IMPLEMENTATION OF TURBULENCE MODELS ON 2D HYBRID GRIDS USING AN EXPLICIT/IMPLICIT MULTIGRID ALGORITHM

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

BY

ALİ EMRE YILMAZ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
AEROSPACE ENGINEERING

SEPTEMBER 2011

Approval of the thesis:

IMPLEMENTATION OF TURBULENCE MODELS ON 2D HYBRID GRIDS USING AN EXPLICIT/IMPLICIT MULTIGRID ALGORITHM

submitted by **ALİ EMRE YILMAZ** in partial fulfillment of the requirements for the degree of **Master of Science in Aerospace Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan ÖZGEN ______________
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Ozan TEKİNALP ______________
Head of Department, **Aerospace Engineering**

Prof. Dr. İsmail H. TUNCER ______________
Supervisor, **Aerospace Engineering Dept., METU**

Examining Committee Members:

Prof. Dr. Yusuf ÖZYÖRÜK ______________
Aerospace Engineering Dept., METU

Prof. Dr. İsmail H. TUNCER ______________
Aerospace Engineering Dept., METU

Assist. Prof. Dr. D. Funda KURTULUŞ ______________
Aerospace Engineering Dept., METU

Dr. Emel MAHMUTYAZICIOĞLU ______________
Aerodynamics Division, TÜBİTAK-SAGE

Assist. Prof. Dr. Oğuz UZOL ______________
Aerospace Engineering Dept., METU

Date: 16 / 09 / 2011

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

Name, Last Name	: Ali Emre YILMAZ
Signature	:

ABSTRACT

IMPLEMENTATION OF TURBULENCE MODELS ON 2D HYBRID GRIDS USING AN EXPLICIT/IMPLICIT MULTIGRID ALGORITHM

YILMAZ, Ali Emre
M.Sc., Department of Aerospace Engineering
Supervisor: Prof. Dr. İsmail H. TUNCER
September 2011, 82 pages

In this thesis study, implementation, numerical stability and convergence rate issues of turbulence modeling are explored. For this purpose, a one equation turbulence model, Spalart-Allmaras, and a two-equation turbulence model, SST k-ω, are adapted to an explicit, cell centered, finite volume method based, structured / hybrid multi grid flow solver, SENSE2D, developed at TUBITAK-SAGE. Governing equations for both the flow and the turbulence are solved in a loosely coupled manner, however, each set of equations are solved using a coupled, semi-implicit solution algorithm. In multigrid solutions, the semi-implicit solution algorithm and the turbulence model equations are employed only in the finest level grid. As a result, stable and convergent numerical solutions are obtained. In order to validate the turbulence models and the semi-implicit solution algorithm implemented, turbulent flow solutions over a flat plate, RAE2822 airfoil and NLR7301 multi element airfoil are performed. The results are compared with the experimental data and the numerical results of the commercial CFD package FLUENT. It is shown that the numerical results obtained by SENSE2D are in good agreement with the experimental data and the FLUENT results. In addition to the turbulence modeling studies, convergence rate studies are also performed by multigrid and semi-implicit solution methods. It is shown that, the convergence rates of the semi-implicit solutions are increased about 5 times for single grid and 35% for multigrid solutions in comparison to the explicit solutions.

Keywords: Turbulence Modeling, Spalart-Allmaras Turbulence Model, SST k-ω Turbulence Model, Implicit Methods, Stability

ÖZ

TÜRBÜLANS MODELLERİNİN İKİ BOYUTLU ÇOK KATMANLI MELEZ ÇÖZÜM AĞLARI ÜZERİNDE AÇIK/ÖRTÜLÜ ALGORİTMA KULLANILARAK UYGULANMASI

YILMAZ, Ali Emre
Yüksek Lisans, Havacılık ve Uzay Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. İsmail H. TUNCER
Eylül 2011, 82 sayfa

Bu tez çalışması kapsamında, türbülans modeli uygulamaları ve uygulama sırasında karşılaşılan sayısal istikrar ve yakınsama oranı sorunları incelenmiştir. Bu amaçla, bir denklemli Spalart-Allmaras ve iki denklemli SST k-ω türbülans modelleri, TÜBİTAK-SAGE tarafından geliştirilen, hücre merkezli, sonlu hacim yöntemi tabanlı, çok katmanlı melez çözüm ağları ile çalışan akış çözücü SENSE2D'ye adapte edilmiştir. Akış ve türbülans denklemleri birbirlerinden ayrı ardışık biçimde çözülmekle beraber, herbir set kendi içlersinde denklemler birbirlerine bağımlı olacak şekilde, yarı-kapalı çözüm algoritması kullanılarak çözülmüştür. Çok katmanlı çözümlerde, yarı örtülü çözüm algoritması ve türbülans model denklemleri sadece en sıkı çözüm ağı seviyesinde kullanılmıştır. Sonuç olarak, istikrarlı ve yakınsak sayısal çözümler elde edilmiştir. Türbülans modellerini ve yarı örtülü çözüm algoritması uygulamasını doğrulamak amacıyla, düz levha, RAE2822 kanat profili ve NLR7301 çok elemanlı kanat profili üzerinde türbülanslı akış çözümleri yapılmıştır. Çözüm sonuçları, deneysel veriler ve ticari bir akış çözücünün, FLUENT, çıktıları ile karşılaştırılmıştır. SENSE2D ile elde edilen sayısal sonuçların, deneysel veriler ve FLUENT sonuçları ile iyi bir uyum içinde olduğu görülmüştür. Tez çalışmaları kapsamında, türbülans modelleme çalışmalarına ek olarak, çok katmanlı ve yarı-kapalı çözüm yöntemleri yakınsama hızları ile ilgili çalışmalar da yapılmıştır. Açık çözümlemeler ile karşılaştırıldığında, yarı-kapalı çözümlemelerin tek katmanlı çözüm ağları üzerinde yaklaşık 5 kat, çok katmanlı çözüm ağları üzerinde %35 daha hızlı yakınsadığı görülmüştür.

Anahtar Kelimeler: Türbülans Modelleme, Spalart-Allmaras Türbülans Modeli, SST k-ω Türbülans Modeli, Kapalı Çözüm Yöntemleri, Kararlılık

To my Parents

ACKNOWLEDGEMENTS

I am heartily thankful to my supervisor, Prof. Dr. İsmail H. TUNCER, whose encouragement, guidance and support from the initial to the final level enabled me to develop an understanding of the subject.

I would like to thank to my voluntary co-supervisors Dr. Emel MAHMUTYAZICIOĞLU and Dr. Mehmet Ali AK for their limitless support during the study.

This work has been supported by TÜBİTAK-SAGE. I wish to thank TÜBİTAK-SAGE for providing the computational power and literature sources. I would also like to convey my thanks to the TÜBİTAK-SAGE Aerodynamic Division members for their understanding.

I would like thank to my parents, Mualla - Necati YILMAZ, and my sister Hande UNCU since they are the indisputable pioneers of any successes in my life.

Lastly, I would like to thank my fairy, Zeynep ERGÜL. Her presence in my life is the main source of the energy that enabled me to complete this study.

TABLE OF CONTENTS

CHAPTERS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF SYMBOLS

C_d	Drag force coefficient
CFL	Courant number
C_l	Lift Force Coefficient
C_p	Pressure Coefficient
M	Mach Number
Re	Reynolds Number
x, y, z	Cartesian Coordinates
y^+	Non-Dimensional Normal Distance
u, v, w	Cartesian Velocities in x, y and z directions
t	Time

Greek Letters

ε	Turbulent Dissipation Rate (k-ε Turbulence Model)
k	Turbulent Kinetic Energy
μ	Dynamic Viscosity
ν	Kinematic Viscosity
ρ	Density
ω	Turbulent Dissipation Rate (SST k-ω Turbulence Model)

CHAPTER 1

INTRODUCTION

Flow solvers are indispensable tools of the aerodynamic design procedures. Today, even though the flow equations are solved with high accuracy, the turbulence equations keep their bottleneck position since the accuracy of the solution is mostly dependent on the prediction of the turbulent quantities. In industrial applications turbulence modeling is widely used due to its practical usage and numerical efficiency.

Two main problems encountered with the turbulence modeling are the slow convergence rate and the instability. While using small time steps makes the solution stable, it increases the convergence time significantly. Full implicit algorithms are the solution to the both instability and slow convergence problem, however, their numerical cost are usually prohibitive. Development of numerically less expensive, and at the same time, fast algorithms is among the current research activities.

The Multigrid (MG) technique is considered to be the most effective technique to achieve a reduction in the CPU cost of explicit flow solvers. The main restriction in front of the convergence rate of an explicit MG solver is the CFL condition, which is limited by 1. Unconditional stability of implicit algorithms can improve the convergence rate of an MG solver. In order not to destroy the numerical cost effectiveness of explicit MG solver, semi implicit algorithm can be implemented to the MG solver instead of implementing fully implicit scheme. Also, coupling flow variables improves the accuracy and convergence rate of the solution.

At TUBITAK–SAGE two and three dimensional, unstructured, finite volume, multi-grid flow solvers are currently being developed. They employ explicit solution algorithms and are capable of solving viscous flows on hybrid grids.

The main objectives of this work are to implement turbulence models and a semi implicit solution algorithm to the MG flow solver developed at TUBITAK-SAGE, SENSE2D, and to examine the improvements on the convergence rate of turbulent flow solutions. For this purpose, one equation turbulence model, Spalart-Allmaras, and two equation turbulence model, SST k-ω are implemented to a MG flow solver and explicit solution algorithm of the flow solver is replaced by a semi-implicit coupled solution algorithm in which flux jacobians are computed numerically.

1.1 Turbulence Modeling

The accurate prediction of the viscous flows, such that boundary layer separation, vorticity and skin friction, has primary importance in the design of the air vehicles. Although, inviscid and laminar flow solutions of the governing equations are generally straight forward and do not introduce any significant difficulties, in the case of turbulent flows numerical solutions may become problematic.

There are several ways so as to predict the turbulent flows. While some of them are capable of resolving all the scales in the flow requiring huge computational resources, some of them find themselves place in the engineering applications due to their reasonable needs for computational resources in the expense of accuracy. In addition, there are also some other techniques in between these two extremities. In Table 1, overview of the turbulence prediction methods are presented according to the increasing level of simplicity.

Table 1 Numerical turbulence prediction methods- Modeling and simulation

DNS			Level 0
LES			Level 1
RANS	2nd Order	RST	Level 2
		ARS	
	1st Order	0-Eq.	Level 3
		1-Eq.	
		2-Eq.	

The most accurate way of predicting the turbulence is the *Direct Numerical Simulation* (DNS). While the results of the DNS solutions are accepted to be exact, other prediction methods include some approximation when compared with DNS. For a DNS solution, number of grid points in the solution domain is proportional to the $Re^{9/4}$ and CPU time is proportional to Re^3. Therefore, in spite of the fast development in computational technologies, in today's world, the practical usage of DNS is only possible for only simplified problems. However, this does not make DNS useless. Results of DNS solutions of simplified problems are used in the validation and improvement of the turbulence models.

Large Eddy Simulation (LES) comes after the DNS and includes first level approximation. Its theory is based on the universal behavior of the small scales. According to the observations, only the large scales of the flow are dependent on the geometry and the small scales are independent of the geometry. Differing from the DNS, in LES only the large scale motions are solved and the small scale effects are approximated by the subgrid scale models. Although, LES requires less computational power when compared with the DNS, it is still computationally expensive and not practical for the engineering applications.

In today's world, the most widely used prediction method for the turbulent flow solutions is the turbulence modeling. When compared with DNS and LES, turbulence modeling requires much less computational power in the expense of high level of approximations. Although this technique houses high level approximations, results of numerical solutions are accurate enough for the engineering applications.

Most commonly used technique in the development of the turbulence models is the decomposition of the velocity and pressure terms in the Navier-Stokes equations into the mean and the fluctuating parts which is called *Reynolds Decomposition*. The decomposition of the density term makes the governing equations much more complicated. Also it is observed that up to Mach number 5 the turbulent characteristics of the flow are not affected by the density fluctuations considerably [1]. Therefore, below this limit, RANS equations are generally solved as if the flow is incompressible without any excessive errors.

Decomposing the flow variables and after some statistical approach, such as time or ensemble averaging, a new equation set similar to the Navier-Stokes equations is obtained. This new set of equations are called the *Reynolds Averaged Navier Stokes* equations (RANS) and presented first in 1985 by Reynolds [2]. Differing from the original Navier-Stokes equations, RANS equations include additional terms called *Reynolds Stresses* originating from the Reynolds decomposition. In its original form, the number of unknowns and the number of equations are equal to each other for Navier-Stokes equations. Therefore, the equation set is complete and can be solved for each flow variable. However, in RANS equations, although the number of the governing equations remains the same, the number of unknowns are more than the equations due to the additional unknowns, Reynolds stresses. In contrary to the original Navier-Stokes equations, RANS equations are not complete. This is called the *Closure Problem*. In order the RANS equations to be solved, additional equations must be introduced so as to complete the equation set.

Reynolds-Stress Transport (RST) and *Algebraic Reynolds-Stress* (ARS) models are the two members of the 2nd order closure models. In RST models all Reynolds stresses are directly evaluated by the solution of the transport equations developed for Reynolds stresses. Similar to the RST model, in ARS model only the two of the transport equations, generally for turbulent kinetic energy and the dissipation rate, are solved and the remaining Reynolds stresses are calculated by some algebraic relations. With this form, ARS models behave like intermediate methods between RST and 1st order models. Detailed explanation of the RST and ARS models can be found in reference [3].

Problems encountered in the implementation and solution of the RST and ARS models and also relatively expensive computational solution nature of the 2nd order models make 1st order models more popular in engineering applications. In these models, Reynolds stresses are directly computed by a scalar called *turbulent eddy viscosity* which is an additive artificial viscosity for the physical laminar viscosity. Boussinesq is the first who related the turbulent shear stress with the mean strain rate in a linear expression. By this hypothesis, turbulent eddy viscosity is calculated from the mean flow solutions and then added to the laminar viscosity for turbulent flow solutions. That is, the viscosity term in the NS equations are simply replaced by the sum of laminar and turbulent viscosities for turbulent flows solutions.

$$\mu = \mu_L + \mu_T \qquad \textbf{(Equation 1)}$$

The eddy viscosity term in Equation 1 is calculated by one of the turbulence models. Turbulence models are separated mainly into 3 categories depending on the number of transport equations solved in the calculation of eddy viscosity. These are: *zero*, *one* and *multiple* equation models.

In zero equation models, no transport equations are solved and the eddy viscosity is calculated by some algebraic relations. Therefore, these models are also named as the algebraic models. Zero equation models are simple to implement and also stingy in the computational resource usage. On the other hand, due to their local nature, zero equation models are lack of performance in the prediction of separated flows. Therefore, they are not used in the industrial applications anymore. The two most popular zero equation models are *Baldwin-Lomax* [4] and *Cebeci-Smith* [5] models.

In contrary to the zero-equation models, one and two equation models take into account the history by the solution of the convection and the diffusion terms in the transport equations. As a result, phenomena such that the boundary layer separation and reattachment are predicted much better than the zero equation models.

The most popular one equation model is the Spalart-Allmaras turbulence model [6]. which solves one transport equation directly for the eddy viscosity. It is especially developed for the external aerodynamic flows. The model is numerically very stable and easy to implement on both structured and unstructured grids. It is capable of modeling the laminar to turbulent transition at a predefined location. Also, its performance under the adverse pressure gradient is also quite high.

In the two equation turbulence models, two transport equations are solved for the two different turbulent transport variables from which the eddy viscosity is calculated algebraically. In most of the two equation turbulence models, one of the transport equations is solved for the turbulent kinetic energy, k. The other transport variable changes according to the turbulence model used. The most popular two equation turbulence models in the engineering applications are the k-ε model of Launder and Spalding [7] and the k-ω model of Wilcox [8].

The best aspects of the k-ε and the k-ω turbulence models are combined in the Shear Stress Transport k-ω turbulence model by Menter [9]. The new model has the capability of switching between the k-ε and the k-ω turbulence models by means of a blending function. In the near wall regions, the model employs the transport equations of the k-ω turbulence model since it does not require damping functions and therefore numerically much more stable than the k-ε turbulence model. Moreover, in the logarithmic region of the turbulent boundary layer, under the adverse pressure gradients and in the compressible flows, k-ω model found to be superior to the k-ε model. However, apart from the near wall effects, in the high Reynolds number regions, k-ω model is strongly sensitive to the free stream value of ω [10]. Therefore, in these regions the model switches to the standard k-ε model which is very accurate in the prediction of wakes, jets and mixing layers.

1.2 Temporal Discretisation

Temporal discretisation is the general name of the time integration techniques that are used in the solution of the Navier-Stokes (NS) equations. Basically there are two different approaches in the time integration of the governing equations: Explicit and implicit time advancement methods.

In explicit schemes, the time integration of the equations is dependent on the already known residuals from the current time level where the residual is defined as the complete right hand side of the governing equations remaining only the time derivative term on the left hand side. Moreover, each cell in the solution domain is integrated in time sequentially which makes the explicit schemes very effective in the computer source usage. The most popular explicit time stepping algorithm so far is the *Runge-Kutta* time stepping scheme.

The main restriction in front of the explicit schemes is the time step size. Explicit schemes are stable up to a certain step size which is dictated by the *Courant-Friedrichs-Lewi* (CFL) condition [11]. According to this condition, at each iteration, any information in the cell must stay in the cell and not cross the boundaries of the cell at the end of the corresponding iteration. Due to the small time steps taken at each iteration, convergence of explicit methods may take longer times when compared with the implicit methods.

Another difficulty encountered with the explicit schemes is related with the *stiffness* which can be defined as the ratio of the smallest to the largest time scale.

For the turbulent variables, changes take place at relatively smaller time scales when compared with the flow variables. In other words, time scales of turbulence and flow equations are different. The common time step size for the explicit solution of both the flow and the turbulence can be chosen according to the smallest time scale, but this time the convergence rate of the flow equations may get extremely low.

A more practical approach is based on the act of separating the time step sizes of the flow and the turbulence. This application allows the flow equations to be integrated in time with larger steps than the turbulence equations, keeping the turbulent solutions stable. Moreover, rapid development of the mean flow triggers the convergence of the turbulence. On the other hand, the smallest time scale of turbulence is formed in a limited region of the whole flow field. Therefore, integrating the turbulence equations with the smallest time step size in the whole domain retards the convergence of the turbulent quantities in the cells in which the larger steps are allowed.

The convergence rate of the turbulence quantities can be improved by imposing time steps specific to each cell. This can be obtained by the *point-implicit* treatment of large source terms of the turbulence equations[12]. By doing so, turbulence quantities in each cell are integrated with the maximum allowable time step that satisfies the stability condition.

It should be kept in mind that, the point implicit method is the way of automatic shifting of the time step sizes in each cell and is not necessarily expected to allow time step sizes larger than the ones used in the explicit methods. The word ‘Point’ in its title stands for the fact that each cell is updated with the information in itself. As a consequence, without receiving information from the neighboring cells, the algorithm works as an explicit algorithm with the maximum allowable time step size in each cell obeying the stability limits. As a consequence, a point implicit method leads to the fastest convergence rate that could be achieved by an explicit approach in the solution of the turbulence equations.

Different than the explicit time stepping methods, in implicit schemes, the time advancement of the variables are dependent on the unknown residuals from the next time level. By this application, solution stays stable up to much larger time step sizes when compared with its explicit counterpart. In other words, the CFL condition does not constitute a restrictive limit for the implicit methods. In addition, the stiffness problem encountered in explicit time stepping methods is directly alleviated by the usage of implicit schemes and it is possible to solve equations with different time scales without any special treatment.

In implicit schemes, the values of the residuals in the next time level are obtained by the *linearization* of the residual terms which requires the *flux Jacobian* calculation. Flux Jacobians are the derivatives of the residual term with respect to the flow variables. On contrary to the explicit schemes, implicit time advancement cannot be performed sequentially. After linearizing the residuals in each cell and writing down the governing equations, a system of equations is obtained. The coefficients of this equation set, which are the flux Jacobians and the time step terms, constitute a large sparse matrix called the *implicit operator*. Solution of the implicit methods requires the inversion of this implicit operator which is a numerically expensive process.

The inversion of the implicit operator can be achieved either by direct or iterative methods. Direct inversion methods such as Gauss elimination or other direct sparse matrix methods are not practical due to their excessive computer source requirements [13]. On the other hand, iterative methods based on either factorization or Krylov-subspace methods widely used in today's numerical applications. The most popular and widely used Krylov-subspace method is the *Generalized Minimal Residual* (GMRES) technique [14].

The inversion of the full implicit operator is numerically expensive and gets much more expensive as the number of elements in the solution domain grows. This directed the researchers to explore numerically more effective methods. One of the shortcuts in order to reduce the numerical effort in inverting the full implicit operator is using only a specific part of the implicit operator instead of the whole matrix. By this application, numerical effort in solving the governing equations is reduced considerably in return the full benefits of implicit schemes. One of the most popular techniques which uses a fraction of the implicit operator in the numerical solutions is the *semi-implicit* method. In this method, as it is in explicit methods, every cell in the solution domain is solved sequentially. Different than the explicit methods, in semi-implicit methods, working variables in the cells are directly updated as long as it is possible. By this way, each cell uses updated neighbor cell values if they are available. From a mathematical point of view, this application directly corresponds to the usage of the lower triangular part of the implicit operator in the solutions. As a consequence, with an explicit numerical cost, some fraction of the implicit method's benefits is achieved.

1.3 Multigrid Strategy

Multigrid method is one of the most powerful acceleration techniques that are used in the solution of the Navier-Stokes equations. It was first developed and used by Brandt [15] for the solution of the elliptic partial differential equations. After, method is used in the solution of the Euler and Navier-Stokes equations.

In a classical Navier-Stokes solution without multigrid application, effects of the boundaries are distributed over the domain as the iterations proceed and this process is continued till the changes in the solution settle down. Originating from this fact, convergence rate of the solution can be increased by increasing the transfer rate of the boundary effects to the inner parts of the domain from the boundaries.

In multigrid applications, different than the classical solution techniques, single grid solutions are followed by the coarse grid solutions. That is, solutions obtained from the fine level are distributed over the coarser grids and solved again. This process is repeated from fine to the coarse and then from coarse to the fine grids till the convergence is achieved. By this way, information is transferred from fine to the coarse and then from coarse to the fine solutions very rapidly which in practice is equivalent to distributing the boundary effects to the inner parts of the domain much more faster than in the case of single grid solutions.

