

IMPLEMENTATION OF THE SPALART-ALLMARAS TURBULENCE MODEL TO A
TWO-DIMENSIONAL UNSTRUCTURED NAVIER-STOKES SOLVER

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

BY
ORHAN AYBAY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE
OF
MASTER OF SCIENCE
IN
MECHANICAL ENGINEERING

DECEMBER 2004

Approval of the Graduate School of Natural and Applied Sciences

Prof. Dr. Canan ÖZGEN
Director

I certify that this thesis satisfies all the requirement as a thesis for the degree of Master of Science.

Prof. Dr. 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 Master of Science.

Prof. Dr. M. Haluk AKSEL
Supervisor

Examining Committee Members

Prof. Dr. Zafer DURSUNKAYA (METU,ME) _____

Prof. Dr. M. Haluk AKSEL (METU,ME) _____

Asst. Prof. Dr. Tahsin ÇETINKAYA (METU,ME) _____

Asst. Prof. Dr. Cüneyt SERT (METU,ME) _____

Yüksel ORTAKAYA (M.Sc. in AEE) (TAI) _____

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.

Orhan Aybay

ABSTRACT

IMPLEMENTATION OF THE SPALART-ALLMARAS TURBULENCE MODEL TO A TWO-DIMENSIONAL UNSTRUCTURED NAVIER-STOKES SOLVER

AYBAY, Orhan
M.Sc., Department of Mechanical Engineering
Supervisor: Prof. Dr. M. Haluk Aksel

December 2004, 84 pages

An unstructured explicit, Reynolds averaged Navier-Stokes solver is developed to operate on inviscid flows, laminar flows and turbulent flows and one equation Spalart-Allmaras turbulence modeling is implemented to the solver. A finite volume formulation, which is cell-center based, is used for numerical discretization of Navier-Stokes equations in conservative form. This formulation is combined with one-step, explicit time marching upwind numerical scheme that is the first order accurate in space. Turbulent viscosity is calculated by using one equation Spalart-Allmaras turbulence transport equation. In order to increase the convergence of the solver local time stepping technique is applied.

Eight test cases are used to validate the developed solver, for inviscid flows, laminar flows and turbulent flows. All flow regimes are tested on NACA-0012 airfoil. The results of NACA-0012 are compared with the numerical and experimental data.

Keywords: Unstructured grid, Navier-Stokes equations, finite volume method, upwind method, Spalart-Allmaras turbulence model.

ÖZ

SPALART-ALLMARAS TÜRBÜLANS MODELİNİN YAPISIZ AĞLI NAVIER-STOKES ÇÖZÜCÜSÜNE UYGULANMASI

AYBAY, Orhan

Yüksek Lisans, Makina Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. M. Haluk Aksel

Aralık 2004, 84 sayfa

Viskoz olmayan akışları, laminar akışları ve türbülanslı akışları hesaplamak için, yapısız ağlı, Reynolds ortalamalı, belirli zaman ilerlemeli Navier-Stokes çözümü geliştirilmiş ve bu çözümü tek denklemli Spalart-Allmaras türbülans modeli eklenmiştir. Korunum biçimindeki Navier-Stokes denklemlerinin sayısal olarak ayrıştırılması için, hücre merkezli sonlu hacim yöntemi kullanılmıştır. Bu formülasyon, uzayda birinci dereceden hassas, tek adımlı ve zaman ilerlemeli ‘Upwind’ sayısal şeması ile birleştirilmiştir. Türbülans viskozitelerinin hesaplanması için tek denklemli Spalart-Allmaras denklemi kullanılmıştır. Çözümenin yakınsama hızını artırmak için yerel zaman adımlama tekniği uygulanmıştır.

Sekiz adet test durumu ile viskoz olmayan akışlar, laminar akışlar ve türbülanslı akışlarda geliştirilen çözümünün doğruluğu test edilmiştir. Bütün testler NACA-0012 uçak kanat kesidi üzerinde yapılmıştır. NACA-0012 çözümleri nümerik ve deneysel sonuçlarla karşılaştırılmıştır.

Anahtar Kelimeler: Yapisız ağ, Navier-Stokes denklemleri, sonlu hacim yöntemi, upwind yöntemi, Spalart-Allmaras türbülans modeli

ACKNOWLEDGEMENTS

I would like to express my gratitude to Prof. Dr. Haluk Aksel, being my supervisor, and for his valuable guidance and support through this study.

I would like to thank to my family, for their loving care and efforts, without their support this work can not be completed.

I would like to thank to Barbaros Çetin, Gökmen Demirkaya, Mustafa Kaya, Burak Sarı, Okan Türkdoğan and N. Deniz Yücel for their friendship and supports.

Special thanks to Maria Turculets for her friendship, encouragement and support.

I want to thank all the colleagues in the CFD group, for their friendship and for sharing their ideas about many issues, giving me the inspiration to complete this work.

Finally, I am also thankful to Asst. Prof. Dr. Cüneyt Sert for his invaluable suggestions and contributions.

TABLE OF CONTENTS

PLAGIARISM.....	iii
ABSTRACT.....	iv
ÖZ.....	vi
ACKNOWLEDGEMENTS.....	viii
TABLE OF CONTENTS.....	ix
LIST OF TABLES.....	xi
LIST OF FIGURES.....	xii
LIST OF SYMBOLS.....	xiv
CHAPTER	
1. INTRODUCTION.....	1
1.1 Background.....	1
1.2 Literature Review.....	4
1.3 Objectives.....	9
1.4 Present Study.....	10
2. GOVERNING EQUATIONS.....	11
2.1 Navier-Stokes Equations.....	11
2.2 Turbulence Modeling.....	15
2.3 Boundary Conditions.....	20
2.3.1 Free Stream Boundary Conditions.....	20
2.3.2 Far Field Boundary Conditions.....	21
2.3.3 Solid Wall Boundary Conditions.....	23
3. NUMERICAL METHOD.....	25
3.1 Cell Based Data Structure.....	25
3.2 Spatial Discretization.....	28

3.2.1	Discretization of Convective Terms.....	32
3.2.2	Viscous Flux Calculations.....	35
3.3	Turbulence Model Discretization.....	38
3.4	Temporal Discretization.....	44
3.4.1	Runge-Kutta Method.....	44
3.4.2	Local Time Stepping.....	46
3.4.3	Residual Smoothing.....	46
4.	RESULTS.....	48
4.1	Euler Solutions.....	49
4.1.1	Test Case 1.....	49
4.1.2	Test Case 2.....	53
4.1.3	Test Case 3.....	56
4.2	Laminar Navier-Stokes Solution.....	59
4.2.1	Test Case 4.....	59
4.2.2	Test Case 5.....	63
4.3	Turbulent Navier-Stokes Solution.....	68
4.3.1	Test Case 6.....	68
4.3.2	Test Case 7.....	71
4.3.3	Test Case 8.....	73
5.	DISCUSSION AND CONCLUSION.....	76
5.1	Summary and Conclusion	76
5.2	Future Recommendations.....	77
	REFERENCES.....	79

LIST OF TABLES

2.1	Riemann invariants calculation and extrapolation.....	22
3.1	Cell based data structure.....	26
3.2	Connectivity of cells to the vertices and neighbouring cells.....	27
4.1	Test case summary.....	48

LIST OF FIGURES

1.1	Interaction between CFD, experiment and theory.....	1
3.1	Example of unstructured triangular grids.....	26
3.2	Example of connectivity of the neighbouring cells.....	27
3.3	Areas considered for averaging the conserved variable in node based scheme.....	31
3.4	Areas considered for averaging the conserved variable in cell based scheme	32
3.5	Initial conditions for the conservation laws.....	34
3.6	One edge of a control volume.....	37
4.1	Unstructured grid consists of 6990 nodes and 13678 cells for Euler solver.....	50
4.2	Close-up grid at the leading edge of the NACA 0012.....	51
4.3	Close-up grid at the trailing edge of the NACA 0012.....	51
4.4	Calculated surface pressure coefficients of subsonic lifting flow.....	52
4.5	Convergence history of subsonic lifting flow test case.....	52
4.6	Calculated surface pressure coefficients of transonic nonlifting flow	54
4.7	Convergence history of transonic nonlifting flow test case.....	54
4.8	Calculated Mach number contours for transonic non lifting test case (from [41]).....	55
4.9	Calculated Mach number contours for transonic non lifting test case (present solution).....	55
4.10	Convergence history of transonic lifting flow test case.....	57
4.11	Calculated surface pressure coefficients of transonic lifting flow	57
4.12	Calculated pressure contours for transonic lifting test case (from [42])...	58

4.13	Calculated pressure contours for transonic lifting test case (present solution).....	58
4.14	Close-up grid around the NACA 0012.....	60
4.15	Calculated Mach number contours for laminar flow at $Re = 500$ (from [42]).....	61
4.16	Calculated Mach number contours for laminar flow at $Re = 500$ (present solution).....	61
4.17	Velocity vectors at the trailing edge for laminar flow at $Re = 500$	62
4.18	Convergence history of laminar flow at $Re = 500$	62
4.19	Calculated Mach number contours for laminar flow at $Re = 5000$ (from [40]).....	64
4.20	Calculated Mach number contours for laminar flow at $Re = 5000$ (present solution).....	64
4.21	Close-up velocity vectors at the trailing edge (from [40]).....	65
4.22	Close-up velocity vectors at the trailing edge (present solution).....	65
4.23	Calculated surface pressure coefficients for laminar flow at $Re = 5000$	66
4.24	Close-up of velocity vectors for separation point (present solution).....	67
4.25	Convergence history of laminar flow at $Re = 5000$	67
4.26	Unstructured grid consists of 19040 nodes and 37798 cells for the turbulent Navier-Stokes solver.....	68
4.27	Close-up grid around the NACA 0012.....	69
4.28	Calculated pressure coefficient distribution on NACA 0012 airfoil for the turbulent subsonic test case 6.....	70
4.29	Convergence history of turbulent flow for the subsonic test case 6.....	71
4.30	Calculated pressure coefficient distribution on NACA 0012 airfoil for the turbulent transonic test case 7.....	72
4.31	Convergence history of turbulent flow for the transonic test case 7.....	73
4.32	Calculated surface pressure coefficients for the turbulent transonic test case 8.....	74
4.33	Convergence history of turbulent flow for the transonic test case 8.....	75

LIST OF SYMBOLS

a	Speed of sound
c	Chord length of the airfoil
C_{b1}	Closure coefficient for a one-equation turbulence model
C_{b2}	Closure coefficient for a one-equation turbulence model
C_{v1}	Closure coefficient for a one-equation turbulence model
C_{t1}	Closure coefficient for a one-equation turbulence model
C_{t2}	Closure coefficient for a one-equation turbulence model
C_{t3}	Closure coefficient for a one-equation turbulence model
C_{t4}	Closure coefficient for a one-equation turbulence model
C_{w1}	Closure coefficient for a one-equation turbulence model
C_{w2}	Closure coefficient for a one-equation turbulence model
C_{w3}	Closure coefficient for a one-equation turbulence model
d	Distance to wall
E	Total energy per unit volume
F	Inviscid flux vector in x-direction
F_v	Viscous flux vector in x-direction
f_{t1}	Closure coefficient for a one-equation turbulence model
f_{t2}	Closure coefficient for a one-equation turbulence model
f_{v1}	Closure coefficient for a one-equation turbulence model
f_{v2}	Closure coefficient for a one-equation turbulence model
f_w	Closure coefficient for a one-equation turbulence model
G	Inviscid flux vector in y-direction
G_v	Viscous flux vector in y-direction
\vec{i}	Unit vector in x-direction
\vec{j}	Unit vector in y-direction
M	Mach number
\vec{n}	Unit normal vector
p	Static pressure
Q	Conservative variable, space coordinate
R	Universal gas constant
Re_a	Reynolds number based on free-stream speed of sound

Re	Reynolds number
S	Source term
t	Time
T	Temperature
u	Velocity component in x-direction
U	Conservative variable vector
v	Velocity component in y-direction
ν	Eddy viscosity
\vec{V}	Velocity vector
x	Spatial variable in physical domain
y	Spatial variable in physical domain

Greek Symbols

γ	Ratio of specific heats
δ	Correction operator, boundary layer thickness
Δt	Time step
ΔV	Cell volume
μ	Absolute viscosity
ρ	Density
τ	Shear stress
ϕ	Dummy variable
$\vec{\omega}$	Vorticity vector
∇	Gradient operator
∂	Partial derivative operator
κ	Von Karman constant

Subscripts

i	Node index
n	Normal direction
v	Viscous component of flux vector
∞	Free stream value

Superscripts

n	Time level
-	Value known from the previous time step
\sim	Primitive form
'	Non-dimensional variable

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

In the last few decades, the world has seen a revolution in technological achievement brought about by computers. Furthermore, the role of computational fluid dynamics (CFD) in engineering applications has become more essential. CFD can be viewed as a new ‘third dimension’ in fluid dynamics today, where the other dimensions are the classical cases of experiment and theory. This relation can be shown in the Figure 1.1 [1]:

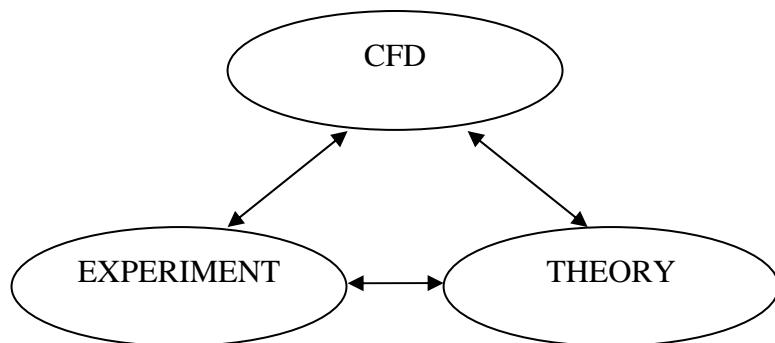


Figure 1.1 Interaction between CFD, experiment and theory

Aerodynamic analysis of aircraft design can be performed using both experimental and theoretical methods. The experimental approach can produce accurate data over a particular range of flow conditions. Scaled models of aircraft or wing sections are constructed and placed in wind

tunnels. Results of experimental origin have been heavily relied on critical aerodynamic designs. Although experimental approach is very accurate, the limitation on the hardware, such as the tunnel size and the difficulty in simulating the prototype flow field, makes it impractical to obtain flow fields for many problems [2].

The theoretical approach approximates realistic non-linear flow field phenomena with simplified mathematical relations. Unlike theoretical methods, numerical methods attempt to resolve all aspects of the flow field including non-linear effects. The task of obtaining solutions to the governing equations of fluid dynamics represents one of the most challenging problems in science and engineering. Generally, the governing equations of fluid dynamics form a set of coupled, nonlinear partial differential equations that must be solved in an irregular domain subject to various initial and boundary conditions.

Since there are very few restrictive assumptions, the CFD approach is superior to the experimental and theoretical approaches in treating complicated configurations. In addition, there are several advantages of CFD over experimental approach such as [3];

- i. Essential reduction of lead times and costs of new designs
- ii. Ability to study very large systems
- iii. Ability to study systems under hazardous conditions
- iv. Ability to obtain detailed results

On the other hand, there are limitations and restrictions for CFD. Flow predictions are as superior as the level of physics that goes into the formulations. That is to say, the quality of the computational results will always depend on the ability to model the physics appropriately and numerical results are always approximate. Furthermore, the cost

effectiveness of a numerical model for aerodynamic analysis is dependent not only on the power of the computer hardware but also on the accuracy and efficiency of the various discretization methods. Several of these methods can offer distinct advantages in cost and flexibility of applications.

Generation of the appropriate grid to support numerical calculations is a challenging, multi-disciplinary problem; it is currently often more time and labor consuming to create the supporting grid than to define and perform the desired simulation. Grid generation makes the problem manageable for computer simulations. There are various ways to define a grid around an object, which are related to the following facts:

- i. Geometric complexity of the object around or inside which the fluid flows
- ii. Mathematical model chosen to solve the problem (i.e. Euler or Navier-Stokes equations)
- iii. Qualitative shape of the flow field (i.e. where large gradients occur, location of shock waves, boundary layers, etc...)

Depending on the above stated considerations, different kinds of algorithms can be employed to produce grids. Grids can be generated in either structured or unstructured form. Structured grids have been used to successfully solve problems in fluid dynamics. In two dimensions, structured grid solvers are extremely useful especially for single element airfoil cases. The majority of the physical domains of interest are nonrectangular and computational domain will require some sort of interpolation for the implementation of the boundary conditions, which causes inaccuracies at the place of the greatest sensitivity [3].

The difficulty of using structured grids while dealing with complex flow problems makes unstructured grids very attractive. The main advantage of the unstructured grid is that it can be used easily to fit irregular, singly-connected domains, as well as multiply-connected domains. Although the third dimension creates a complicated situation, it is still possible to generate single block, unstructured grids around complex three dimensional objects. However, the major drawback of unstructured grid solvers is that they are typically slower than their structured counterparts and require more memory. To address these problems, techniques like residual smoothing and local time stepping are usually added to speed up unstructured solvers.

Unstructured solvers have historically used triangular elements in two dimensions since triangle is the simplest shape that can be used to fill a space. Triangles however, can lead to inaccurate solutions in the boundary layers of viscous flows meaning that quadrilateral cells are desirable in some grid regions.

1.2 LITERATURE REVIEW

The inviscid Navier-Stokes equations, commonly referred as the Euler equations are solved on structured grids in 1981 by Jameson, Schimdt, and Turkel [4]. Their algorithm solved the Euler equations on a structured grid using a central differencing finite volume solver that achieved second order spatial accuracy in space. After discretization, the Euler equations were reduced to a system of ordinary differential equations which were explicitly marched in time to a steady state solution using a four step Runge-Kutta time marching routine. The artificial dissipation scheme used was a version of scalar dissipation, sometimes referred to as the Jameson Schimdt and Turkel scheme, that is still used today in many solvers. In the mid-eighties, Jameson and Mavriplis [5] introduced a finite-volume solver for the Euler equations on regular triangular meshes. This solver was later

expanded to include fully unstructured meshes, local time stepping and residual smoothing [6]. Local time stepping allowed each control volume to be marched explicitly through time at its largest permissible time step. Residual smoothing allowed the algorithm to be marched in time at more than twice the original rate.

Several authors improved the solver presented by Jameson and Mavriplis. Much of the work done in resolving viscous flows on unstructured grids. In competition with the finite-volume/artificial dissipation solvers, upwinded solvers have been developed by several authors. Desideri and Dervieux [7] developed one of the earliest upwinding schemes on unstructured grids in 1988.

Laminar and turbulent flows are the major flow regimes for any fluid flow. Most flows in nature and in engineering applications are turbulent. Exact numerical simulation at relatively low Reynolds numbers is possible [8]. On the other hand, this is not the case for turbulent flows, in which velocity and other flow properties exhibit random fluctuations. Thus, contrary to laminar flow, which is regular and deterministic, turbulent flow is irregular and chaotic [9]. Properties of turbulence can be stated briefly as below:

- i. Turbulence is irregular or random
- ii. Turbulence arises at large Reynolds numbers
- iii. Turbulence is intrinsically three dimensional
- iv. Turbulence dissipates energy and diffusive
- v. Turbulence is flow dependent

The viscous regions which arise during realistic flight conditions are characterized by turbulence. Instabilities in the shear layers create turbulent fluctuations of the flow field properties. If accurate models of the fluid dynamics about aerodynamic bodies in realistic conditions are to be achieved, the effects of turbulence must be included. A common approach in turbulence modeling is to include an additional eddy viscosity in the conservation equations. The eddy viscosity can be modeled by one of the available methods. In the present study, the eddy viscosity is modeled by Spalart-Allmaras turbulence modeling.

Until recently, algebraic turbulence models were used for most cases and they perform very well for specific geometries. Because of rapid advances in computer speeds, improvements in flow solvers and grid generation algorithms, the aerodynamics community feels the need for a new generation of turbulence models. These models are more accurate, robust and efficient than the algebraic models. While CFD applications involved more complex flows and configurations, the algebraic models often produced awkward results [10]. For example, Baldwin-Lomax [11] model made Navier-Stokes calculations possible in situations that are awkward for the Cebeci-Smith [12] model, because the thickness of the boundary layer is not well defined. The Johnson-King [13] model has shown more accurate predictions of shock/boundary layer interactions, compared with the Baldwin-Lomax and Cebeci-Smith models. The algebraic models treat the whole boundary-layer as a single tightly-coupled module and they are boundary-layer models in spirit [14]. Furthermore, these models produce incorrect results for detached and multiple shear layers. At the implementation level, the difficulty occurs because of their non-local nature. Their success depends on surveying the velocity or vorticity profile on a smooth grid line roughly orthogonal to the surface. When unstructured grids are used, this becomes awkward [15].

