turbulent Flows*", AIAA Journal, Vol.36, No.11, pp. 1975-1982, November, 1998

18. Frink, N. T., “*Assessment of an Unstructures-Grid Method for Predicting 3-D Turbulent Viscous Flows*”, AIAA 96-0292, 1996
19. Wang, Q., Massey, S. J., Abdol-Hamid, K. S., and Frink, N. T., “*Solving Navier-Stokes Equations with Advanced Turbulence Models on Three-Dimensional Unstructured Grids*”, AIAA 99-0156, 1999
20. Frink, N. T., Pirzadeh, S. Z., and Parikh, P. , “*An Unstructured-Grid Software System for Solving Complex Aerodynamic Problems*”, NASA-CP-3291, May, 1995
21. Batina, J. T., “*Implicit Upwind Solution Algorithms for Three-Dimensional Unstructured Meshes*”, AIAA Journal, Vol.31, No.5, pp. 801-805, May, 1993
22. Batina, J. T., “*Unsteady Euler Algorithm with Unstructured Dynamic Mesh for Complex-Aircraft Aerodynamic Analysis*”, AIAA Journal, Vol.29, No.3, pp. 327-333, March, 1991
23. Yaniv, S., “*Navier-Stokes Calculations for Rotating Configurations: Implementation for Rockets*”, Journal of Spacecraft and Rockets, Vol.33, No.5, pp. 756-758
24. Barakos, G., and Drikakis, D., “*An Implicit Unfactored Method for Unsteady Turbulent Compressible Flows with Moving Boundaries*”, Computers & Fluids, Vol.28, pp. 899-922, 1999
25. Barth, T. J., “*Aspects of Unstructured Grids and Finite-Volume Solvers for the Euler and Navier-Stokes Equations*”, NASA Ames Research Center, Moffet Field, Ca., USA, Von Karman Institute for Fluid Dynamics, Lecture Series 1994-05, Computational Fluid Dynamics, March 21-25, 1994

26. Venkatakrishnan V., *“Implicit Schemes and Parallel Computing in Unstructured Grid CFD”*, ICASE Report No. 95-28 NASA Langley Research Center , 1995
27. Jothiprasad, G., Mavripilis, D. J., and Caughey, D. A., *“Higher Order Time Integration Schemes for the Unsteady Navier-Stokes Equations on Unstructured Meshes”*, NASA/CR-2002-211967 ICASE Report No. 2002-44, 2002
28. Oktay, E., Akay, H.U., and Uzun, A., *“Parallelized Three-Dimensional Unstructured Euler Solver for Unsteady Aerodynamics”*, Journal of Aircraft, Vol.40, No.2, pp. 348-354, March-April, 2003
29. Oktay, E., Akay, H.U., and Uzun, A., *“A Parallelized 3D Unstructured Euler Solver for Unsteady Aerodynamics”*, AIAA-2002-0107, January, 2002
30. Oktay E., and Akay, H. U., *“CFD Predictions of Dynamic Derivatives for Missiles”*, AIAA 2002-0276, January, 2002
31. Dadone, A., and Grossman, B., *“Characteristic-Based, Rotated Upwind Scheme for the Euler Equations”*, AIAA Journal, Vol.30, No.9, pp. 2219-2226, September, 1992
32. Liou, M. S., *“Ten Years in the Making-AUSM-family”*, AIAA 2001-2521, June, 2001
33. Pan, D., and Cheng, J. C., *“Upwind Finite-Volume Navier-Stokes Computations on Unstructured Triangular Meshes”*, AIAA Journal, Vol.31, No.9, pp. 1618-1625, September, 1993

34. Anderson, W. K., and Bonhaus D. L., "*An Implicit Upwind Algorithm for Computing Turbulent Flows on Unstructured Grids*", Computers Fluids, Vol.23, No.1, pp. 1-21, 1994

35. Bonhaus D.L., "*An Upwind Multigrid Method for Solving Viscous Flows on Unstructured Triangular Meshes*", Ms. Thesis, University of Cincinnati, August, 1993

36. Bermudez, A. And, Vazquez, M.E., "*Upwind Methods for Hyperbolic Conservation Laws with Source Terms*", Computers Fluids, Vol.23, No.8, pp. 1049-1071, 1994

37. Roe, P. L., "*Discrete Models for the Numerical Analysis of Time-Dependent Multidimensional Gas Dynamics*", Journal of Computational Physics, Vol.63, pp. 458-476, 1986

38. Roe, P. L., "*Approximate Riemann Solvers, Parameter Vectors and Difference Schemes*", Journal of Computational Physics, Vol.43, pp. 357-372, 1981

39. Lacor, C., and Hirsch, C., "*Genuinely Upwind Algorithms for the Multidimensional Euler Equations*", AIAA Journal, Vol.30, No.1, pp. 55-63, January, 1992

40. Steger J.L., and Warming, R.F., "*Flux Vector Splitting of the Inviscid Gas Dynamic Equations with Applications to Finite-Difference Methods*", Journal of Computational Physics, Vol.40, pp. 263-293, 1981

41. Van Leer, B., "*Flux Vector Splitting for the Euler Equations*", Lecture Notes in Physics, Vol.170, Springer-Verlag, Berlin, pp. 507-512, 1982

42. Godunov, S.K., “*A Difference Method for the Numerical Computation of Discontinuous Solutions of Hydrodynamic Equations*”, Math Sbornik, Vol.47, pp. 271-306, 1959
43. Osher, S., and Solomon, F., “*Upwind Schemes for Hyperbolic Systems of Conservation Laws*”, Mathematics of Computation, Vol.38, No.158, pp. 339-377, 1982
44. Toro, E. F., *Riemann Solvers and Numerical Methods for Fluid Dynamics*, 2nd Edition, Springer, 1999
45. Harten, A., “*High Resolution Schemes for Hyperbolic Conservation Laws*”, Journal of Computational Physics, Vol.49, pp. 357-393, 1983
46. Tomaro, R. F., and Strang, W. Z., “*An Implicit Algorithm for Solving Time Dependent Flows on Unstructured Grids*”, AIAA 97-0333, 1997
47. Mavriplis, D.J., “*Solution of the Two-Dimensional Euler Equations on Unstructured Triangular Meshes*”, Ms. Thesis, Princeton University, June, 1987
48. Zhao, Y., and Tai, C. H., “*Higher-Order Characteristics-Based Method for Incompressible Flow Computation on Unstructured Grids*”, AIAA Journal, Vol.39, No.7, pp. 1280-1287, July, 2001
49. Bruner, C., W., S., “*Parallelization of the Euler Equations on Unstructured Grids*”, Ph.D. Thesis, Virginia Polytechnic Institute and State University, May, 1996