From a mathematical point of view, accelerating effect of the multigrid applications can be explained as follows:

1. Due to the large cell volumes, larger time step sizes can be taken in the integration of the governing equations of the coarse grids. This leads to a faster convergence.
2. For iterative schemes, at each time step variables are updated with the calculated residuals. As the solution converges, changes in the flow variables go to zero in parallel with the decrease in the residual terms. Therefore, convergence rate of the solution is related with how fast the solver damps the residuals and makes them reach zero.

   In the earlier stages of the iterations, the changes in the working variables are high due to the large residual terms. This results in a sharp decrease in the residual terms that indicates high convergence rate towards the final solution. However, as the iterations proceed, changes in the working variables get smaller and smaller due to the shrinking residuals and as a result the convergence rate of the solution decreases. Therefore, the key for the high convergence rates is dealing with the large residuals. Multigrid techniques are applied to achieve this goal.

   In a multigrid solution, variables and residuals in the fine level are transferred to the coarse levels, that is, working variables and residuals of a group of fine cells are gathered in bigger cells. Summation of the small residuals from the fine cells results in the relatively large residual in the bigger cell. Since the residual is large, changes in the working variables in the coarse cells are large as in the initial stages of the iterations in the fine level and the convergence rate is high. In other words, instead of dealing with each small residual in separate cells of fine level, multigrid techniques combine small residuals from different cells in one bigger cell and damp these residuals at a time resulting in an accelerated convergence.

The main idea behind all multigrid methods is the same and explained briefly in the previous paragraphs. However, in application there are two different approaches: Geometrical multigrid (MG) and the Algebraic Multigrid (AMG).

As in the case of all other methods using grids in the discretisation of the solution domain, a physical fine grid is generated for both MG and AMG applications. In MG applications, in addition to the fine level grid, coarse level grids must be generated and stored to be used in the calculations. Contrary to the MG methods, in AMG applications which are especially developed for the implicit schemes, coarse level grids are not generated physically and the multigrid theory is applied to the implicit operator virtually. For this purpose, an implicit operator is constructed for the fine level solutions. For the coarse level solutions, a coarsening matrix is applied to the implicit operator of the fine level and the dimensions of the resulting implicit operator of the coarse level are rearranged according to the rules of the applied multigrid method.

1.4 Objectives and Summary

SENSE2D is a two-dimensional, finite volume, multigrid laminar flow solver on hybrid grids which is developed by TÜBİTAK-SAGE. It employs Roe's upwind flux differencing scheme for convective fluxes and three stage Runge-Kutta scheme for time integration. The grid coarsening is achieved by agglomerating the unstructured/hybrid cells based on their localization on a quadtree data.

The aim of this study is to bring SENSE2D the capability of dealing with the turbulent flows by means of implementing the turbulence models. For this purpose, two most popular one and two-equation turbulence models, namely Spalart-Allmaras and SST k-ω, are adapted.

At the initial stages of the study, it seemed to be easy to implement the models to the code since the equations are posted everywhere and there exists many commercial and in-house codes dealing with the turbulent flows. However, after many unsuccessful tries it is realized that the implementation is not straightforward as expected.

Introduction of the turbulence model equations to SENSE2D resulted in a stiff, unstable algorithm. Due to the large source terms of the turbulence model equations, explicit time integration techniques remained inadequate in having a stable, convergent solution.

The action taken against the instability problem is the treatment of the source terms of the turbulence model equations in a point implicit manner. By this application, solutions are able to be stabilized.

Following the stabilization of the solutions, another problem encountered with the turbulence modeling is the slow convergence rates. The point implicit method is basically an explicit algorithm and stabilizes the solution by reducing the explicit time step size of the turbulence model equations in an automatic way. Therefore, it is required to implement some sort of implicit method that associates several cells for the solution of the equations in order to overcome the slow convergence rate problem. For this purpose, a semi-implicit solution algorithm is implemented for both the solution of the flow and the turbulence model equations. Moreover, the flow and the turbulence model equation sets are solved in a loosely coupled manner, but, each set is solved coupled in itself.

With the aid of the methods explained, stable and convergent turbulent flow solutions are obtained by Spalart-Allmaras and SST k-ω turbulence models. Moreover, convergence rate of the solver is increased about 5 times in the single-grid solutions.

In order to complete the study, the semi-implicit solution algorithm and turbulence models are brought up and running for the multigrid applications. For this purpose, both of them are applied in the finest level grid. With the implementation of the semi-implicit algorithm, 35% of convergence rate increase is obtained in the multigrid solutions.

CHAPTER 2

METHOD

In this chapter, CFD methodology related with the scope of this study is explained in detail. In addition, some other methods that are not used directly but related with the study are mentioned briefly or referenced so as to ensure completeness.

2.1 Governing Equations

SENSE2D is a FVM based NS solver. Application of FVM requires the division of the solution domain into regions with finite volumes. Then the governing equations are obtained by the application of the *conservation law* to the volumes.

According to the conservation law, amount of any quantity enclosed by a finite volume at rest is changed depending on the net flux across the boundaries and the sources of that quantity in that volume. According to this description, general conservation law for an arbitrary quantity, W, can be expressed as

$$\frac{\partial}{\partial t}\int_{\Omega} W\, d\Omega + \oint_{\partial\Omega}\left[\left(\overrightarrow{F_C} - \overrightarrow{F_v}\right).\vec{n}\right]dS = \int_{\Omega} Q\, d\Omega \qquad \textbf{(Equation 2)}$$

Where Ω is the volume of the cell, dS is the finite piece of the control surface, $\overrightarrow{F_C}$ and $\overrightarrow{F_v}$ are the net convective and viscous fluxes across dS and Q is the source of W in the control volume.

Then, governing equations of the fluid flow are derived by the application of the general conservation equation, Equation 2, to the mass, momentum and energy.

Table 2 Governing Equations of the Flow

Conservation of Mass $\frac{\partial}{\partial t}\int_{\Omega} \rho \, d\Omega + \oint_{\partial\Omega} \rho(\vec{v}.\vec{n})\, dS = 0$
Conservation of Momentum $\frac{\partial}{\partial t}\int_{\Omega} \rho\vec{v}\, d\Omega + \oint_{\partial\Omega} \rho\vec{v}\,(\vec{v}.\vec{n})\, dS = \int_{\Omega} \rho\vec{f}\, d\Omega - \oint_{\partial\Omega} p\,\vec{n}\, dS + \oint_{\partial\Omega} \bar{\bar{\tau}}.\vec{n}\, dS$
Conservation of Energy $\begin{aligned} &\frac{\partial}{\partial t}\int_{\Omega} \rho\vec{E}\, d\Omega + \oint_{\partial\Omega} \rho E\,(\vec{v}.\vec{n})\, dS \\ &\qquad = \oint_{\partial\Omega} k(\nabla T.\vec{n})\; dS + \int_{\Omega} \left(\rho\vec{f}.\vec{v} + \dot{q}_h\right) d\Omega - \oint_{\partial\Omega} p\,(\vec{v}.\vec{n})\, dS \\ &\qquad + \oint_{\partial\Omega} (\bar{\bar{\tau}}.\vec{v}).\vec{n}\, dS \end{aligned}$

Governing equations of the flow are given in Table 2. In these equations, ρ is the density, $\vec{v}$ is the velocity vector, p is the isotropic pressure component, $\bar{\bar{\tau}}$ is the viscous stress tensor, $\vec{f}$ is the body force, ∇T is the temperature gradient, k is the thermal conductivity coefficient and $\dot{q}_h$ is the time rate of heat transfer per unit mass. Detailed information about the derivation of the governing fluid flow equations from conservation law can be found in references [11] and [16].

2.2 Numerical Solutions of the Governing Equations

In practice, the numerical solution of the conservation law equation, Equation 2, and the resulting flow governing equations requires the replacement of integral and differential terms with simple algebra.

The solution domain is composed of a number of elements of which the volumes are constant and independent of time during the calculations. In addition, assuming a finite time increment, the corresponding change in the flow variable in the control volume will also be finite. Applying these assumptions to the time derivative term in Equation 2 the following form is obtained.

$$\frac{\partial}{\partial t}\int_{\Omega} W \, d\Omega \approx \frac{\Delta \mathrm{W}}{\Delta \mathrm{t}}\Omega \qquad \textbf{(Equation 3)}$$

The surface integral of the flux terms on the left hand side of Equation 2 can be replaced by the sum of the fluxes across the faces of the control volume. That is,

$$\oint_{\partial\Omega} [(\overrightarrow{F_C} - \overrightarrow{F_v}).\vec{n}]dS \approx \sum_{k=1}^{N_f} (\overrightarrow{F_C} - \overrightarrow{F_v})_k \Delta S_k \qquad \textbf{(Equation 4)}$$

where N_f is the number of the faces, *k* is the face index of the control volume and ΔS_k is the area of face *k*.

Lastly, the source term is assumed to be constant in the control volume. Therefore,

$$\int_{\Omega} Q \, d\Omega \approx Q\Omega \qquad \textbf{(Equation 5)}$$

Gathering previous three equations, one can get the suitable form of the conservation law equation for the numerical solutions such that,

$$\frac{\Delta \mathrm{W}}{\Delta \mathrm{t}}\Omega = -\sum_{k=1}^{N_f} (\overrightarrow{F_C} - \overrightarrow{F_v})_k \Delta S_k + Q\Omega \qquad \textbf{(Equation 6)}$$

The complete right hand side of Equation 6 is called the *residual* and denoted by *R.* In the governing equations of the flow the residual is equal to the net flux across the cell boundaries. However, in the transport equations of any turbulence models, the residual term may also contain the source terms in addition to the flux terms.

Using this final definition and rewriting Equation 6 for cell *I*, the most common form of the conservation equation is obtained as

$$\frac{\Delta W_I}{\Delta t}\Omega_I = -R_I = -\sum_{k=1}^{N_f}\left(\overrightarrow{F_C} - \overrightarrow{F_v}\right)_k \Delta S_k + (Q\Omega)_I \qquad \textbf{(Equation 7)}$$

The left and the right hand sides of Equation 7 can be solved either by simultaneously or sequentially. In this study, time integration is performed just after the calculation of the right hand side of the governing equations, which is in fact a sequential approach. This method is called the *method of lines*. By doing so, depending on the case, different discretisation techniques can be applied in space and time in order to obtain more accurate results.

2.2.1 Spatial Discretisation

Spatial discretisation is a phrase used in place of the numerical approximation of the convective and the viscous fluxes as well as the source terms. *Finite difference*, *finite element* and the *finite volume* methods are the three spatial discretisation techniques that are widely used in the CFD applications. In SENSE2D, the governing equations of the flow are discretisized in space by FVM as well as the transport equations of the turbulence models' done. Therefore, in this section the attention is focused on this topic.

The FVM is applied to the integral form of the NS equations, that is, its theory is based on the general conservation law given by Equation 2. In SENSE2D, the construction of the control volumes is performed by the *cell centered* scheme. In a cell centered FVM method, control volumes are the same with the grid and the variables are calculated at the centers of the cells.

After constructing the control volumes in the solution domain, the next step is the calculation of the convective and the viscous fluxes on the faces of the control volumes. Since the conservative variables are located at the cell centers, accurate calculation of the fluxes at the faces, especially the convective ones, requires the interpolation of the flow variables or the fluxes which is not a straightforward process.

Due to their stationary nature, the only way of approximating the viscous fluxes is employing the *central schemes*. Note that the central and the cell-centered schemes are totally different concepts. The central schemes are performed either by the flux or the variable averaging approaches.

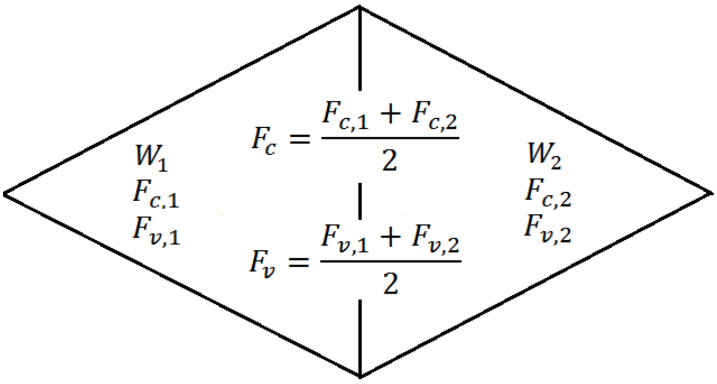

W_1
$F_{c.1}$
$F_{v,1}$
$F_c = \frac{F_{c,1} + F_{c,2}}{2}$
$F_v = \frac{F_{v,1} + F_{v,2}}{2}$
W_2
$F_{c,2}$
$F_{v,2}$

Figure 1 Calculation of the viscous fluxes by flux averaging.

The first way of calculating the viscous fluxes on the surfaces of the control volumes is based on the direct averaging of the cell centered viscous fluxes. In this method, viscous fluxes are calculated at the cell centers with the use of the cell centered variables. After, approximate fluxes on the faces are calculated by the arithmetic average of the fluxes of the neighbor cells. This, method is illustrated in Figure 1.

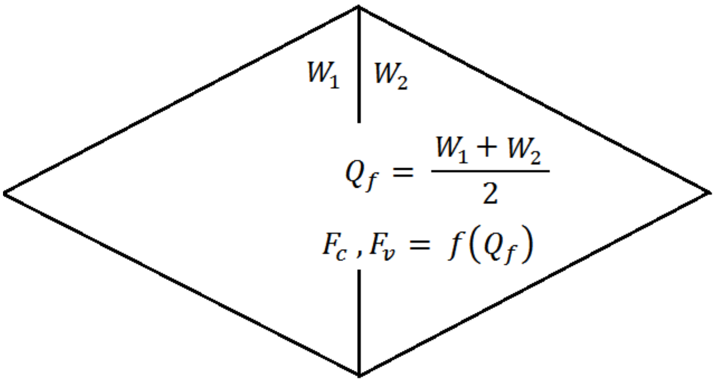


Figure 2 Calculation of the viscous fluxes by variable averaging.

The second approach used in the calculation of the viscous fluxes on the faces of the control volumes is based on the averaging of the variables. In this approach, the approximate values of the variables on the common face are obtained by taking the arithmetic average of the cell centered values. Then, the viscous fluxes on the common face are calculated by using these averaged variables. This, method is illustrated in Figure 2. In the calculations of SENSE2D, variable averaging method is used.

The treatment of the convective fluxes is not as simple as in the case of the viscous fluxes. In the evaluation of the convective fluxes, the influence of the upstream and the downstream may not necessarily be the same and generally differentiated by one of the *upwind schemes* depending on the local Mach regime. The upwind schemes can be divided into four main groups:

- *Flux Vector Splitting,*
- *Flux Difference Splitting,*
- *Total Variation Diminishing (TVD),*
- *Fluctuation Splitting* schemes.

The detailed information about the upwind schemes given above can be found in reference [17]. The Roe's flux difference splitting method is proved to be quite accurate in the computation of the boundary layers and capturing the discontinuities, such as shock waves. Therefore, it is used in the computation of the convective fluxes of the flow governing equations throughout this study.

Although the spatial discretisation of the flow governing equations are mostly performed by one of the upwind methods touched in the previous paragraph, more simple upwind methods, such as *simple upwinding,* can be used in the computation of the turbulent convective fluxes since the turbulent quantities can be defined as local. In other words, the upstream and the downstream influences on these quantities are expected to be similar. In this application, the convective fluxes of the turbulent quantities are calculated depending on the direction of the averaged normal velocity vector at the surface. If the averaged normal velocity vector is right-going, that is directed towards the right cell, the convective flux of the turbulence quantity at that face is calculated by using the left cell's variable and just the opposite in the case of left-going velocity vectors.

The last step in the calculation of the complete residual, if exists, is the computation of the source terms. The source terms are generally accepted as constant throughout the cell and evaluated by simply multiplying the source magnitude with the volume of the cell.

It is appropriate to finish this section with the gradient calculation method which is necessary for the computation of the viscous fluxes and the turbulent quantities. One of the widely used gradient evaluation methods is based on the Green-Gauss theorem. The method approximates the gradient of any scalar in a control volume by integrating its value over the surface of the control volume and then dividing the result by the volume. The Green-Gauss theorem is formulated in Equation 8.

$$\nabla W \approx \frac{1}{\Omega} \int W \, dS \qquad \textbf{(Equation 8)}$$

After the calculation of the net convective and the viscous fluxes as well as the source terms in the cells, the residual term can be evaluated by simply adding these terms. The governing equations are then ready to be advanced in time.

2.2.2 Temporal Discretisation

Temporal discretisation is the general name of the techniques that are used in the integration of the governing equations in time. At each time step, variables are updated by a difference, $\Delta W,$ in the form of

$$W_I^{n+1} = W_I^n + \Delta W_I \qquad \textbf{(Equation 9)}$$

Where ΔW_I is calculated from Equation 7 after the calculation of the residual and the time advancement.

Depending on the treatment of the residual term, temporal discretisation of the governing equations may have the form of explicit or implicit.

2.2.2.1 Explicit Temporal Discretisation

In explicit time stepping, residuals are calculated at the same time level with the flow variables.

$$\frac{\Omega_I}{\Delta t_I} \Delta W_I^n = -R_I^n \qquad \textbf{(Equation 10)}$$

After the calculation of all residuals, each cell in the domain is updated separately by its own residual independent of already updated variables in the neighbor cells. In other words, solution is based only on already known residuals of the cell itself.

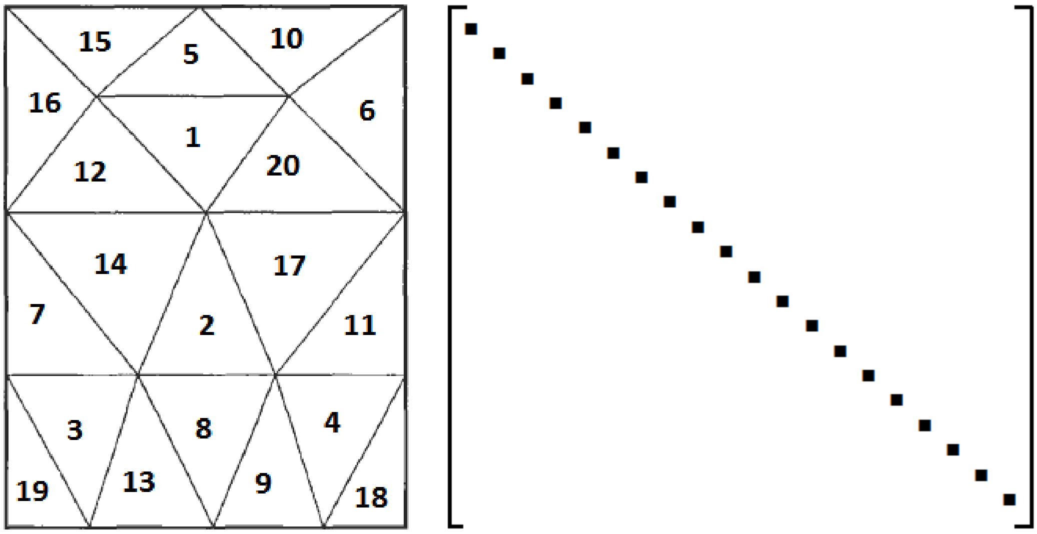


Figure 3 Unstructured mesh (left) and its explicit operator (right).

In Figure 3, an arbitrary domain with 20 unstructured cells and its explicit operator is demonstrated. Although it is unnecessary to represent an explicit operator in matrix form, it has importance in seeing the big picture.

Each line in the operator matrix stores the cell-neighbor relation. While the diagonal elements in the operator matrix represent each cell to be updated in the domain, the off-diagonal elements represent their neighbors. In each line, the marked cells correspond to the cells that are used in the update of the diagonal cell. In explicit time integration, the only cell needed so as to update a cell is the cell itself. Therefore, only the diagonal elements are marked in Figure 3.

One of the most popular explicit time stepping methods so far is the *Runge-Kutta* method [18], in which the time advancement is performed in multi stages.

$$
\begin{aligned}
W_I^{(0)} &= W_I^n \\
W_I^{(1)} &= W_I^{(0)} - \alpha_1 \frac{\Delta t_I}{\Omega_\mathrm{I}} R_I^{(0)} \\
W_I^{(2)} &= W_I^{(0)} - \alpha_2 \frac{\Delta t_I}{\Omega_\mathrm{I}} R_I^{(1)} \\
&\vdots \\
W_I^{(n+1)} &= W_I^{(m)} \\
&= W_I^{(0)} - \alpha_m \frac{\Delta t_I}{\Omega_\mathrm{I}} R_I^{(m-1)}
\end{aligned}
\qquad \textbf{(Equation 11)}
$$

In Equation 11, formulation of an m-stage Runge-Kutta time stepping is presented, where α terms are the stage coefficients. Different than the classical Runge-Kutta method, only the first and last residuals are stored in order to reduce memory allocations.

2.2.2.2 Implicit Temporal Discretisation

Different than the explicit schemes, in the implicit schemes residual is calculated in the next time level

$$
\frac{\Omega_I}{\Delta t_I} \Delta W_I^n = -R_I^{n+1} \qquad \textbf{(Equation 12)}
$$

In order to solve Equation 12, R_I^{n+1} must be calculated. For this purpose, the term R_I^{n+1} is linearized as follows

$$
R_I^{n+1} \approx R_I^n + \left(\frac{\partial R_I}{\partial W}\right)^n \Delta \mathrm{W}^n \qquad \textbf{(Equation 13)}
$$

where the derivative term on the right hand side of Equation 13 is the *flux Jacobian*. Evaluation of the flux Jacobian is one of the most critical aspects in the implicit solution algorithms.

Flux Jacobians can be obtained either by analytically or numerically. Analytical derivation of the flux Jacobians, especially the viscous flux Jacobians, is generally challenging if not impossible. Also, they must be derived for each flux discretisation method separately. Even if the analytical derivations of the Jacobians are performed by hand calculation or symbolic mathematic tools, numerical effectiveness of the code can be poor [19].

Alternative for the analytical Jacobian is the numerical Jacobian. Numerical Jacobians are easy to implement but less accurate. In this technique, Jacobians are calculated by using the finite-difference method with small perturbation technique such that,

$$\frac{\partial R_i}{\partial W_j} \approx \frac{R_i(W_j + h) - R_i(W_j)}{h_j} \qquad \textbf{(Equation 14)}$$

The step size h in the evaluation of Jacobian is in the form of [20]

$$h_j = \sqrt{\varepsilon}\, max(|W_j|, typ\ W_j)\ sign\ (W_j) \qquad \textbf{(Equation 15)}$$

where ε is the machine epsilon and $typ\ q_j$ is the typical size of q_j which is taken as 1 in this study. Machine epsilons for single and double precision calculations are $1.19\text{x}10^{-7}$ and $2.2\text{x}10^{-16}$ [21].