The problems with the algebraic models have led to the use of transport equations models such as k - ε turbulence model. Two-equation models are known to provide improved results over algebraic models in many cases. However, limited computer resources allowed their use only in two-dimensional cases or for a single-element three-dimensional geometry such as an inlet, nozzle or wing [10]. They are also far from having shown a decisive advantage on the prediction of shock/boundary layer interactions or separated flows. They require finer grids near a wall and they demand non-trivial upstream and free-stream conditions for the turbulence variables. The near-wall problems often lead to the use of wall functions which loose their justification in the case of separation.

The original idea of using a single transport or one equation model governing directly the development of the eddy viscosity field came from Nee and Kovasznay [16] in the late sixties. The Nee-Kovasznay [16] model is not a local model, since it uses a characteristic length which scales to boundary-layer thickness in the outer region. About twenty years later, Baldwin-Barth [17] rediscovered the idea and propose the Baldwin-Barth [17] model which is a combined form of the k - ε model, through some further assumptions. To that extent its growth may be restricted.

Inspired by the work of Baldwin-Bart [17], Spalart-Allmaras [14] developed a new one-equation model which was derived through empirical arguments and has been shown to be effective in aerodynamic applications. The derivation employs certain empiricism and arguments of dimensional analysis, Galilean invariance and selective dependence on the molecular viscosity, more details can be found in literature [14]. The model yields fairly rapid convergence to steady state. The wall and free stream boundary conditions are trivial. Furthermore, the low cost and robustness of the eddy viscosity-transport formulations make this model very attractive with respect to algebraic models.

The methodology, implementation and validation of the Spalart-Allmaras turbulence model are discussed in references [18-27].

A suitable grid must be generated throughout the domain of solution, before implementation of any numerical model. The generation of unstructured grids has followed from a number of algorithms that permit a high degree of automation and the grid systems are usually generated directly within the physical space. The domain of solution is usually divided into triangles or quadrilaterals (or any kind of polygon) in two-dimensions, whereas pyramids or tetrahedra are used in three-dimensional cases [2]. Mavriplis [28], Barth [29], Hoffman [2] and Peraire [30] have all produced reports reviewing numerous unstructured grid generation methods. The main advantage of the unstructured grid is that it can be used easily to fit irregular, singly-connected domains, as well as multiply-connected domains. Unstructured grid generation methods have usually followed two approaches, the ‘Advancing Front’ and ‘Delaunay’ methods. The advancing front method fills a domain by creating new triangles at the leading edge of a group of preexisting triangles. The front begins with the discretized airfoil body and outer boundary and proceeds into the computational domain. The advantages of the Advancing front scheme are that the scheme is simple and straightforward, it is relatively easy to implement for numerical applications and it can triangulate concave domains without any difficulty or additional effort. However, the major drawbacks of this scheme are that the scheme is not as efficient as some of the other triangulation schemes and control over grid quality is limited. The Delaunay point insertion method assumes that a valid grid exists and proceeds to insert new points according to the Delaunay criterion. If a circle is drawn through the three nodes at the vertices of a triangle, the Delaunay criterion is met if no other node in the grid is within the circle. The most important aspect of this scheme is its

efficiency, as well as the quality of the generated grid. The disadvantage of the scheme is associated with triangulations of concave domains.

1.3 OBJECTIVES

The objective of this work is to develop a two-dimensional unstructured grid solver that is capable of solving inviscid, viscous and turbulent flows over a wide range of Reynolds numbers and incident angles for single-element airfoils. The solver should be flexible enough in its operation to accommodate all such demands. The solver will operate on triangular grids and it will use a finite-volume type discretization, and an explicit time-marching scheme. To accelerate convergence, residual smoothing and local time stepping will be implemented. Such goals are included in the following list of objectives which are used to guide the development of this solver:

- i. The first objective of this work is to develop an unstructured finite volume solver that produces accurate solutions of inviscid flows over two-dimensional airfoil profiles.
- ii. The second objective of this thesis is to implement the viscous terms of Navier-Stokes equations.
- iii. The third objective of the present study is to implement the Spalart-Allmaras turbulence model.
- iv. The final objective is to solve a wide range of test cases over laminar and turbulent cases on triangular element grids for inviscid, viscous and turbulent flows.

1.4 PRESENT STUDY

The present study is divided into five chapters that cover both the details of the solver and a series of test cases used to evaluate its performance. The solution of the compressible two-dimensional Navier-Stokes equations through the finite-volume methodology is given in Chapter 2. Implementation of the turbulence model and boundary conditions are also reviewed in this chapter. The next chapter concentrates on the discretization of viscous fluxes, the time marching scheme and the solution procedures for the Spalart-Allmaras turbulence model. Solutions to a number of aerodynamic problems for inviscid, viscous and turbulent flows are presented in Chapter 4. The last chapter reviews the significant features and contributions of this research. The conclusions and future directions of research are also presented in Chapter 5.

CHAPTER 2

GOVERNING EQUATIONS

2.1 NAVIER-STOKES EQUATIONS

Navier-Stokes equations describe the conservation of mass and the conservation of momentum. The simultaneous solution of these equations provides the density, the components of the momentum flux in the x and y Cartesian coordinate directions and the total energy of the fluid. In this study, compressible Navier-Stokes equations are considered with respect to a stationary reference frame, with no external body force and heat generation. The governing equations written in two-dimensional conservative form are given by the following expression:

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} = \frac{\partial F_v}{\partial x} + \frac{\partial G_v}{\partial y} + S \quad (2.1)$$

where

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E \end{bmatrix} \quad (2.2)$$

$$F = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(E + p) \end{bmatrix} \quad (2.3)$$

$$G = \begin{bmatrix} \rho \\ \rho uv \\ \rho v^2 + p \\ v(E + p) \end{bmatrix} \quad (2.4)$$

$$F_v = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{yx} \\ \tau_{xx}u + \tau_{xy}v \end{bmatrix} \quad (2.5)$$

$$G_v = \begin{bmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ \tau_{yx}u + \tau_{yy}v \end{bmatrix} \quad (2.6)$$

$$S = [0 \ 0 \ 0 \ 0]^T \quad (2.7)$$

where u and v are the two velocity components in the x and y directions, respectively, ρ is the density, p is the static pressure and E is the total energy per unit volume and is defined as

$$E = \frac{p}{\gamma - 1} + \rho \left(\frac{u^2 + v^2}{2} \right) \quad (2.8)$$

The non-dimensionalization is based on free-stream values. The algorithm is intended for use over a range of initial conditions and airfoil geometries. The input parameters are the free-stream Mach number, incident angle and Reynolds number. To investigate each of these parameters independently, the governing equations are written in non-dimensional form. The chord length of the airfoil, denoted as c , and the free-stream values of density, ρ_∞ , speed of sound, a_∞ , dynamic viscosity, μ_∞ and temperature, T_∞ are used to non-dimensionalize as follows and the details of the non-dimensionalization can be found in literature [31]:

$$x' = \frac{x}{c} \quad y' = \frac{y}{c} \quad (2.9)$$

$$u' = \frac{u}{a_\infty} \quad v' = \frac{v}{a_\infty} \quad (2.10)$$

$$\mu' = \frac{\mu}{\mu_\infty} \quad \rho' = \frac{\rho}{\rho_\infty} \quad (2.11)$$

$$T' = \frac{T}{T_\infty} \quad p' = \frac{p}{\rho_\infty a_\infty^2} \quad (2.12)$$

The prime notation indicates the non-dimensional form of the variable and is omitted from all subsequent variable references. Note that the Reynolds number Re_a is based on the free-stream speed of sound, and differs from the traditional Reynolds number, Re . This relation is a Reynolds number based on the free-stream speed of sound and chord length. It is related to the traditional form of the Reynolds number, Re through the free-stream Mach number, M_∞ .

$$Re = \frac{\rho_\infty c u_\infty}{\mu_\infty}, \quad M_\infty = \frac{u_\infty}{a_\infty} \quad (2.13)$$

$$Re = M_\infty Re_a, \quad Re_a = \frac{\rho_\infty c a_\infty}{\mu_\infty} = \frac{Re}{M_\infty} \quad (2.14)$$

The shear stress components are expressed as;

$$\tau_{xx} = (\mu + \mu_t) \frac{M_\infty}{Re} \left(\frac{4}{3} \frac{\partial u}{\partial x} - \frac{2}{3} \frac{\partial v}{\partial y} \right) \quad (2.15)$$

$$\tau_{yy} = (\mu + \mu_t) \frac{M_\infty}{Re} \left(\frac{4}{3} \frac{\partial v}{\partial y} - \frac{2}{3} \frac{\partial u}{\partial x} \right) \quad (2.16)$$

$$\tau_{yx} = (\mu + \mu_t) \frac{M_\infty}{Re} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \quad (2.17)$$

$$\tau_{xy} = \tau_{yx} \quad (2.18)$$

where u and v are the velocity components in the x and y directions, respectively, ρ is the density, p is the static pressure, μ is the laminar viscosity, μ_t is the turbulent viscosity, M_∞ is the free-stream Mach number and Re is the Reynolds number.

Pressure is related to the conservative flow variables U by the equation of state for a perfect gas:

$$p = (\gamma - 1) \left(e - \frac{1}{2} \rho (u^2 + v^2) \right) = \rho R T \quad (2.19)$$

Sutherland's law is used to relate the kinematic viscosity μ to the temperature

$$\frac{\mu}{\mu_\infty} = \left[\frac{T_\infty + 110}{T + 110} \right] \left(\frac{T}{T_\infty} \right)^{\frac{3}{2}} \quad (2.20)$$

where T stands for the temperature and T_∞ is the free-stream temperature. T and T_∞ are expressed in Kelvin.

2.2 TURBULENCE MODELING

In reality, aircraft wing sections are used exclusively in flow regimes that are characterized by Reynolds numbers well above the critical value for turbulence. Simulation of many complex features of the flows of practical importance needs the capture of its turbulent behavior. Viscosity plays a major role in those cases and the randomness of the flow can be modeled using turbulence modeling. Viscosity can be viewed as consisting of two contributions: laminar and turbulent. The laminar viscosity is usually a function of temperature and can be estimated using Sutherland's formula [31]. On the other hand, the turbulent viscosity is a function of the flow and needs to be evaluated by using one of the turbulence models. In this thesis, the turbulent viscosity is estimated by using the Spalart-Allmaras [14] one equation model in high Reynolds number flows. The Spalart-Allmaras turbulence model used in this algorithm solves a differential expression for turbulence parameters \tilde{v} . The model in conservative non-dimensional form is

$$\begin{aligned}
\frac{\partial \tilde{v}}{\partial t} + u \frac{\partial \tilde{v}}{\partial x} + v \frac{\partial \tilde{v}}{\partial y} = & \frac{C_{b1}}{Re_a} [1 - f_{t2}] \tilde{S} \tilde{v} + \\
& \frac{1}{Re_a} \frac{1}{\sigma} \left[\nabla \cdot ((v + \tilde{v}) \nabla \tilde{v}) + C_{b2} (\nabla \tilde{v})^2 \right] - \\
& \frac{1}{Re_a} \left[C_{w1} f_w - \frac{C_{b1}}{\kappa^2} f_{t2} \right] \left[\frac{\tilde{v}}{d} \right]^2 + Re_a f_{t1} \Delta U^2
\end{aligned} \tag{2.21}$$

The dynamic eddy viscosity ν_t is obtained from

$$\nu_t = \tilde{v} f_{v1} \tag{2.22}$$

where the viscous damping function f_{v1} is given by

$$f_{v1} = \frac{X^3}{X^3 + C_{v1}^3} \tag{2.23}$$

with

$$X = \frac{\tilde{v}}{v} \tag{2.24}$$

The turbulent viscosity, μ_t is computed from

$$\mu_t = \rho \tilde{v} f_{v1} \tag{2.25}$$

The production term $\frac{C_{b1}}{\text{Re}_a} [1 - f_{t2}] \tilde{S} \tilde{v}$ is modeled with

$$\tilde{S} = S f_{v3} + \frac{\tilde{\nu}}{\kappa^2 d^2} f_{v2} \quad (2.26)$$

$$f_{v2} = (1 + \frac{X}{C_{v2}})^{-3} \quad (2.27)$$

$$f_{v3} = \frac{(1 + X f_{v1})(1 - f_{v2})}{\max(X, 0.001)} \quad (2.28)$$

$$f_{t2} = c_{t3} \exp(-c_{t4} X^2) \quad (2.29)$$

C_{b1}, C_{t4}, C_{t3} and κ are constants, d is the distance from the wall, κ is the von Karman constant. The turbulence production variable S is defined as the magnitude of the vorticity in the flow field

$$S = \sqrt{2\Omega_{ij}\Omega_{ij}} \quad (2.30)$$

where Ω_{ij} is the mean rate of rotation tensor and is defined by

$$\Omega_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i} \right) \quad (2.31)$$

The justification for the default expression for S is that when the model was formulated, turbulence is found only near walls where vorticity is generated. On the other hand, it has been acknowledged that one should also take into consideration the effects of the mean strain on the turbulence production.

$$S = |\Omega_{ij}| + C_{prod} \min(0, |S_{ij}| - |\Omega_{ij}|) \quad (2.32)$$

where $C_{prod} = 2.0$, $|\Omega_{ij}| = \sqrt{2\Omega_{ij}\Omega_{ij}}$, $|S_{ij}| = \sqrt{2S_{ij}S_{ij}}$

with the main strain rate S_{ij} , is defined as

$$S_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad (2.33)$$

In the original version of Spalart-Allmaras turbulence modeling, \tilde{S} was defined in such a way that it could become negative. This case could disturb the closure coefficient value of r and stall the convergence. Therefore, the function f_{v3} is introduced and the definition of the function f_{v2} has been changed. This modification was suggested by Spalart to prevent \tilde{S} from being negative.

The variables in the destruction term $\frac{1}{Re_a} \left[C_{w1} f_w - \frac{C_{bl}}{\kappa^2} f_{t2} \right] \left[\frac{\tilde{v}}{d} \right]^2$ are given as follows:

$$f_w(r) = g \left[\frac{1 + C_{w3}^6}{g^6 + C_{w3}^6} \right]^{1/6} \quad (2.34)$$

$$g = r + C_{w2}(r^6 - r) \quad (2.35)$$

$$r = \frac{\tilde{v}}{\tilde{S} \kappa^2 d^2} \quad (2.36)$$

where C_{w1} , C_{w2} and C_{w3} are constants. For large r , the function f_w reaches a constant, so large values of r can be truncated to 10, the details can be found in literature [14].

The trip term $\text{Re}_a f_{t1} \Delta U^2$ that allows one to specify the boundary layer transition location explicitly can be expressed with

$$f_{t1} = c_{t1} g_t \exp(-c_{t2} \frac{w_t^2}{\Delta U^2} (d^2 + g_t^2 d_t^2)) \quad (2.37)$$

where D_t is the distance to the nearest trip point, w_t is the vorticity at the wall at the trip point, ΔU is the norm of the difference between the velocity at the trip (zero if the wall stationary) and $g_t = \min(0.1, \Delta U / \omega_t \Delta x)$ with Δx being the spacing along the wall at the trip point.

Although the model allows the transition point localization, computations in the present study are assumed to be fully turbulent, so this term is not used. Therefore, the values of f_{t1} and f_{t2} , which are associated with these terms, will be taken as zero. The various constants used in this model have the following values:

$$C_{b1} = 0.1355, C_{b2} = 0.622, \sigma_v = \frac{2}{3}, C_{v1} = 7.1, C_{w2} = 0.3,$$

$$\kappa = 0.41, C_{t4} = 2, C_{t3} = 1.1, C_{t2} = 2, C_{t1} = 1,$$

$$C_{w1} = \frac{C_{b1}}{\kappa^2} + \frac{1}{\sigma_v} (1 + C_{b2}) \quad (2.38)$$

2.3 BOUNDARY CONDITIONS

Proper boundary conditions must be specified to obtain a unique and accurate solution. The provided information about the boundary conditions will determine the problem to be solved. The Navier-Stokes equations contain second-order viscous stress terms and gradients of flow variables. The presence of these terms allow the specification of not only Dirichlet boundary conditions but also Neumann boundary conditions. The boundary conditions for different flow situations are discussed below.

2.3.1. Free-Stream Boundary Conditions

Free-stream boundary conditions specify the flow-field conditions that would exist in the absence of an airfoil or other body. Several conditions are specified by the user and several others can be calculated from the user supplied parameters. The free-stream conditions are used to initialize the entire flow field before the solution starts and used to calculate the farfield boundary conditions throughout the solution process. The free-stream boundary conditions provided in non-dimensional form. The free stream velocity components are determined from the angle of attack α and the free-stream Mach number M_∞ which are both supplied to the code by the user as:

$$u_\infty = M_\infty \cos\alpha , \quad (2.39)$$

$$v_\infty = M_\infty \sin\alpha \quad (2.40)$$

The free-stream pressure, density and speed of sound can be derived from the non-dimensionalization being used.

$$\rho_\infty = 1, \quad p_\infty = \frac{1}{\gamma}, \quad a_\infty = 1 \quad (2.41)$$

The free-stream total energy can then be calculated from equation (2.18) as:

$$E_\infty = \frac{p_\infty}{\gamma - 1} + \frac{1}{2} \rho_\infty (u_\infty^2 + v_\infty^2) \quad (2.42)$$

2.3.2 Far Field Boundary Conditions

The selection of computational domain for external flows will include artificial boundaries set at the far field which could be either inflow or outflow. The airflow over the airfoil generates disturbances in the free-stream conditions and these disturbances are convected and diffused throughout the domain eventually reaching the outer boundaries. The location of far field boundary should be set as far away as possible. However, placing the far field boundary more than 100 chords from the airfoil surface is not an applicable option. Since, it requires either a large number of nodes in the outer regions near the boundary or much stretched cells, both of which may reduce the performance of the code. Therefore, from efficiency and accuracy points of view, the far field boundary must be set reasonably to reduce the size of the computational domain. Furthermore, non-reflecting boundary conditions must be used to allow the disturbances to leave the computational domain otherwise they could reflect leading to large oscillations and preventing the convergence of the solver. Riemann invariants are the well-known non-reflecting boundary conditions. Riemann invariants are given as:

$$R_1 = U_n - \frac{2a}{\gamma - 1} \quad (2.43)$$

$$R_2 = U_n + \frac{2a}{\gamma - 1} \quad (2.44)$$

$$R_3 = \frac{p}{\rho^\gamma} \quad (2.45)$$

$$R_4 = U_t \quad (2.46)$$

where U_n is the flow velocity normal to the boundary and U_t is the flow velocity tangential to the boundary. For the flow leaving the solution domain U_n is positive and for the flow entering the solution domain U_n is negative.

The four Riemann invariants are either calculated from free-stream values or extrapolated from interior values by simple zeroth order extrapolation. Boundary conditions are usually implemented by ghost cells. The process is summarized in Table 2.1.

Table 2.1 Riemann invariants calculation and extrapolation

Boundary type	Normal Velocity	Free-stream conditions	Extrapolated conditions
Subsonic Inflow	$-a < U_n < 0$	R_1, R_3, R_4	R_2
Subsonic Outflow	$0 < U_n < a$	R_1	R_2, R_3, R_4
Supersonic Inflow	$U_n < -a$	R_1, R_2, R_3, R_4	
Supersonic Outflow	$a < U_n$		R_1, R_2, R_3, R_4

Since the far field boundary is at a limited distance from the airfoil the flow-field at the boundary will not exactly match with the free-stream values. Simply imposing the free-stream Riemann invariant values can cause an inaccurate solution leading to inaccurate lift and drag values. On the other hand, a correction can be imposed on the far field boundary conditions to simulate a larger computational domain. The correction approximates the effects of the airfoil as a point vortex located at the quarter-chord point. This technique is called a circulation correction. The circulation is used to calculate corrected free-stream velocities and speed of sound for point at the far field boundary. The new velocities and speed of sound should be replaced with the free-stream values in the calculation of the Riemann invariants. However, the circulation correction topic is beyond the scope of this study so it can be implemented in future to develop the present solver and to work on a larger computational domain. The detailed discussion of the circulation correction can be found in Thomas *et al.* [32].

2.3.3 Solid Wall Boundary Conditions

In viscous flow problems, such as air moving over an airfoil surface, the region adjacent to the airfoil inside the boundary layer will be dominated by the effects of viscosity. The boundary conditions on the airfoil surface are chosen to accurately reflect such phenomena. In real flows, the intermolecular forces between the fluid and airfoil surface will give rise to the ‘no-slip’ condition, where the velocity of the fluid drops to a negligible value [35]. In the numerical code this allows the velocity components on the surface of the airfoil to be set to zero. By defining the velocities on the airfoil surface to rigid values, the momentum equations are not required to be solved over control volumes centered about surface nodes, saving computational resources. The turbulent eddy viscosity $\tilde{\nu}$ is also constrained by the effects of the wall. On the airfoil surface due to turbulence effects are reduced, the eddy viscosity can be set to $\tilde{\nu} = 0$ [14].