50. Uzun Ali, "*Parallel Computations of Unsteady Euler Equations on Dynamically Deforming Unstructured Grids*", Ms. Thesis, Purdue University, August, 1999
51. Anderson, W. K., "*Grid Generation and Solution Method for Euler Equations on Unstructured Grids*", NASA-TM-4295, April, 1992
52. Mineck, R.E., "*Application of an Unstructured Grid Navier-Stokes Solver to a Generic Helicopter Body*", NASA/TM-1999-209510, August, 1999
53. Koobus, B., Farhat, C., and Tran, H., "*Computation of Unsteady Viscous Flows Around Moving Bodies Using the k - ϵ Turbulence Model on Unstructured Dynamic Grids*", *Comp. Methods Appl. Mech. Eng.*, Vol.190, pp. 1441-1466, 2000
54. Haselbacher, A., McGuirk, J.J., and Page, G.J., "*Finite Volume Discretization Aspects for Viscous Flows on Mixed Unstructured Grids*", *AIAA Journal*, Vol.37, No.2, pp. 177-184, February, 1999
55. Ghizawi, N., and Abdallah, S., "*Parallel Processing Scheme for the Navier-Stokes Equations, Part 1: Scheme Development*", *AIAA Journal*, Vol.36, No.11, pp. 2013-2019, November, 1998
56. Ghizawi, N., and Abdallah, S., "*Parallel Processing Scheme for the Navier-Stokes Equations, Part 2: Parallel Implementation*", *AIAA Journal*, Vol.36, No.11, pp. 2020-2026, November, 1998
57. Akay, H. U., Oktay, E., Li, Z., and He, X., "*Parallel Computing for Aeroelasticity Problems*", AIAA-2003-3511, 2003

58. Ramamurti, R., and Löhner, R., “*A Parallel Implicit Incompressible Flow Solver Using Unstructured Meshes*”, *Computers & Fluids*, Vol.25, No.2, pp. 119-132, 1996
59. Parikh, P., “*Application of a Scalable, Parallel, Unstructured-Grid-Based Navier-Stokes Solver*”, AIAA 2001-2584, June, 2001
60. Luke, E., A., “*A Rule-Based Specification System for Computational Fluid Dynamics*”, Ph.D. Thesis, Mississippi State University, December, 1999
61. Dolean, V., and Lanteri, S., “*A Domain Decomposition Approach to Finite Volume Solutions of the Euler Equations on Unstructured Triangular Meshes*”, *International Journal for Numerical Methods in Fluids*, Vol.37, pp. 625-656, 2001
62. Hu, Y. F., Emerson, D. R., Ashworth, W., Maguire, K. C. F., and Blake, R. J., “*Parallelizing FLITE3D - A Multigrid Finite Element Euler Solver*”, *International Journal for Numerical Methods in Fluids*, Vol.35, pp. 809-827, 2001
63. Karypis, G. and Kumar, V., “*METIS - A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices, Version 3.0.*”, Manual, University of Minnesota and Army HPC Research Center, 1997
64. Champigny, P., and D'Espiney, P., “*Numerical Simulation of Vortex Flows around Missile Configurations*”, RTO-AVT, May, 2001
65. Wang, Q., Massey, S. J., and Abdol-Hamid, K. S., “*Implementation of Advanced Two Equation Turbulence Models in the USM3D Unstructured Flow Solvers*”, NASA CR-2000-210102, April, 2000

66. Wilcox, D. C., *Turbulence Modeling for CFD*, DCW Industries, Inc., La Canada, California, 1998
67. Baldwin, B. S. and Lomax, H., “ *Thin Layer Approximation and Algebraic Model for Separated Turbulent Flows*”, AIAA Paper 78-257, Jan. 1978
68. Cebeci, T., and Smith, A. M. O., *Analysis of Turbulent Boundary Layers*, Academic Press, New York, 1974
69. Weinacht, P., Sturek, W. B., and Schiff, L. B., “*Navier-Stokes Predictions of Pitch Damping for Axisymmetric Projectiles*”, Journal of Spacecraft and Rockets, Vol.34, No.6, pp. 753-761, November-December, 1997
70. Park, M. A., and Green, L. L., “*Steady-State Computation of Constant Rotational Rate Dynamic Stability Derivatives*”, AIAA 2000-4321, 2000
71. Kandil, O. A., and Chuang, H. A., “*Unsteady Navier-Stokes Computations Past Oscillating Delta Wing at High Incidence*”, AIAA Journal, Vol. 28, No.9, pp. 1565-1572, September, 1990
72. Nietubicz, C. J., Sturek, W. B., and Heavey, K. R., “*Computations of Projectile Magnus Effect at Transonic Velocities*”, AIAA Journal, Vol.23, No.7, pp. 998-1004, July, 1985
73. Park, M. A., Green, L. L., Montgomery, R. C., and Raney, D. L., “*Determination of Stability and Control Derivatives Using Computational Fluid Dynamics and Automatic Differentiation*”, AIAA 99-3136, 1999
74. Koobus, B., and Farhat, C., “*Second-Order Time-Accurate and Geometrically Conservative Implicit Schemes for Flow Computations on Unstructured*