In Equation 13, subscripts of all the terms except W are denoted as *I* indicating the belonging to the cell itself. In an implicit method, residual term must include at least the Jacobian of the cell itself. In addition, residual term may contain Jacobians of some or all of the neighboring cells depending on the type of implicit method used. Therefore, subscripts of W terms changes according to the implicit method. That is why the subscript regions of the W terms are remained blank.

Assuming a two-dimensional unstructured domain with triangular cells and including the effects of all neighboring cells as well as the cell itself on the residual term in Equation 13, the residual formulation of the full-implicit method is obtained as follows

$$R_I^{n+1} \approx R_I^n + \left(\frac{\partial R_I}{\partial W_I}\right)^n \Delta \mathrm{W}_I^n + \left(\frac{\partial R_I}{\partial W_{I1}}\right)^n \Delta \mathrm{W}_{I1}^n + \left(\frac{\partial R_I}{\partial W_{I2}}\right)^n \Delta \mathrm{W}_{I2}^n + \left(\frac{\partial R_I}{\partial W_{I3}}\right)^n \Delta \mathrm{W}_{I3}^n \quad \textbf{(Equation 16)}$$

where *I1, I2* and *I3* stand for the respective neighbors of cell *I*. Substituting Equation 16 in Equation 12, governing equation for cell *I* is obtained in the form of

$$\left[\frac{\Omega_I}{\Delta t_I} + \left(\frac{\partial R_I}{\partial W_I}\right)^n\right] \Delta \mathrm{W}_I^n + \left(\frac{\partial R_I}{\partial W_{I1}}\right)^n \Delta \mathrm{W}_{I1}^n + \left(\frac{\partial R_I}{\partial W_{I2}}\right)^n \Delta \mathrm{W}_{I2}^n + \left(\frac{\partial R_I}{\partial W_{I3}}\right)^n \Delta \mathrm{W}_{I3}^n = -R_I^n \quad \textbf{(Equation 17)}$$

Equation 17 includes 4 unknowns and cannot be solved without having extra equations. That is, differing from the explicit methods, in the full-implicit methods all cells in the domain are required to be solved simultaneously.

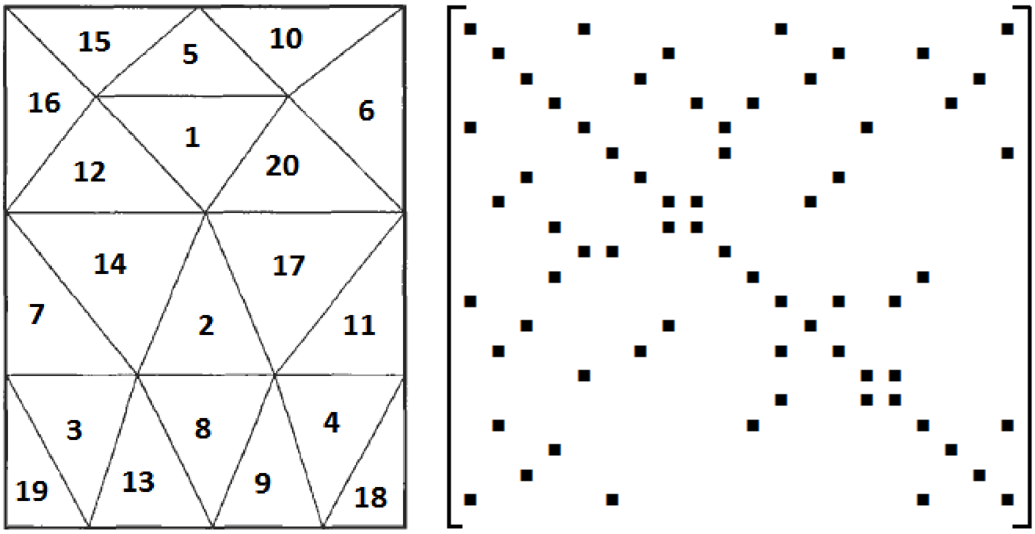


Figure 4 Unstructured mesh and its implicit operator (Nearest neighbor stencil)

Applying Equation 17 to all cells in the domain, a large sparse matrix with a structure similar to the one given in Figure 4 is obtained. This matrix is called the *implicit operator*. In order to solve this linear system, inverse of the implicit operator must be computed. This can be done either by the direct inversion or the iterative methods. Generally, direct inversion methods such as Gauss elimination or some direct sparse matrix methods [22] are not desired for this type of applications due to their high computer source requirements. Instead, iterative methods are used since they are more practical for this type of applications due to low computer source needs. Some of the most popular iterative sparse matrix solvers are Alternating Direction Implicit (ADI) [23], Gauss-Seidel[24], and GMRES[14].

In a *point-implicit* method, the effects of the neighboring cells are ignored and only the values of the cell itself are used. Updating the formulation in Equation 13 according to this rule, the residual formulation of the point-implicit scheme is obtained.

$$R_I^{n+1} \approx R_I^n + \left(\frac{\partial R_I}{\partial W_I}\right)^n \Delta \mathrm{W}_I^n \qquad \textbf{(Equation 18)}$$

Substituting Equation 18 in Equation 12 and rearranging, the following point-implicit scheme is obtained

$$\left[\frac{\Omega_I}{\Delta t_I}+\left(\frac{\partial R_I}{\partial W_I}\right)^n\right]\Delta \mathrm{W}_I^n = -R_I^n \qquad \textbf{(Equation 19)}$$

where the term in the brackets on the left hand side is the implicit operator of the point implicit method.

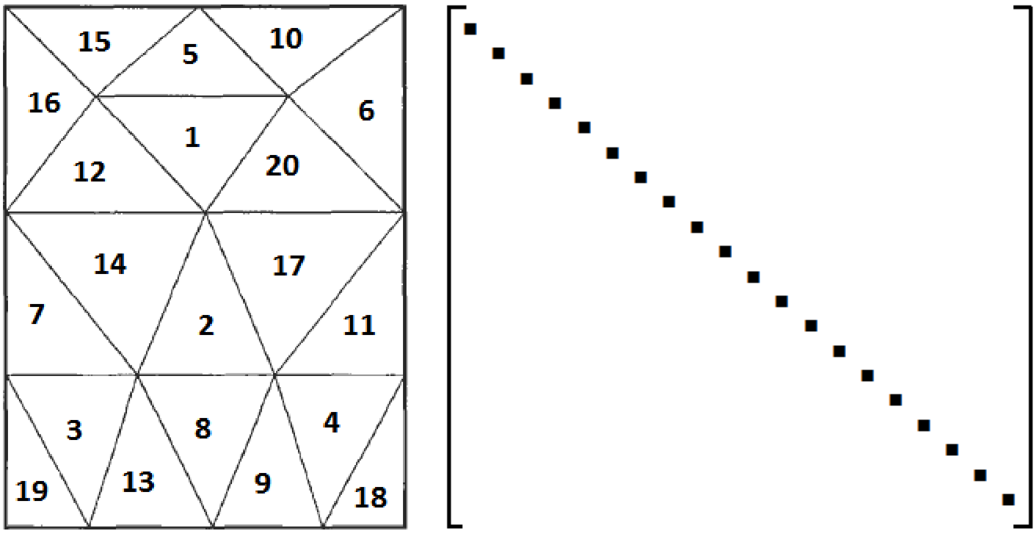


Figure 5 Unstructured mesh (left) and its semi-implicit operator (right).

Figure 5 shows the unstructured grid distribution and the related implicit operator for point-implicit methods. Comparing Figure 3 and Figure 5, it is seen that the structures of the matrix operators for the explicit and the point-implicit methods are exactly the same. That is, for both the explicit and the point-implicit schemes time advancement of the variables in a cell depends only on the properties of the cell itself, not its neighbors. Difference comes from the fact that, diagonal elements of the implicit operator of the point-implicit formulation also include the Jacobian terms in addition to the Ω/Δt term when compared to the diagonal elements of the explicit operator. Note that, Jacobian term in a point implicit method accompanies the time step size, therefore acts as a time-like term. It may increase or the decrease the effective time step size in the cells depending on the flow. Different than changing the CFL number manually, point implicit methods change the time step size locally and automatically. By this method, governing equations, especially with *stiff* terms, are made more stable in time. On the other hand, since the update of the flow variables in a cell is dictated only by the cell own properties and not related with the neighbor cells, equivalent time step size, which is composed of the explicit time step size and the Jacobian term, must obey the CFL condition. Therefore, point implicit method can be defined as a kind of explicit method which is more stable in time compared to the classical version.

There is also another method between the explicit and the full-implicit methods. Examining Figure 3 and Figure 4, a semi implicit operator can be constructed.

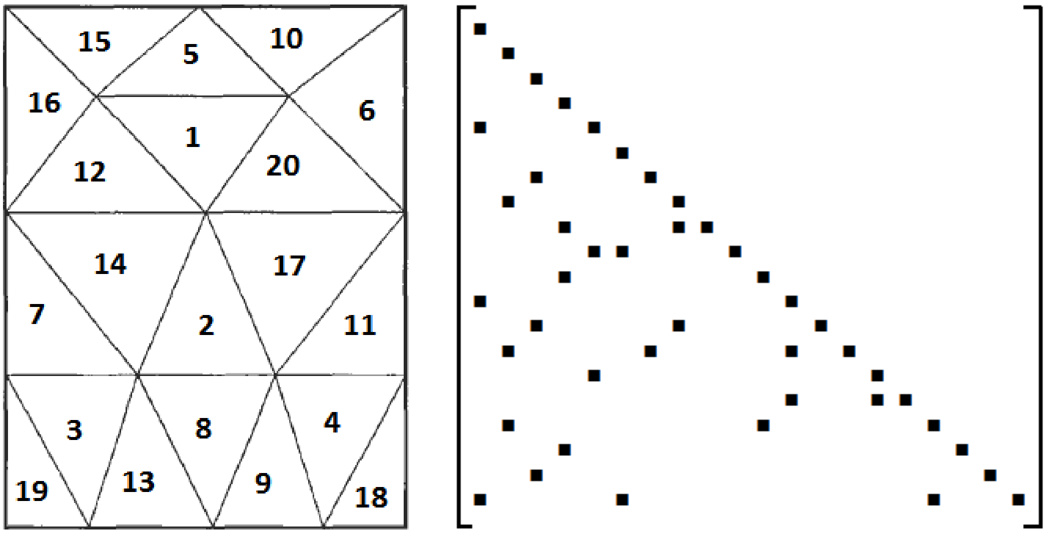


Figure 6 Unstructured mesh and its semi-implicit operator (Nearest neighbor stencil)

Semi-implicit operator In Figure 6 is composed of the diagonal and the lower diagonal elements. Although, it seems random to construct a matrix operator standing between the explicit and the full-implicit operators as demonstrated in Figure 6, in fact, there is a specific reason in using the diagonal and the lower diagonal elements especially.

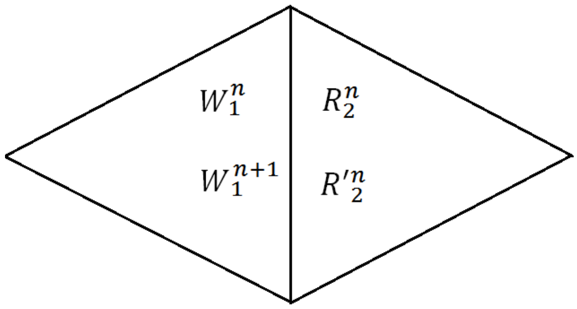


Figure 7 Two Neighboring Cells (Explanation of Semi-Implicit Method)

In Figure 7 two neighboring cells are illustrated. In the first scenario, the residual of cell 2, R_2^n, is calculated by using the outdated value of cell 1, W_1^n. In the second scenario, residual of cell 2, R'^n_2, is calculated by using the updated value of cell 1, W_1^{n+1}. In cell 2, the difference between the residuals of the two scenarios directly arises due the change in the W variable in cell 1 from one scenario to the other. In other words, R'^n_2 results from the change in R_2^n due to the change in W_1 in time. Formulating,

$$R'^n_2 - R_2^n = \frac{\partial R_2^n}{\partial W_1^n}\left(W_1^{n+1} - W_1^n\right) \qquad \textbf{(Equation 20)}$$

Rearranging Equation 20

$$R'^n_2 = R_2^n + \frac{\partial R_2^n}{\partial W_1^n}\,\Delta W_1^n \qquad \textbf{(Equation 21)}$$

Comparing Equation 21 with Equation 16, it is seen that $R'^n_2 = R_2^{n+1}$ with only lower index neighbors included.

During semi-implicit solution, every cell in the domain is visited and updated sequentially. Differing from an explicit time stepping, the updated value of each cell is immediately used in the update of upcoming cells. By this way, without calculating any Jacobians and performing extra work, lower diagonal of full-implicit operator is solved with a cost of an explicit scheme.

A different but related topic with this section is the coupling of the flow variables. Up to this point, only the influences of the neighboring cells to each other are discussed in the scope of the implicit algorithms. However, the influences of the flow variables to each other can also be taken into account in the solution of the NS and turbulence model equations.

In the preceding paragraphs, the link between the cell and its neighbors is created using the Jacobians. Moreover, the Jacobians are accepted as the derivatives of the residual terms with respect to the conservative variables of the same kind in the cell itself as well as in the neighbor cells. In other words, the residual of the conservation of mass equation is assumed to be affected only by the changes in the density variable of the cell itself and the neighbor cells. As a result, the effect of the changes in the other conservative variables on the density residual term is ignored.

Rewriting Equation 12 after replacing the general scalar, W, by the density variable, the implicit form of the conservation of mass equation is obtained.

$$\frac{\Omega_I}{\Delta t_I}\Delta\rho_I^n = -R_I^{n+1} \qquad \textbf{(Equation 22)}$$

Linearizing the residual term with respect to the density term in the cell itself, the point implicit formulation of the conservation of mass equation is obtained.

$$\frac{\Omega_I}{\Delta t_I}\Delta\rho_I^n = -\left(R_I^n + \frac{\partial R_I}{\partial \rho_I}\Delta\rho_I^n\right) \qquad \textbf{(Equation 23)}$$

In addition, the change in the density variable can also be related with the other conservative variables in the form of Equation 24.

$$\frac{\Omega_I}{\Delta t_I}\Delta\rho_I^n = -\left(R_I^n + \frac{\partial R_I}{\partial \rho_I}\Delta\rho_I^n + \frac{\partial R_I}{\partial \rho u_I}\Delta\rho u_I^n + \frac{\partial R_I}{\partial \rho v_I}\Delta\rho v_I^n + \frac{\partial R_I}{\partial \rho e_I}\Delta\rho e_I^n\right) \qquad \textbf{(Equation 24)}$$

Equation 24 has four unknowns and cannot be solved without having extra equations. Treating the other governing equations in the same manner, the equation set is completed and can be solved simultaneously.

Note that the method introduced so far is the *point-implicit-coupled* scheme, that is, neighbor relations are ignored. Nevertheless, it requires the simultaneous solution of four equations. Therefore, coupling of the flow variables in full implicit algorithms is numerically very expensive.
In the semi-implicit calculations of SENSE2D, only the diagonal and the lower diagonal elements of the full-implicit operator are used. The effects of the diagonal elements are taken into account by the treatment of the residual terms in a point-implicit-coupled manner. Moreover, the lower-diagonal elements are counted in the calculations by using the updated values immediately in the calculations of the remaining cells.

2.3 Turbulence Modeling

In the scope of this thesis work, two popular turbulence models, Spalart-Allmaras and SST k-ω, are implemented to the Navier-Stokes solver, SENSE2D.

2.3.1 Spalart-Allmaras Turbulence Model

For the solution of the turbulent flows, one equation Spalart-Allmaras turbulence model [25] is implemented in its nondimensional form. The turbulent viscosity is nondimensionalized by the free stream viscosity, υ_∞. and directly calculated from the transport equation in the form of Equation 25.

$$\frac{\partial \tilde{\upsilon}}{\partial t} + \nabla \cdot (V\tilde{\upsilon}) = c_{b_1} S\tilde{\upsilon} + \frac{M}{Re}\frac{c_{b_1} f_{v_2}}{K^2}\left(\frac{\tilde{\upsilon}}{d}\right)^2 - \frac{M}{Re} c_{w_1} f_w \left(\frac{\tilde{\upsilon}}{d}\right)^2 + \frac{M}{Re}\frac{1}{\sigma}\left[\nabla \cdot \left((\upsilon_L + \tilde{\upsilon})\nabla\tilde{\upsilon}\right) + c_{b_2}(\nabla\tilde{\upsilon})^2\right] \qquad \textbf{(Equation 25)}$$

Where,

$$\upsilon_t = \tilde{\upsilon}\, f_{v1},$$

$$C_{b1} = 0.1355, \quad C_{b2} = 0.622, \quad C_{w2} = 0.3, \quad C_{w3} = 2, \quad \kappa = 0.41, \quad \sigma = {}^{2}/_{3},$$

$$C_{w1} = \frac{C_{b1}}{\kappa^2} + \frac{(1 + C_{b2})}{\sigma}, \quad f_{v1} = \frac{X^3}{X^3 + C_{v1}^3}, \quad f_{v2} = 1 - \frac{X}{1 + X\, f_{v1}},$$

$$f_w = g\left[\frac{1 + C_{w3}^6}{g^6 + C_{w3}^6}\right], \quad X = \frac{\tilde{\upsilon}}{\upsilon}, \quad g = r + C_{w2}(r^6 - r), \quad r = \frac{\tilde{\upsilon}}{\tilde{S}\,\kappa^2\, d^2}\frac{M}{Re}$$

$$\tilde{S} = S + \frac{\tilde{\upsilon} f_{v2}}{\kappa^2\, d^2}\frac{M}{Re}, \quad S = \left|\Omega_{ij}\right|, \Omega_{ij} = \frac{1}{2}\left(\frac{\partial u}{\partial y} - \frac{\partial v}{\partial x}\right),$$

$$d = Wall\ Distance$$

In its complete form Spalart-Allmaras turbulence model has extra terms such that transition damping of production and transition source of turbulence. With these additional terms, the model has the capability of predicting the laminar to turbulent transition effects at the predefined transition locations. In this study, transition terms are omitted and the solutions are performed fully turbulent.

The second term on the left hand side of Equation 25 is the convective flux and discretisized by simple upwinding. The terms on the right hand side are production, dissipation, diffusive (viscous) flux and the non-conservative diffusion. The production, dissipation and non-conservative diffusion terms are the source terms and assumed to be constant in a cell. Similar to the convective flux, diffusive flux is also discretisized by the simple upwind scheme.

One of the widely accepted modifications to the Spalart-Allmaras turbulence model is applied through the magnitude of the vorticity, S, which is a scalar measure of deformation. As given in the Equation 25, S term contains the effect of vorticity which is identical to the strain rate in rotational flows. Therefore, even if the vorticity is zero in regions such as stagnation lines, due to the unphysical production caused by finite mean strain rate, production term can be over predicted. In order to overshoot this problem, the effect of both the vorticity and the strain are included in the definition of S and given in reference [26] such that

$$S \equiv |\Omega_{ij}| + C_{prod} \min\left(0, |\mathrm{S}_{ij}| - |\Omega_{ij}|\right) \qquad \textbf{(Equation 26)}$$

where

$C_{prod} = 2.0$

with the mean strain rate is defined as

$$\mathrm{S}_{ij} = \frac{1}{2}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \qquad \textbf{(Equation 27)}$$

Explicit implementation of the turbulence models is generally problematic and needs special treatment. Large source terms of the turbulence models change the flow variables very rapidly and immediately cause the explicit solution diverge. One of the solutions to this problem is the treatment of the source terms in an implicit manner [27]. Applying Equation 25 to a cell with a finite volume and treating the source term implicitly Equation 28 is obtained.

$$\frac{\Omega_\mathrm{I}}{\Delta t_I} \Delta\tilde{v}_I^n = -\left[\sum_{k=1}^{N_F} (F_c^n - F_v^n)_k \Delta\mathrm{S_k} - \Omega_\mathrm{I}\mathrm{Q_I^{n+1}}\right] \qquad \textbf{(Equation 28)}$$

Since the value of the source term, Q, is not known at time level n+1 it must be approximated by linearization

$$Q_I^{n+1} = Q_I^n + \left(\frac{\partial Q}{\partial \tilde{\upsilon}}\right)_I^n \Delta\tilde{\upsilon}_I^n \qquad \textbf{(Equation 29)}$$

Substituting Equation 29 in Equation 28 and rearranging the terms the following expression is obtained

$$\left[\frac{1}{\Delta t_I} - \left(\frac{\partial Q}{\partial \tilde{\upsilon}}\right)_I^n\right]\Delta\tilde{\upsilon}_I^n = -\frac{1}{\Omega_\mathrm{I}}\left[\sum_{k=1}^{N_F}(F_c^n - F_v^n)_k \Delta \mathrm{S_k} - \Omega_\mathrm{I}\mathrm{Q_I^n}\right] \qquad \textbf{(Equation 30)}$$

$$= -\frac{1}{\Omega_\mathrm{I}} R_I^n$$

As it is expressed in section 2.2.2.2, the formula in Equation 30 is a point-implicit scheme since the implicit operator on the left hand side is only dependent on the values of the cell itself. Different than the flow governing equations, instead of linearizing the complete right hand side of Equation 30, in turbulence model equations only the source terms are linearized.

In his original study, Spalart added the Jacobian term to the left hand side matrix in Equation 30 only if its contribution is positive [25]. Mathematically, this makes the implicit operator diagonally dominant which makes the solution more stable. From a physical point of view, suppose that

$$\frac{1}{(\Delta t_I)_\mathrm{eqv}} = \left[\frac{1}{\Delta t_I} - \left(\frac{\partial Q}{\partial \tilde{\upsilon}}\right)_I^n\right] \qquad \textbf{(Equation 31)}$$

where, $(\Delta t_I)_\mathrm{eqv}$ is the equivalent step size that the variables sense as the solution proceeds. Positive contribution of the Jacobian term increases the total value in the square brackets of Equation 31 which yields a smaller $(\Delta t_I)_{\mathrm{eqv}}$. By this way, as the flow governing equation are integrated with larger time steps, turbulence governing equation are integrated with smaller time steps and the moderate changes in the source terms prevents solution from diverging.

In this study, similar to the case of the flow governing equations, semi-implicit solution algorithm is used in the integration of turbulence governing equations. By this way convergence rate of the turbulent field is increased in parallel with the increase of the time step size that can be used in the solutions.