The temperature on the wall can be specified as:

$$T = T_{wall} \quad (2.47)$$

or

$$-k \frac{\partial T}{\partial n} = q_{wall} \quad (2.48)$$

where q_{wall} is the wall heat flux.

In this study walls are modeled as adiabatic, $q_{wall} = 0$ and equation (2.48) simplifies to

$$\frac{\partial T}{\partial n} = 0 \quad (2.49)$$

At the solid boundary with no slip boundary condition, the pressure at the wall is obtained by solving the momentum equation in the normal direction, which reduces to the following form

$$\frac{\partial p}{\partial n} = 0 \quad (2.50)$$

CHAPTER 3

NUMERICAL METHOD

The solution procedures of the governing equations that are presented in the previous chapter are described in the following sections. The data structure used to store the grid information is discussed in Section 3.1. The numerical discretization of the convective terms and viscous flux calculations are addressed in Section 3.2. Turbulence model discretization is discussed in Section 3.3. Lastly, Section 3.4 deals with the temporal discretization including Runge-Kutta explicit scheme, local time stepping and residual smoothing.

3.1 CELL BASED DATA STRUCTURE

The grid information can be stored in two different ways (i) Cell based data structure (ii) Cell face based data structure. In two-dimensional cases the cell face based data structure, which is used in three-dimensional cases, will change to an edge based one. Cell based data structure lists the connectivity of each cell to the vertices and the neighbouring cells. On the other hand, the edge based data structure lists the connectivity of each edge to its vertices and neighbours. In the present work, cell based data structure will be used to store the grid information. Consider a two dimensional grid that consists of four cells, as shown in Figure 3.1.

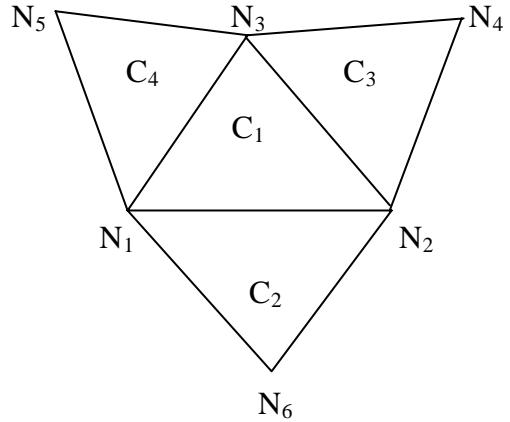


Figure 3.1 Example of unstructured triangular grids

The nodes are denoted by $N_1, N_2, N_3, N_4, N_5, N_6$ and the cells by C_1, C_2, C_3, C_4 . In the cell based data structure the grid is represented as shown in Table 3.1. The nodes that form a cell are usually ordered in either a clockwise or a counter-clockwise direction. In this study, the counter-clockwise ordering of nodes was chosen.

Table 3.1 Cell based data structure

Cell Number	Vertex 1	Vertex 2	Vertex 3
C_1	N_1	N_2	N_3
C_2	N_1	N_6	N_2
C_3	N_2	N_4	N_3
C_4	N_3	N_5	N_1

In the present thesis, to store grid information one should also supply the connectivity of each cell to the vertices and neighbour cells. This approach is best explained by an example.

Consider the two dimensional grid which consist of ten cells as shown in Figure 3.2.

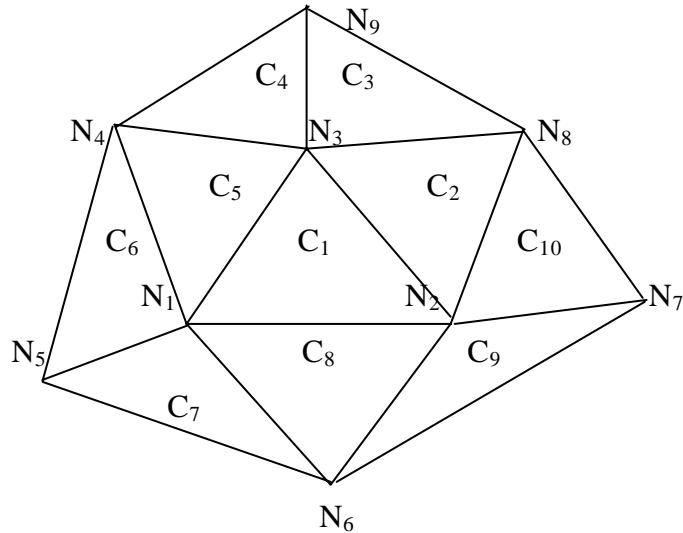


Figure 3.2 Example of connectivity of the neighbouring cells

The nodes are denoted by N_1, N_2, \dots, N_9 and the cells by C_1, C_2, \dots, C_{10} . In the cell based data structure the connectivity of each cell to the vertices and neighbouring cells is represented as shown in Table 3.2.

Table 3.2 Connectivity of cells to the vertices and neighbouring cells

Cell Number	Vertex 1	Vertex 2	Vertex 3	Neigh. 1	Neigh. 2	Neigh. 3
C_1	N_1	N_2	N_3	C_2	C_5	C_8
C_2	N_2	N_8	N_3	C_3	C_1	C_{10}
C_5	N_3	N_4	N_1	C_6	C_1	C_4
C_8	N_2	N_1	N_6	C_7	C_9	C_1

Note that in the present solver, for the far field (outer) boundary condition value of -1 is used instead of any cell number and for the inner boundary condition, i.e. on the airfoil surface, value of -2 is used.

3.2 SPATIAL DISCRETIZATION

The governing Navier-Stokes equations presented in the previous chapter are written in a differential form describing a continuum flow field. To extend the conservation of the dependent variables to arbitrary regions of the real space domain, it is necessary to write these relations in integral form. Considering an arbitrary control volume Ω , integration of a generalized conservation equation yields the following expression;

$$\int_{\Omega} \frac{\partial \rho \phi}{\partial t} dx dy + \int_{\Omega} \nabla \cdot H ds = \int_{\Omega} S dx dy \quad (3.1)$$

where

$$H = F + G - (F_v + G_v) \quad (3.2)$$

In this relation, ϕ represents any conservative variable on a per unit mass basis, such as the x velocity component u , the y velocity component v , the total energy $\frac{E}{\rho}$ or the turbulence variable \tilde{v} and in the mass conservation equation the value ϕ will be 1. In equation (3.1) H term stands for both convective and diffusive fluxes for each dependent variable. The S term in equation (3.1) represents the source terms for each conservation equations. For laminar or turbulent flows, the mass, momentum and energy equations will not contain source terms. On the other hand, for turbulent flows that solve the turbulence variable, source terms will be presented.

The integral equation (3.1) can be further simplified so that it is more reasonable to discretization methods. The control volumes over which the integral equation is solved are fixed in space with no movement of nodes or

edges during solution. The Leibnitz theorem then allows the term $\int_{\Omega} \frac{\partial \rho \phi}{\partial t} dx dy$ to be written as a function of time. The divergence term can also

be expressed in a simplified format. Gauss' theorem in the plane allows the integral of the flux divergence to be expressed as a line integral of the flux through the control volume boundaries [33]. This allows a simplified evaluation of the convective and diffusive fluxes through the reduction of the order of differentiation. Instead of calculating the spatial derivatives of the convective fluxes, the simplified form only requires the calculation of the flux components along the outer boundaries of the control volume. Representing the outer boundary of the arbitrary control volume as $\partial\Omega$ the integral equation (3.1) can be written as follows;

$$\frac{d}{dt} \int_{\Omega} \rho \phi dx dy + \iint_{\partial\Omega} H ds = \int_{\Omega} S dx dy \quad (3.3)$$

Equation (3.3) is the starting point for the transformation from a completely analytical problem to a discrete one. The integrals of the equation are approximated as sum and products of the flux terms using a process commonly referred to as discretization, thus making them suitable for solution through numerical means. The temporal and spatial terms can be addressed separately, allowing a solution process to use various combinations of methods.

The finite volume method begins with the discretization of equation (3.3). The dependent variables $\rho\phi$ in the first term of the equation are continuous over each control volume. The discretization commences with the replacement of this continuous variable with one that represents the average value over the control volume. The first term can then be approximated by

$$\frac{d}{dt}(\rho\phi A) \quad (3.4)$$

where $\rho\phi$ is the average value over the control volume. The term A is the area of the control volume, which is constant and can be removed from the temporal differentiation. The source term of equation (3.3) has an identical formulation to the temporal term and can be treated in the same manner.

The term $\iint_{\Omega} H.ds$ in equation (3.3) represents the net flux through the

closed boundary of the control volume. In its present form, this term requires that exact expressions for the boundary fluxes be provided to complete its evaluation. Therefore, it must also be approximated to allow a numerical formulation. This implies that some assumption must be made about the representation of the flux across the control volume boundaries. One of the most common methods is to use a trapezoidal formulation to approximate the integral. A linear interpolation of the flux along an edge is made based on values obtained at either end of the edge. As an example, if the flux at the midpoint of an edge j is denoted as f_j , and the flux values at either end of the edge f_i, f_{i+1} are evaluated using both convective and diffusive fluxes, then

$$f_j = \frac{1}{2}(f_i + f_{i+1}) \quad (3.5)$$

This method of flux evaluation can be performed very quickly using only the information at the ends of each edge. This allows a numerical algorithm based on this formulation to be computationally efficient in terms of computer effort and memory.

A finite volume based scheme by Barth [34] and Whitaker [35] is one of the common approaches used for solving the governing equations on unstructured grids. There are basically two different approaches for storing the conserved variables; one is a node based approach while the other one is a cell based approach. The shaded regions in Figure 3.3 and 3.4 show the areas that are used for estimating the averaged values of the conserved variables for the two different approaches. In the node based method the area considered for this averaging is the area around the nodes. For two dimensional cases this area is taken as the area enclosed by the polygon formed by connecting the cell centers of the neighbouring cells to the midpoint of the corresponding edges. In the second approach, the values stored at the cell center are taken as the cell averaged values. The area for the averaging is taken as that of the cell itself.

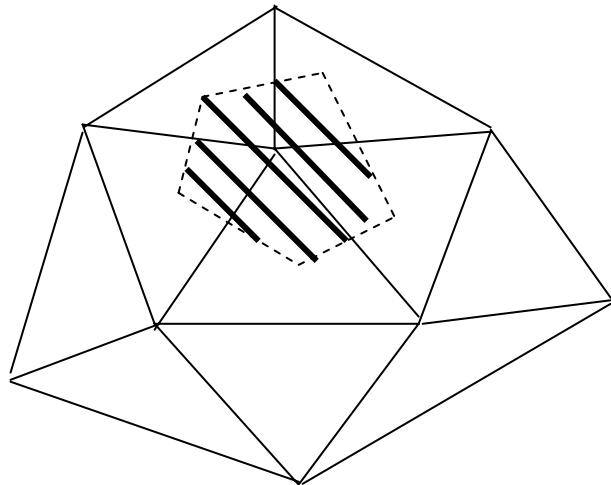


Figure 3.3 Areas considered for averaging the conserved variable in node based scheme

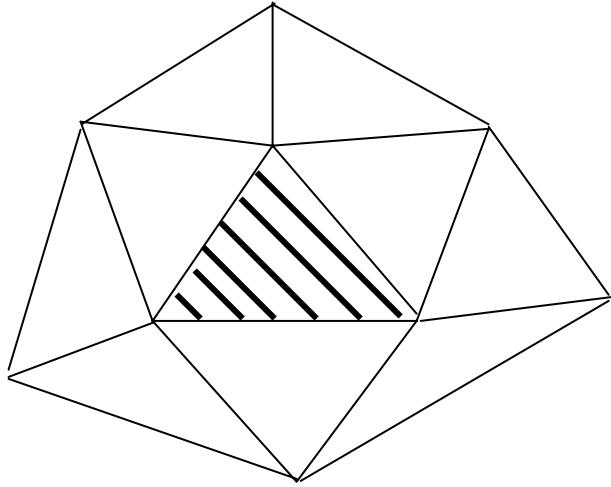


Figure 3.4 Areas considered for averaging the conserved variable in cell based scheme

In the present study, the cell centered approach is used for storing the conserved variables and for the subsequent solution procedure.

3.2.1 Discretization of Convective Terms

The governing equations in differential form without the viscous and source terms are as follow

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} = 0 \quad (3.6)$$

Integrated equation (3.6) over an triangular mesh. Thus, one has

$$\int_{\Omega} \left(\frac{\partial U}{\partial t} \right) dx dy + \int_{\Omega} \left(\frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} \right) dx dy = 0 \quad (3.7)$$

Using Green's Theorem which converts area integrals to the line integral. Equation (3.7) can be written as

$$\int_{\Omega} \left(\frac{\partial U}{\partial t} \right) dx dy + \oint_{\Omega} (F dy - G dx) = 0 \quad (3.8)$$

Using the cell centered finite volume approach, a discretized form of equation (3.8) is

$$(U^{n+1}_{center} - U^n_{center}) = - (F_{ij} \Delta y - G_{ij} \Delta x) \quad (3.9)$$

The indices i and j denote the cell and face numbers respectively, n stands for the time level, Δy and Δx represent the increment. The right-hand-side of equation (3.9) represents the flux balance for the i^{th} cell.

The differential form of the equation (3.6) is hyperbolic in time; so that the flux evaluation is based on the direction of propagation of the information. This is done using unwinding based on the eigenvector of the governing equations.

The numerical flux crossing a cell face is calculated as the exact solution of the approximate Riemann problem by Roe [36]. The basic principle behind the Roe's approximate Riemann solver is explained below using the one dimensional hyperbolic system of conservation laws,

$$\frac{\partial q}{\partial t} + \frac{\partial f}{\partial x} = 0 \quad (3.10)$$

with the initial conditions given in Figure 3.5

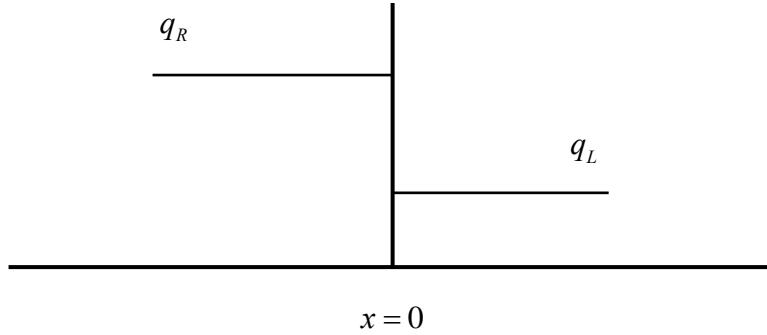


Figure 3.5 Initial conditions for the conservation laws

The above conservation laws can be written as

$$\frac{\partial q}{\partial t} + a \frac{\partial q}{\partial x} = 0 \quad (3.11)$$

$$\text{where } a = \frac{\partial f}{\partial q}$$

Roe's approximate Riemann solver finds the exact solution of the following linear hyperbolic system.

$$\frac{\partial q}{\partial t} + \bar{a} \frac{\partial q}{\partial x} = 0 \quad (3.12)$$

where \bar{a} is a locally constant matrix and has to satisfy the following properties [39].

- v. In order to satisfy the hyperbolicity of the system, \bar{a} should have real eigenvalues.
- vi. In order to be consistent with the original Jacobian matrix, $a(Q)$, \bar{a} should exhibit the behaviour $\bar{a} = a(Q)$.

- vii. In order to satisfy the conservation property, \bar{a} should be chosen to be $F(Q_R) - F(Q_L) = \bar{a}(Q_R - Q_L)$.

The jump in the values of the flow variables across the face and is defined as

$$\Delta(\) = (\)_R - (\)_L.$$

3.2.2 Viscous Flux Calculations

In the present work, the solver uses a finite volume discretization. The Navier-Stokes equations (2.1) can be discretized by integrating around the boundary of each control volume. Gauss's divergence theorem in two dimensions allows us to transform the area integrals in equation (3.13) into surface integrals around the boundary of the control volume:

$$\frac{d}{dt} \int_{\Omega} U dA + \oint_{\partial\Omega} (F + G) ds = \frac{1}{Re_a} \oint_{\partial\Omega} (F_v + G_v) ds + \int_{\Omega} S dA \quad (3.13)$$

where U represents the vector of conserved variables, F and G are the inviscid flux, F_v and G_v are the viscous flux and the turbulence model source terms are represented in S :

$$S = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ S_t \end{bmatrix} \quad (3.14)$$

The solver uses several approximations, using these approximations the Navier-Stokes equation (3.13) can be simplified as follow;

$$\Omega \frac{d}{dt} U + \iint_{\Omega} (F + G) ds = \frac{1}{Re_a} \iint_{\Omega} (F_v + G_v) ds + AS \quad (3.16)$$

with the fact that the grid does not change in time. The line integrals in equation (3.16) may now be discretized as the sum of line integrals over each edge in the control volume.

The flux is assumed to be constant along each control volume edge and the value used is the average from the two primary grid nodes on either side of the edge. The averaging is a good approximation for the viscous stresses, since they are diffusive in nature. The equations are written in spatially discrete form as:

$$\begin{aligned} \Omega \frac{d}{dt} U + \sum_{j=1}^n (F_{xj} \delta y - F_{yj} \delta x) + \\ \sum_{j=1}^n (G_{xj} \delta y - G_{yj} \delta x) = \\ \frac{1}{Re_a} \left[\sum_{j=1}^n ((F_v)_{xj} \delta y - (F_v)_{yj} \delta x) \right] \\ + \frac{1}{Re_a} \left[\sum_{j=1}^n ((G_v)_{xj} \delta y - (G_v)_{yj} \delta x) \right] + AS \\ + AS \end{aligned} \quad (3.17)$$

where the sums are over the edges in the control volume and the fluxes at each edge are calculated from

$$F_{xj} = \frac{(F_{xi} + F_{xk})}{2}, \quad (F_v)_{xj} = \frac{((F_v)_{xi} + (F_v)_{xk})}{2} \quad (3.18)$$

$$F_{yj} = \frac{(F_{yi} + F_{yk})}{2}, \quad (F_v)_{yj} = \frac{((F_v)_{yi} + (F_v)_{yk})}{2} \quad (3.19)$$

$$G_{xj} = \frac{(G_{xi} + G_{xk})}{2}, \quad (G_v)_{xj} = \frac{((G_v)_{xi} + (G_v)_{xk})}{2} \quad (3.20)$$

$$G_{yj} = \frac{(G_{yi} + G_{yk})}{2}, \quad (G_v)_{yj} = \frac{((G_v)_{yi} + (G_v)_{yk})}{2} \quad (3.21)$$

where j stands for the current edge, i is the central node and k is the neighbour associated with the edge j . The edge components δx and δy are precomputed for each edge and the control volume boundary is traversed in the counter clock wise direction, as shown in Figure 3.6.

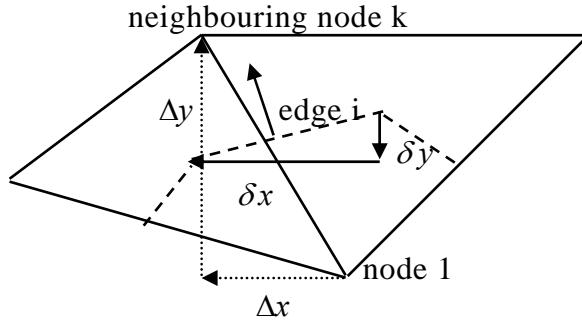


Figure 3.6 One edge of a control volume

In order to compute the viscous fluxes F_v and G_v flow variable derivatives i.e. velocity gradients must be calculated. Velocity gradients are calculated using Green's Gradient Theorem which is given as follows;

$$\nabla \phi = \frac{1}{\Omega} \int_{\partial\Omega} \phi ds \quad (3.22)$$

and after discretization of equation (3.22) following form is obtained;

$$\nabla \phi = \frac{1}{\Omega} \sum_1^3 \phi_j (\Delta y_{i,j} \vec{i} - \Delta x_{i,j} \vec{j}) \quad (3.23)$$

where ϕ represents the velocity variable.