- Dynamic Meshes*”, Computer Methods in Applied Mechanics & Engineering, Vol.170, pp. 103-129, 1999
75. Lesoinne, M., and Farhat, C., “*Geometric Conservation Laws for Flow Problems with Moving Boundaries and Deformable Meshes, and Their Impact on Aeroelastic Computations*”, Computer Methods in Applied Mechanics & Engineering, Vol.134, pp. 71-90, 1996
 76. Farhat, C., Geuzaine, P., and Grandmont, C., “*The Discrete Geometric Conservation Law and the Nonlinear Stability of ALE Schemes for the Solution of Flow Problems on Moving Grids*”, Journal of Computational Physics, Vol.174, pp. 669-694, 2001
 77. Geuzaine, P., Grandmont, C., and Farhat, C., “*Design and Analysis of ALE Schemes with Provable Second-Order Time-Accuracy for Inviscid and Viscous Flow Simulations*”, Journal of Computational Physics, Vol.191, pp. 206-227, 2003
 78. Guillard, H., and Farhat, C., “*On the Significance of the Geometric Conservation Law for Flow Computations on Moving Meshes*”, Comp. Methods Appl. Mech. Eng., Vol.190, pp. 1467-1482, 2000
 79. Trepanier, J. Y., Reggio, M., Paraschivoiu, M., and Camarero R., “*Unsteady Euler Solutions for Arbitrarily Moving Bodies and Boundaries*”, AIAA Journal, Vol.31, No.10, pp. 1869-1876, October, 1993
 80. Trepanier, J. Y., Reggio, M., Zhang, H., and Camarero R., “*A Finite-Volume Method for the Euler Equations on Arbitrary Lagrangian-Eulerian Grids*”, Computers & Fluids, Vol.20, No.4, pp. 399-409, 1991

81. Zhang, H., Reggio, M., Trepanier, J. Y., and Camarero R., “*Discrete Form of the GCL for Moving Meshes and Its Implementation in CFD Schemes*”, *Computers & Fluids*, Vol.22, No.1, pp.9-23, 1993
82. Michler, C., De Sterck, H., and Deconinck, “*An Arbitrary Lagrangian Eulerian Formulation for Residual Distribution Schemes on Moving Grids*”, *Computers & Fluids*, Vol.32, pp. 59-71, 2003
83. Demirdzic, I., and Peric, M., “*Finite Volume Method for Prediction of Fluid Flow in Arbitrarily Shaped Domains with Moving Boundaries*”, *International Journal for Numerical Methods in Fluids*, Vol.10, pp. 771-790, 1990
84. Thomas, P.D., and Lombard, C.K., “*Geometric Conservation Law and Its Application to Flow Computations on Moving Grids*”, *AIAA Journal*, Vol.17, pp. 1030-1037, 1979
85. Smith, R.W., “*AUSM(ALE): A Geometrically Conservative Arbitrary Lagrangian–Eulerian Flux Splitting Scheme*”, *Journal of Computational Physics*, Vol.150, No.1, pp. 268-286, 1999
86. Khelil, S., B., Guillen, P., Lazareff, M., and Lacau, R., G., “*Numerical Simulation of Roll Induced Moment of Cruciform Tactical Missiles*”, *Aerospace Science Technologies*, Vol.5, pp. 109-124, 2001
87. Moreau, V., “*Arbitrary Lagrangian Eulerian (ALE) Formulation: Application for Euler and Navier-Stokes Equations*”, 1997
88. Sarrate, J., Huerta, A., and Donea, J., “*Arbitrary Lagrangian-Eulerian Formulation for Fluid-Multi Rigid Bodies Interaction Problems*”, *Computational Mechanics: New Trends and Applications*, 1998

89. Sarrate, J., Huerta, A., and Donea, J., “*Arbitrary Lagrangian-Eulerian Formulation for Fluid-Rigid Body Interaction*”, *Comp. Methods Appl. Mech. Eng.*, Vol.190, pp. 3171-3188, 2001
90. Van Haaren, M. J., Stoker, H. C., Van den Boogard, A. H., and Huetink, J., “*The ALE-Method with Triangular Elements: Direct Convection of Integration Point Values*”, *International Journal for Numerical Methods in Engineering*, Vol.49, pp. 697-720, 2000
91. Spalart, P. and Allmaras, S. “*A One-Equation Turbulence Model for Aerodynamic Flows*”, Technical Report AIAA-92-0439, American Institute of Aeronautics and Astronautics, 1992
92. Deck, S., Duveau, P., D’Espiney, P., and Guillen, P. “*Development and Application of Spalart-Allmaras One Equation Turbulence Model to Three-Dimensional Supersonic Complex Configurations*”, *Aerospace Science and Technology*, Vol.6, pp. 171-183, 2002
93. Ashford, G.A., and Powell, K.G., “*An Unstructured Grid Generation and Adaptive Solution Technique for High-Reynolds-Number Compressible Flows*”, von Karman Institute for Fluid Dynamics, Lecture Series 1996-06, March, 1996
94. Saxena, S.K., and Nair, M.T., “*Implementation and Testing of Spalart-Allmaras Model in a Multi-Block Code*”, AIAA-2002-0835, January, 2002
95. Rogers, D. F., and Adams, J. A., *Mathematical Elements for Computer Graphics*, 2nd Edition, McGraw-Hill, 1990,
96. NASA Technical Memorandum 74058 – Transonic Static and Dynamic Stability Characteristics of a Finned Projectile Configuration

97. NASA Technical Paper 1078 – Supersonic Aerodynamic Characteristics of A Sparrow III Type Missile Model With Wind Controls and Comparison With Existing Tail-Control Results
98. Haliloğlu, M. U., “*Development of Three-Dimensional Object-Oriented Navier-Stokes Solver Using C++ Programming Language*”, Ms. Thesis, METU, 2000
99. Aksel M. H., *Notes on Fluid Mechanics*, METU
100. Van den Berg, B., and Gooden, J. H. M., “*Low-Speed Pressure and Boundary Layer Measurement Data for the NLR 7301 Airfoil Section with Trailing Edge Flap*”
101. Mavriplis, D., and Jameson, A., “*Multigrid Solution of the Navier-Stokes Equations on Triangular Meshes*”, AIAA Journal, Vol. 28, No.8, pp. 1415-1425, August 1990
102. Wu, J.C, Wang, C.M., and Tuncer, I.H., “*Unsteady Aerodynamics of Rapidly Pitched Airfoils*”, AIAA-86-1105, 1986.
103. Hoffmann, K.A., and Chiang, S.T., *Computational Fluid Dynamics Volume III, Fourth Edition*, Engineering Education System Publications, 2000.
104. Pan, D., and Cheng, J.C., “*Upwind Finite-Volume Navier-Stokes Computations on Unstructured Triangular Meshes*”, AIAA Journal, Vol. 31, No. 9, pp. 1618-1625, September, 1993
105. Thibert, J.J., Grandjacques, M., and Ohman, L.H., “*Experimental Data Base for Computer Program Assessment*”, AGARD AR-138, May 1979