For external flows, typical inflow boundary condition for the eddy viscosity in Spalart-Allmaras turbulence model is generally taken as the one tenth of the laminar viscosity as long as the transition is modeled. However, in this study since the transition terms of the Spalart-Allmaras turbulence model are omitted and therefore the solutions are performed fully turbulent, the inflow boundary condition for the turbulent viscosity is chosen between 1 and 10 [28]. At solid walls, $\tilde{\upsilon}$ is set to zero. At outflow boundaries eddy viscosity is extrapolated from the interior cells. In addition, for a stable solution, turbulent field is initialized from the inflow boundary condition of the turbulent viscosity at the start of the solution.

2.3.2 SST k-ω Turbulence Model

In addition to the one-equation turbulence model, Spalart-Allmaras, one of the most popular two-equation turbulence models, SST k-ω, is implemented to SENSE2D in its non-dimensional form. The turbulence quantities, k and ω, are non-dimensionalized by a_∞^2 and a_∞^2/ν_∞ respectively. The turbulent viscosity is calculated by Equation 32.

$$\upsilon_t = \min\left(\frac{k}{\omega}, \frac{Re}{M}\frac{a_1 k}{\Omega\, F_2}\right) \qquad \textbf{(Equation 32)}$$

where k and ω are calculated by the solution of the transport equations in the form of Equation 33 and Equation 34.

$$\frac{\partial k}{\partial t} + \nabla \cdot (Vk) = \left(\frac{M}{Re}\right) \upsilon_t \Omega^2 - \frac{Re}{M} \beta' k\omega + \left(\frac{M}{Re}\right) \nabla \cdot \left(\left(\upsilon_L + \frac{\upsilon_T}{\sigma_k}\right) \nabla k\right)$$ **(Equation 33)**

$$\frac{\partial \omega}{\partial t} + \nabla \cdot (V\omega) = \left(\frac{M}{Re}\right) \gamma \Omega^2 - \frac{Re}{M} \beta \omega^2 + \left(\frac{M}{Re}\right) \nabla \cdot \left(\left(\upsilon_L + \frac{\upsilon_T}{\sigma_\omega}\right) \nabla \omega\right) + \left(\frac{M}{Re}\right) 2(1 - \mathrm{F}_1) \sigma_{\omega 2} \frac{1}{\omega} (\nabla k . \nabla \omega)$$ **(Equation 34)**

In both transport equations the second term on the left hand side is the convective term. The terms on the right hand sides of the transport equations are the production, dissipation and the conservative diffusion terms respectively. The last tem of the ω-equation is the cross diffusion term responsible for the switching between the k-ε and the k-ω turbulence models by means of F_1 term.

The constants in the transport equations are calculated from $\Phi = F_1 \Phi_1 + (1 - F_1) \Phi_2$ where $\Phi's$ are the constants. The auxiliary constants in the calculation of the Φ terms are given in Equation 35.

$$\begin{array}{ll} \gamma_1 = 0.5532 & \gamma_2 = 0.4403 \\ \sigma_{k1} = 1.1765 & \sigma_{k2} = 1.0 \\ \sigma_{\omega 1} = 2.0 & \sigma_{\omega 2} = 1.1682 \\ \beta_1 = 0.075 & \beta_2 = 0.0828 \end{array}$$

(Equation 35)

$$\kappa = 0.41, \qquad a_1 = 0.31, \qquad \beta' = C_\mu = 0.09$$

$$F_1 = \tanh(\Gamma^4)$$

$$\Gamma = \min[\max(\Gamma_1, \Gamma_3), \Gamma_2]$$

$$\Gamma_1 = \left(\frac{M}{Re}\right)^2 \frac{500\,\upsilon_L}{\mathrm{d}^2\Omega} \qquad \Gamma_2 = \frac{4\rho\sigma_{\omega 2}k}{\mathrm{d}^2\mathrm{CD}_{\mathrm{k}\omega}} \qquad \Gamma_3 = \left(\frac{M}{Re}\right)\frac{\sqrt{\mathrm{k}}}{C_\mu\,\omega\,d}$$

$$\mathrm{CD}_{\mathrm{k}\omega} = \max\left(\rho\frac{2\sigma_{\omega 2}}{\omega}(\nabla k.\nabla\omega), 1\mathrm{E}-20\right)$$

$$F_2 = \tanh(\Pi)$$

$$\Pi = \max(2\Gamma_3, \Gamma_1)$$

The SST k-ω turbulence model is also implemented in a semi-implicit manner due to the stability considerations explained in sub-section 2.3.1. The method used in the implementation of the SST k-ω turbulence model is almost the same with the Spalart-Allmaras' and therefore is not repeated here. The only difference comes from the coupling of the k and the ω equations which is not applicable for the one-equation Spalart-Allmaras turbulence model. The resulting 2 by 2 system is solved simultaneously by Kramer's rule.

The no-slip wall boundary condition for the turbulent kinetic energy is equal to zero since the mean and the fluctuating components of the velocity vector vanish at the solid walls. The wall boundary condition for ω is given in Equation 36.

$$\omega_{wall} = \frac{60\,\upsilon_L}{\beta_1 d^2}$$

(Equation 36)

The freestream values of the k and ω are taken from the reference [29] and applied to the inflow boundaries.

$$k_{\infty} = 9\,x\,10^{-9}$$
$$\omega_{\infty} = 1\,x\,10^{-6}$$ **(Equation 37)**

At the outflow boundaries k and ω are extrapolated from the interior cells. In addition, for a stable solution, turbulent field is initialized from the inflow boundary conditions of k and ω at the start of the solution.

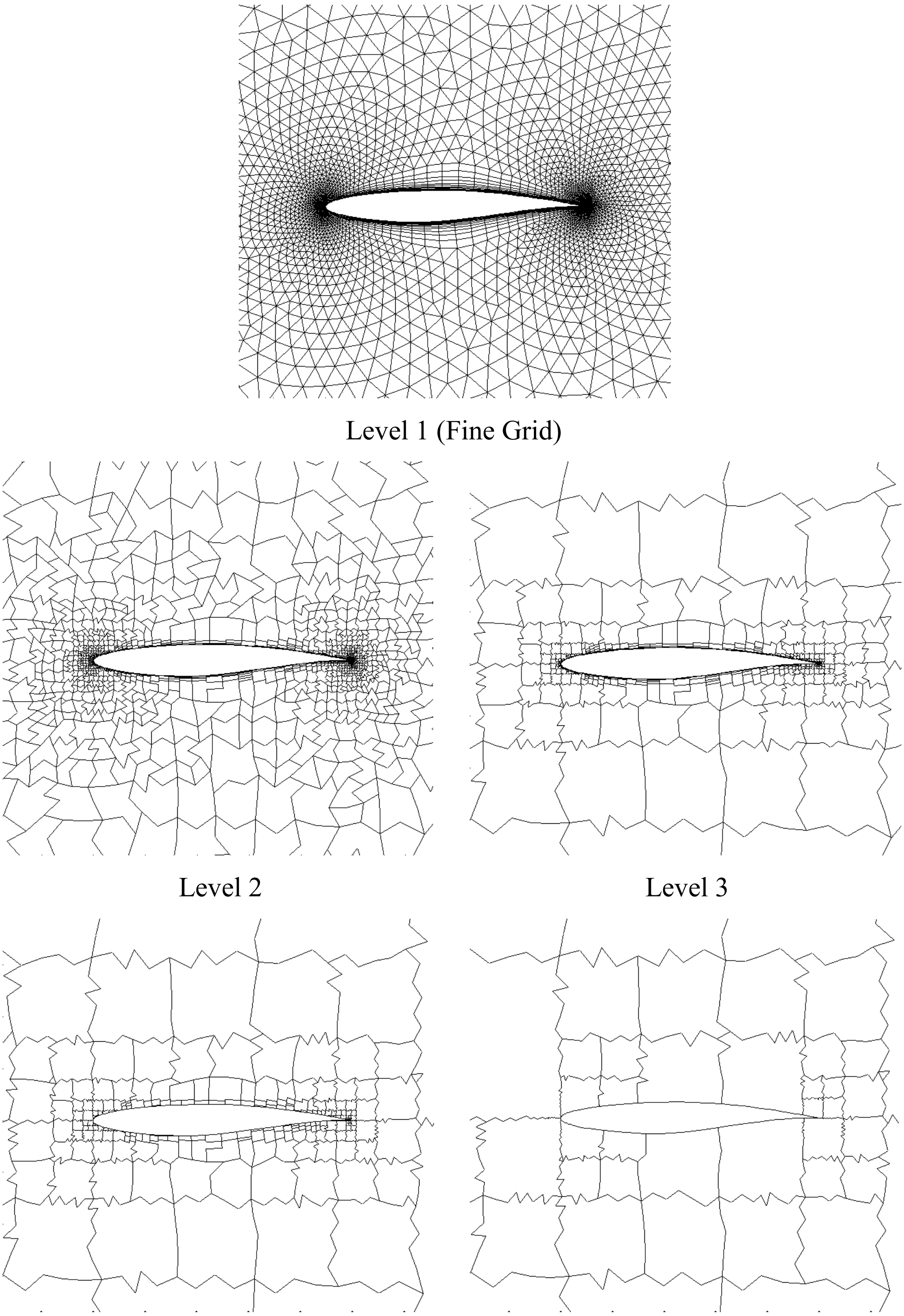
2.4 Multigrid Methodology

A basic geometric multigrid application consists of three main steps:

- Generation of the coarse cells
- Transfer of the fine level solutions to the coarse levels and obtaining solution in the coarse levels
- Transfer of the coarse level solutions to the fine level and obtaining solution in the fine level

The first step in the application of the multigrid methods is the generation of the coarse level grids. One of the most effective coarsening methods is so called the *agglomeration technique* and introduced in references [30] and [31] in detail. "The New Grid Coarsening Method Based on Quadtree / Octree Data Structure" is used in the MG applications of this study. The grid coarsening with agglomeration method is illustrated in Figure 8. The details of the method are given in reference [32].

The next step after the grid coarsening in multigrid solutions is the transfer of the fine level solutions to the coarser levels which is named as "*restriction*". In restriction step, both the flow variables and the residual terms are transferred to the coarse level.



Level 1 (Fine Grid)

Level 2

Level 3

Level 4

Level 5

Figure 8 Grid Coarsening by the Agglomeration Method

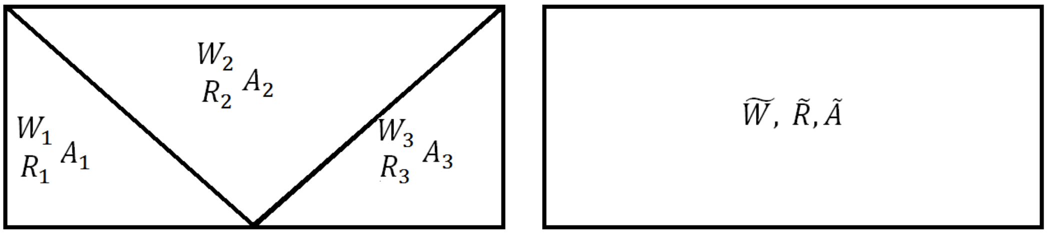


Figure 9 Transfer of Flow Variables and Residuals from Fine to Coarse Levels

In Figure 9, transfer of the flow variables from the fine to the coarse level is illustrated. The value of the flow variable at the center of the coarse grid is simply calculated by the area weighted averaging of the fine level values, that is:

$$\widetilde{W} = \frac{W_1A_1 + W_2A_2 + \ W_3A_3}{\tilde{A} = A_1 + \ A_2 + \ A_3} \qquad \textbf{(Equation 38)}$$

Different than the flow variables, residuals are transferred from the fine to the coarse level by direct summation, that is:

$$\tilde{R} = \ R_1 + \ R_2 + \ R_3 \qquad \textbf{(Equation 39)}$$

In the coarse level grids, before integrating in time a corrective step must be applied to the flow variables, $\widetilde{W}$, due to the error introduced by the area weighted averaging during the variable transfer from the fine to the coarse level grid.

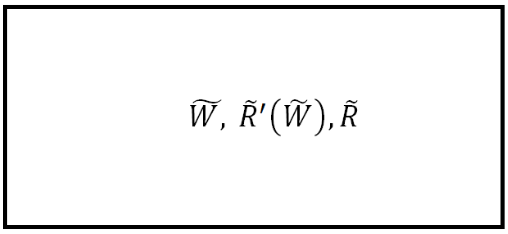


Figure 10 Two Different Residuals in a Coarse Grid

In Figure 10, a coarse grid is illustrated just after the variable and residual transfer from the fine level. $\tilde{R}$ is the residual transferred from the fine level and calculated by Equation 39. Besides that, $\tilde{R}'(\widetilde{W})$ is the residual directly calculated in the coarse level by the use of transferred variables, $\widetilde{W}$. If the value of the transferred variable is able to be predicted exactly, then $\tilde{R}$ and $\tilde{R}'(\widetilde{W})$ are expected to be the same. However, averaging of the flow variables always introduces some error proportional to the difference between $\tilde{R}$ and $\tilde{R}'(\widetilde{W})$ and therefore transferred variable needs correction in the form of Equation 40.

$$\frac{(\widetilde{W}_c - \widetilde{W})}{\Delta t} = \tilde{R}'(\widetilde{W}) - \tilde{R} \qquad \textbf{(Equation 40)}$$

In Equation 40, $\widetilde{W}_c$ is the corrected variable in the coarse level after transferred from the fine level. The right hand side of Equation 40 is named as the *forcing function* which is the difference between the transferred residual from the fine level grid and the calculated residual in the coarse level grid. After the correction of the working variables in the coarse level grid, one of the temporal discretisation techniques is applied so as to update the variables. This type of multigrid applications is called *the Full Approximation Storage (FAS)* method. FAS methods are quite effective in the solution of the non-linear equations since the nonlinearities in the system are directly carried to the coarse level grids by restriction step.

Instead of applying the correction and update steps separately, both calculations can be performed in a single step. Assuming a simple euler integration in the coarse level grid, the resulting relation is in the form of Equation 41.

$$\frac{(\widetilde{W}^{n+1} - \widetilde{W}^{n})}{\Delta t} = \left[\tilde{R}'(\widetilde{W}) - \tilde{R}\right] + \tilde{R}'(\widetilde{W}) = -\tilde{R} \qquad \textbf{(Equation 41)}$$

In Equation 41, it is seen that the first iteration in the coarse level grid is directly dependent on the residual transferred from the fine level grid. As a result, the accuracy of the fine level is retained in the coarse level grid calculations.

After performing several time-integrations in the coarse level grids, the solution of the coarse level grids are transferred back to the fine level grids by various interpolation techniques. This process is named as the *prolongation* which can be seen as the inverse of the restriction.

The multigrid solution is obtained after the successive application of the restriction and the prolongation steps until a converged solution is obtained.

One of the problems encountered in the application of the multigrid methods is the implementation of the turbulence models. Today most of the turbulence models use the wall distance as the key parameter in the calculation of the production and the dissipation terms of the turbulence quantities. Moreover, in order to have a stable solution with accurate results, the boundary layer has to be resolved sufficiently. Although the usage of the wall functions decreases the need for the fine meshes in the near wall regions, the wall distance of the large, agglomerated cells used in the multigrid methods is still beyond the applicability limit of the wall functions. Therefore, it is hard to employ turbulence models in the coarse grid levels.

One of the solutions introduced to this problem is employing the turbulence models in the finest level grid only. Then only the flow variables are worked during the multigrid processes. By doing this, the turbulence quantities are included in the multigrid calculations by means of the flow variables. However, it is still required that the grid is capable of resolving the turbulent boundary layer in the finest level.

A similar approach is followed in the implementation of the semi-implicit solution algorithm to the code. The flow and the turbulence variables are solved implicitly in the finest grid level only. Then the variables and the residuals are carried to the coarse levels by means of the flow variables. A schematic of the process is presented in Figure 11.

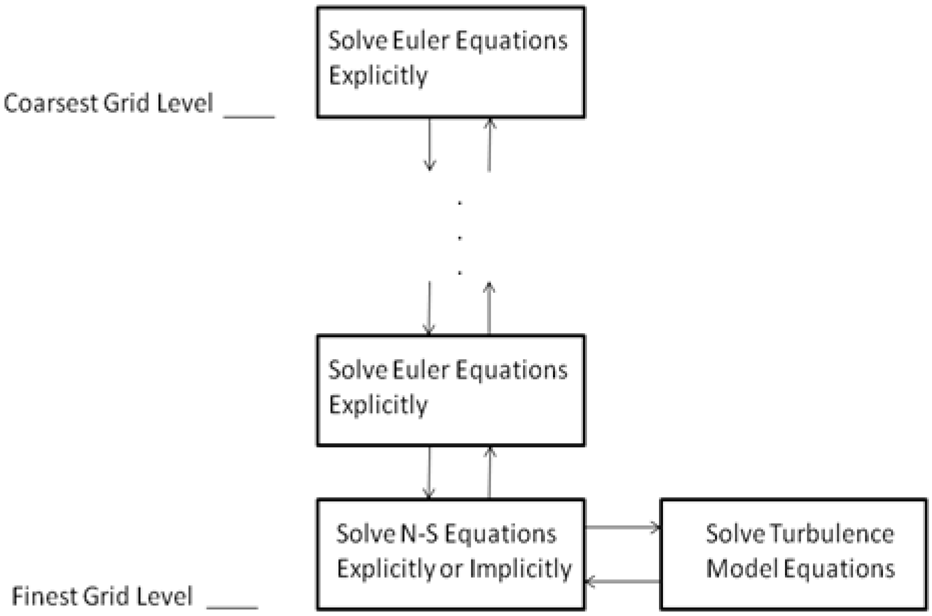

Solve Euler Equations Explicitly
Coarsest Grid Level
Solve Euler Equations Explicitly
Solve N-S Equations Explicitly or Implicitly
Solve Turbulence Model Equations
Finest Grid Level

Figure 11 Multigrid Solutions-A Simple Flow Chart

Since the time advancement methods may differ between the finest and the coarse grid levels, a simple correction must be applied before transferring the residual terms from the finest to the coarse grid level in implicit solutions. Employing implicit time integration in the finest grid level, the solution ends up with a residual term in the next time level, R^{n+1}. However, in the coarse grid levels only the explicit Runge-Kutta time stepping is performed. Therefore, direct transfer of the implicit residuals from the finest to the coarse level grid results in an explicit scheme with an implicit residual term on the right hand side. However, this is inconsistent with the Runge-Kutta approach. In order to overcome this discrepancy, three stage Runge-Kutta time integration is performed one time before each restriction step in the finest level grid.

CHAPTER 3

RESULTS AND DISCUSSION

In this chapter, the numerical results obtained by SENSE2D are presented and discussed. For this purpose, turbulent flow solutions are obtained over a flat plate, RAE2822 airfoil and NLR7301 multi-element airfoil. The studies performed can be summarized in three steps.

Firstly, implemented semi-implicit algorithm is verified by the comparison of the turbulent velocity profiles over a flat plate obtained separately by the explicit and the semi-implicit time advancement methods. Moreover, the improvement in the convergence rate of the turbulent flow solutions on a single grid is investigated in the same case with the comparison of the convergence times of the explicit and the semi-implicit solutions.

Secondly, in multigrid solutions, a study is performed in order to find the optimum number of iterations in the fine and the coarse level grids for the fastest convergence rate. For this purpose a solution matrix with different number of iterations in the fine and the coarse level grids is constructed and solved using the explicit and the implicit multigrid algorithms separately. At the end of this study, the performance of the semi-implicit algorithm in the multigrid applications is investigated by comparing the fastest convergence rates obtained by the explicit and the implicit multigrid solutions.

Finally, for the validation of the implemented Spalart-Allmaras and SST k-ω turbulence models aerodynamic coefficients such as C_l, C_d, C_p, and the velocity profiles are compared with the commercial CFD code FLUENT results and the experimental data as long as they are available. Also, various contour plots are presented where required.

3.1 Turbulent Flow Solution over a Flat Plate

In this case, two dimensional turbulent flow solutions over a flat plate are studied on single grid. The free stream conditions for the case are given in Table 3.

Table 3 Free Stream Conditions - Flat Plate Solution

Mach Number	Re Number	Angle of Attack
0.2	12E+6	0^0

For the turbulent flow solutions over a flat plate, a rectangular solution domain is discretized with a hybrid grid (Figure 12). The flat plate is placed at the middle of the bottom edge which is defined as a no-slip wall. The rest of the bottom edge is defined as the symmetry plane. The outer edges are treated as the inflow-outflow boundary depending on the direction of the flow. Grid and boundary types are shown in Figure 12. The computational grid consists of 8181 nodes and 9320 cells. The distance from the first layer to the nearest wall is a 2E-6 unit.

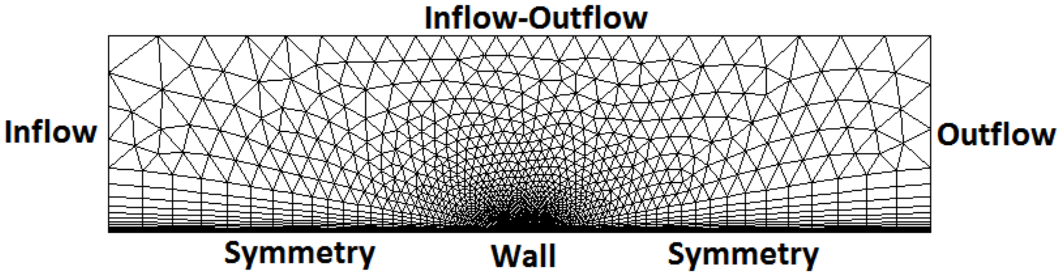


Figure 12 Solution Domain – Flat Plate

As an initial assessment, turbulent flow solutions over a flat plate are performed by explicit and semi-implicit methods using Spalart-Allmaras turbulence model. The CFL number in the explicit solution algorithm is limited by 0.95. In semi-implicit solution, the CFL number is increased up to 8 maintaining the stability. L2-norm of the flow residuals and the turbulent viscosity residual histories of the explicit and the semi-implicit methods are shown in Figure 13.