3.3 TURBULENCE MODEL DISCRETIZATION

The turbulence model is discretized in the same way as the Navier-Stokes flux terms. The turbulence model contains mostly first derivatives of flow variables which can easily be computed as in equation (3.23). However, the diffusion term contains second derivative terms;

$$\frac{1}{Re_a} \frac{1}{\sigma} [\nabla \cdot ((\nu + \tilde{\nu}) \nabla \tilde{\nu}] \quad (3.24)$$

In the present study these derivatives are calculated in the same way as the Navier-Stokes viscous fluxes. The Spalart-Allmaras model solves a second order partial differential equation for the variable $\hat{\tilde{\nu}}$ and the turbulent kinematic viscosity $\tilde{\nu}_t$ is estimated from $\hat{\tilde{\nu}}$ by multiplying a damping function f_{v1} . The \wedge represents dimensional quantities. The Spalart-Allmaras one equation turbulence model in dimensional form is given in equation (3.25) [14],

$$\begin{aligned} \frac{D\hat{\tilde{\nu}}}{Dt} &= C_{b1} (1 - f_{t2}) \hat{S} \hat{\tilde{\nu}} + \frac{1}{\sigma} \left(\nabla \cdot (\hat{\tilde{\nu}} + \hat{\tilde{\nu}}) \nabla \hat{\tilde{\nu}} \right) + C_{b2} (\nabla \hat{\tilde{\nu}})^2 \\ &- \left(C_{w1} \hat{f}_w - \frac{C_{b1}}{x^2} f_{t2} \right) \left(\frac{\hat{\tilde{\nu}}}{\hat{d}} \right)^2 + f_{t1} (\Delta \hat{U})^2 \end{aligned} \quad (3.25)$$

where $\hat{\tilde{\nu}}$ is the laminar kinematic viscosity and \hat{d} is the distance to the nearest wall. For simplicity, the tripping terms are neglected and the

equation is solved in the fully turbulent mode. This is done by setting f_{t1} and f_{t2} to zero. Equation (3.25) then becomes

$$\begin{aligned} \frac{D\hat{\tilde{v}}}{Dt} &= C_{b1}\hat{S}\hat{\tilde{v}} + \frac{1}{\sigma}\left(\nabla.\left(\left(\hat{v} + \hat{\tilde{v}}\right)\nabla\hat{\tilde{v}}\right) + C_{b2}\left(\nabla\hat{\tilde{v}}\right)^2\right) \\ &\quad - C_{wl}\hat{f}_w\left(\frac{\hat{\tilde{v}}}{\hat{d}}\right)^2 \end{aligned} \quad (3.26)$$

Having expanded the total derivative, equation (3.36) is rewritten as;

$$\begin{aligned} \frac{\partial\hat{\tilde{v}}}{\partial\hat{t}} &= -\underline{V}.\nabla\hat{\tilde{v}} + C_{b1}\hat{S}\hat{\tilde{v}} + \frac{1}{\sigma}\left(\nabla.\left(\left(\hat{v} + \hat{\tilde{v}}\right)\nabla\hat{\tilde{v}}\right) + C_{b2}\left(\nabla\hat{\tilde{v}}\right)^2\right) \\ &\quad - C_{wl}\hat{f}_w\left(\frac{\hat{\tilde{v}}}{\hat{d}}\right)^2 \end{aligned} \quad (3.27)$$

where \underline{V} is the velocity variable.

The above equation has to be non-dimensionalized before it can be solved numerically, since the governing equations are in non-dimensional form. The non-dimensional parameters are the same as those used for non-dimensionalization of the Navier-Stokes equations. After non-dimensionalization, equation (3.25) becomes

$$\begin{aligned} \frac{\partial\tilde{v}}{\partial\hat{t}} &= -\underline{V}.\nabla\tilde{v} + C_{b1}\bar{S}\tilde{v} + \frac{1}{\sigma Re_a}\left(\nabla.\left(\left(v + \tilde{v}\right)\nabla\tilde{v}\right)\right) \\ &\quad + \frac{C_{b2}}{\sigma Re_a}\left(\nabla\tilde{v}\right)^2 - \frac{C_{wl}f_w}{Re_a}\left(\frac{\tilde{v}}{d}\right)^2 \end{aligned} \quad (3.28)$$

Correspondingly the variables appearing in the above equations will be modified as

$$X = \frac{\tilde{\nu}}{\nu}, \quad \nu_t = \tilde{\nu} f_{v1},$$

$$f_{v1} = \frac{X^3}{X^3 + C_{v1}^3}, \quad f_{v2} = 1 - \frac{X}{1 + X f_{v1}},$$

$$\bar{S} = S + \frac{1}{Re_a} \left(\frac{\tilde{\nu}}{x^2 d^2} \right) f_{v2}, \quad f_w = g \left[\frac{1 + C_{w3}^6}{C_{w3}^6} \right]^{\frac{1}{6}},$$

$$g = r + C_{w2} (r^6 - r), \quad r = \frac{1}{Re_a} \left(\frac{\tilde{\nu}}{\bar{S} x^2 d^2} \right),$$

$$C_{b1} = 0.1355, \quad \sigma = \frac{2.0}{3.0}, \quad C_{b2} = 0.622, \quad x = 0.41,$$

$$C_{w1} = \frac{C_{b1}}{x^2} + \frac{1 + C_{b2}}{\sigma}, \quad C_{w2} = 0.3,$$

$$C_{w3} = 2.0 \text{ and } C_{v1} = 7.1 \quad (3.29)$$

The different terms appearing in equation (3.28) can be divided into convective, production, diffusion and destruction terms. So the transport equation for the turbulent viscosity parameter can be given as

$$\begin{aligned} \text{Time rate of change of viscosity } (\tilde{\nu}) &= -\text{Convection of } \tilde{\nu} + \text{Production of } \tilde{\nu} \\ &\quad + \text{Diffusion of } \tilde{\nu} - \text{Destruction of } \tilde{\nu} \end{aligned}$$

where the convection is given by $\underline{V} \cdot \nabla \tilde{\nu}$. Production of the turbulent viscosity is due to vorticity and is given by $C_{bl} \bar{S} \tilde{\nu}$. The diffusion term is a function of the gradient of the viscosity parameter and is defined as

$$\frac{1}{\sigma Re_a} (\nabla \cdot ((\nu + \tilde{\nu}) \nabla \tilde{\nu})) + \frac{C_{b2}}{\sigma Re_a} (\nabla \tilde{\nu})^2 \quad (3.30)$$

Neglecting the gradient of laminar viscosity, the diffusion terms can be written as,

$$\frac{1+C_{b2}}{\sigma Re_a} (\nabla \cdot ((\nu + \tilde{\nu}) \nabla \tilde{\nu})) - \frac{C_{b2}}{\sigma Re_a} (\nu + \tilde{\nu}) \nabla^2 \tilde{\nu} \quad (3.31)$$

The destruction term for the turbulent viscosity parameter is taken as inversely proportional to the square of the distance to the solid wall and is given by,

$$\frac{C_{w1} f_w}{Re_a} \left(\frac{\tilde{\nu}}{d} \right)^2 \quad (3.32)$$

Equation (3.28) with the simplified diffusion terms can be written as,

$$\begin{aligned} \frac{\partial \tilde{\nu}}{\partial \hat{t}} = & -\underline{V} \cdot \nabla \tilde{\nu} + C_{bl} \bar{S} \tilde{\nu} + \frac{(1+C_{b2})}{\sigma Re_a} (\nabla \cdot ((\nu + \tilde{\nu}) \nabla \tilde{\nu})) \\ & + \frac{C_{b2}}{\sigma Re_a} (\nu + \tilde{\nu})^2 \nabla^2 \tilde{\nu} - \frac{C_{w1} f_w}{Re_a} \left(\frac{\tilde{\nu}}{d} \right)^2 \end{aligned} \quad (3.33)$$

Integrating equation (3.33) over a control volume yields;

$$\begin{aligned}
\int_{\Omega} \frac{\partial \tilde{v}}{\partial t} d\Omega &= - \int_{\Omega} \underline{V} \cdot \nabla \tilde{v} d\Omega + \int_{\Omega} C_{b1} \bar{S} \tilde{v} d\Omega \\
&+ \frac{(1+C_{b2})}{Re_a \sigma} \int_{\Omega} \nabla \cdot ((\nu + \tilde{\nu}) \nabla \tilde{v}) d\Omega \\
&- \frac{C_{b2}}{Re_a \sigma} \int_{\Omega} (\nu + \tilde{\nu}) \nabla^2 \tilde{v} d\Omega - \frac{C_{wl}}{Re_a} \int_{\Omega} f_w \left(\frac{\tilde{\nu}}{d} \right)^2 d\Omega
\end{aligned} \quad (3.34)$$

Using the divergence theorem for the surface integral, equation (3.34) can be transformed to,

$$\begin{aligned}
\int_{\Omega} \frac{\partial \tilde{v}}{\partial t} d\Omega &= - \int_{\Omega} \underline{V} \cdot \nabla \tilde{v} d\Omega + \int_{\Omega} C_{b1} \bar{S} \tilde{v} d\Omega \\
&+ \frac{(1+C_{b2})}{Re_a \sigma} \iint_{\partial\Omega} (\nu + \tilde{\nu}) \nabla \tilde{v} \cdot \underline{n} ds \\
&- \frac{C_{b2}}{Re_a \sigma} \int_{\Omega} (\nu + \tilde{\nu}) \nabla^2 \tilde{v} d\Omega - \frac{C_{wl}}{Re_a} \int_{\Omega} f_w \left(\frac{\tilde{\nu}}{d} \right)^2 d\Omega
\end{aligned} \quad (3.35)$$

As in the case of Navier-Stokes equations, equation (3.35) is also solved using a cell centered scheme. The values of \tilde{v} , stored at the cell centers, are assumed to be cell averaged values. The discretized form of equation (3.35) is

$$\begin{aligned}
\frac{\tilde{\nu}^{n+1} - \tilde{\nu}^n}{\Delta t_i} V_i = & - \sum_{j=1}^k \left(U^+ \tilde{\nu}_i + U^- \tilde{\nu}_j \right) ds_j + C_{bl} (\bar{S} \tilde{\nu})_i V_i \\
& + \frac{(1+C_{b2})}{\sigma \text{Re}_a} \sum_{j=1}^k (\nu + \tilde{\nu})_{ij} (\nabla \tilde{\nu} \cdot \vec{n})_{ij} ds_j \\
& - \frac{C_{b2}}{\sigma \text{Re}_a} (\nu + \tilde{\nu})_i \nabla^2 \tilde{\nu}_i V_i \\
& - \frac{C_{wl}}{Re_a} \left(f_w \left(\frac{\tilde{\nu}}{d} \right)^2 \right)_i \forall_i
\end{aligned} \tag{3.36}$$

where k is the number of edges of cell i , \vec{n} is the unit normal to the face, ds_j is the length of the edge j , $n_{(j)}$ is the cell that shares the j^{th} edge of the cell i and \forall_i is the cell volume. The variables U^+ and U^- are defined as

$$U^+ = \frac{1}{2} (U + |U|) \tag{3.37}$$

$$U^- = \frac{1}{2} (U - |U|) \tag{3.38}$$

where U is the contravariant velocity. The laminar and turbulent viscosities at the cell faces are taken as the average values of those on either side of the cell face.

In equation (3.39) the gradient of the turbulent viscosity parameter $\tilde{\nu}$ is calculated by Green's Gradient Theorem. The same calculation type is performed in equation (3.22).

$$\nabla \tilde{\nu} = \frac{1}{\Omega} \iint_{\partial\Omega} \tilde{\nu} d\mathbf{s} \tag{3.39}$$

3.4 TEMPORAL DISCRETIZATION

This section will address the treatment of the time dependent terms in a manner that permits a solution of the coupled set of governing equations. Explicit time march methods, such as the Runge-Kutta routines rely on the transient nature of the equations to integrate the system from one point in time to the next. The simplicity and low computer memory requirements are the main advantageous of such methods for large problems.

3.4.1 Runge-Kutta Method

For explicit time integration, a four stage Runge-Kutta method is employed as discussed in Jameson *et. al* [4]. The flow variables at $(n+1)^{th}$ time step are obtained from the variables at the $(n)^{th}$ time step in four stages. The Runge-Kutta explicit time marching method can be classified as ‘one-step’ methods since they advance the system in time using only the information provided in the previously computed time step. The start up procedure is also very simple since one-step methods need only an initial solution at $t=0$. The scheme begins with the calculation of the fluxes and source terms using the dependent variables U determined at an initial point at time t_0 . Thus, the dependent variables denoted as U^{t_0} and for example at the second stage in Runge-Kutta scheme by $U^{(2)}$. The residual R over a control volume k at time step t_0 is defined as;

$$R_k(U^{t_0}) = \sum_{j=1}^m (F_{t_j} \Delta y_j - G_{t_j} \Delta x_j) + A_k S \quad (3.40)$$

where the terms Δx_j and Δy_j are the length components of the edge j in the x and y directions, respectively. This flux term is summed over all m

edges that form the outer boundary of control volume k and A_k is the control volume area.

The four stage Runge-Kutta routines computes the dependent variables at the new time level $U^{t_0+\Delta t}$ using the following sequence of relaxation steps,

$$U^{(0)} = U^{t_0}$$

$$U^{(1)} = U^{(0)} - \alpha_1 \frac{\Delta t}{A_k} \left[R_k(U^{(0)}) \right]$$

$$U^{(2)} = U^{(0)} - \alpha_2 \frac{\Delta t}{A_k} \left[R_k(U^{(1)}) \right]$$

$$U^{(3)} = U^{(0)} - \alpha_3 \frac{\Delta t}{A_k} \left[R_k(U^{(2)}) \right]$$

$$U^{(4)} = U^{(0)} - \alpha_4 \frac{\Delta t}{A_k} \left[R_k(U^{(3)}) \right]$$

$$U^{(t_0+\Delta t)} = U^{(4)} \quad (3.41)$$

where Δt is the time step and the coefficients used are $\alpha_1 = 0.15$, $\alpha_2 = 0.3275$, $\alpha_3 = 0.57$ and $\alpha_4 = 1.0$. These weighting coefficients are available in literature and have been determined for structured upwind codes in Turkel *et al.* [37].

3.4.2 Local Time Stepping

Explicit time marching schemes such as the four-stage Runge-Kutta have an inherent limit on the rate at which they can advance in time. This is a direct result of the limited domain of influence prescribed by such schemes. The residuals that determine the flow characteristics at a point are functions of the local conditions at the previously determined time level U^0 . Disturbances can not therefore travel faster than one local node spacing between the two successive time steps. This limitation is known as the Courant-Friedrichs-Lowy (CFL) condition. For stability of the present scheme, CFL number must stay below 1.0 for every cell in the computational domain.

3.4.3 Residual Smoothing

The CFL number for the solver is limited by the Runge-Kutta time step method that is used. Residual smoothing can be used to increase the stability region for the method, allowing a higher CFL and faster convergence. The residual smoothing method enhances the support of the algorithm by providing each node with a residual correction based on an averaging procedure of the surrounding residuals [5]. The residual used in the convergence enhancement method is defined as;

$$R_i^{(new)} = R_i^{(old)} + \varepsilon \sum_{k=1}^n (R_k^{(new)} - R_i^{(new)}) \quad (3.42)$$

The above equation can be written in a more useful form as

$$(1 + \varepsilon n) R_i^{(new)} - \varepsilon \sum_{k=1}^n R_k^{(new)} = R_i^{(old)} \quad (3.43)$$

$$R_i^{(new)} = \frac{R_i^{(old)} + \varepsilon \sum_{k=1}^n R_k^{(new)}}{(1 + \varepsilon n)} \quad (3.44)$$

where ε is suggested a value of 0.5 in Mavriplis and Jameson [6]. In the present study residual smoothing technique is applied for all the cells in the domain of the problem at every stage of the Runge-Kutta time cycle. A reduction of three orders of the magnitude of the density residuals is the convergence criteria for the present solver.

CHAPTER 4

RESULTS

The present code is tested for inviscid, viscous and turbulent flows over a wide range of Reynolds numbers and incident angles for single element airfoils. This chapter presents the results of eight test cases based on aerodynamic problems. Three test cases will be used to validate the developed two-dimensional Euler code. These test cases are subsonic lifting flow, transonic nonlifting flow and transonic lifting flow. Two laminar test cases are included that contain both separated and attached boundary layers. Three turbulent test cases are presented which provide a range of physically realistic conditions. The present solver will be tested for subsonic and transonic turbulent flows. The following table summarizes the details of each test cases.

Table 4.1 Test case summary

Test Cases	Airfoil	Free Stream Mach Number	Angle of Attack	Reynolds Number	Reference
Case 1	NACA 0012	0.5	3.0°		[5]
Case 2	NACA 0012	0.8	0.0°		[5], [38]
Case 3	NACA 0012	0.8	1.25°		[5], [38], [42]
Case 4	NACA 0012	0.8	10.0°	500	[39]
Case 5	NACA 0012	0.5	0.0°	5000	[39], [40]
Case 6	NACA 0012	0.3	4.04°	1.86×10^6	[41]
Case 7	NACA 0012	0.753	1.95°	3.88×10^6	[41]
Case 8	NACA 0012	0.799	2.26°	9.00×10^6	[43]

The present code is written by using Fortran programming language. Two dimensional unstructured grids are obtained by using Fastran software and data digitizer is used to get points from scanned images. The test cases are discussed in the following order. Two-dimensional Euler solutions are presented in Section 4.1 and two-dimensional Navier-Stokes solutions for laminar and turbulent flows are presented in Sections 4.2 and 4.3 respectively.

4.1 EULER SOLUTIONS

4.1.1 Test Case 1

The first test case for the two dimensional unstructured Euler solver is a subsonic lifting flow. The test case was performed by Jameson and Mavriplis [5]. Two dimensional Euler calculations are validated by computing the flow over the NACA 0012 airfoil. The free stream Mach number is 0.5 and the angle of attack is 3 degrees. The grid used for this simulation consists of 6990 nodes and 13678 cells as show in Figure 4.1. In order to validate the performance of the Euler solver the same grid will be used in test cases 2 and 3. The grid spacing at the leading edge is 0.0025 and at the trailing edge is 0.003. The grids near the nose of the airfoil and the tail of the airfoil are shown in Figure 4.2 and Figure 4.3 respectively. There are 31 nodes on the far-field boundary and 271 nodes on the airfoil. The far-field boundary is located 10 chords out from the airfoil. For the present subcritical test case, plot of surface pressure distribution obtained by Jameson and Mavriplis [5] and calculated results of pressure coefficient C_p are shown in Figure 4.4. Present results of C_p plot for upper and lower surfaces are the same as the ones obtained by Jameson and Mavriplis [5]. For the present test case a shock-free solution is obtained. There are no shock waves on the NACA 0012 airfoil.

Rapid convergence is achieved for the CFL value of 0.9 and a reduction of 3 orders of the magnitude of the density residuals is achieved in 9690 time steps as shown in Figure 4.5. A reduction of the density residuals of 3.8 orders of magnitude is achieved for 16000 time steps and the present code is not converged anymore after 18000 time steps as shown in Figure 4.5. Intel Pentium III 730 MHz computer with 128 MB RAM is used for the test case 1 and 10 iterations are achieved in 1 second.