106. DeSpirito, J., and Heavey, K. R., “*CFD Computation of Magnus Moment and Roll Damping Moment of a Spinning Projectile*”, AIAA-2004-4713, August, 2004

107. Sethian, J.A., *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, Cambridge University Press, 1996

APPENDIX A

SAMPLE INPUT DATA

1.1	Inflow Mach Number
1	Viscous Model 0:Inviscid; 1:Viscous
0	Turbulence 0:Laminar; 1:Spalart-Allmaras
5000	Reynolds #
1.	Free stream Spalart-Allmaras Working Variable; anutin
1.	Inflow Pressure [Non-dim]
1.	Inflow Temperature [Non-dim]
1.	Outflow Pressure [Non-dim](For Cascade Flows)
10.0, 0.0	Alpha and Beta [°]
0., 0., 0.	Center of Mass [m]
12.69	Reference Length [m]
1	Implicit Residual Smoothing 0:Closed; 1:Open
1.5	CFL #
10000	Number of Time Step
10	Frequency for the Result to be Printed on Screen
1.e10	Maximum Physical Time For Run [sec](For Unsteady Problems)
5.	Order of Magnitude For The Residual To Be Reduced
1000	Frequency for the Solution to be Saved
1 q.end	Input Data File is Used or Not; File Name
3	Tecplot Output
30	Number of Partitions
1	iloc 0:local, 1:global time stepping (for ALE)
1	irotn 0:no rotation 1:rotation (for ALE)
0.0093	Reduced Frequency (Rotation)Around Axis of Projectile (for ALE)
0.	Reduced Frequency (Rotation)Around Axis of Trajectory (for ALE)

APPENDIX C

UNSTRUCTURED GRID GENERATION FOR A PROJECTILE

Unstructured mesh on a projectile without fins for viscous flow calculations is generated by post processing the structured mesh generated by CFD-GEOM. It is impossible to handle a multi domain mesh for this purpose because CFD-GEOM gives i-j-k ordered output. Because of this reason it is tried to obtain a single zone viscous structured mesh over a projectile.

First a 2-D computational mesh is generated over the half of the projectile, Figure 165.

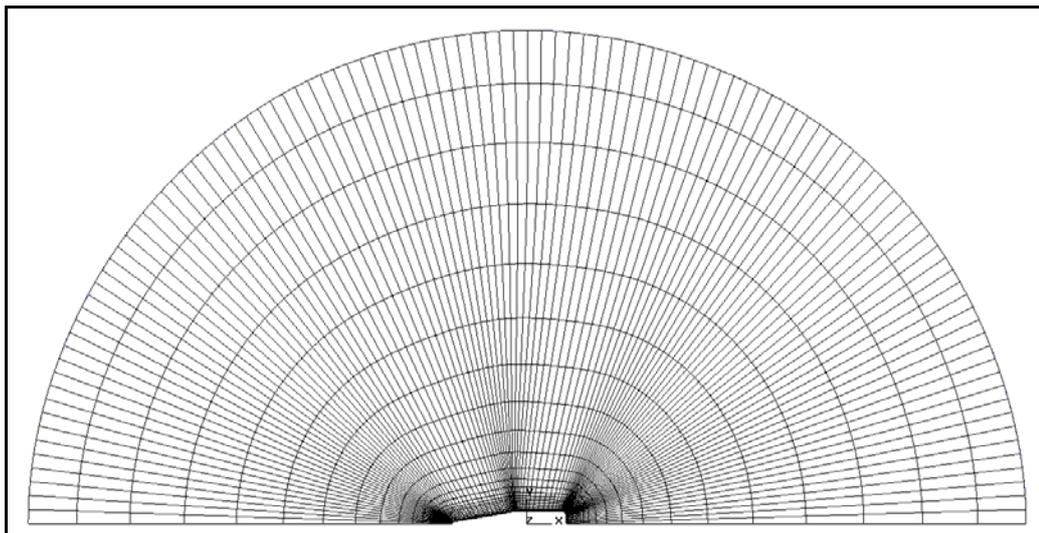


Figure 165. 2-D Structured Computational Mesh Around a Projectile

Then 3-D computational mesh is obtained by rotating this 2-D mesh around x-axis. Generated mesh is reoriented in order to make i direction coincident with x-axis, j-direction coincident with y-axis and k-direction coincident with z-axis, Figure 166.

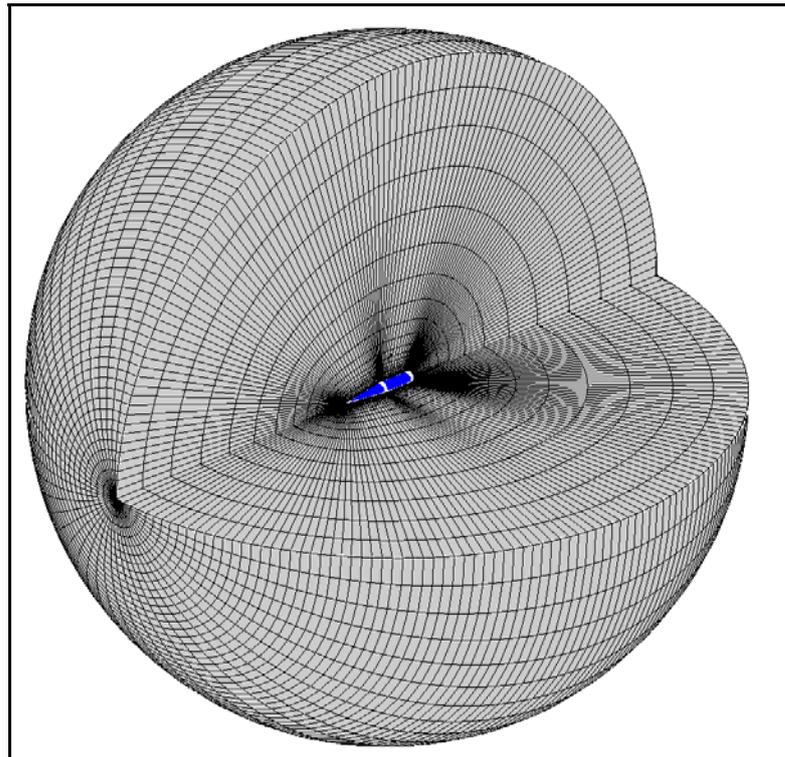


Figure 166. 3-D Structured Computational Mesh Around a Projectile

Then 3-D structured mesh data is exported from CFD-GEOM in i-j-k PLOT3D format. This mesh can be treated as a mesh composed of hexahedral elements and prismatic elements at vicinity of $y = 0$ and $z = 0$, Figure 167.