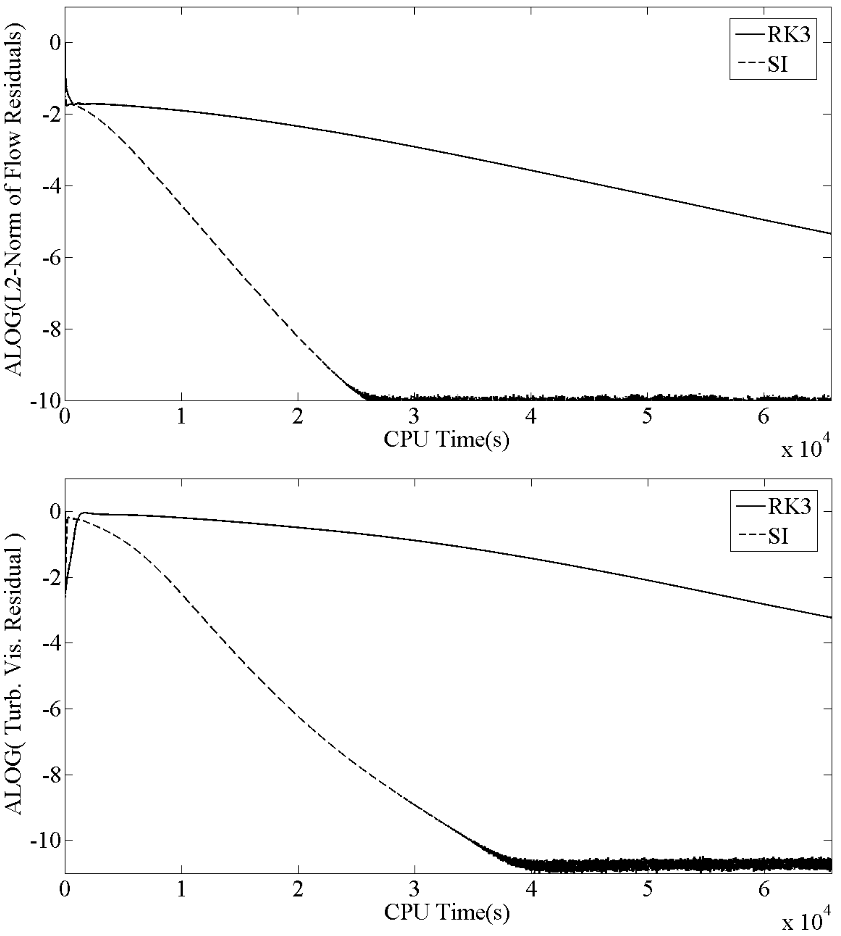


Figure 13 Residual History (Flat Plate, SA)

For a five order drop in the L2-norm of the flow residuals, with respect to the CPU time, it takes explicit solution approximately six times longer in comparison to the semi-implicit solution. This ratio grows as the residuals drop further. Similarly, a considerable acceleration in the convergence of the turbulent variable is seen in Figure 13.

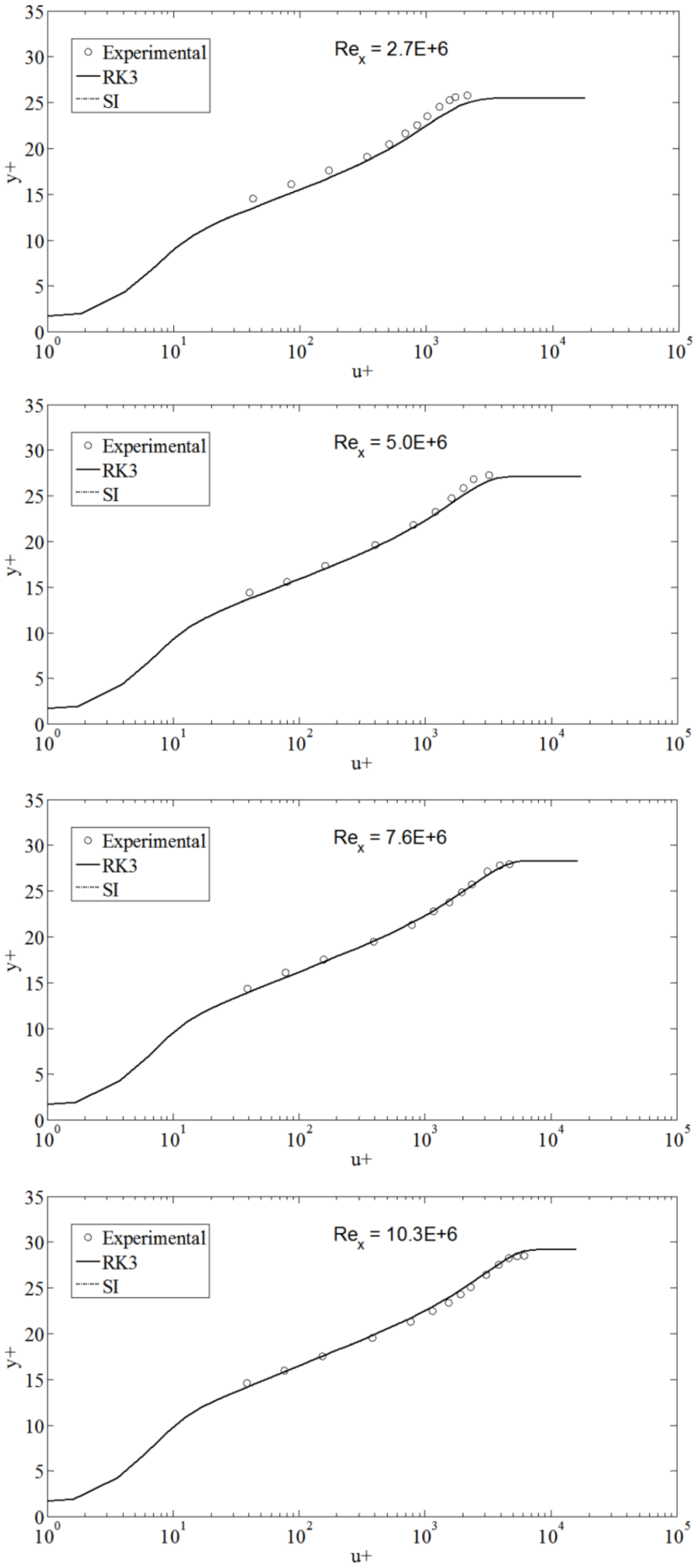


Figure 14 Turbulent Velocity Profiles (Flat Plate, SA)

In Figure 14, turbulent velocity profiles obtained numerically by Spalart-Allmaras turbulence model are compared with the experimental ones by Patel, Rodi, and Scheuerer [33] at four different sections. In all sections explicit and semi-implicit solutions are in exact agreement with each other and similar to the experimental data.

In the numerical solution of the turbulent flow over a flat plate, SST k-ω turbulence model is also performed with semi-implicit time integration. The computational domain and the free stream conditions are the same with the Spalart-Allmaras solutions. In Figure 15, convergence history of the turbulent flow solution over a flat plate is given in terms of the flow and turbulence quantity residuals.

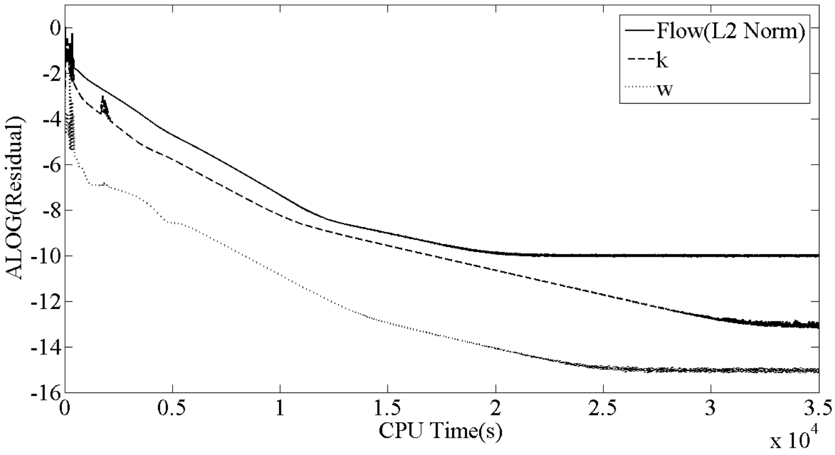


Figure 15 Residual History (Flat Plate, SST k-ω)

From Figure 15, it is seen that the residual drop of the flow and the turbulence quantities are sufficient to be converged. Comparing the L2-norms of the flow residuals from Figure 13 and Figure 15, it can be concluded that the convergence rates of the Spalart-Allmaras and the SST k-ω turbulence models are identical to each other.

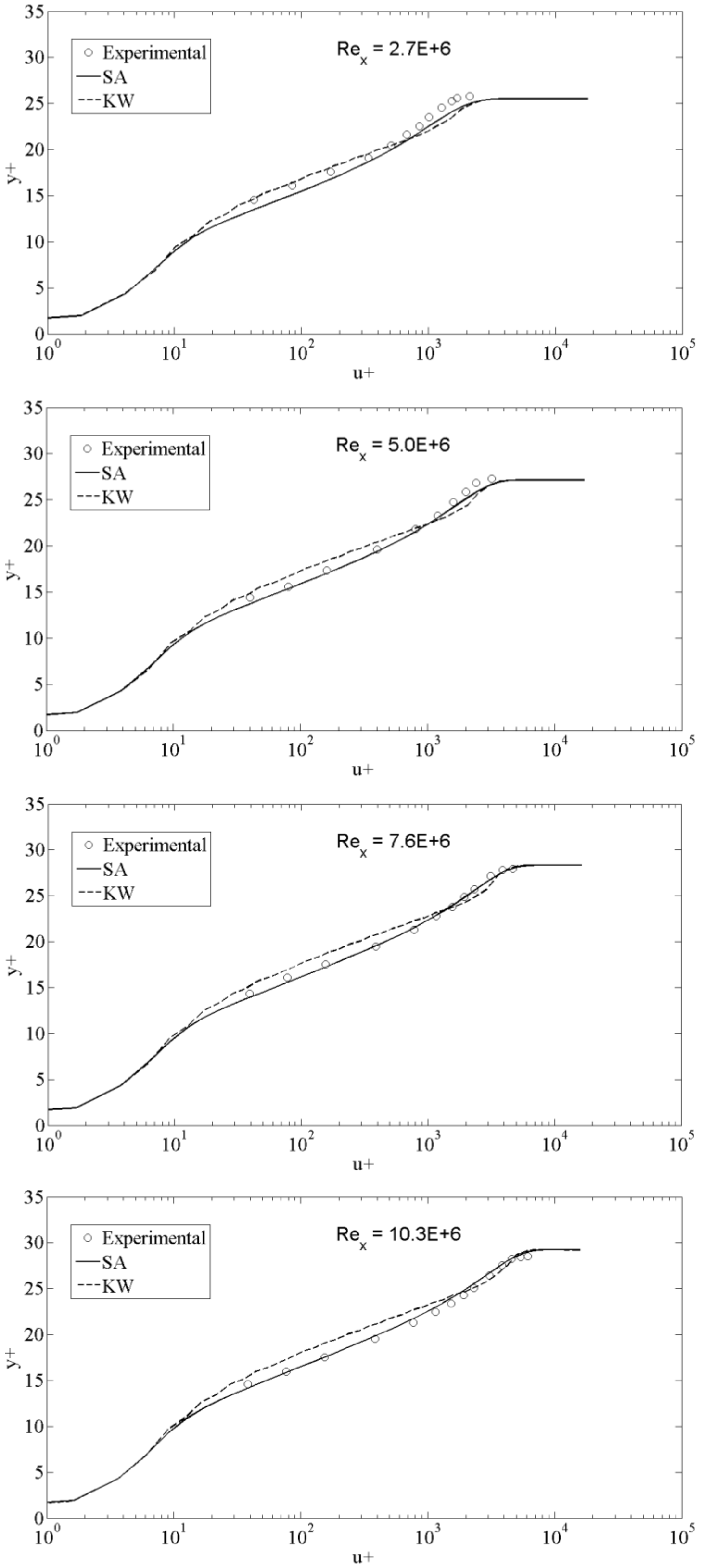


Figure 16 Turbulent Velocity Profiles (Flat Plate, SST k-ω)

In Figure 16, the non-dimensional turbulent velocity profiles over a flat plate, obtained by SST k-ω turbulence model, are compared with the numerical and the experimental data given in Figure 14. In the viscous sublayer, the profiles obtained by the Spalart-Allmaras and the SST k-ω turbulence models are exactly the same. However, in the fully turbulent logarithmic region, SST k-ω turbulence model slightly underpredicts the velocities.

3.2 Turbulent Flow Solution over RAE2822 Airfoil

In this case, two dimensional, turbulent, high subsonic flow over RAE2822 airfoil case is studied with multigrid application. The free stream conditions for the case are presented in Table 4.

Table 4 Free Stream Conditions- RAE2822 Solution

Mach Number	Re Number	Angle of Attack
0.6	6.3E+6	2.57^0

For the solution of the turbulent flow over RAE2822 airfoil, O-type solution domain with hybrid meshes is created in GAMBIT 2.4.6 version. The chord length of the RAE2822 airfoil is taken as 1 unit while the smaller and larger radii of the elliptical domain are taken as 10 and 20 units. Solution domain consists of 8746 nodes and 12871 face elements. The distance from the first layer of the boundary layer to the nearest wall is 5E-6 units.

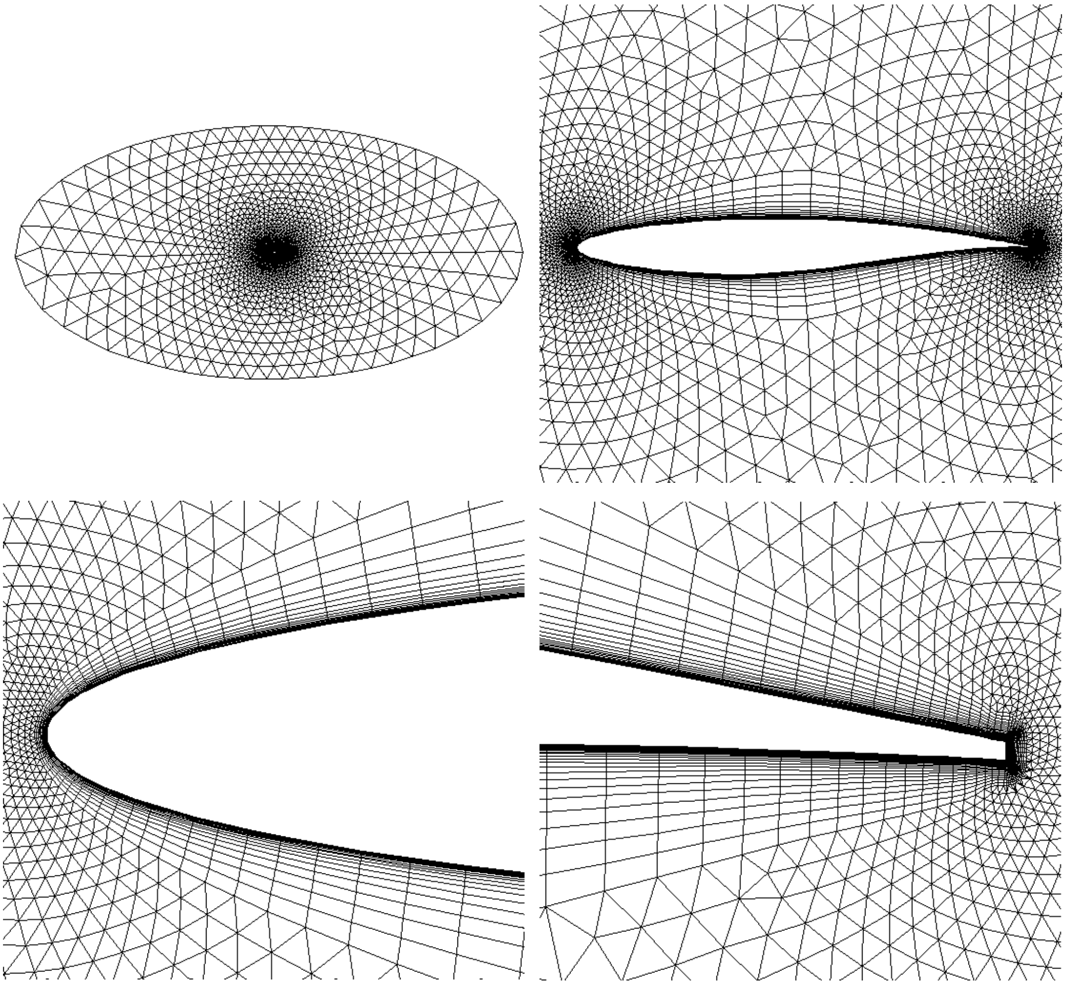


Figure 17 Solution Domain – RAE2822 Airfoil

In the current study the CFL number on the coarse level grids is kept at 0.95. In explicit solutions the maximum CFL number at the fine grid level is 0.95. Whereas in the semi-implicit solution the maximum CFL number with a stable and convergent solution is limited at 3.5.

Before comparing the convergence rates of the explicit and the implicit MG solution algorithms, a set of solutions are performed in order to optimize the number of iterations at the fine and the coarse grid levels for the best convergence rate. For this purpose, a five order drop in the L2-Norm of the flow residuals is taken as the convergence criterion which mostly provides converged aerodynamic loads. The solution matrix and the related CPU times in seconds for the convergence are given in Table 5 and Table 6 respectively for the explicit and the implicit multigrid solutions. The F# and C# terms are used in place of the number of iterations at the fine and the coarse level grids while X's and O's stand for the divergent and non-convergent solutions respectively. The convergence rate studies are performed using the Spalart-Allmaras turbulence model.

Table 5 Convergence times (RAE2822, explicit-MG, SA)

	C1	C3	C5	C7	C10	C20
F1	O	X	X	X	X	X
F3	**1490**	O	O	X	X	X
F5	**1344**	O	O	O	X	X
F7	**1285**	O	O	O	X	X
F10	**1148**	1290	O	O	X	X
F20	**1175**	1183	O	O	O	X

Table 6 Convergence times (RAE2822, implicit-MG, SA)

	C1	C3	C5	C7	C10	C20
F1	**855**	O	X	X	X	X
F3	**605**	O	O	X	X	X
F5	**546**	639	O	X	X	X
F7	**518**	584	O	O	X	X
F10	**475**	519	569	O	X	X
F20	513	**479**	490	512	546	X

From Table 5 and Table 6, it is seen that the fastest convergence for both the explicit-MG and implicit-MG methods is obtained by iterating 10 times in the fine and 1 time in the coarse grid levels. Optimum input parameters for each method is summarized in Table 7.

Table 7 Optimized Parameters List- RAE2822 Airfoil Solutions

	CFL (Coarse)	CFL (Fine)	# of Iterations (Coarse)	# of Iterations (Fine)
Explicit	-	0.95	-	-
Explicit-MG	0.95	0.95	1	10
Implicit	-	3.5	-	-
Implicit-MG	0.95	3.5	1	10

The complete residual histories of the fastest solutions obtained by the explicit and the implicit multigrid algorithms are presented in Figure 18.

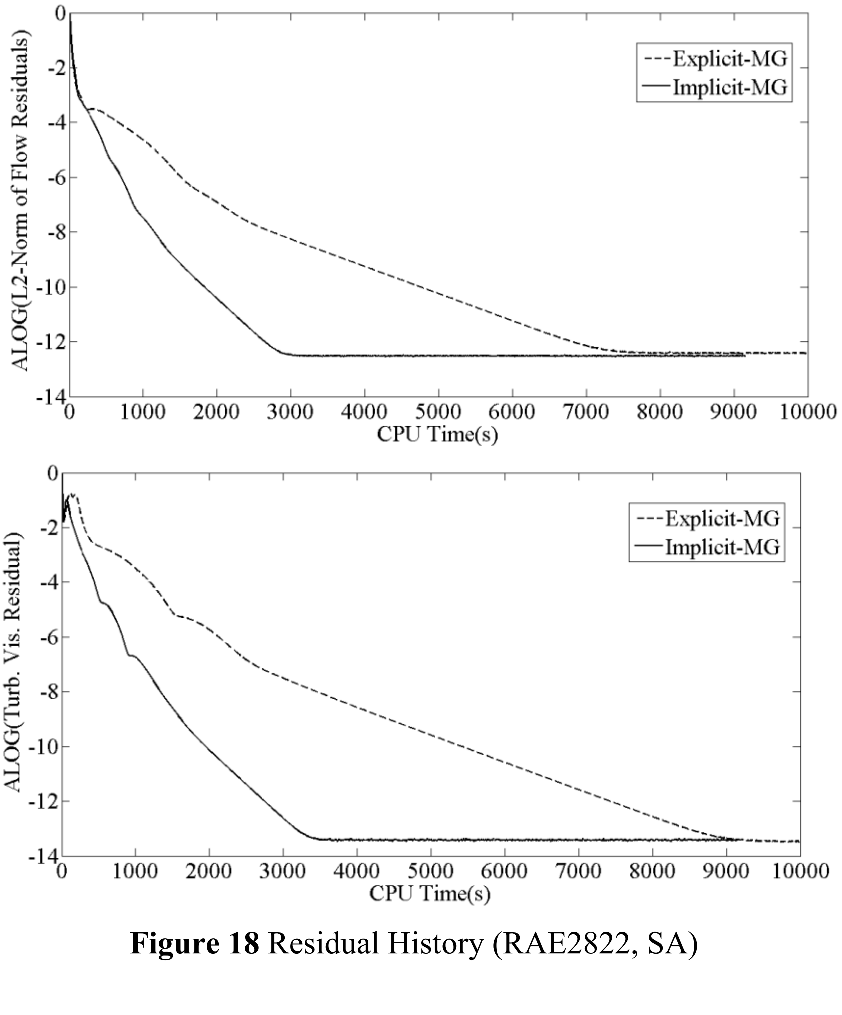


Figure 18 Residual History (RAE2822, SA)

For a five order drop in the L2-Norm of the flow residuals, implicit-MG method is found to be 2.4 times faster than the explicit-MG method.