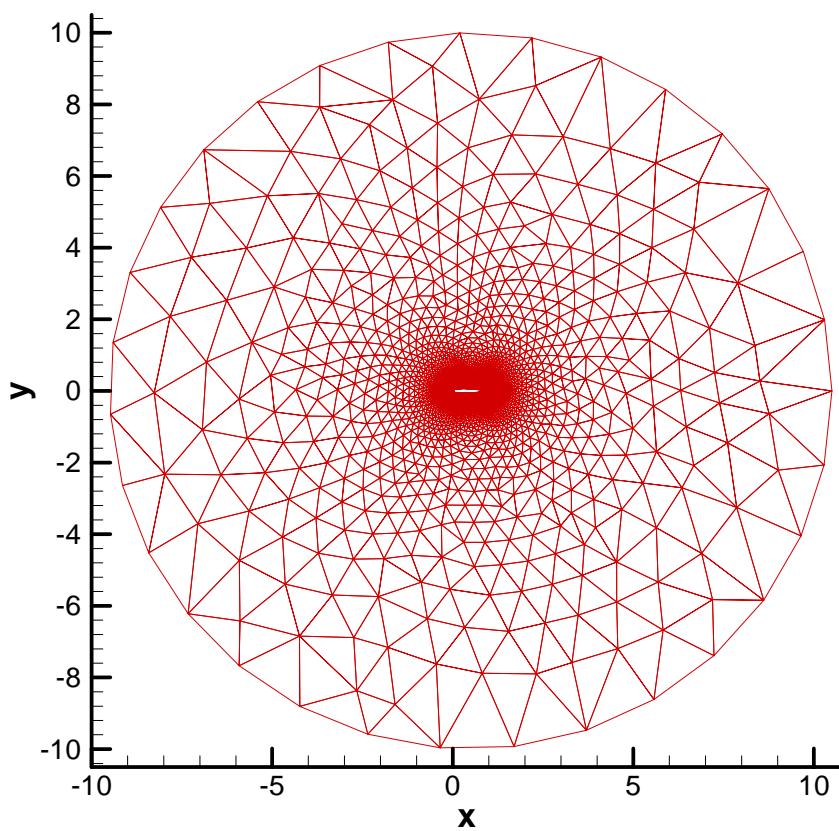


Figure 4.1 Unstructured grid consists of 6990 nodes and 13678 cells
for the Euler solver

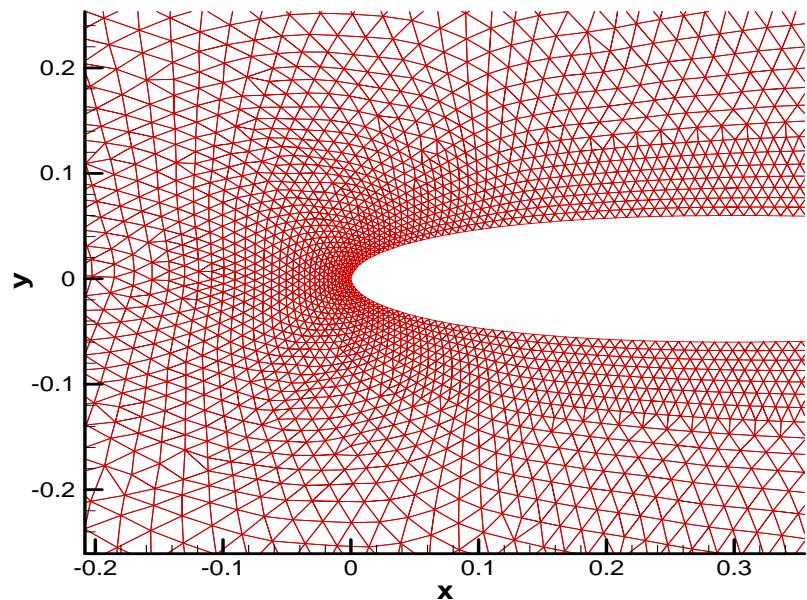


Figure 4.2 Close-up grid at the leading edge of the NACA 0012

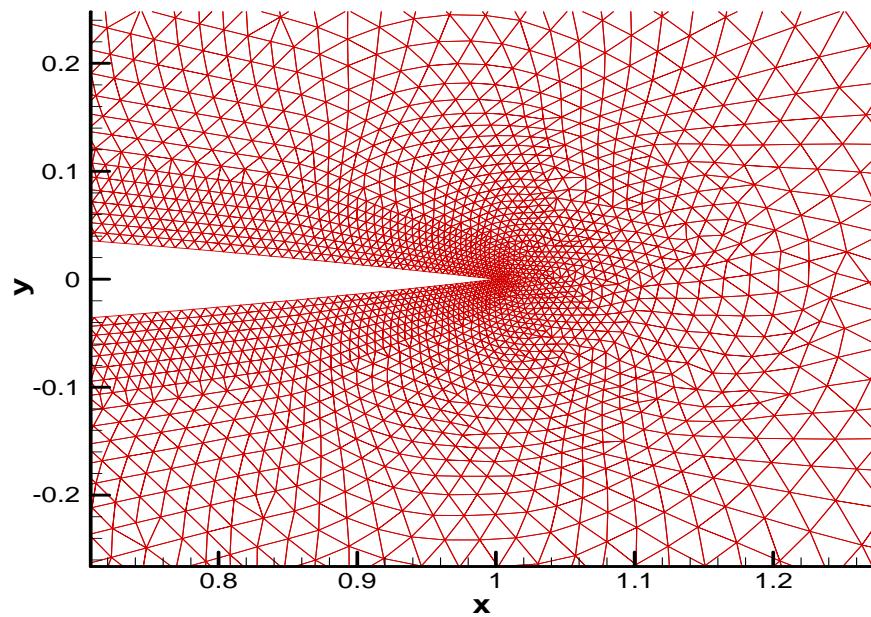


Figure 4.3 Close-up grid at the trailing edge of the NACA 0012

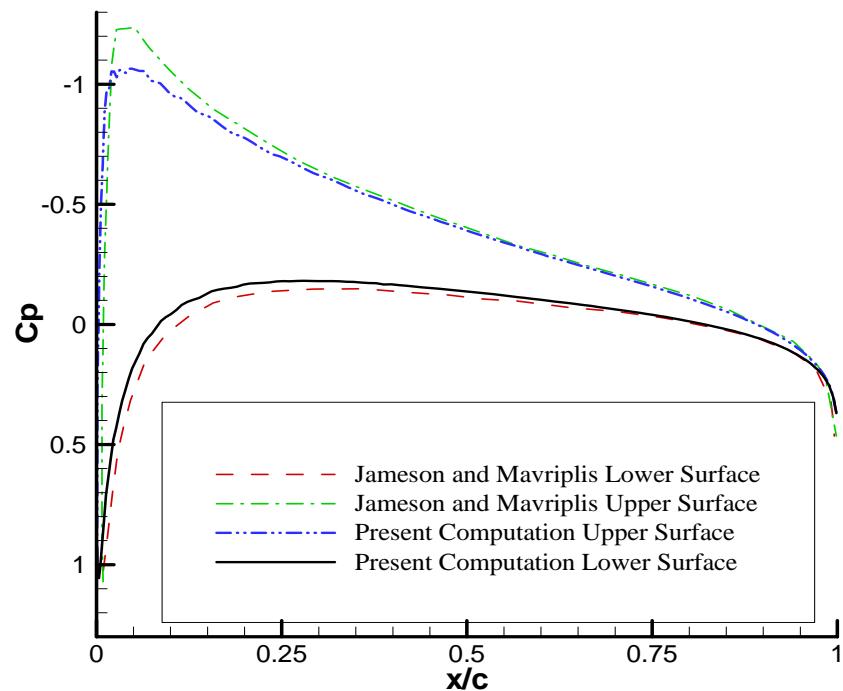


Figure 4.4 Calculated surface pressure coefficients of subsonic lifting flow

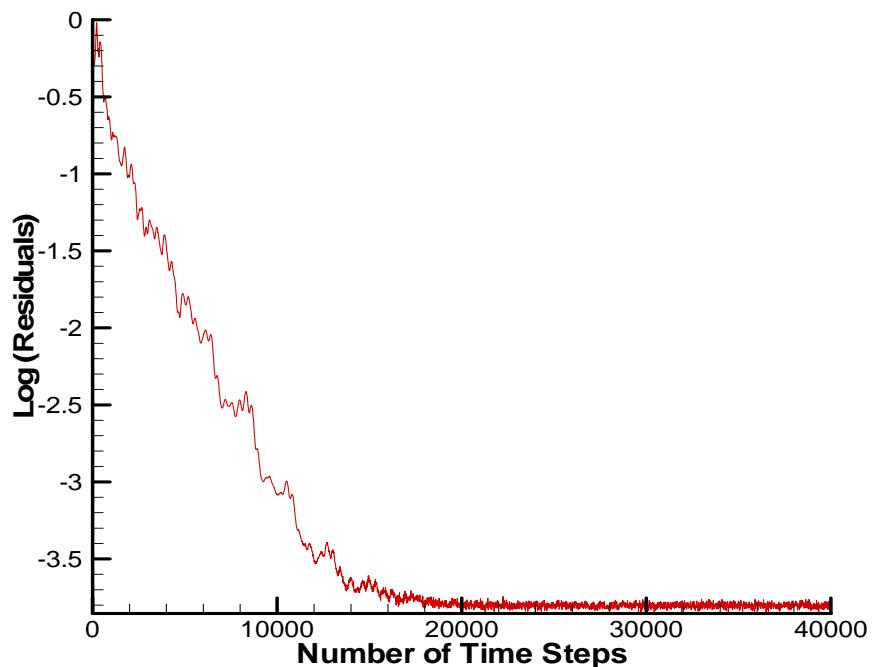


Figure 4.5 Convergence history of subsonic lifting flow test case

4.1.2 Test Case 2

The second test case is a transonic nonlifting flow. The present test case uses the same grid as in the previous one with a free stream Mach number of 0.8 at 0.0 degrees incidence. Plot of calculated surface pressure coefficient obtained by Jameson and Mavriplis [5] and the present solution of surface pressure coefficient C_p is given in Figure 4.6. The C_p plots are not the same as the ones obtained by Jameson and Mavriplis [5]. This is mainly because of the application of different numerical schemes. The shock location is approximately at 45% chord at the surface of the airfoil. The maximum pressure loss occurs at the down stream of the shock that can be seen in Figure 4.6. A reduction of the density residuals of 3 orders of magnitude is achieved for 4910 time steps as shown in Figure 4.7. For this case, the CFL value is equal to 0.9. A reduction of 3.3 orders of the magnitude of the density residuals is achieved for 6000 time steps and the convergence of present code is maintained constant after 6000 time steps as shown in Figure 4.7. Intel Pentium IV 2.4 GHz computer with 512 MB RAM is used for the test case 2 and 20 iterations are achieved in 1 second. Mach number contours obtained by Jameson and Yoon [38] are shown in Figure 4.8 for comparison. The calculated Mach number contours are shown in Figure 4.9. The shock location at the surface of the airfoil can be seen clearly in Figure 4.9. The contour levels are the same as the ones obtained by Jameson and Yoon [38].

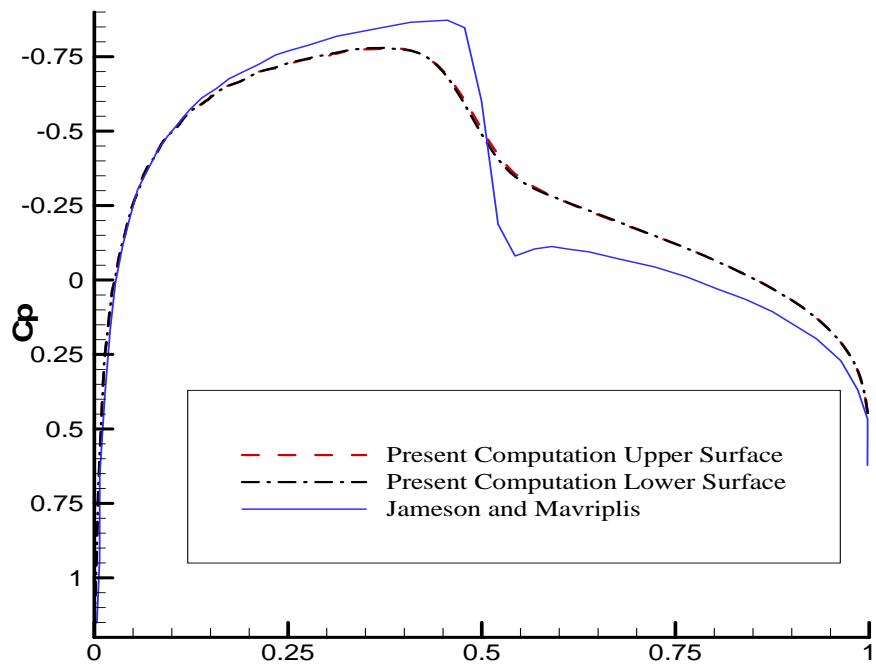


Figure 4.6 Calculated surface pressure coefficients of transonic nonlifting flow

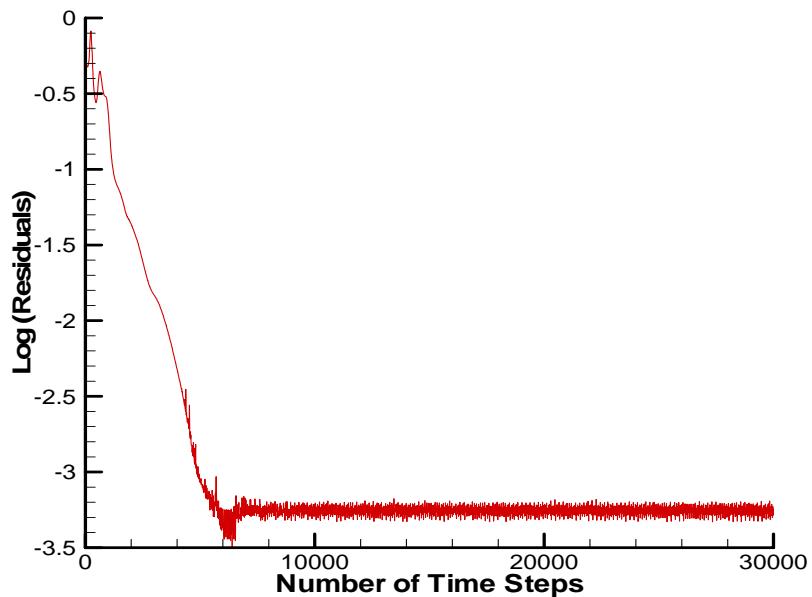


Figure 4.7 Convergence history of transonic nonlifting flow test case

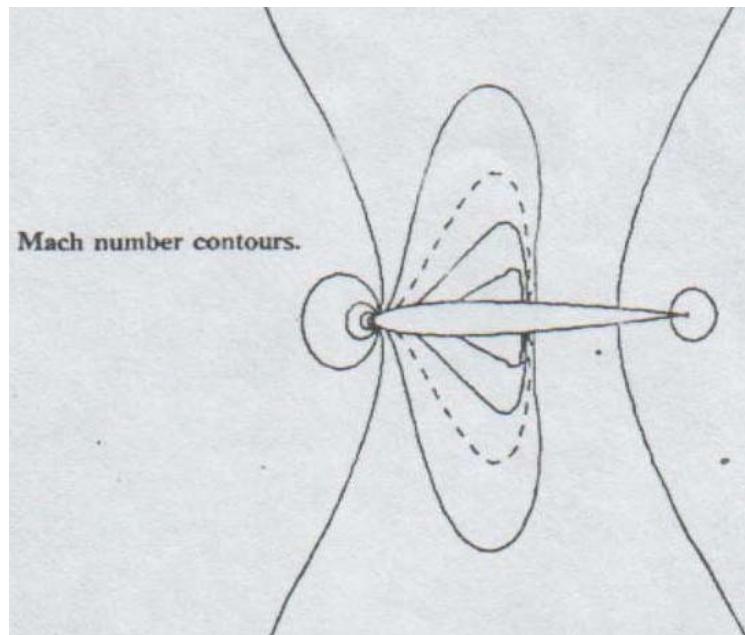


Figure 4.8 Calculated Mach number contours for transonic non lifting test case (from [41])

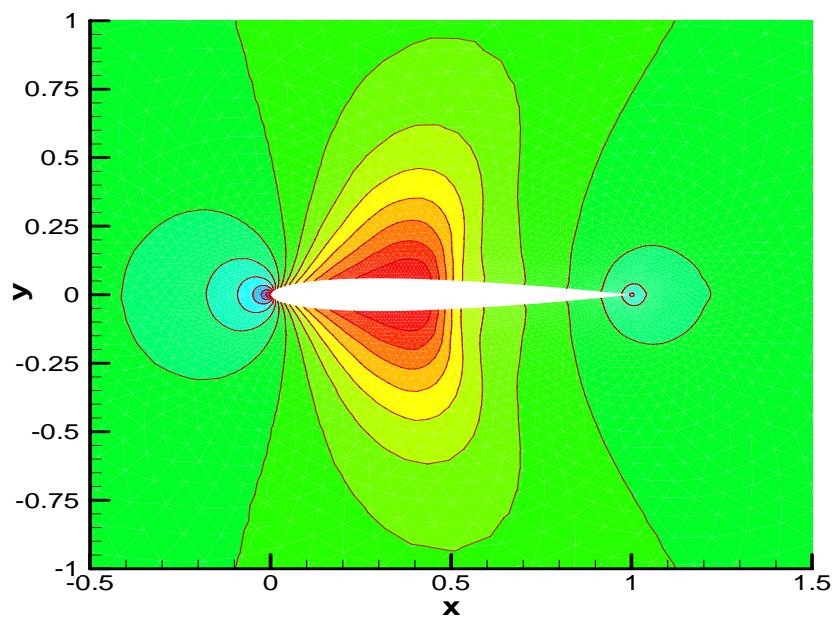


Figure 4.9 Calculated Mach number contours for transonic non lifting test case (present solution)

4.1.3 Test Case 3

The last test case for the two dimensional unstructured Euler solver is a transonic lifting flow. The test case was found in references [5] and [38]. The free stream Mach number is 0.8 and the angle of attack is 1.25 degrees. The same grid is used as in the previous one for the present test case. Time stepping on all grids uses the same CFL number and identical residual averaging. The CFL number is set to 0.9. A reduction of 3 orders of the magnitude of the density residuals is achieved in 6180 time steps as shown in Figure 4.10 and the present code is not converged anymore after 9000 time steps as shown in Figure 4.10. Intel Pentium IV 2.4 GHz computer with 512 MB RAM is used for the test case 3 and 20 iterations are achieved in 1 second. Plot of calculated surface pressure coefficient obtained by Jameson and Mavriplis [5] and the computed surface pressure coefficient plot is given in Figure 4.11. From the comparison of the surface pressure coefficient plots the strong upper surface shock is clearly visible and the small lower surface shock can be observed also. In addition, the pressure contour plot obtained by Pulliam [42] is presented in Figure 4.12. The present solution of pressure contours are given in Figure 4.13. The sharp upper surface shock is captured well and its location is approximately at 60% chord at the upper surface of the airfoil.

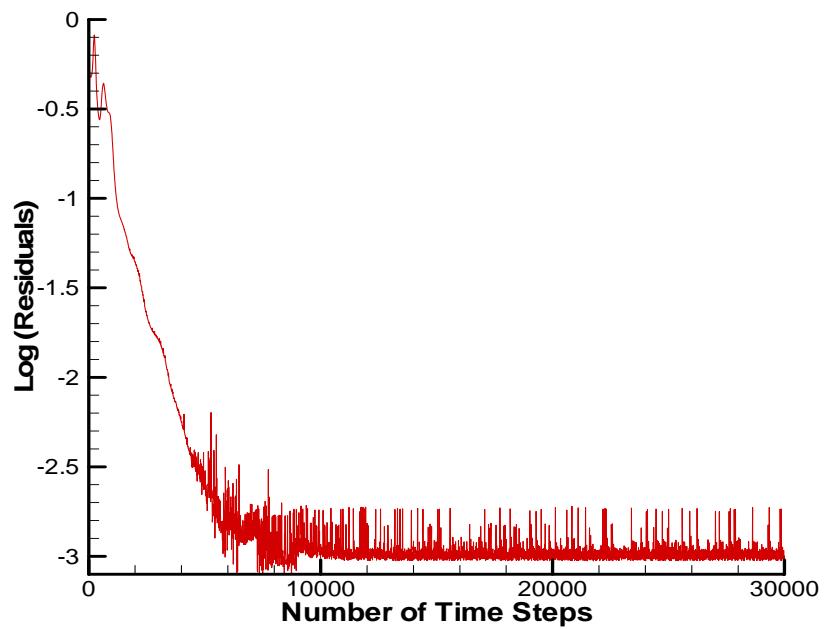


Figure 4.10 Convergence history of transonic lifting flow test case

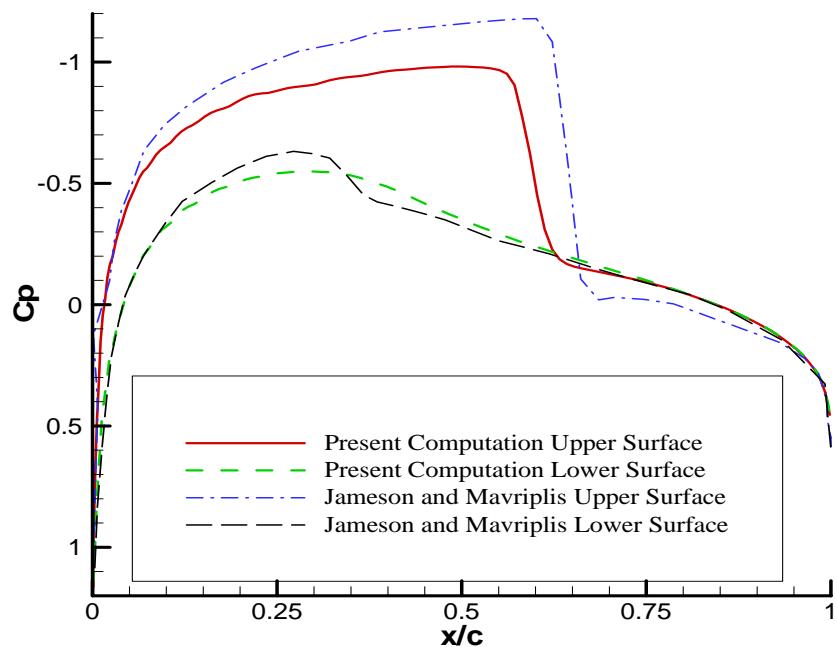


Figure 4.11 Calculated surface pressure coefficients of transonic lifting flow

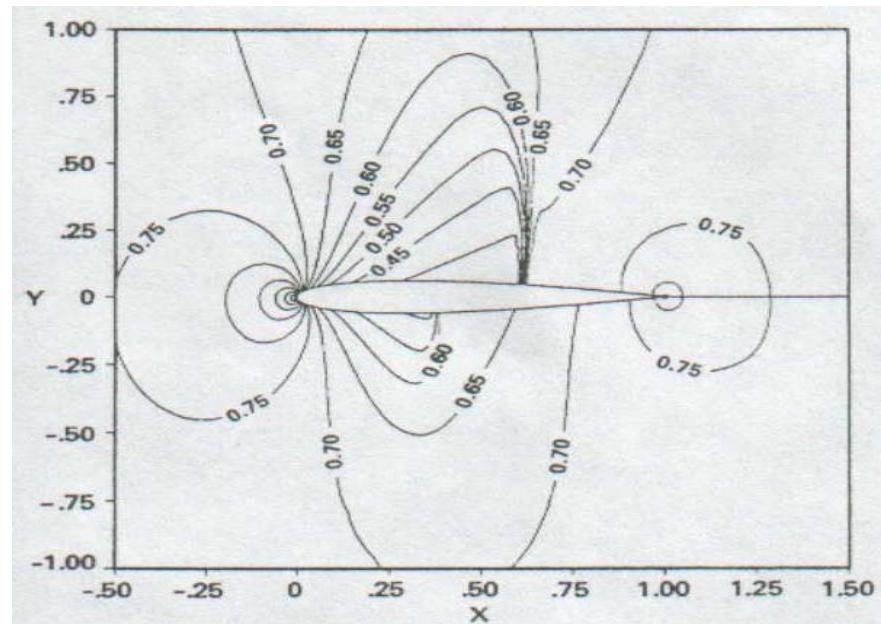


Figure 4.12 Calculated pressure contours for transonic lifting test case (from [42])