Hexahedral elements are divided into six tetrahedral elements where prismatic elements are divided into three tetrahedral elements. Special treatment is

given to the beginning and end of the projectile surface where prismatic elements occur. Normal directions of the faces of the generated tetrahedral elements are forced to be directing outside of the element.

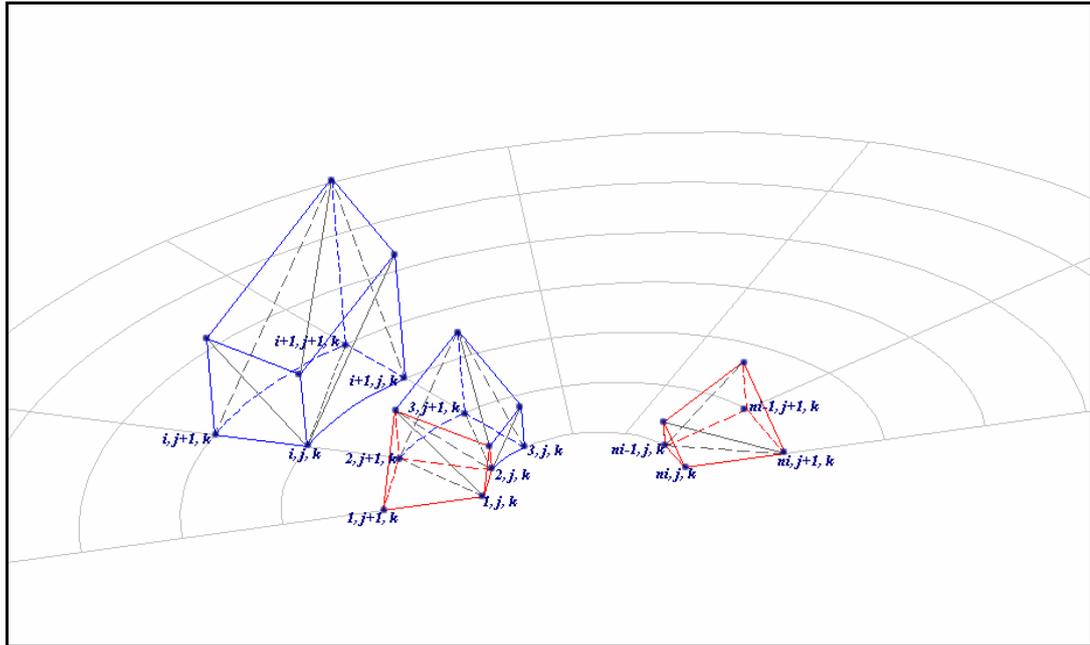


Figure 167. Hexahedral and Prismatic Elements on 3-D Structured Mesh

Illustration of generating three unstructured tetrahedral elements from a prismatic element at the nose of the projectile is given in Figure 168. Resulting connectivity for three tetrahedral elements is shown in Table 6.

Illustration of generating three unstructured tetrahedral elements from a prismatic element at the base of the projectile is given in Figure 169. Resulting connectivity for three tetrahedral elements is shown in Table 7.

Illustration of generating six unstructured tetrahedral elements from a hexahedral element is given in Figure 170. Resulting connectivity for six tetrahedral elements is shown in Table 8.

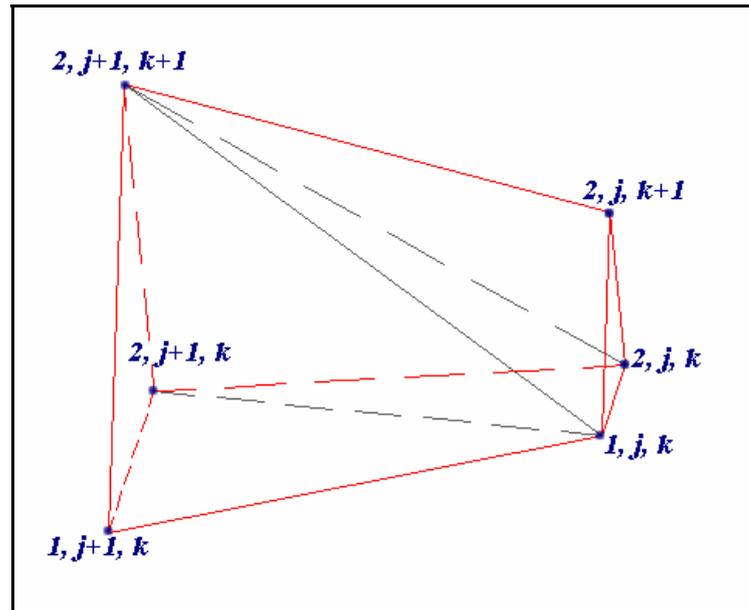


Figure 168. Generating Tetrahedral Elements from the Prismatic Element at the Nose of the Projectile

Table 6. Connectivity for Three Tetrahedral Elements Obtained from the Prismatic Element at the Nose of the Projectile

Element #	i1	i2	i3	i4
1	2, j+1, k	2, j, k	2, j+1, k+1	1, j, k
2	2, j+1, k+1	2, j, k	2, j, k+1	1, j, k
3	1, j+1, k	2, j+1, k	2, j+1, k+1	1, j, k