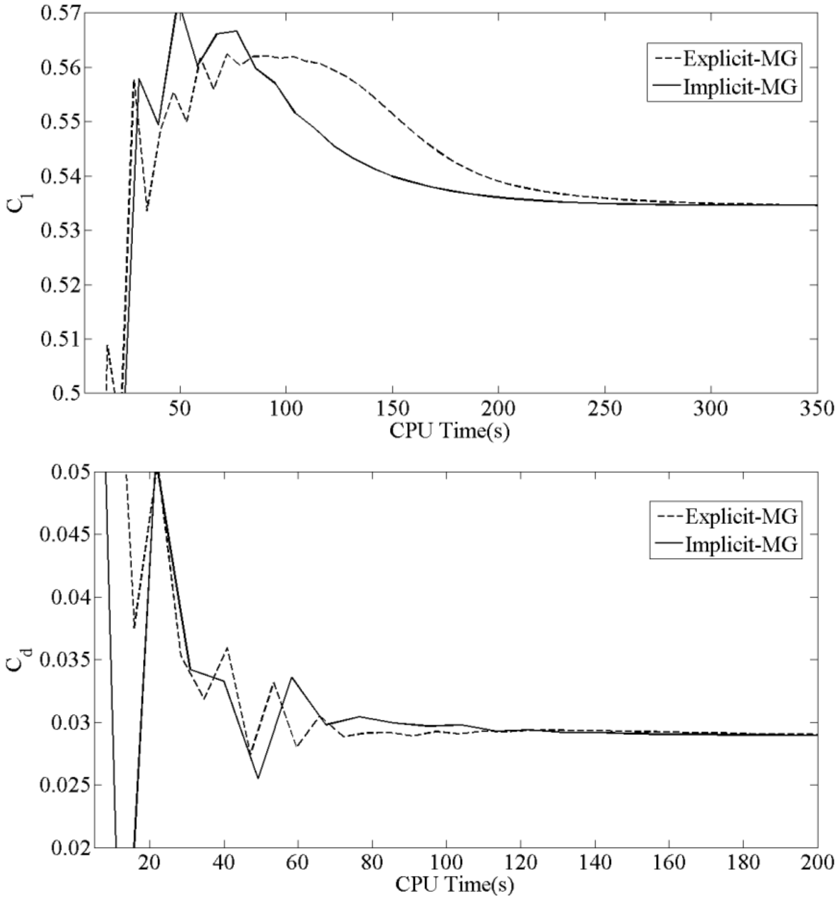
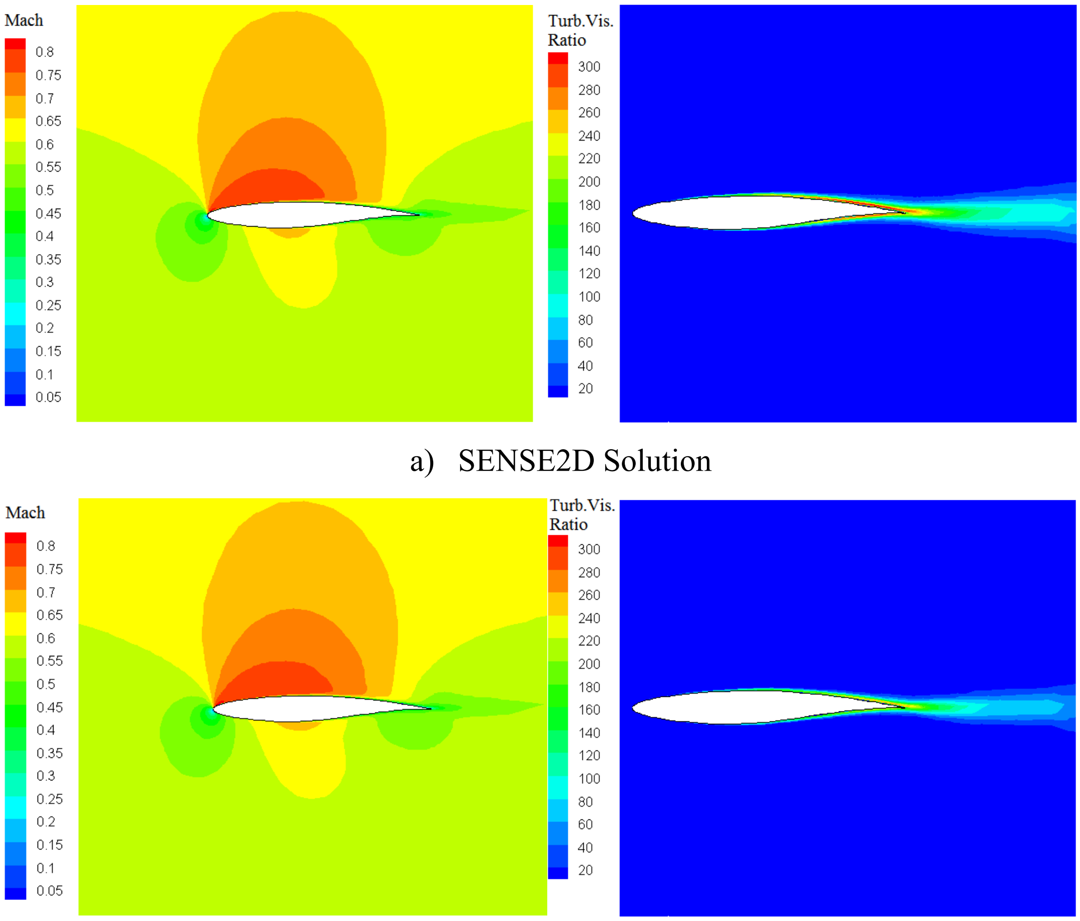


Figure 19 Loads History (RAE2822, SA)

In Figure 19, the lift and the drag coefficients' convergence histories are presented. Similar to the residual case, implicit-MG method is observed to be 1.5-2 times faster than the explicit-MG method in lift convergence. The drag coefficient convergence rates of the implicit-MG and explicit-MG methods are similar. Being the fastest algorithm, results of the implicit-MG solutions will be presented from this point on. The solutions obtained by the SA turbulence model will be presented first. After, the results of the SST k-ω turbulence model will be presented in a similar manner.



a) SENSE2D Solution

b) FLUENT Solution

Figure 20 Mach Number & Turbulent Viscosity Ratio Fields (RAE2822, SA)

The Mach number and the turbulent viscosity ratio fields of the turbulent flow over RAE2822 airfoil obtained by SENSE2D are compared with the same fields by FLUENT in Figure 20. The views of the fields from both solutions are similar to each other both qualitatively and quantitatively.

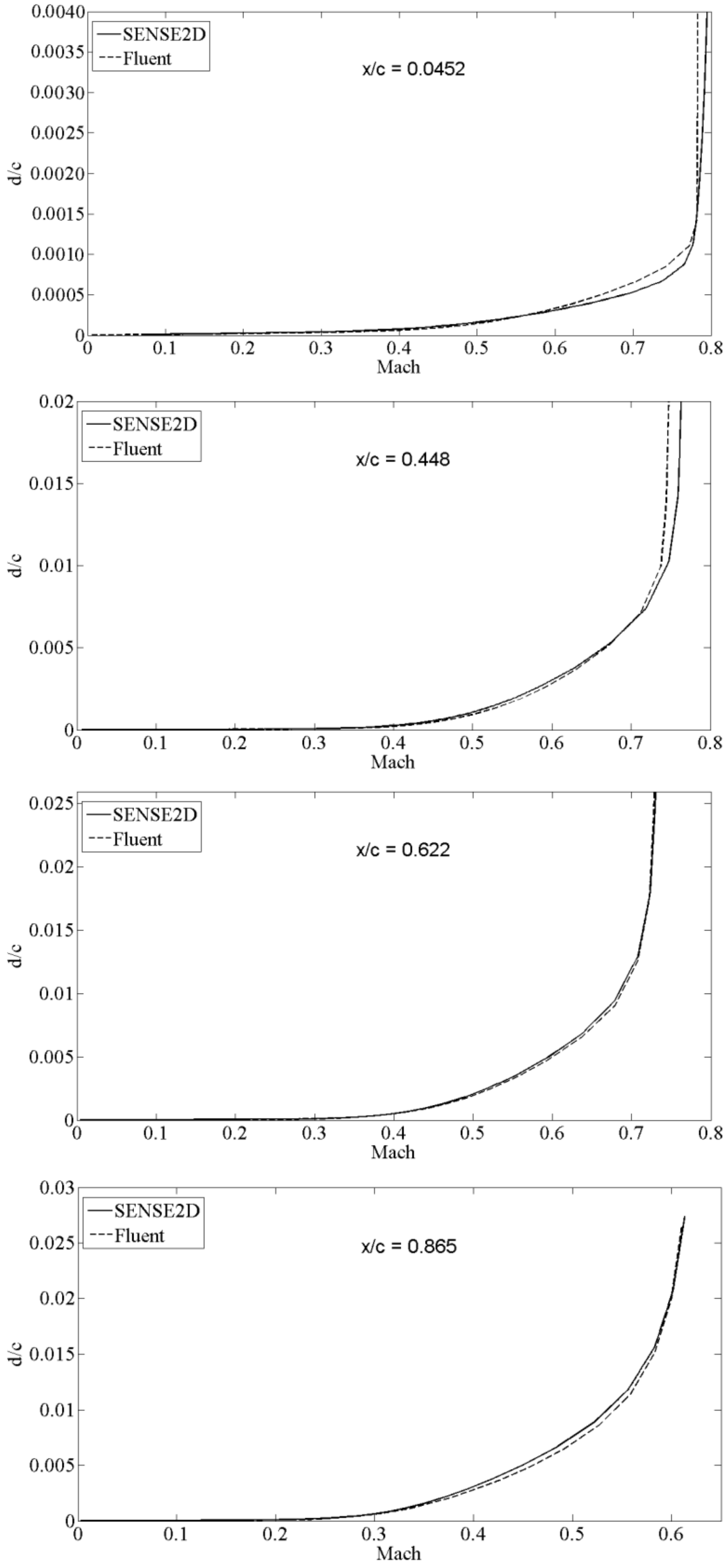


Figure 21 Turbulent Velocity Profiles (RAE2822, SA)

In Figure 21, velocity profiles at four different sections of the upper surface of the RAE2822 airfoil are given and compared with the FLUENT solutions. All the profiles are in good agreement.

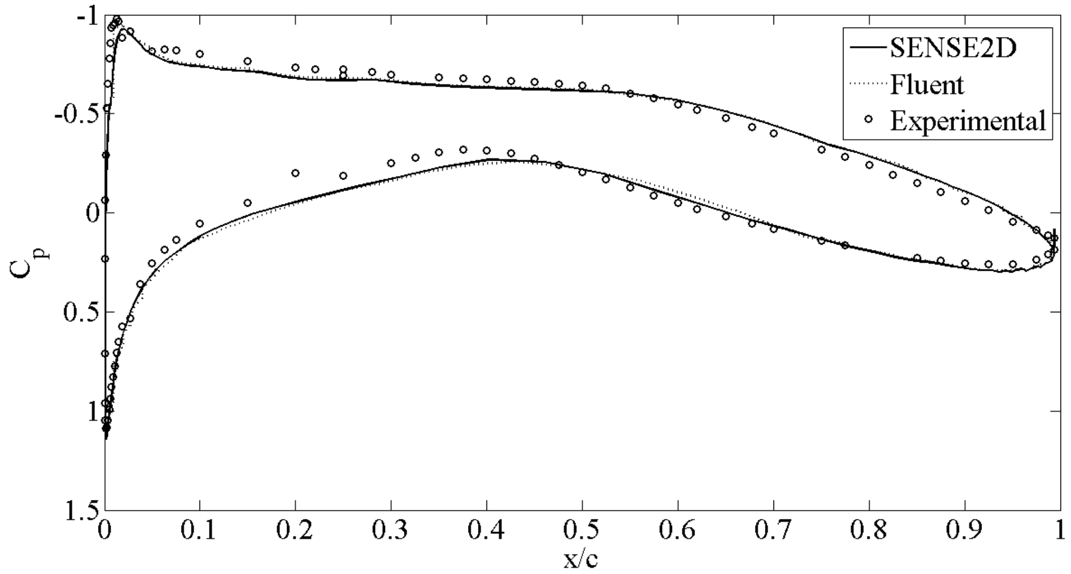
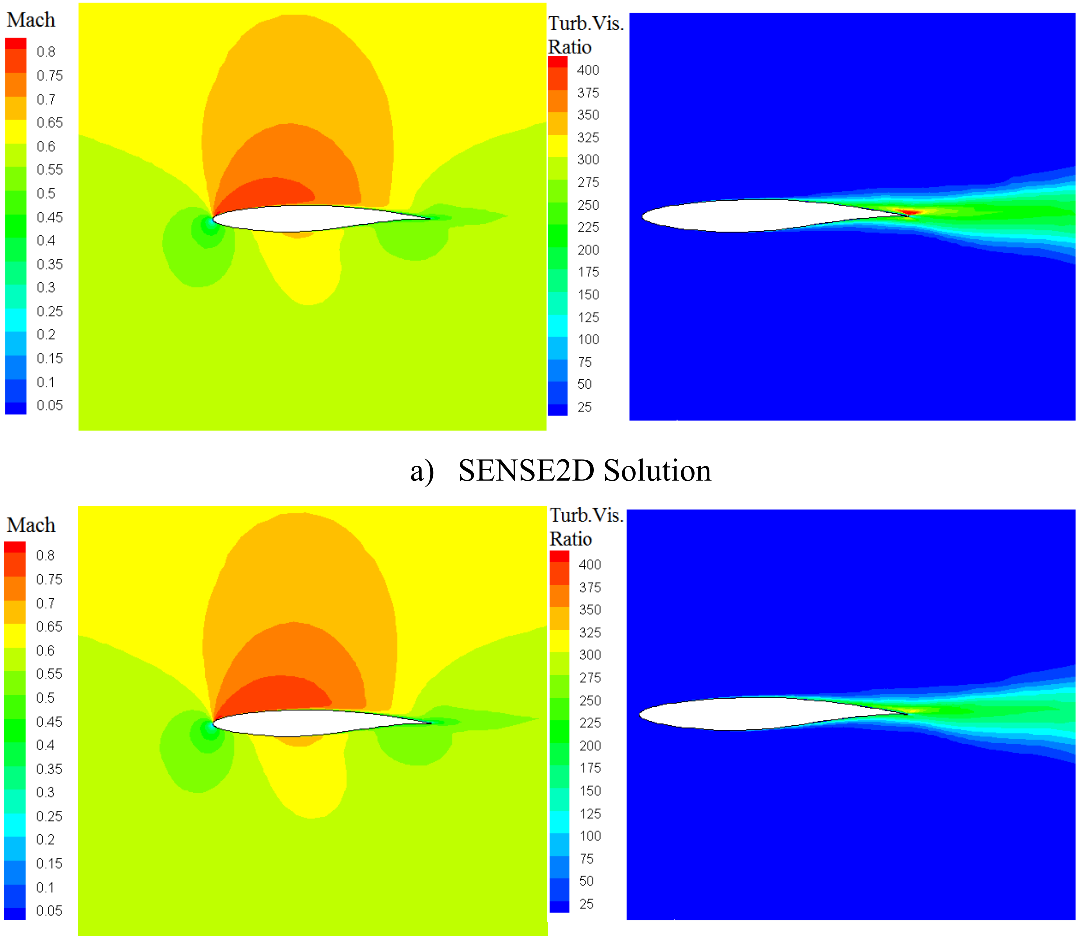


Figure 22 Pressure Coefficient Distribution (RAE2822, SA)

The pressure coefficient distribution over the RAE 2822 airfoil is given in Figure 22. In parallel to the good agreement between the boundary layer velocity profiles, the pressure distributions predicted by FLUENT and SENSE2D agree quite well.

Except the convergence rate study, all the solutions performed with the SA turbulence model are repeated with the SST k-ω turbulence model for validation purposes.



a) SENSE2D Solution

b) FLUENT Solution

Figure 23 Mach Number & Turbulent Viscosity Ratio Fields(RAE2822,SST k-ω)

The Mach number and the turbulent viscosity ratio fields of SENSE2D and FLUENT solutions are presented in Figure 23. Mach contours of both solvers are in good agreement both qualitatively and quantitatively. However, although the fields of turbulent viscosity ratio look similar, SENSE2D overpredicts the field in some regions when compared with FLUENT results.

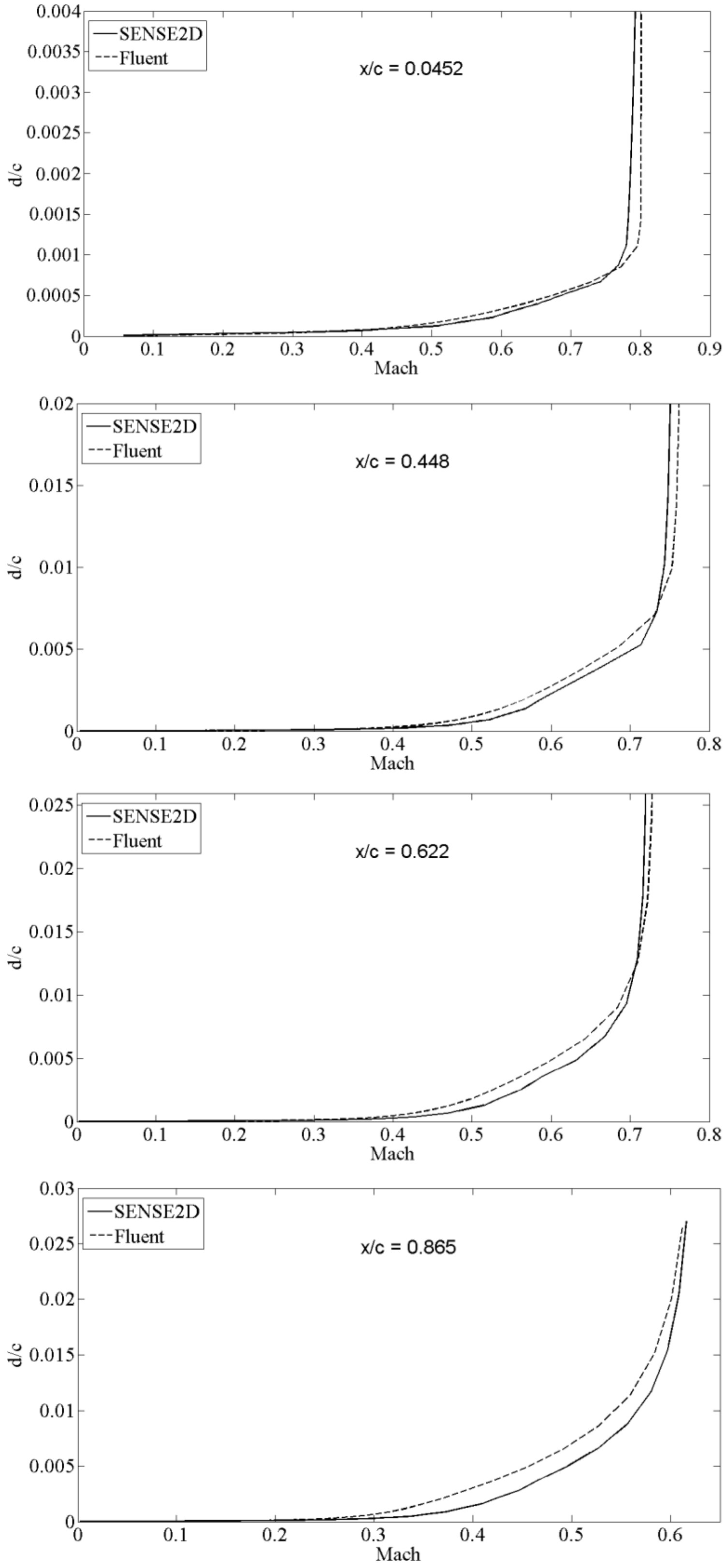


Figure 24 Turbulent Velocity Profiles (RAE2822, SST k-ω)

In Figure 24, turbulent velocity profiles obtained by SENSE2D are compared to the FLUENT results. Although the velocity profiles from each solver are almost on top of each other, it observed that the turbulent velocity profiles obtained by the SST k-ω turbulence model are not as smooth as the ones obtained by SA turbulence model.

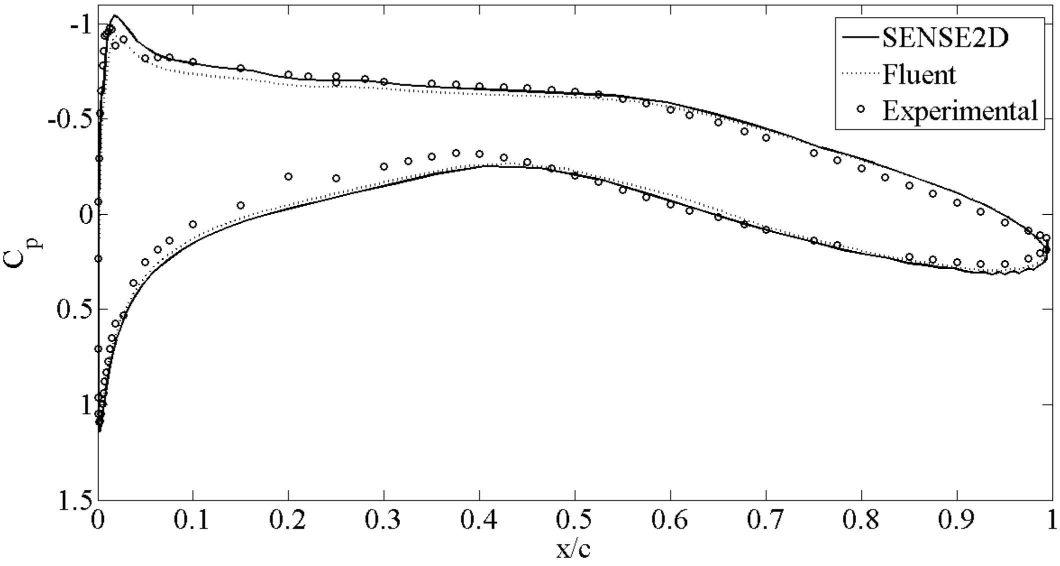


Figure 25 Pressure Coefficient Distribution (RAE2822, SST k-ω)

The pressure coefficient distribution obtained by the SST k-ω model is given in Figure 25 and compared with FLUENT results and experimental data. All three data sets are quite similar to each other. Moreover, in the suction region of the airfoil, SENSE2D is in better agreement with the experimental data than FLUENT is.

Table 8 Aerodynamic Coefficients (RAE2822, SA, SST k-ω)

	Spalart-Allmaras		SST k-ω	
Coefficients	SENSE2D	FLUENT	SENSE2D	FLUENT
C_l	0.535	0.547	0.529	0.538
C_d	0.0289	0.0290	0.0284	0.0286

Aerodynamic load coefficients of the RAE2822 airfoil are given in Table 8. The values obtained from the SENSE2D and the FLUENT solvers are almost the same as a result of the accurately predicted velocity profiles and the pressure distribution.