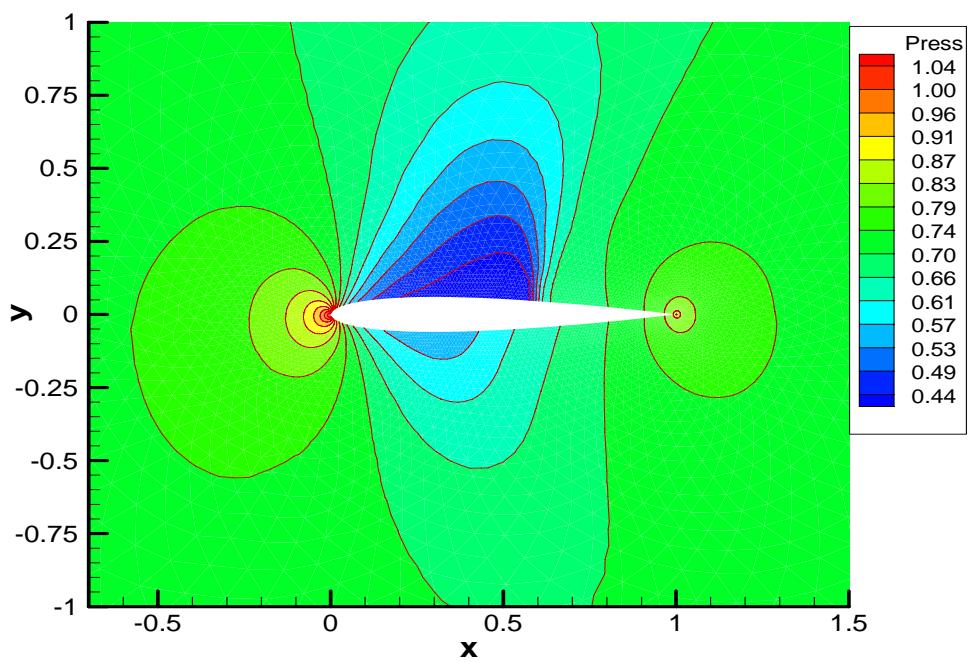


Figure 4.13 Calculated pressure contours for transonic lifting test case (present solution)

4.2 LAMINAR NAVIER-STOKES SOLUTION

4.2.1 Test Case 4

In this test case, which is taken from Mavriplis and Jameson [39], the Mach number is 0.8, the angle of attack is 10 degrees and Reynolds number is 500 over a NACA 0012 airfoil. As the Reynolds number increases, the viscous regions generally become thinner over streamlined bodies and the gradients in the normal direction within these regions increase. Therefore, viscous effects can be observed in thin boundary layer and wake regions for high Reynolds number flows. This case was selected to observe the viscous effects clearly in thick boundary layers and well developed wake region. The grid employed for this test case is same as the previous one which is shown in Figure 4.1. It contains 6990 nodes and 13678 cells. There are 31 nodes on the far-field boundary and 271 nodes on the airfoil. The far-field boundary is located 10 chords out from the airfoil. The grid spacing at the nose is 0.0025 and at the tail is 0.003. The grid near the airfoil surface is shown in Figure 4.14.

The present laminar flow test case involves very low Reynolds number flows. For this case the flow is dominated by viscous effects. The calculated Mach number contours obtained by the Mavriplis and Jameson [42] are presented in Figure 4.15 and the Mach number contours of the present solver are given in Figure 4.16. The contour levels are the same as the ones obtained by Mavriplis and Jameson [39]. The separation of the boundary layer occurs on the top surface of the airfoil and the recirculation region after the separation of the boundary layer is also captured well at the downstream of the airfoil. In Figure 4.17, the trailing edge of the airfoil is zoomed in order to see the details of the recirculation region near the trailing edge of the airfoil.

Rapid convergence is achieved for the CFL value of 0.9 and a reduction of 3 orders of the magnitude of the density residuals is achieved in 5870 time steps as shown in Figure 4.18. A reduction of 3.3 orders of the magnitude of the density residuals is achieved for 10000 time steps and the convergence of present code is oscillated between 2.8 and 3.3 value of a reduction orders of the magnitude of the density residuals after 10000 time steps as shown in Figure 4.18. Intel Pentium IV 2.4 GHz computer with 512 MB RAM is used for the test case 4 and 10 iterations are achieved in 2 seconds.

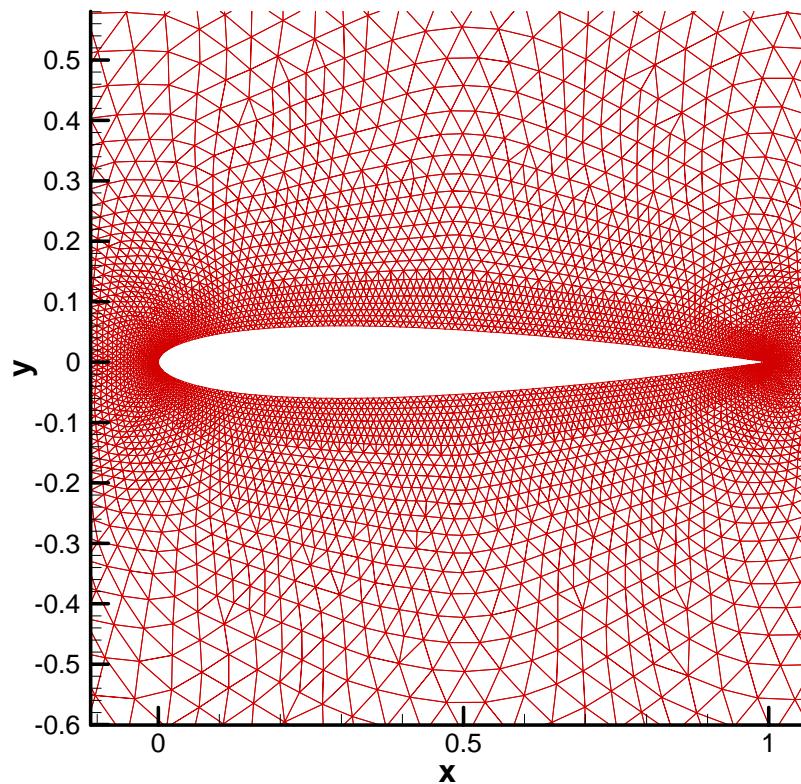


Figure 4.14 Close-up grid around the NACA 0012

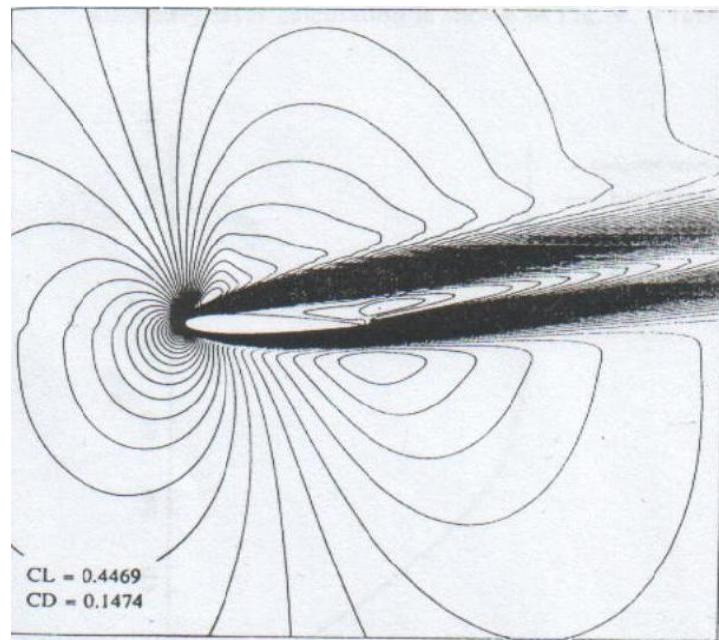


Figure 4.15 Calculated Mach number contours for laminar flow at
 $Re = 500$ (from [39])

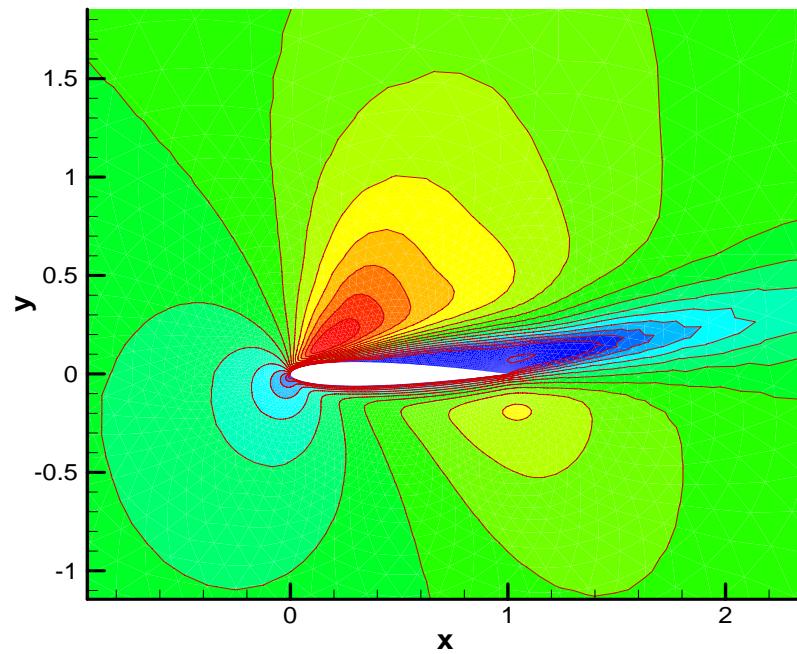


Figure 4.16 Calculated Mach number contours for laminar flow at
 $Re = 500$ (present solution)

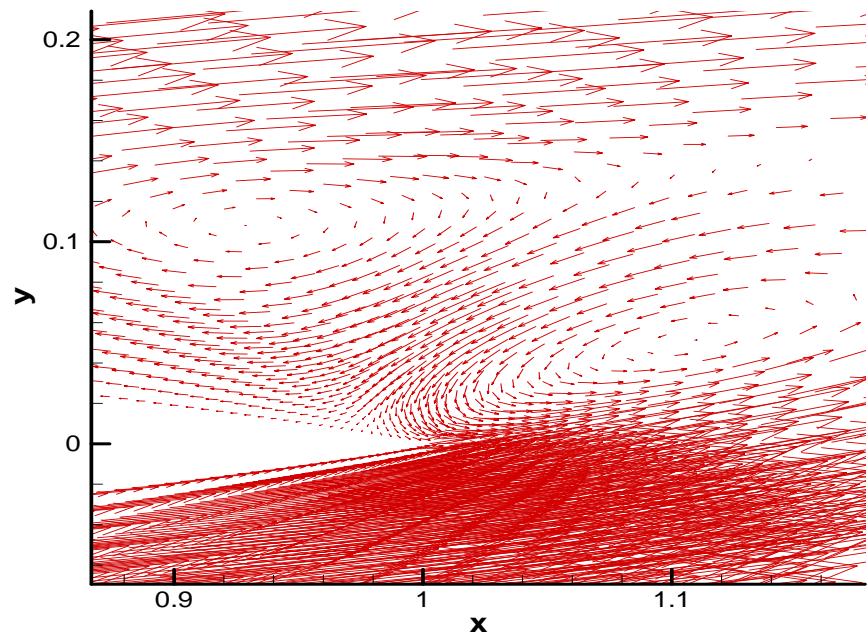


Figure 4.17 Velocity vectors at the trailing edge for laminar flow at
 $Re = 500$

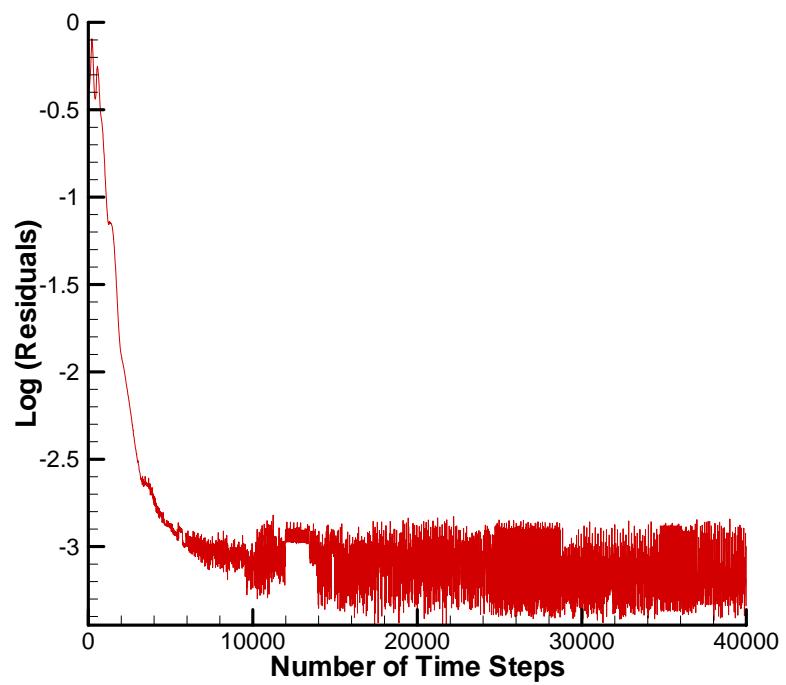


Figure 4.18 Convergence history of laminar flow at $Re = 500$

4.2.2 Test Case 5

The next test case is again a laminar flow over NACA 0012 airfoil. The present test case uses the same grid as in the previous one with a free stream Mach number of 0.5 and Reynolds number of 5000 at 0.0 degrees incidence. For the present test case, viscous effects are not as strong as in the previous test case. The boundary layers are attached to the airfoil and they are fairly thin. Plot of calculated Mach number contours obtained by Crumpton *et al.* [40] is shown in Figure 4.19 and the present solution of Mach number contours is given in Figure 4.20. The contour levels are the same as the ones obtained by Crumpton *et al.* [40]. Thin boundary layer and wake regions can be seen clearly in Figure 4.20. Velocity vectors at the trailing edge obtained by Crumpton *et al.* [40] is presented in Figure 4.21 and the velocity vectors of the present solver at the trailing edge is given in Figure 4.22. The velocity vectors at the trailing edge of the present solver and Crumpton *et al.* [40] are quite similar. The main feature of the solution is a recirculation region near the trailing edge. The recirculation region after the separation of the boundary layer is captured well and the back flow near the trailing edge can be seen clearly in Figure 4.22.

Plot of surface pressure coefficient obtained by Mavriplis and Jameson [39] and the present solution of surface pressure coefficient C_p is given in Figure 4.23. The C_p plot is not same as the one obtained by Mavriplis and Jameson [39]. In the present solution separation point is caught at 89% chord of the airfoil in Figure 4.24. However, in the validation cases separation point starts at 80% chord of the airfoil. Since the separation point is caught at downstream of the validation cases there is pressure change is observed in the present study.

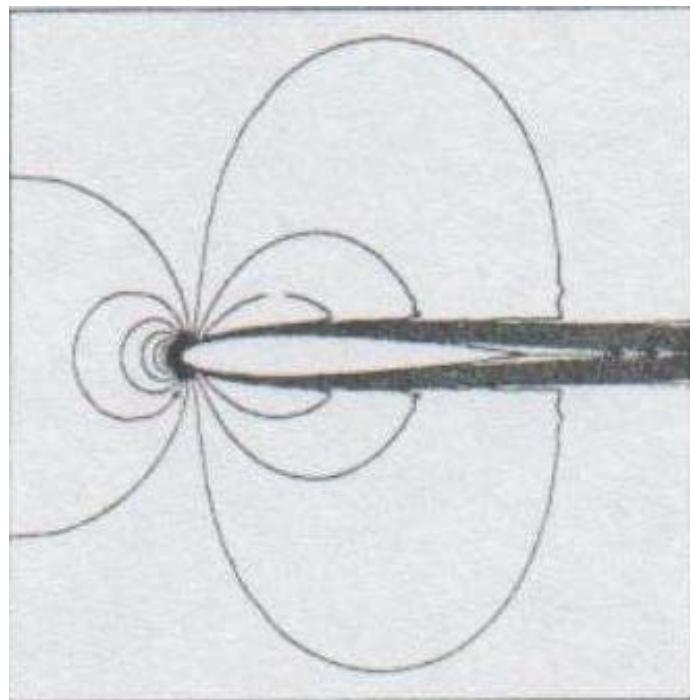


Figure 4.19 Calculated Mach number contours for laminar flow at
 $Re = 5000$ (from [40])

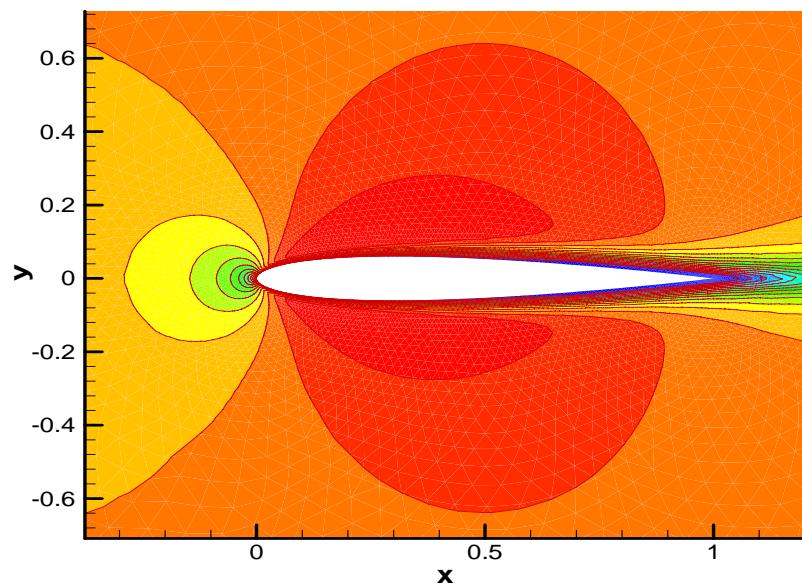


Figure 4.20 Calculated Mach number contours for laminar flow at
 $Re = 5000$ (present solution)

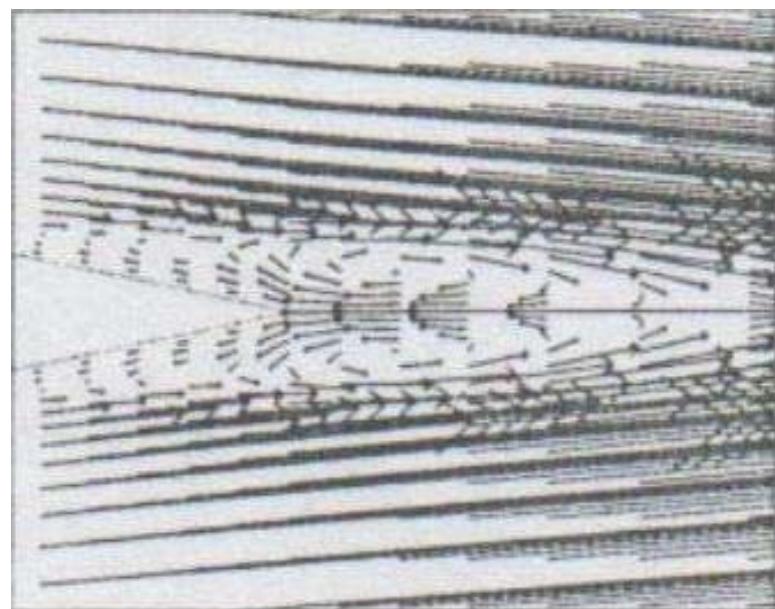


Figure 4.21 Close-up of velocity vectors at the trailing edge (from[40])