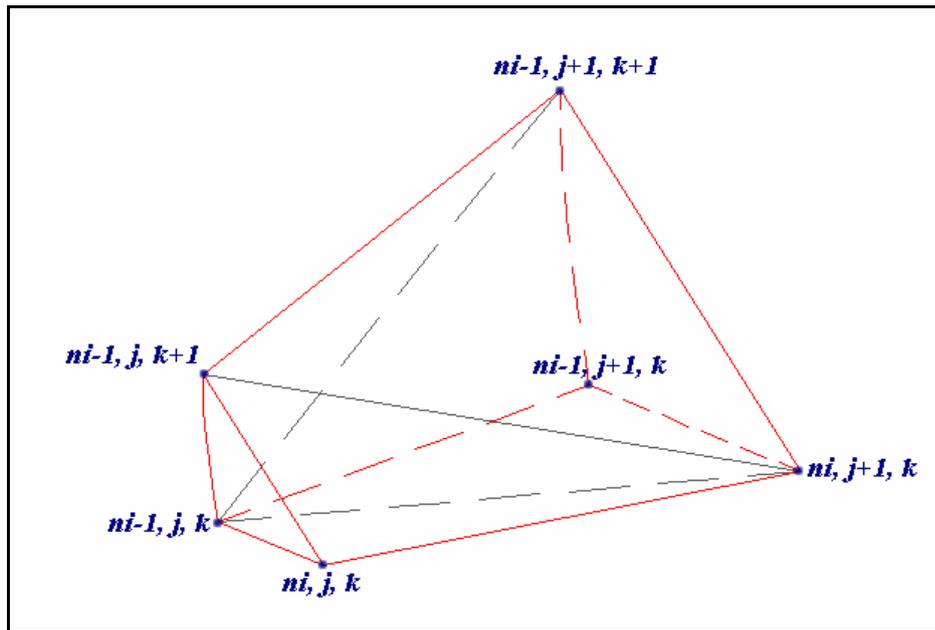


Figure 169. Generating Tetrahedral Elements from the Prismatic Element at the Base of the Projectile

Table 7. Connectivity for Three Tetrahedral Elements Obtained from the Prismatic Element at the Base of the Projectile

Element #	i1	i2	i3	i4
1	$ni-1, j, k$	$ni-1, j+1, k$	$ni-1, j+1, k+1$	$ni, j+1, k$
2	$ni-1, j, k+1$	$ni-1, j, k$	$ni-1, j+1, k+1$	$ni, j+1, k$
3	ni, j, k	$ni-1, j, k$	$ni-1, j, k+1$	$ni, j+1, k$

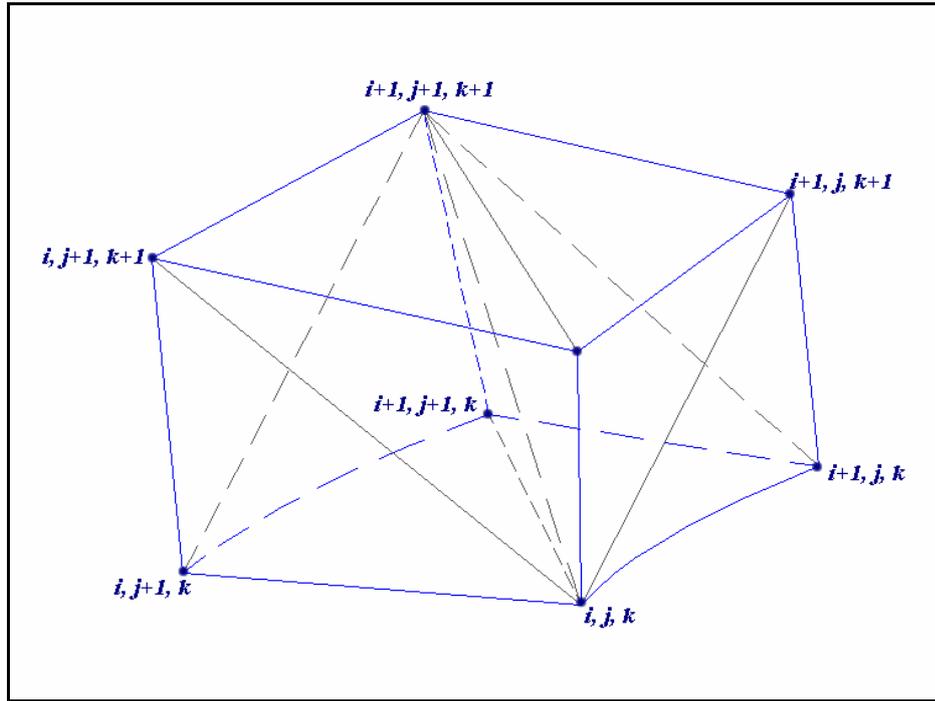


Figure 170. Generating Tetrahedral Elements from a Hexahedral Element

Table 8. Connectivity for Six Tetrahedral Elements Obtained from a Hexahedral Element

Element #	i1	i2	i3	i4
1	$i+1, j+1, k$	$i+1, j, k$	$i+1, j+1, k+1$	i, j, k
2	$i+1, j+1, k+1$	$i+1, j, k$	$i+1, j, k+1$	i, j, k
3	$i, j+1, k$	$i+1, j+1, k$	$i+1, j+1, k+1$	i, j, k
4	$i+1, j+1, k+1$	$i+1, j, k+1$	$i, j, k+1$	i, j, k
5	$i, j+1, k+1$	$i+1, j+1, k+1$	$i, j, k+1$	i, j, k
6	$i, j+1, k$	$i+1, j+1, k+1$	$i, j+1, k+1$	i, j, k

APPENDIX D

MIMIMUM WALL DISTANCE CALCULATION IN THE SPALART-ALLMARAS TURBULENCE MODEL

The procedure followed for the calculation of minimum distance of cell centers of the computational elements to the wall boundary which is extremely important for Spalart-Allmaras turbulence model is stated below in order:

1. First all the faces on the wall boundary are marked. Four neighbors of all of the tetrahedral control volumes have already been determined in the solver. It is known whether a face of the control volume is wall or not.
2. Coordinates of the center of these wall faces are calculated by averaging the coordinates of the nodes forming the face:

$$\begin{aligned}x_{\text{wall,center}} &= \frac{x(\text{node1}) + x(\text{node2}) + x(\text{node3})}{3} \\y_{\text{wall,center}} &= \frac{y(\text{node1}) + y(\text{node2}) + y(\text{node3})}{3} \\z_{\text{wall,center}} &= \frac{z(\text{node1}) + z(\text{node2}) + z(\text{node3})}{3}\end{aligned}\tag{D.1}$$

3. An overall loop is turned over the total number of control volumes and the coordinates of the centroids of tetrahedral control volumes are calculated again by averaging the coordinates of the nodes that are belonging to that control volume:

$$\begin{aligned}
x_{n,\text{center}} &= \frac{x(\text{node1}) + x(\text{node2}) + x(\text{node3}) + x(\text{node4})}{4} \\
y_{n,\text{center}} &= \frac{y(\text{node1}) + y(\text{node2}) + y(\text{node3}) + y(\text{node4})}{4} \\
z_{n,\text{center}} &= \frac{z(\text{node1}) + z(\text{node2}) + z(\text{node3}) + z(\text{node4})}{4}
\end{aligned} \tag{D.2}$$

4. For all control volumes distance from the control volume centroid to the marked wall faces are calculated one by one:

$$\text{dist}(n) = \sqrt{(x_{n,\text{centr}} - x_{\text{wall,centr}})^2 + (y_{n,\text{centr}} - y_{\text{wall,centr}})^2 + (z_{n,\text{centr}} - z_{\text{wall,centr}})^2} \tag{D.3}$$

5. Minimum value of these distances and corresponding wall face is stored for each control volume.
6. An extra check has to be performed for highly stretched elements in order to eliminate the problem of calculating wrong minimum distances. This situation is illustrated in Figure 171. The following condition has to be satisfied for the Spalart-Allmaras turbulence model.

$$\text{dist}(1) < \text{dist}(2) < \text{dist}(3) < \text{dist}(4) \tag{D.4}$$

Note that these distances which are calculated by equation (D.3) are not exactly the minimum distances to the wall. Following steps will correct this approximation.

7. Control volumes which have a minimum wall distance less than a certain value (0.1 is taken for M910 projectile) are treated again. For these control volumes the equation of minimum distance between a point and plane is used. The equation of a plane, the wall face in our case, is defined as:

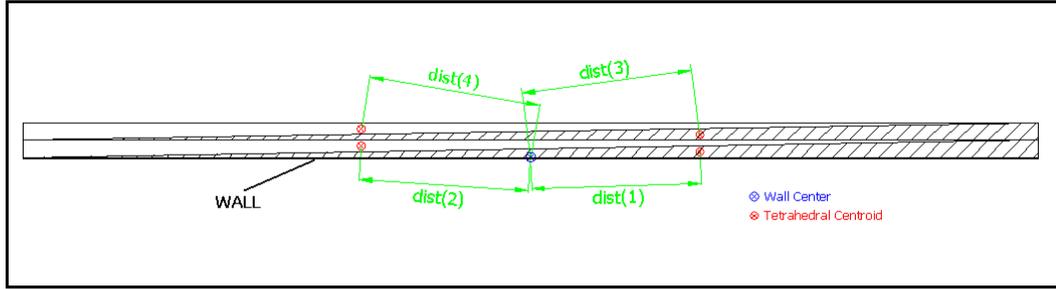


Figure 171. Minimum Wall Distance Calculation for Highly Stretched Elements Near Wall Boundary

$$A \cdot (x - x_{\text{wall,center}}) + B \cdot (y - y_{\text{wall,center}}) + C \cdot (z - z_{\text{wall,center}}) = 0 \quad (\text{D.5})$$

$$A \cdot x + B \cdot y + C \cdot z + D = 0 \quad (\text{D.6})$$

$$\text{where } D = -A \cdot x_{\text{wall,center}} - B \cdot y_{\text{wall,center}} - C \cdot z_{\text{wall,center}} \quad (\text{D.7})$$

and A, B and C are the coefficients of surface normal vector of the wall face.

The equation of the minimum distance between the centroid of a control volume and the wall plane is defined as:

$$\text{dist}(n) = \frac{|A \cdot x_{n,\text{centroid}} + B \cdot y_{n,\text{centroid}} + C \cdot z_{n,\text{centroid}} + D|}{\sqrt{A^2 + B^2 + C^2}} \quad (\text{D.8})$$

This step provides the correct calculation of the minimum wall distances for highly stretched elements near wall boundary, Figure 172. The condition defined in Equation (D.4) is satisfied by this method.

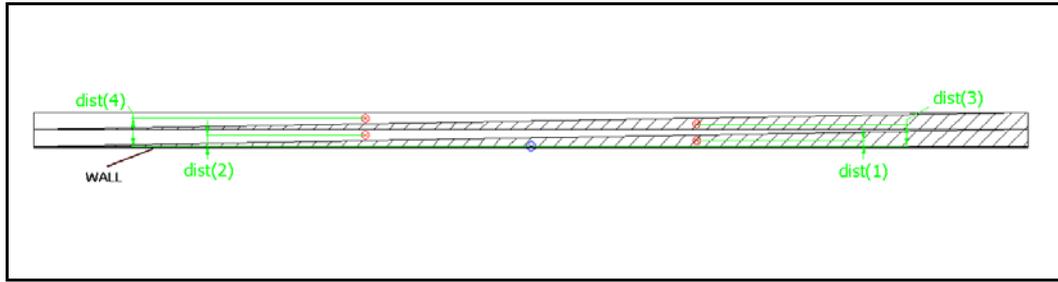


Figure 172. Minimum Wall Distance Calculation for Highly Stretched Elements
Near Wall Boundary - Corrected

CURRICULUM VITAE

PERSONAL INFORMATION :

Surname : GÖNÇ

Name : L. Oktay

Date of birth : 09 September 1973

Place of birth : Ankara - TURKEY

Nationality : Turkish

Permanent address : 6. Sok. 38/5
Bahçelievler
Ankara ,TURKEY Tel: +90 (312) 223 10 03
Mobile Tel: +90 (532) 343 03 26