The pointed, sharp trailing edges of the airfoils can cause highly skew meshes in wedge type boundary layers which are also used in this study. Therefore, instead of using the original airfoil geometries, a small portion of the sharp trailing edges of the airfoils are cut resulting in a bump trailing edge without affecting the overall results (Figure 17). As a result, high quality meshes with small skewness values are obtain in these problematic regions. However, in Figure 23, it is seen that the turbulent viscosity ratio values in the trailing edge region are overpredicted by SENSE2D when compared to the results of FLUENT using SST k-ω turbulence model. Although the same meshes are used in the SENSE2D and the FLUENT solutions, in order to be sure that the difference is not arising from the bump trailing edges, the same case is repeated with original RAE2822 airfoil geometry with a pointed, sharp trailing edge. Moreover, a fully structured mesh is used in the solutions so as to avoid skew meshes on the trailing edge. The solution domain is given in Figure 26.

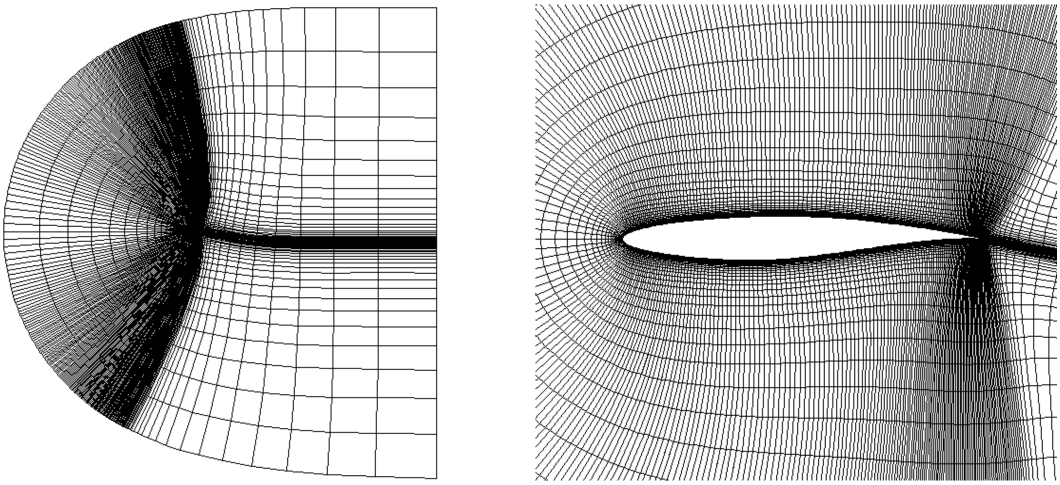


Figure 26 Solution Domain – RAE2822 Airfoil (Sharp Trailing Edge)

The flow conditions for this case are the same with the previous RAE2822 solutions and given in Table 4. In order to investigate the effect of the trailing edge cut, turbulent viscosity ratio fields from two solutions are compared to each other.

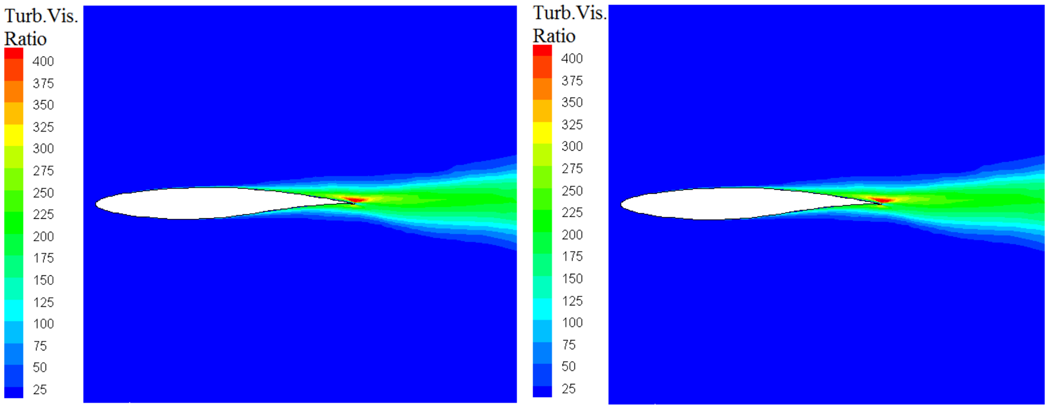


Figure 27 Comparison of Sharp and Bump Trailing Edges

In Figure 27, two solutions obtained by SENSE2D solver with SST k-ω turbulence model are compared to each other. The left hand side field belongs to the results of the flow solutions over RAE2822 airfoil with bump trailing edge. On the other hand, the right hand side field belongs to the results of the flow solutions over the same airfoil with pointed and sharp trailing edge. As it is clear from the figures that the cut of the sharp trailing edges of the airfoils due to the meshing considerations does not affect the overall result at all as long as the size of the cut is small when compared with the overall chord of the airfoil. As a result, the overprediction of the turbulent viscosity ratio field is not originating from this treatment.

3.3 Turbulent Flow Solution over NLR7301 Two Element Airfoil

In this subsection, two dimensional turbulent flow over NLR7301 two element airfoil case is studied with multigrid application. The free stream conditions for the case are presented in Table 9.

Table 9 Free Stream Conditions- NLR7301 Solution

Mach Number	Re Number	Angle of Attack
0.185	2.51E+6	6^0

For the solution of the turbulent flow over NLR7301 airfoil, O-type solution domain with hybrid meshes is created in GAMBIT 2.4.6 version. The chord length of the airfoil is taken as 0.95 units for the main part and 1.2 for the whole airfoil while the smaller and larger radii of the elliptical domain are taken as 10 and 20 units. Solution domain consists of 17415 nodes and 20213 face elements. The distance from the first layer of the boundary layer to the nearest wall is 4E-6 units.

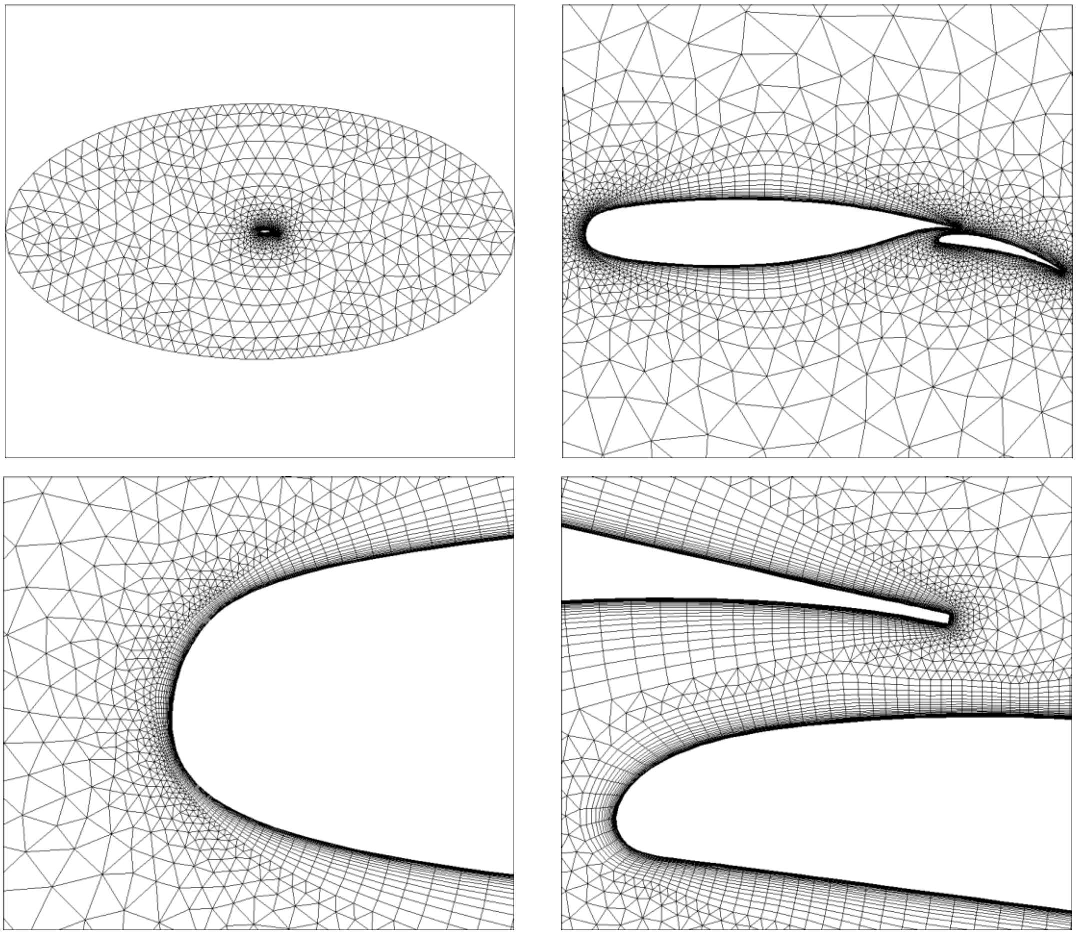


Figure 28 Solution Domain – NLR7301 Airfoil

In the current study the CFL number on the coarse grid levels is kept at 0.65. In explicit solutions the maximum CFL number at the fine grid level is 0.95. Whereas in the semi-implicit solution the maximum CFL number with a stable and convergent solution is limited at 8.

Before comparing the convergence rates of the explicit and the implicit multigrid solution algorithms, a study similar to the one performed in RAE2822 case is repeated here in order to optimize the number of iterations at the fine and the coarse grid levels for the best convergence rate. For this purpose SA is used as the turbulence model. A five order drop in L2-Norm of the flow residuals is taken as the convergence criterion which mostly provides converged aerodynamic loads. The solution matrix and the related CPU times in seconds for the convergence are given in Table 10 and Table 11.

Table 10 Convergence times (NLR7301, explicit-MG, SA)

	C1	C3	C5	C7	C10	C20
F1	X	**3463**	4776	6136	X	X
F3	O	**1940**	2327	2743	3549	X
F5	O	**1712**	2000	2223	2581	X
F7	2017	**1804**	1877	1995	2227	X
F10	2255	1900	**1775**	1900	2029	X
F20	3061	2617	2345	2185	**2119**	X

Table 11 Convergence times (NLR7301, implicit-MG, SA)

	C1	C3	C5	C7	C10	C20
F1	X	X	X	**1426**	X	X
F3	X	**1511**	1769	2005	2214	X
F5	O	**1341**	1422	1545	1647	X
F7	1291	**1277**	1366	1438	1498	X
F10	1292	**1192**	1238	1309	1321	X
F20	1416	1350	1293	1273	**1252**	X

From Table 10 and Table 11, it is seen that the fastest convergence for the explicit-MG method is obtained by iterating 5 times in the fine and 3 times in the coarse grid levels. Whereas for the implicit-MG case the fastest convergence is obtained by iterating 10 times in the fine and 3 times in the coarse grid levels. Note that these parameters are not the same with the ones obtained in the RAE2822 case. The optimum input parameters for each method is summarized in Table 12.

Table 12 Optimized Parameter List- NLR7301 Two Element Airfoil Solutions

	CFL (Coarse)	CFL (Fine)	# of Iterations (Coarse)	# of Iterations (Fine)
Explicit	-	0.95	-	-
Explicit-MG	0.65	0.95	3	5
Implicit	-	8	-	-
Implicit-MG	0.65	8	3	10

The complete residual histories of the fastest solutions obtained by the explicit and the implicit multigrid algorithms are presented in Figure 29.

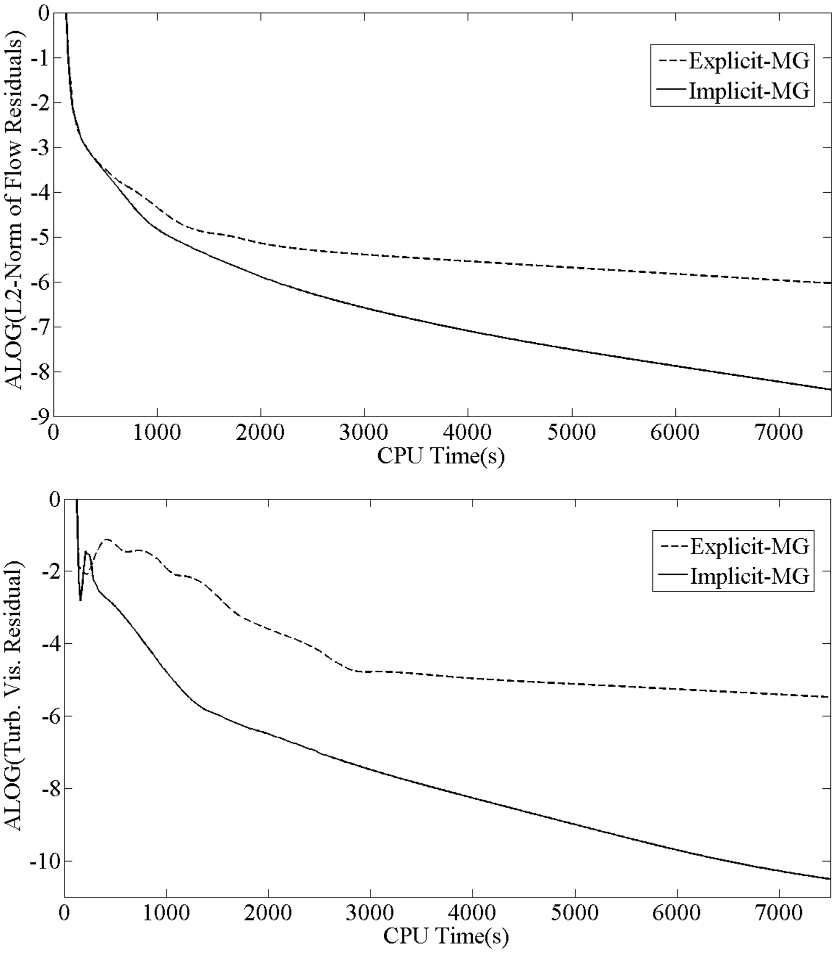


Figure 29 Residual Histories (NLR7301)

For a five order drop in the L2-Norm of the flow residuals, implicit-MG method is found to be 35% faster than the explicit-MG method.

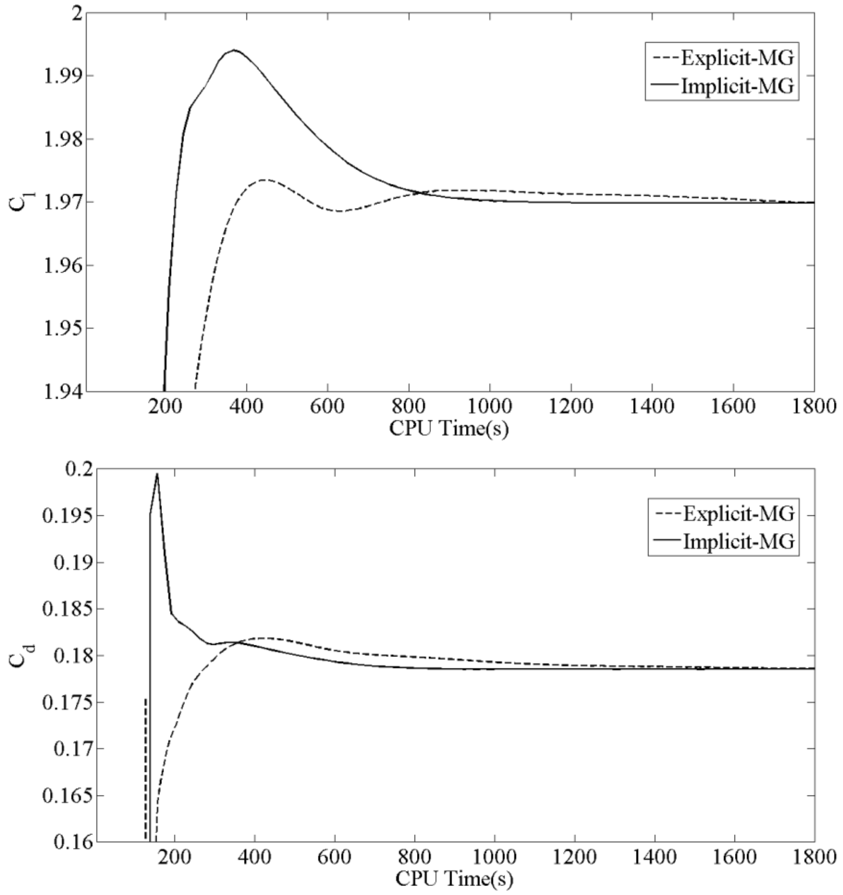
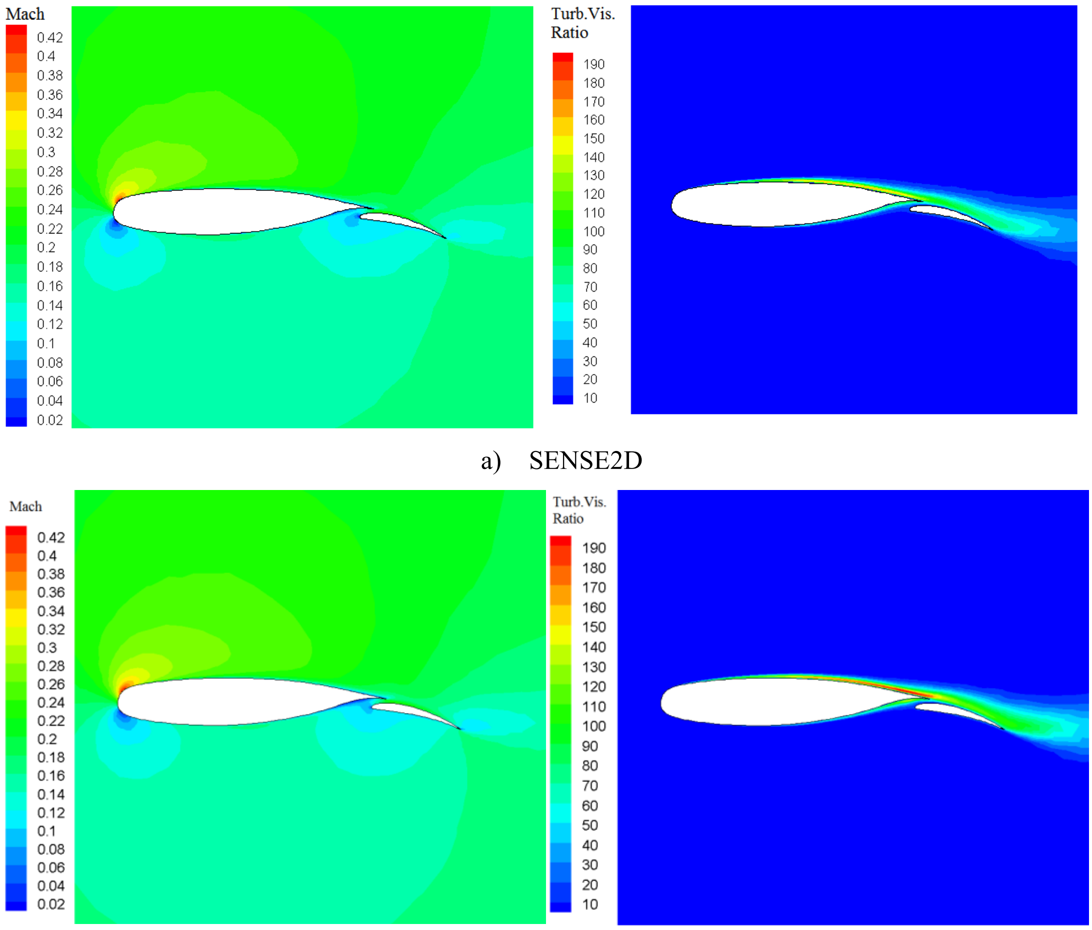


Figure 30 Load Convergence Histories (NLR7301,SA)

In Figure 30, the lift and the drag coefficients' convergence histories are presented. Similar to the residual case, the implicit-MG method is observed to be 1.3-1.5 times faster than the explicit-MG method in both lift and drag convergences. Being the fastest algorithm, results of the implicit-MG solutions will be presented from this point on.



a) SENSE2D

b) FLUENT

Figure 31 Mach Number & Turbulent Viscosity Ratio Fields (NLR7301,SA)

In Figure 31, the Mach number and the turbulent viscosity ratio fields of SENSE2D and FLUENT solutions are presented. The views of the fields from both solutions are in good agreement with each other.

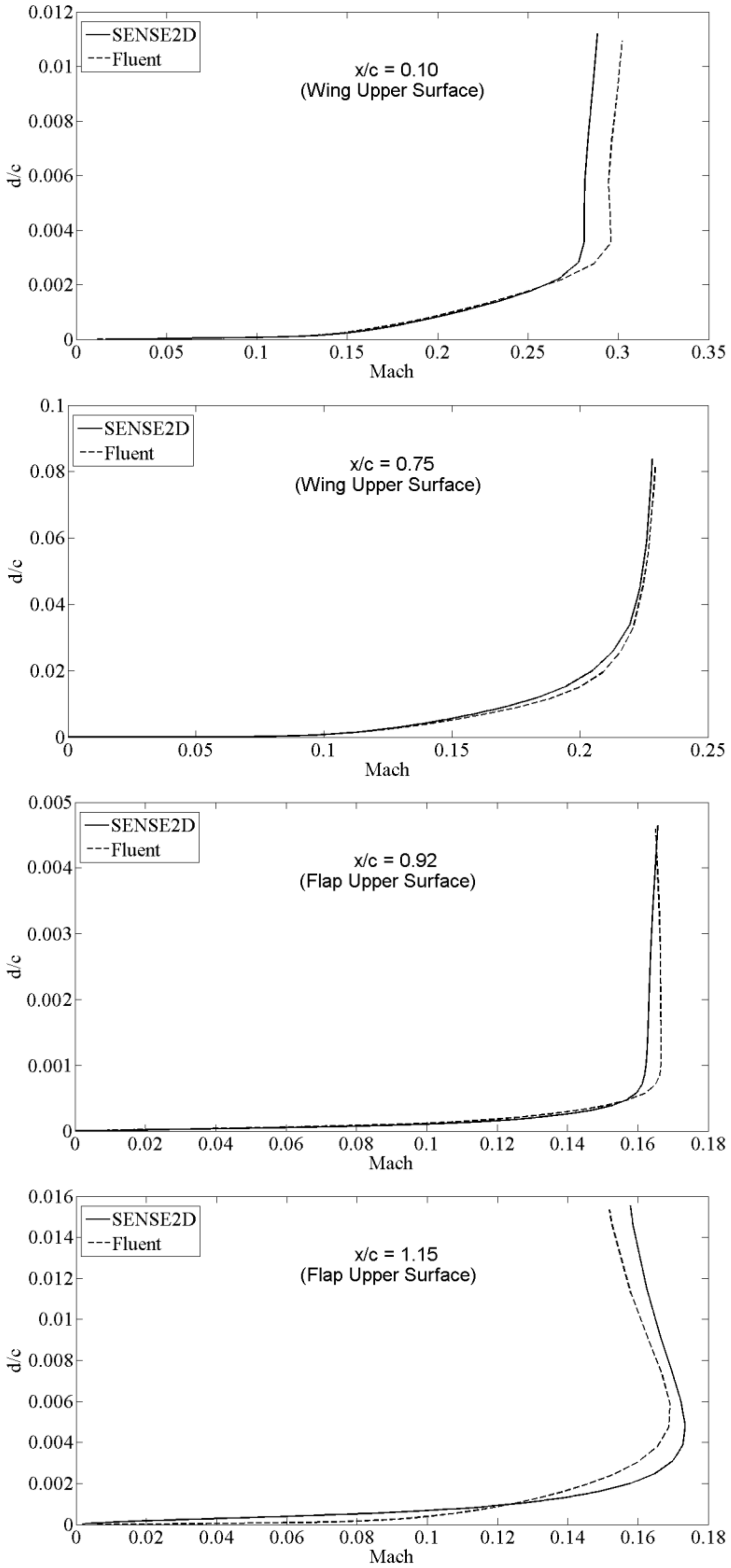


Figure 32 Turbulent Velocity Profiles (NLR7301, SA)

In Figure 32, turbulent velocity profiles obtained by SENSE2D are compared to the FLUENT results. The velocity profiles from each solver are on top of each other, even in the gap between the main wing and the flap where the boundary layers of two elements are overlapping.