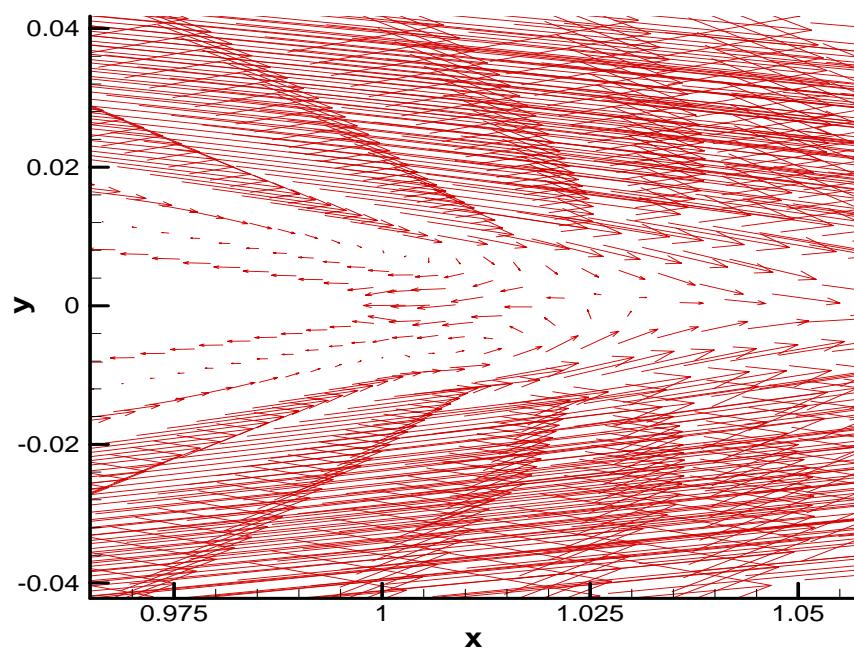


Figure 4.22 Close-up of velocity vectors at the trailing edge
(present solution)

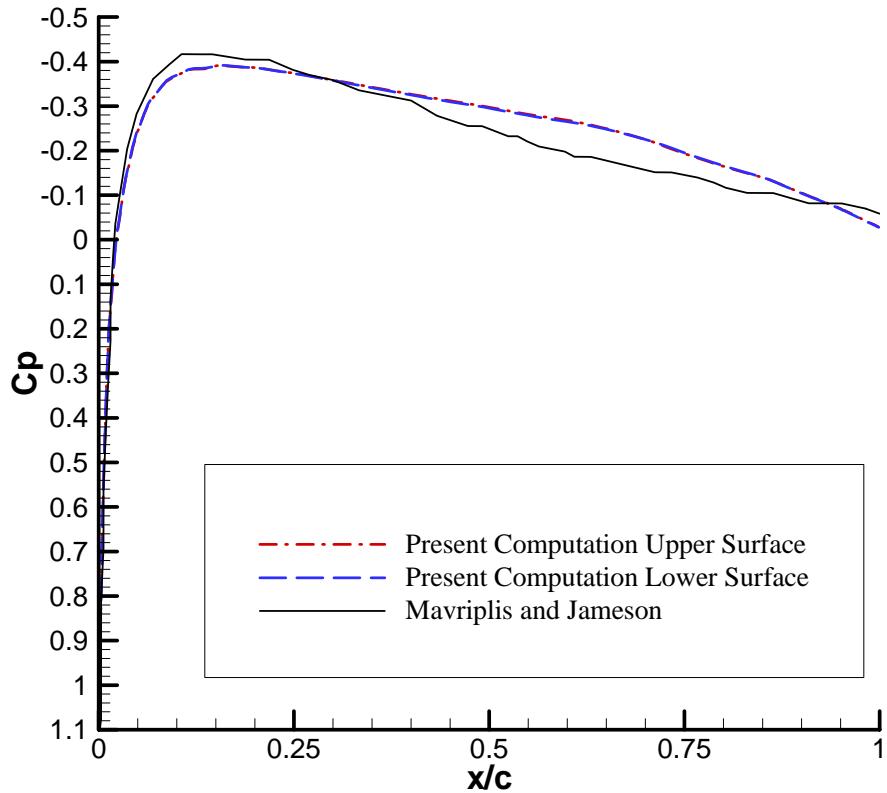


Figure 4.23 Calculated surface pressure coefficients for laminar flow
at $Re = 5000$

A reduction of the density residuals of 3 orders of magnitude is achieved in 9120 time steps as shown in Figure 4.25. For this case, the CFL value is equal to 0.9. A reduction of the density residuals of 3.8 orders of magnitude is achieved for 16000 time steps and the present code is not converged anymore after 16000 time steps as shown in Figure 4.25. Intel Pentium IV 2.4 GHz computer with 512 MB RAM is used for the test case 5 and 10 iterations are achieved in 2 seconds.

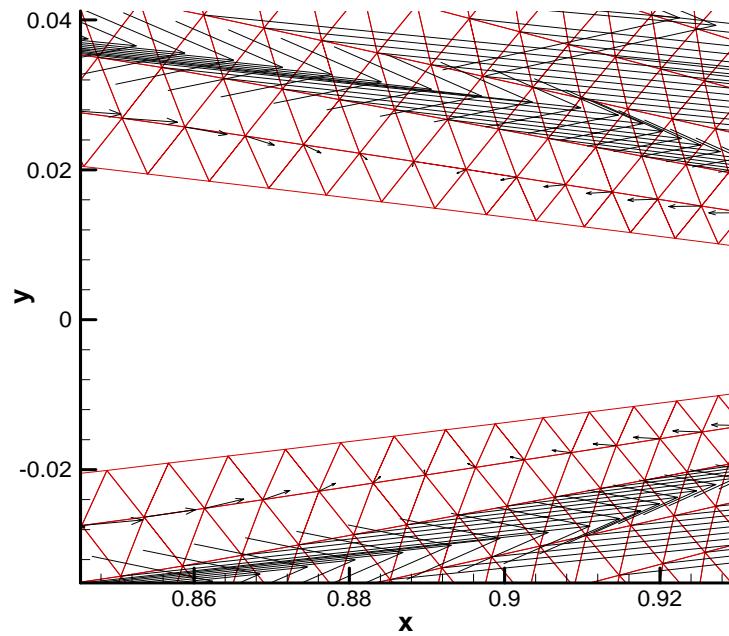


Figure 4.24 Close-up of velocity vectors for separation point
(present solution)

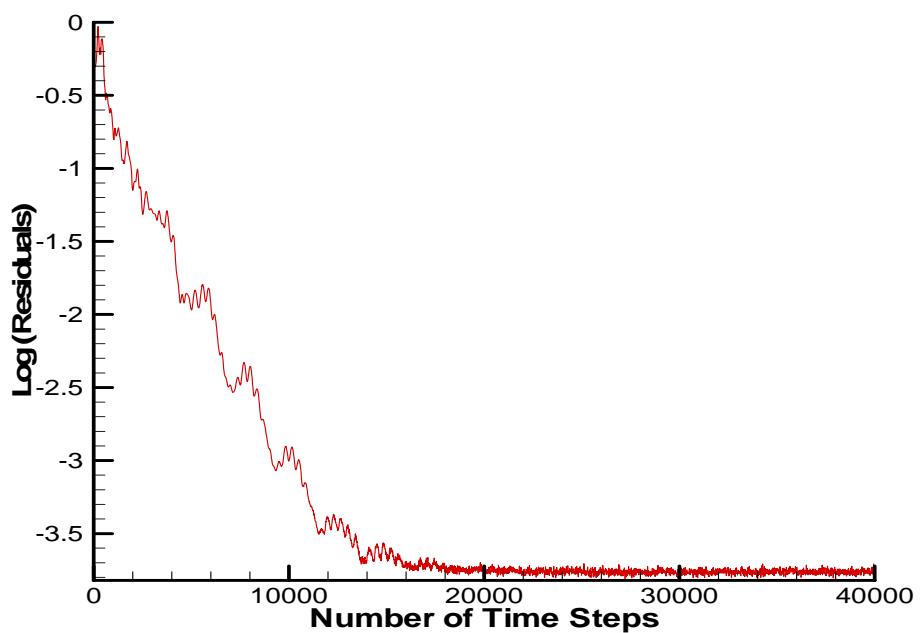


Figure 4.25 Convergence history of laminar flow at $Re = 5000$

4.3 TURBULENT NAVIER-STOKES SOLUTION

4.3.1 Test Case 6

The sixth test case is a turbulent test case. The flow conditions were a free stream Mach number of 0.3. In the present test case, the Reynolds number is set at a value of 1.86×10^6 and an angle of attack of 4.04 degrees. In such realistic flows, the boundary layer surrounding the airfoil is expected to be very thin relative to the previous test cases. The initial node spacing near the airfoil must be small. The mesh used for this test case and for the following test cases is presented in Figure 4.26. It contains 19040 nodes and 37798 cells. There are 88 nodes on the far-field boundary and 194 nodes on the airfoil. The far-field boundary is located 15 chords out from the airfoil. The grid around the airfoil is shown in Figure 4.27

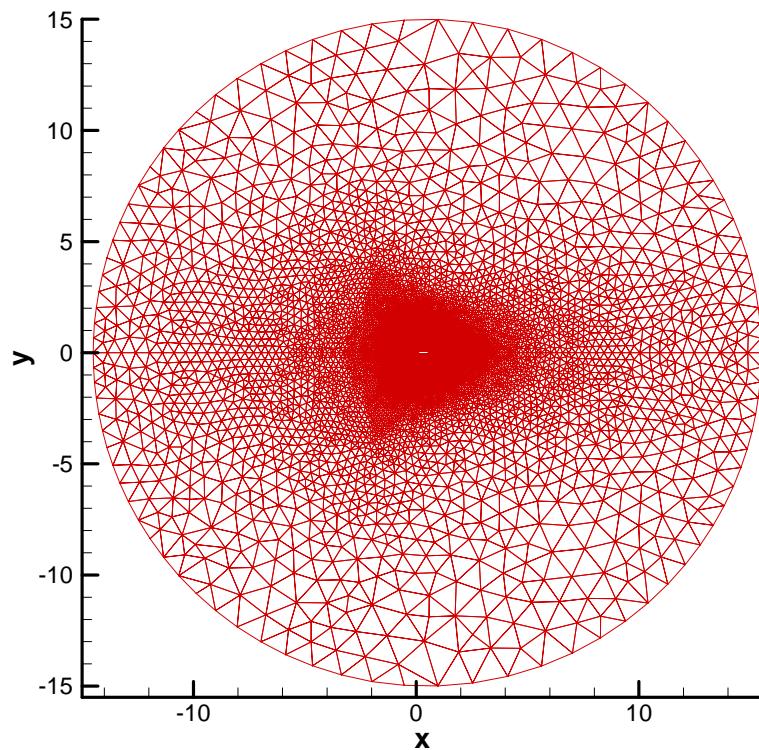


Figure 4.26 Unstructured grid consists of 19040 nodes and 37798 cells for the turbulent Navier-Stokes solver

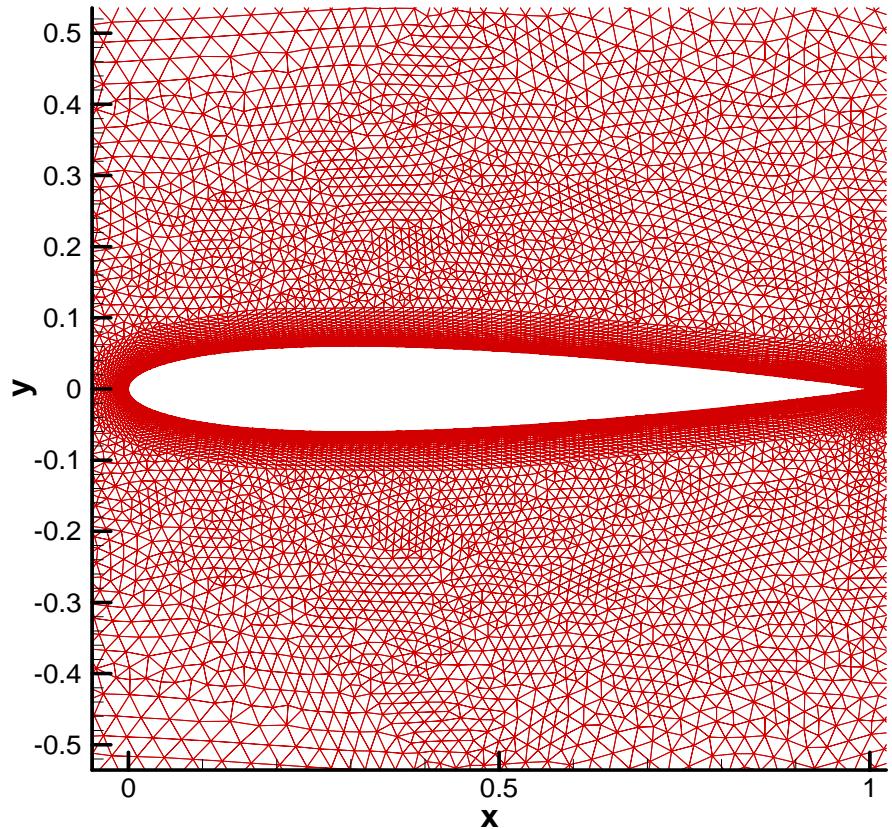


Figure 4.27 Close-up grid around the NACA 0012

The computed pressure coefficient is compared with the experimental data which is found in AGARD Advisory Report no. 138 [41] in Figure 4.28. The present solution and the experimental data are almost identical. For the subsonic turbulent flow test case, solution of the present solver is reasonable.

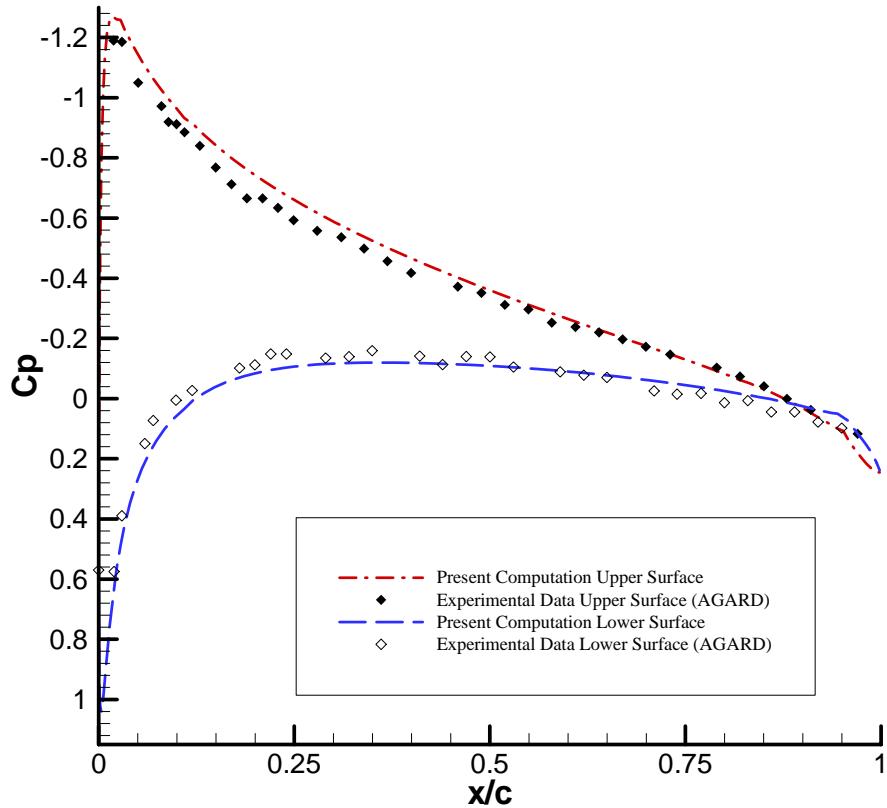


Figure 4.28 Calculated pressure coefficient distribution on NACA 0012 airfoil for the turbulent subsonic test case 6

A reduction of 3 orders of the magnitude of the density residuals is achieved in 20130 time steps as shown in Figure 4.34. The CFL number is set to 0.9 for this test case. A reduction of 3.5 orders of the magnitude of the density residuals is achieved for 30000 time steps and the convergence of present code is maintained constant after 30000 time steps as shown in Figure 4.29. Intel Pentium IV 2.4 GHz computer with 512 MB RAM is used for the test case 6 and 10 iterations are achieved in 7 seconds.

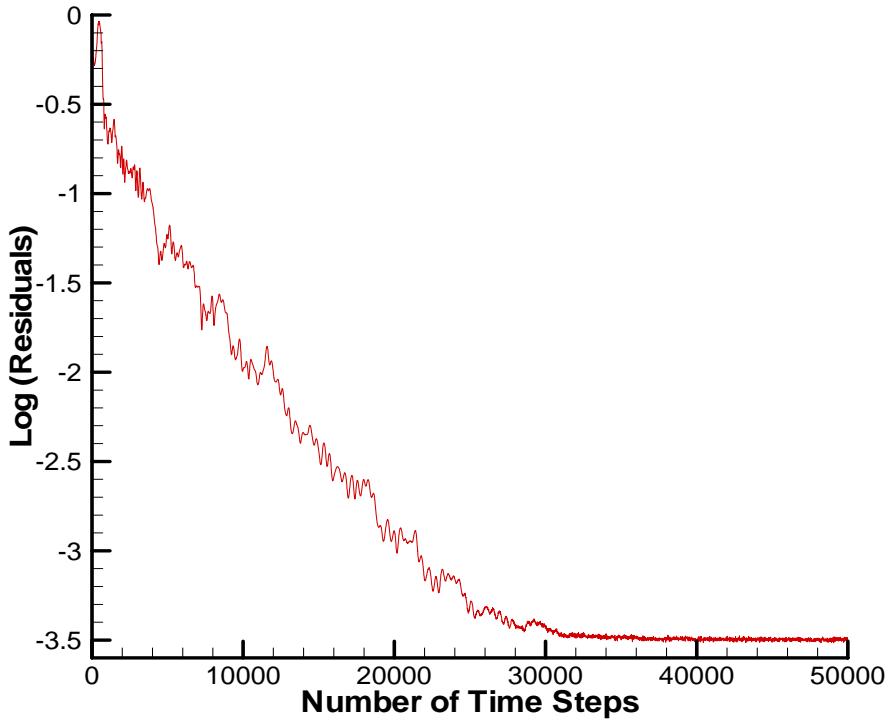


Figure 4.29 Convergence history of turbulent flow for the subsonic test case 6

4.3.2 Test Case 7

The next test case is again taken from the AGARD Advisory Report no. 138 [41]. The Mach number is increased to 0.753 and the angle of attack is 1.95 degrees. The Reynolds number is 3.88×10^6 . The same grid is also used for the present test case. The Reynolds number is a reasonable number for a real airfoil in flight and as with the previous turbulent case, the thin boundary layer requires a small grid spacing near the airfoil wall. Figure 4.30 shows the comparison of pressure coefficient along the airfoil surface obtained from the present solver and from the experimental data. One essential feature of the solution is a shock wave located near the midpoint of the upper surface of the airfoil. The maximum pressure loss occurs in the boundary layer at the downstream of the shock as shown in Figure 4.30. In

addition, the location of the shock wave is clearly evident in the experimental results by the sudden jump in pressure along the upper surface of the airfoil.

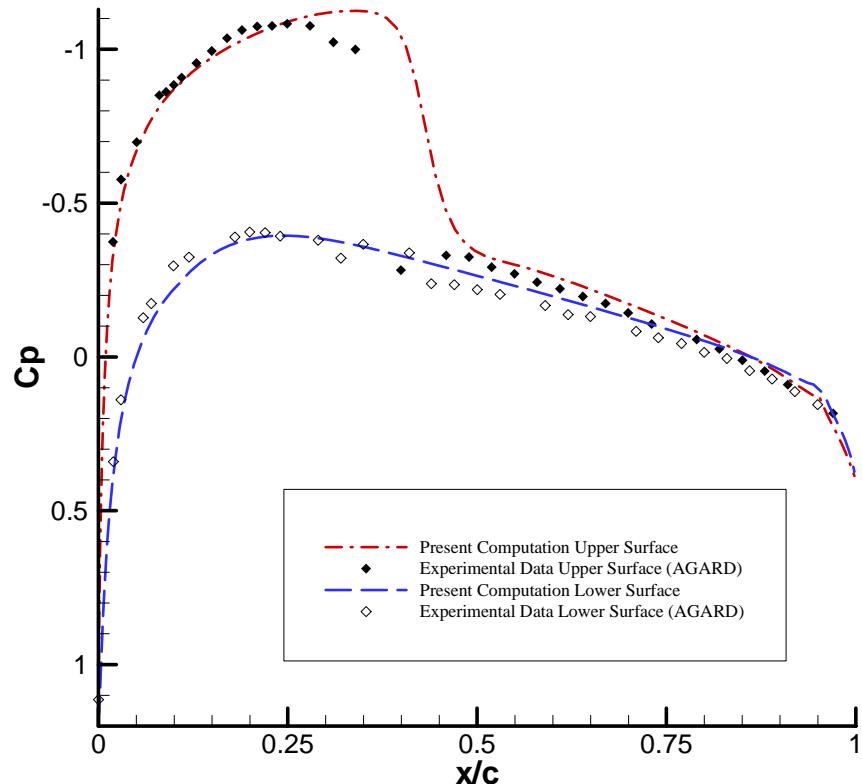


Figure 4.30 Calculated pressure coefficient distribution on NACA 0012 airfoil for the turbulent transonic test case 7

Local time stepping is used to accelerate convergence. CFL number is taken as 0.9 until 3 orders of residual decrease to convergence with 14540 time steps as shown in Figure 4.31. A reduction of 3.4 orders of the magnitude of the density residuals is achieved for 28000 time steps and the convergence of present code is oscillated between 3.2 and 3.4 value of a reduction orders of the magnitude of the density residuals after 36000 time steps as shown in Figure 4.31. Intel Pentium IV 2.4 GHz computer with 512 MB RAM is used for the test case 7 and 10 iterations are achieved in 6 seconds.