Work address : TÜBİTAK-SAGE
Turkish Scientific & Technical Research Council
Defense Industries Research & Development Institute
P.K. 16, 06261 Mamak – ANKARA / TURKEY
Tel: +90 (312) 590 91 07
Fax: +90 (312) 590 91 48-49
E-mail: lgonc@sage.tubitak.gov.tr
oktaygonc@yahoo.com

Marital status : Single

Profession : Mechanical Engineer, B.Sc., M.Sc. METU

Memberships : Chamber of Mechanical Engineers
METU Alumni Society

EDUCATION :

<i>School</i>	<i>Years</i>	<i>Diploma, Grade and Subject</i>
<u>Primary School</u>		
Hürriyet İlkokulu Çankaya, Ankara, TÜRKİYE	1979 - 1981	
T.E.D. Ankara Koleji İlkokulu Çankaya, Ankara, TÜRKİYE	1981 - 1984	Primary School Diploma Honors
<u>Secondary & High School</u>		
Gazi Anatolian High School Yenimahalle, Ankara, TÜRKİYE	1984 - 1991	Secondary & High School Diploma / in math. & science division / Honors (9.09 / 10)
<u>University</u>		
METU Department of Mechanical Engineering İnönü Blv., Ankara, TÜRKİYE	1991 - 1995	B.Sc. , Fluid Mechanics and Thermodynamics, High Honors (3.76 / 4.00) * Second degree in ME
METU Department of Mechanical Engineering İnönü Blv., Ankara, TÜRKİYE	1995 - 1998	M.Sc. , “ Axial Compressor and Turbine Stage Design using Through Flow & Blade- To-Blade Codes” (3.78 / 4.00)
METU Department of Mechanical Engineering İnönü Blv., Ankara, TÜRKİYE	1998 - 2005	Phd. , “Computation of External Flow Around Rotating Bodies” (4.00 / 4.00)

LANGUAGE SKILLS :

Native Language : Turkish
Other Languages : English (Bilingual standard)

AWARDS:

Prof. Dr. Mustafa N. Parlar Education & Research Foundation
1999 - METU Best Thesis of the Year June – 1999

Türk Isı Bilimi ve Tekniği Derneği 1999 – Encouragement Award (TÜBİTAK-SAGE Internal Ballistics Division) 1999

PUBLICATIONS:

1. Gonc, L. O., and Ucer, A.S., “Axial Compressor and Turbine Stage Design Using Through-Flow and Blade-To-Blade Codes”, XIV. International Symposium on Airbreathing Engines, Florence, Italy, IS-7046, September 1999.
2. Gönç, L.O., Ak, M. A., Tuncer, I.H., Başoğlu, O.,and Işık, T., “TÜBİTAK-SAGE’de Geliştirilen Sayısal Akışkanlar Mekaniği Yetenekleri”, ABAQUS Kullanıcılar Toplantısı, İstanbul, TÜRKİYE, 5 December 2003, pp 197-202.
3. Gönç, L.O., Ak, M.A., Tuncer, I.H., Başoğlu, O.,Işık, T.,and Sümer, B., “TÜBİTAK-SAGE’de Geliştirilen Sayısal Akışkanlar Mekaniği Yetenekleri”, Kayseri V. Havacılık Sempozyumu , 13-14 May 2004, pp. 62-66, Kayseri, TÜRKİYE.
4. Ulaş, A., AK, M.A., Orhan F.E., Gündüz, D.E., Gönç, L.O., Sümer, B., Püskülcü, G.,and Yıldırım, C., ”Design and Testing of a Hydrogen Peroxide-Kerosene Liquid Rocket Engine”, Istanbul Space Propulsion Workshop 2004, 10-12

June 2004, İstanbul, TÜRKİYE.

5. Ulaş, A., AK, M.A., Orhan F.E., Gündüz, D.E., Gönç, L.O., Sümer, B., Püskülcü, G., and Yıldırım, C., “Türkiye’nin Uzaya Bağımsız Ulaşımının Anahtarı: İlk Milli Hidrojen Peroksit - Kerosen Sıvı Yakıtlı Roket Motoru Tasarımı”, SAVTEK 2004, SAVUNMA TEKNOLOJİLERİ KONGRESİ, 24-25 June 2004, Ankara, TÜRKİYE.

TRAINING :

Summer Training :

- 1.) AKDÖKÜM Inc.. , İzmir, TÜRKİYE July - August 1993
- 2.) ASELSAN Inc. Microwave & System Tech. Div. Ankara, TÜRKİYE
July - August 1994

Technical Training :

- 1.) AutoCAD R - 12
Middle East Technical University Ankara, TÜRKİYE Sep. – Dec. 1994
- 2.) ISO 9000 Quality Assurance Management Model
TÜBİTAK-SAGE Ankara, TÜRKİYE July-1999
- 3.) Time Management
TÜBİTAK-SAGE Ankara, TÜRKİYE February-2000
- 4.) Occupational Health And Safety
TÜBİTAK-SAGE Ankara, TÜRKİYE March-2001
- 5.) Configuration Management

TÜBİTAK-SAGE Ankara, TÜRKİYE October-2001

6.) Basic Quality Concepts & Institution Culture

TÜSSİDE Gebze, Kocaeli, TÜRKİYE March-2002

7.) CFD-FASTRAN / CFD-GEOM / CFD-VIEW Training

CFDRC Hunstville, Alabama, USA August-2002

8.) CFD-FASTRAN / CFD-GEOM / CFD-VIEW User Conference

CFDRC Hunstville, Alabama, USA May-2003

9.) ICHMT THMT-03 International Symposium on Turbulence, Heat and Mass Transfer, Antalya, TÜRKİYE October-2003

WORK EXPERIENCE :

<i>Organization</i>	<i>Years</i>	<i>Job</i>
<i>METU</i>		
Department of Mechanical Engineering İnönü Blv., Ankara, TÜRKİYE	Aug. 1995 – Sep. 1998	Research Assistant
<i>Organization</i>		
<i>TÜBİTAK-SAGE</i>		
Motor & Warhead Technologies Group / Propulsion Division Ankara, TÜRKİYE	Oct. 1998 – ...	Senior Research Engineer