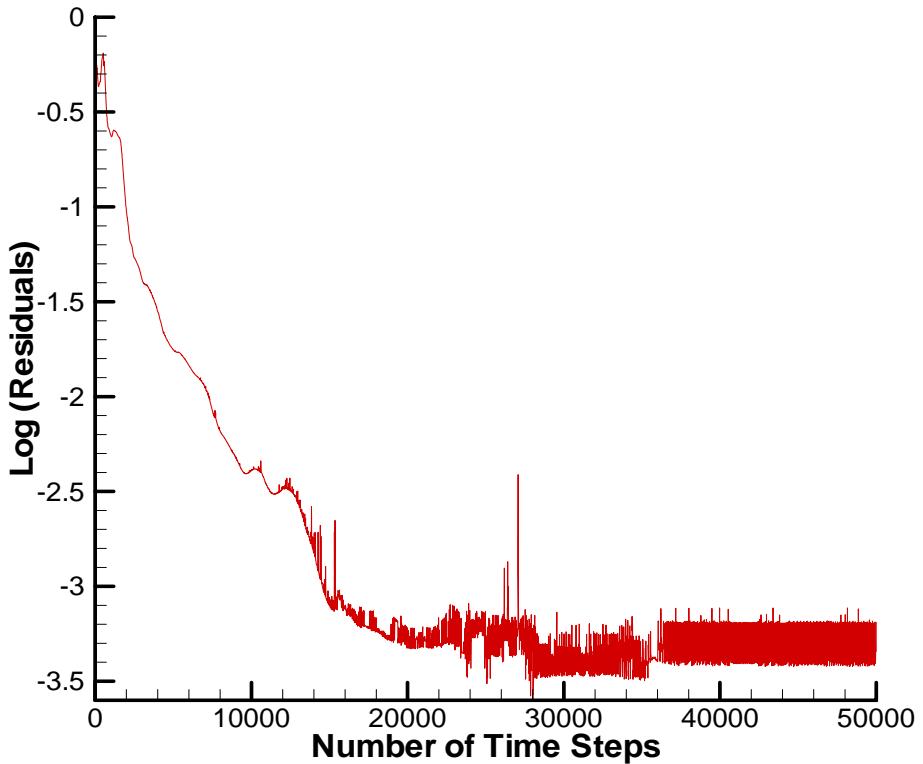


Figure 4.31 Convergence history of turbulent flow for the transonic test case 7

4.3.3 Test Case 8

In the last test case, which is a transonic test of the Spalart-Allmaras turbulence model over NACA 0012, the flow conditions used were a free stream Mach number of 0.799, an angle of attack of 2.26 degrees and Reynolds number 9.0×10^6 which is a reasonable number for a real airfoil in flight. The grid used for this simulation is same as the previous grid. Strong shock waves may cause complete boundary layer separation. Present turbulent Navier-Stokes solver is tested in a shock-boundary layer interaction. The same test case was performed by Holst [43]. Plots of surface pressure distribution obtained by Holst [43] and the present result of

surface pressure coefficient C_p are given in Figure 4.32. The computed surface pressure coefficient is compared with the C_p plots of Holst [43]. The shock location is captured with a slight deviation in the present solution. The maximum pressure loss occurs in the boundary layer at the downstream of the shock. Clustering grid around 55% chord airfoil may overcome the deviation.

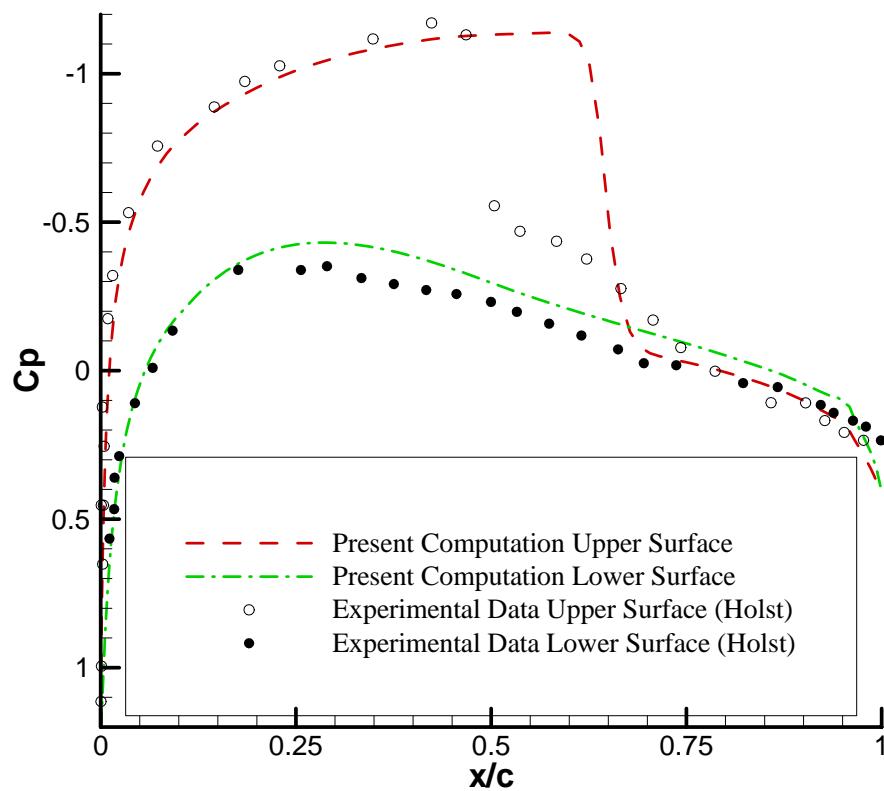


Figure 4.32 Calculated surface pressure coefficients for turbulent flow transonic test case 8

Convergence is achieved for the CFL value of 0.9 and a reduction of 3 orders of the magnitude of the density residuals is achieved in 15400 time steps as shown in Figure 4.33. A reduction of 3.3 orders of the magnitude of the density residuals is achieved for 30000 time steps and the convergence of present code is maintained constant after 30000 time steps as shown in Figure 4.33. Intel Pentium IV 2.4 GHz computer with 512 MB RAM is used for the test case 8 and 10 iterations are achieved in 6 seconds.

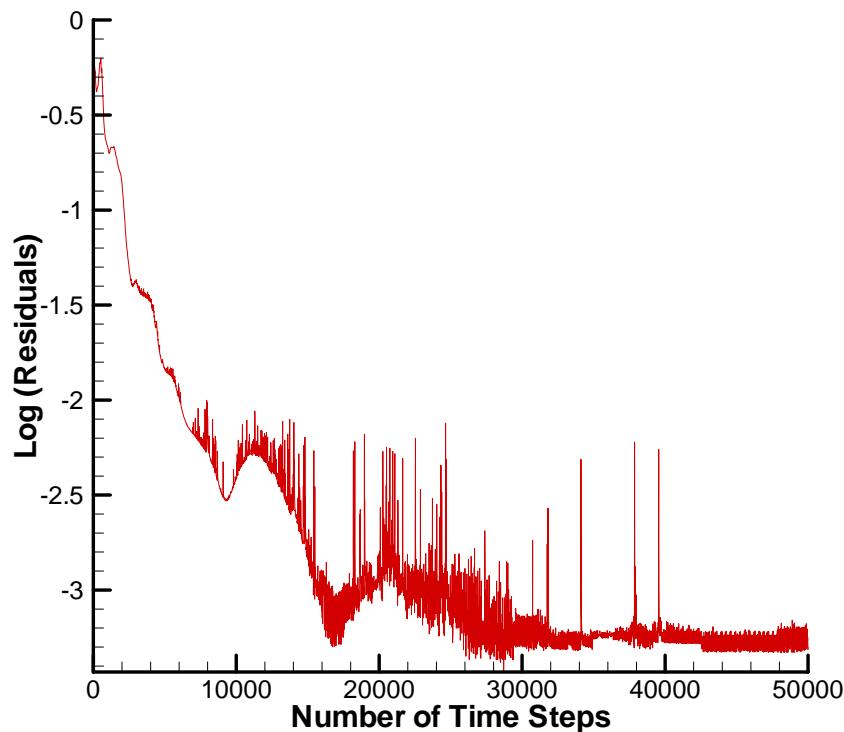


Figure 4.33 Convergence history of turbulent flow for the transonic test case 8

CHAPTER 5

DISCUSSION AND CONCLUSION

5.1 SUMMARY AND DISCUSSION

In this study, a two-dimensional Euler solver is developed to operate on inviscid flows and a two-dimensional Navier-Stokes solver is developed to operate on laminar flows and turbulent flows. Unstructured grids with a triangular element are used to define the computational domain. Four-stage Runge-Kutta method is applied as an explicit-time marching scheme. The time-marching solver is accelerated by applying local time stepping procedure. Moreover, residual smoothing technique is used to accelerate the solver further. The finite volume formulation is cell-centered. For the calculation of the eddy viscosity terms at cell centers Spalart-Allmaras turbulence model is implemented to the Navier-Stokes solver.

The developed solver is tested with eight different cases. Three test cases are used to validate the developed two-dimensional Euler code. These test cases are subsonic lifting flow, transonic nonlifting flow and transonic lifting flow. Two laminar flow test cases are used to validate the Navier-Stokes code. Laminar flow test cases are chosen especially for separated and attached boundary layers. Lastly, three test cases are used to test the implemented Spalart-Allmaras turbulence model from accuracy point of view. The results of these test cases are compared with the experimental data and the solutions from literature. With the results obtained from the Euler solver test cases, it can be

concluded that the present solver gives better results for subsonic flows. In transonic test cases the present solutions are reasonable. There are small deviations in surface pressure coefficient plots, which is thought to be mainly resulting from grid clustering and location of the far-field boundary. The results for the laminar flow test cases are in good agreement with other results in the literature. Separated and attached boundary layers are observed clearly. For the attached boundary layer results, deviation at the location of separation point is observed. Grid clustering can yield a better solution for this test case. The last test cases are the turbulent flow test cases. The results of developed solver are compared with the experimental data and the solutions from literature. For the subsonic turbulent test case the results are reasonable and match with the experimental ones. For the transonic test case a small deviation is observed from surface pressure coefficient plot about the location of the strong shock wave at the upper surface of the airfoil. The first order spatial accuracy can be improved to achieve better results for the transonic cases.

5.2 FUTURE RECOMMENDATIONS

The cell base data structured is used to store grid information. For the cell base data structured, the run times are obtained quite long. Cell face based data structure can be used to achieve convergence in short time. Moreover, a parallel-computing technique can be applied to the developed solver. Lastly, using multigrid algorithm will speed up the convergence. They can increase the speed of the code greatly so that geometrically more complex and three-dimensional cases can be investigated easily.

If the numerical scheme is changed to an implicit one, there will not be any stability problem. Therefore, using implicit scheme will eliminate the limitation of the maximum allowable CFL number. This will definitely improve the convergence rate of the solver.

Accurate results can be obtained if more suitable boundary conditions are applied to the developed solver. Smaller computational domains can be studied when the circulation correction techniques are used in the developed solver. Mesh refinement techniques can be used to optimise the solution.

Finally, a two-equation turbulence model can be implemented to the present code in order to investigate the turbulence phenomenon better. Heat transfer equations can be also implemented to the code. Therefore, the problems with heat transfer can be studied also.

REFERENCES

- [1] Anderson, J.D., “*Computational Fluid Dynamics: The Basics with Applications*”, Int. Ed., McGraw-Hill Singapore, 1995.
- [2] Hoffman, K.A., Chiang, S.T., “*Computational Fluid Dynamics*”, Vol. 1-2-3, Engineering Education System Publication, USA, August 2000.
- [3] Versteeg, H.K., “*An Introduction to Computational Fluid Dynamics, The Finite Volume Method*”, Longman, 1995.
- [4] Jameson, A., Schmidt, D., and Turkel, E., “*Numerical Solutions of the Euler Equations by Finite Volume Method Using Runge-Kutta Time-Stepping Schemes*”, AIAA 14th Fluid and Plasma Dynamics Conference, AIAA-81-1259, June 1981.
- [5] Jameson, A. and Mavriplis D. “*Finite Volume Solution of the Two-Dimensional Euler Equations on a Regular Triangular Mesh*”, AIAA Journal, Vol. 24, No. 4, April 1986, pp. 611-618.
- [6] Mavriplis, D. and Jameson, A., “*Multigrid Solution of the Two-Dimensional Euler Equations on Unstructured Triangular Meshes*”, AIAA Paper 87-0353, January 1987.
- [7] Desideri, J.A. and Dervieux, A., “*Compressible Flow Solvers Using Unstructured Grids*”, VKI Lecture Series, 1988-05, pp. 1-115.

- [8] Lumley, J. L. “*Turbulence Modeling*”, Journal of Applied Mechanics, Vol. 50, December 1983, pp. 1097-1103.
- [9] Chen, C.J., Jaw, S.Y., “*Fundamentals of Turbulence Modeling*”, Taylor & Francis, USA, 1998.
- [10] Mani, M., Willhite, P., Ladd, J., “*Performance of One-Equation Turbulence Models in CFD Applications*”, 26th AIAA Fluid Dynamics Conference, AIAA 95-2221, June 1995.
- [11] Baldwin, M.S. & Lomax, H., “*Thin-Layer Approximation and Algebraic Model for Separated Flows*”, AIAA-78-257.
- [12] Cebeci, T. and Smith, A.M.D., “*A Finite-Difference Method for Calculating Compressible Laminar and Turbulent Boundary Layers*”, Journal of Basic Engineering, Vol. 29, No.3 pp. 523 -535.
- [13] Johnson, D.A. and King, L.S., “*A Mathematically Simple Turbulence Closure Model for Attached and Separated Turbulent Boundary Layers*”, AIAA Journal, Vol.23, No.11, 1985, pp. 1684-1692.
- [14] Spalart, P.R. and Allmaras, S.R., “*A One-Equation Turbulence Model for Aerodynamics Flows*”, Boeing Commercial Airplane Group, Seattle, Washington, 1992.
- [15] Saxena, S.K. and Nair, M.T., “*Implementation and Testing of Spalart-Allmaras Model In a Multi-Block Code*”, 40th AIAA Aerospace Sciences Meeting and Exhibit, AIAA 2002-0835, January 2002.

- [16] Nee, V.W. and Kovasznay, L.S.G., “*Simple Phenomenological Theory of Turbulent Shear Flows*”, Physics of Fluids, Vol.12, No.3, 1969, pp.473-484.
- [17] Bald, B.S. and Barth, T.J., “*A One Equation Turbulence Transport Model for High Reynolds Number Wall Bounded Flows*”, NASA TM 102847, 1990.
- [18] Anderson, W.K. and Bonhaus, D.L., “*Aerodynamic Design on Unstructured Grids for Turbulent Flows*”, NASA TM 112867, June 1997.
- [19] Seror, S., Rubin T., Peigin, S. and Epstein, B., “*Implementation and Validation of the Spalart-Allmaras Turbulence Model for Parallel Processing on PC's Cluster*”, 21st Applied Aerodynamic Conference, June 2003.
- [20] Deck, S., Duveau, P., Espiney, P. and Guillen, P., “*Development and Application of Spalart-Allmaras One Equation Trubulence Model to Three-Dimensional Supersonic Complex Configurations*”, Aerospace Science and Technology, June 2002, pp. 171-183.
- [21] Frink, N.T., “*Assessment of an Unstructured-Grid Method for Predicting 3-D Trubulent Viscous Flows*”, 34th Aerospace Sciences Meeting, AIAA-96-0292, January 1996.
- [22] Shima, E., Egami, K. and Amano, K. “*Navier-Stokes Computation of A High Lift System using Spalart-Allmaras Turbulence Model*”, 32nd Aerospace Sciences Meeting & Exhibit, AIAA-94-0162, January 1994.

- [23] Anderson, W.K. and Bonhaus, D.L., “*An Implicit Upwind Algorithm for Computing Trubulent Flows on UnstructuredGrids*”, Computer Fluids, Vol.23, No.1, 1994, pp.1-21.
- [24] Walsh, P.C., “*Adaptive Solution of Viscous Aerodynamics Flows Using Unstructured Grids*”, Doctor of Philosophy, University of Toronto, Canada, 1998.
- [25] Koomullil, R.P., “*Flow Simulation System for Generalized Static and Dynamic Grids*”, Doctor of Philosophy, Mississippi State University, USA, May 1007.
- [26] Greiner, K., “*A Finite Volume Solver for Viscous Turbulent Flows on Mixed Element Unstructured Meshes*”, Doctor of Philosophy, University of Toronto, Canada, !998.
- [27] Rango, S.D., “*Higher Order Spatial Discretization for Turbulent Aeodynamics Flows*”, Doctor of Philosophy, University of Toronto, Canada, 2001.
- [28] Mavriplis, D., “*Unstructured Grid Techniques*”, Annu. Rev. Fluid Mech., Vol. 29, 1997, pp. 473-514.
- [29] Barth, T.J., “*Aspects of Unstructured Grids and Finite-Volume Solvers for the Euler and Navier-Stokes Equations*”, Von Karman Inst. For Fluid Dynamics, Lecture Series, 1994-05.
- [30] Peraire, J., Morgan, K., Piero, J., “*Unstructured Mesh Methods for CFD*”, Imperial College of Science Technology and Medicine, London, U.K., I.C. Aero Report 90-04, June, 1990.

- [31] Anderson, D.A., Tannehill, J.C., and Pletcher, R.H., “*Computational Fluid Mechanics and Heat Transfer*”, Series in Computational Methods in Mechanics and Thermal Sciences, Hemisphere Publishing Corporation, !984.
- [32] Thomas, J.L. and Salas, M.D., “*Far-Field Boundary Conditions for Transonic Lifting Solutions to the Euler Equations*”, AIAA Journal, Vol.24, No.7, pp.1074-1080.
- [33] Hirsch, C., “*Numerical Computation of Internal and External Flows*”, Vol.1, John Wiley and Sons, Chichester, 1988, pp.237-264.
- [34] Barth, T.J. and Jespersen, D.C., “*The Design and Application of Upwind Scheme on Unstructured Meshes*”, AIAA 89-0366, 1989.
- [35] Whitaker, D.L., “*Three Dimensional Unstructured Grid Euler Computations Usind A Fully Implicit Upwind Method* ”, AIAA 93-3337, 1993.
- [36] Roe, P.L., “*Approximate Riemann Solvers, Parameter Vector and Difference Schemes* ”, Journal of Computational Physics, Vol. 43, pp. 357-372, 1981.
- [37] Turkel, E, and Van Leer, B., “*Flux-Vector Splitting and Runge-Kutta Methods for the Euler Equations* ”, ICASE Report Number 84-27, NASA CR 172415, June 1984.
- [38] Jameson, A, and Yoon, S., “*Multigrid Solution of the Euler Equations Using Implicit Schemes* ”, AIAA Journal, Vol.24, No.11, pp.1737-1743, November 1986.

- [39] 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.
- [40] Crumpton P.I., Mackenzie, J.A. and Morton, K.W., “*Cell Vertex Algorithms for the Compressible Navier-Stokes Equations*”, Journal of Computational Physics, No. 109, pp. 1-15, 1993.
- [41] Thibert, J.J., Grandjacques, M. and Ohman, L.H., “*Experimental Data Base for Computer Program Assessment*”, AGARD AR-138, May 1979.
- [42] Pulliam, T.H. “*Artificial Dissipation Models for the Euler Equations*”, AIAA Journal, Vol.24, No.12, pp.1931-1940, December 1986.
- [43] Holst, T.L. “*Viscous Transonic Airfoil Workshop Compendium of Results*”, AIAA Paper, 87-01460, 1987.