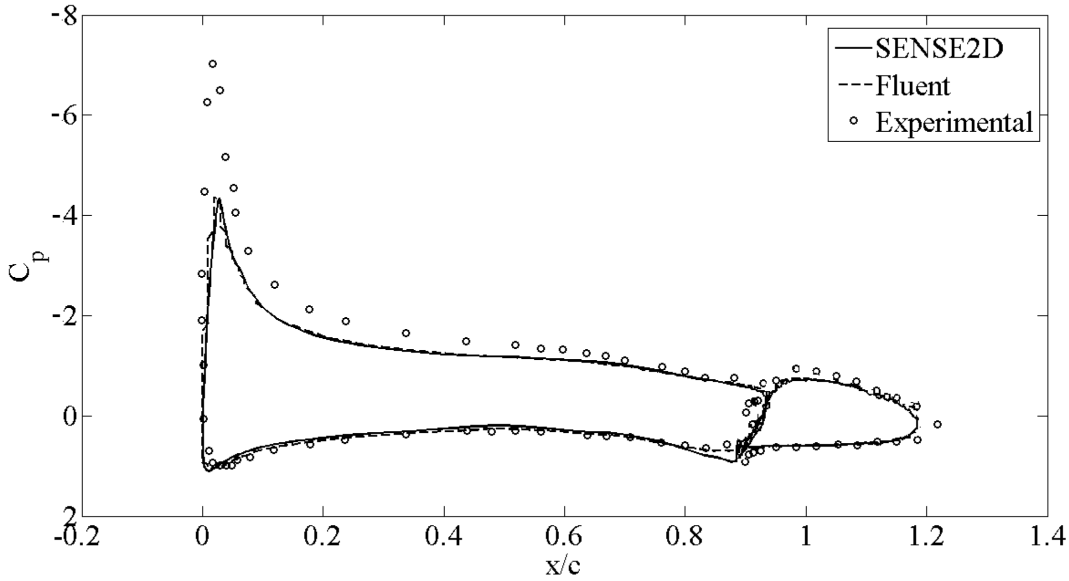
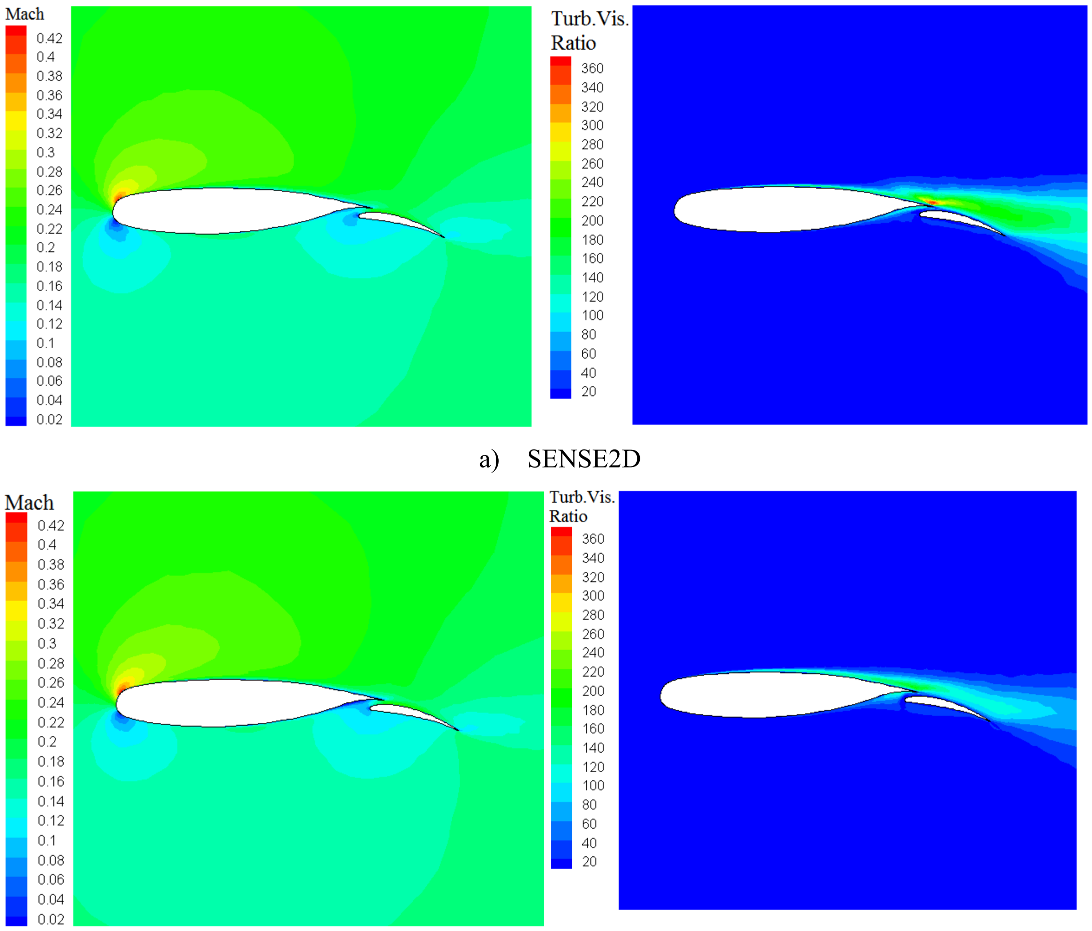


Figure 33 Pressure Coefficient Distribution (NLR7301, SA)

As a final accuracy assessment for SA turbulence model, the pressure coefficient distribution over the NLR 7301 multi element airfoil is given in Figure 33. Similar to the boundary layer velocity profiles, the pressure distributions predicted by SENSE2D and FLUENT agree quite well. However, they both underpredict the suction pressure at the leading edge of the main airfoil slightly due to the first order solutions. The spikes in the pressure coefficient at the trailing edges of the main wing and the flap are attributed to the cut trailing edges employed in the present study for the generation of O-type boundary layer grids.

Except the convergence rate study, all the solutions performed with the SA turbulence model are repeated with the SST k-ω turbulence model for validation purposes.



a) SENSE2D

b) FLUENT

Figure 34 Mach Number & Turbulent Viscosity Ratio Fields(NLR7301,SST k-ω)

In Figure 34, the Mach number and the turbulent viscosity ratio fields of SENSE2D and FLUENT solutions are presented. The Mach contour fields from both solutions are in good agreement with each other. On the other hand, SENSE2D is overpredicting the turbulent viscosity ratio field when compared to the FLUENT results.

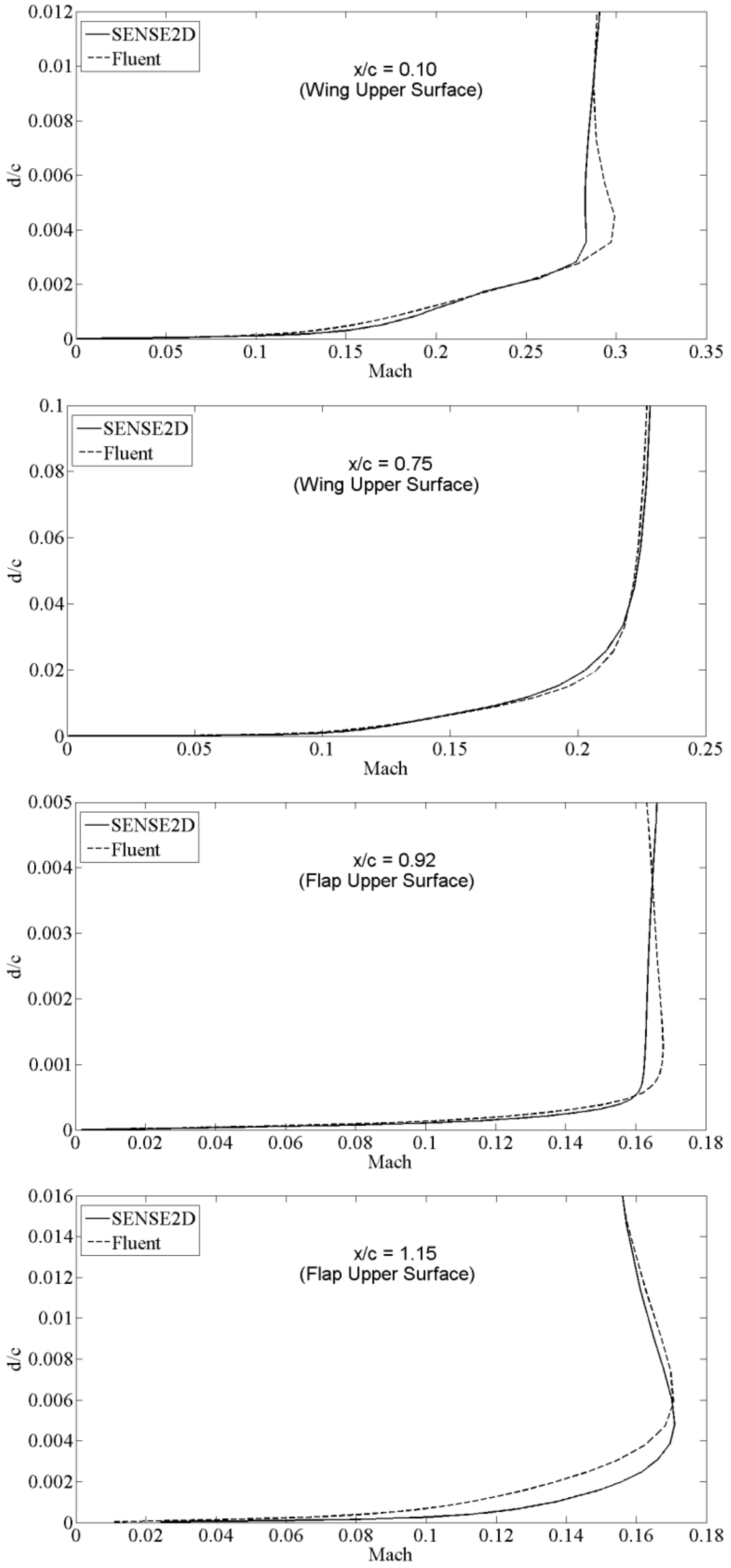


Figure 35 Turbulent Velocity Profiles (NLR7301, SST k-ω)

In Figure 35, turbulent velocity profiles obtained by SENSE2D are compared to the FLUENT results. The turbulent velocity profiles from each solver are quite similar.

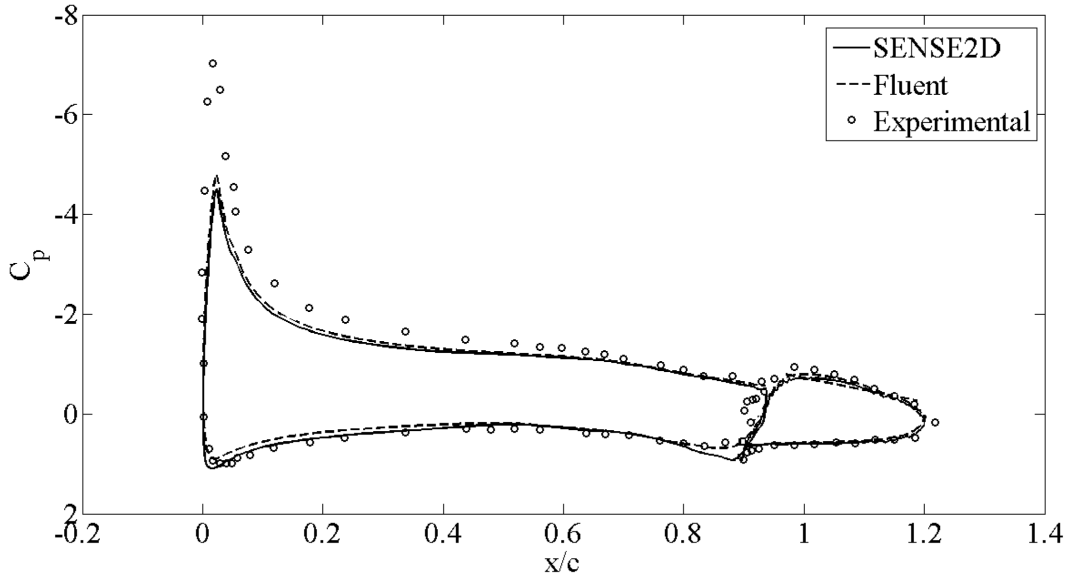


Figure 36 Pressure Coefficient Distribution (NLR7301, SST k-ω)

In Figure 36, pressure coefficient distribution over NLR7301 airfoil is presented. In parallel with the similar velocity field predictions, pressure distributions from SENSE2D and FLUENT solvers are also on top of each other. Moreover, both solutions agree with the experimental data except for the suction peak region where higher order spatial discretisation methods are required.

Table 13 Aerodynamic Coefficients (NLR7301, SA, SST k-ω)

	Spalart-Allmaras		SST k-ω	
Coefficients	SENSE2D	FLUENT	SENSE2D	FLUENT
C_l	1.97	1.99	1.99	2.02
C_d	0.1785	0.1932	0.132	0.136

Aerodynamic load coefficients of the NLR7301 airfoil are given in Table 13. The values obtained from SENSE2D and FLUENT are quite close to each other.

CHAPTER 4

CONCLUSIONS

In this study, the one and the two-equation turbulence models, namely Spalart-Allmaras and SST k-ω, are implemented to the multigrid NS solver, SENSE2D. Understanding the whole algorithm lying behind the solver in detail and implementing the turbulence model equations to this solver with robust and stable methods are the challenging aspects of this study.

Pure explicit treatment of the turbulence model equations generally results in instable solutions and it is proper to treat the large turbulence source terms at least point implicitly so as to stabilize the solutions. However, although the point implicit treatment of the source terms of turbulence model equations is a solution for the stability problem, convergence rates remain low with this application due to the small time scale of the turbulence. Therefore, more advanced implicit methods that accounts for the relationship between the neighboring cells is better to use so as to improve the convergence rates. The implementation of the full-implicit methods is the direct solution for both the stability and the slow convergence problem. However, it is hard to implement to a multigrid solver and can be appreciated as another research topic. Instead, multigrid flow solutions with explicit-like implicit time integration methods, such as semi-implicit method, are good alternatives of the full-implicit algorithms. The semi-implicit algorithms are found to be 5-10 times faster than their explicit counter parts with respect to the residual convergence rates. In addition, it improved the convergence rate of the explicit multigrid solver 35% in price of an explicit solver. As a result, as long as the time accuracy is not the primary objective, it is better to use semi-implicit methods instead of the explicit time stepping methods, such as Runge-Kutta, with regard to the improved stability and the convergence rate.

From the numerical point of view, the Spalart-Allmaras turbulence model has found to be easier to implement since it requires less correlations in the calculation of the turbulent viscosity and can be integrated through the whole domain. Moreover, at the near wall regions where the other two equation turbulence models suffer from instability, due to its well defined, unique boundary condition, Spalart-Allmaras turbulence model is very stable so that in most of the applications it results in convergent solutions without any limitations. The SST k-ω turbulence model can also be classified as stable. It can also be integrated in the whole domain including the boundary layer thanks to its well defined algebraic solutions in the near wall regions. Limiting the turbulence quantity k, by means of preserving its positivity, in the early stages of the solutions it is possible to have satisfactory results.

During this thesis study, the author also spent so much time in implementing the *low-Reynolds-number k-ε* and the various *two-layer k-ε* turbulence models, but, was not able to obtain any results with the semi-implicit method applied in this study. In addition, application of the *strang-splitting* method, which is also used in the solution of the highly stiff chemically reacting flows, did not ended up with satisfactory results either.

Being verified with the numerical and the experimental data of flow over flat plate and two different airfoils, implementation of Spalart-Allmaras and SST k-ω turbulence models to SENSE2D has completed successfully. As future works, one of the primary objectives is the repetition of the work performed in the scope of this study for the three-dimensional NS solver, SENSE3D. Moreover, implementation of the semi-implicit algorithm to the coarse level grids is expected to improve the convergence rate further. Spalart-Allmaras and SST k-ω are the two most popular turbulence models thanks to their superior accuracy. Therefore, there is no need to attempt more in order to implement other one and two equation turbulence models. Instead, beyond the modeling, simulation based turbulence prediction methods, such as Detached Eddy Simulation (DES) or Large Eddy

Simulation (LES), can be considered as near future works. Between these two, especially DES is becoming an industrial standard in between the turbulence prediction methods in parallel with the developments in the computer power and most probably will replace the turbulence modeling in the next five years. Also, it stands in between the turbulence modeling and the LES. Therefore, it can be seen as the next step just after gaining experience with turbulence modeling.

To conclude, during this study the author get detailed knowledge about the physics of turbulence and its modeling and also brought an in-house flow solver the capability of dealing with the turbulent flows. However, the most important return of the study to the author is the experience gained in the CFD field beyond turbulence modeling. Working with an in-house code, the author had a detailed knowledge about the whole theory lying behind an NS solver and created necessary background for working on the more advanced topics in CFD.

REFERENCES

[1] Morkovin, M.V.: *Effects of Compressibility on Turbulent Flow*. The Mechanics of Turbulence, Gordon & Breach, New York, 1964.

[2] Reynolds, O.: *On the Dynamical Theory of Incompressible Viscous Fluids and the Determination of the Criterion*. Phil. Trans. Roy. Soc., London, Series A 186 (1985), pp. 123-164.

[3] Speziale, C.G.: *A Review of Reynolds Stress Models for Turbulent Shear Flows*. ICASE Report 95-15, 1995.

[4] Baldwin, B.S.; Lomax, H.J.: *Thin Layer Approximation and Algebraic Model for Seperated Turbulent Flow*. AIAA Paper 78-257, 1978.

[5] Smith, A.M.O.; Cebeci, T.: *Numerical Solution of the Turbulent Boundary Layer Equations*. Douglas Aircraft Division Report 33735, 1967.

[6] Spalart, P.; Allmaras, S.: *A One-Equation turbulence Model for Aerodynamic Flows*. AIAA Paper 92-0439, 1992.

[7] Launder, B.E.; Spalding, B.: *The Numerical Computation of Turbulent Flows*. Computational Methods in Applied Mechanics and Engineering, 3 (1974), pp. 269-289.

[8] Wilcox, D.C.: *Reassesment of the Scale Determining Equation for Advanced Turbulence Models*. AIAA Journal, 26 (1988), pp. 1299-1310.

[9] Menter, F.R.: *Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applicaitions.* AIAA Paper 93-2906, 1993.

[10] Menter, F.R.: *Influence of Free Stream Values on k-ω Turbulence Model Predicitions*. AIAA Journal, 30 (1992), pp. 1651-1659.

[11] Blazek, J.: *Computational Fluid Dynamics: Principles and Applications*. Elsevier, 2001.

[12] Bussing, T.R.A.; Murman, E.M.: *Finite Volume Method for the Calculation of Compressible Chemically Reacting Flows*. AIAA Journal, 26 (1988), pp. 1070-1078.

[13] Vanden, K.J.; Whitfield, D.L.: *Direct and Iterative Algorithms for the Three-Dimensional Euler Equations*. AIAA Paper 93-3378, 1993.

[14] Saad, Y.; Schulz, M.H.: *GMRES: A Generalized Minimum Residual Algorithm for Solving Nonsymmetric Linear Systems*. SIAM Journal on Scientific and Statistical Computing, 7 (1986), pp. 856-869.

[15] Brandt, A.: *Guide to Multigrid Development. Multigrid Methods Lecture Notes in Mathematics, No 960*. Springer Verlag, New York, 1981.

[16] Versteeg, H.K.; Malalasekera, W.: *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*. Prentice Hall, 1995.

[17] Laney, C.B.: *Computational Gas Dynamics*. Cambridge University Press, 1998.

[18] Jameson, A.; Schmidt, W.; Turkel, E.: *Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes*. AIAA Paper 81-1259, 1981.

[19] Orkwis, P.D.; Vanden, K.J.: *On the Accuracy of Numerical versus Analytical Jacobians*. AIAA Paper 94-0176, 1994.

[20] Dennis, J.E.; Schnabel, R.B.: *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall, 1983.

[21] Ezertaş, A.A.: *Sensitivity Analysis Using Finite Difference and Analytical Jacobians*. M.S. Thesis, Middle East Technical University, 2009.

[22] George, A.; Liu, J.W.: *Computer Solution of Large Sparse Positive Definite Systems.* Prentice Hall Series in Computational Math., 1981.

[23] Briley, W.R.; McDonald, H.: *Solution of the Multi-Dimensional Compressible Navier-Stokes Equations by a Generalized Implicit Method*. Journal of Computational Physics, 24 (1997), pp. 372-397.

[24] Chapra, S.C.; Canale, R.P.: *Numerical Methods for Engineers*. Mc Graw Hill, 2002.

[25] Spalart, S.R.; Allmaras, S.A.: *A one Equation Turbulence Model for Aerodynamic Flows*. AIAA Paper 92-0439, 1992.

[26] Dacles-Mariani, J.; Zilliac, G.G.; Chow, J.S.; Bradshaw, P.: *Numerical/Experimental Study of a Wingtip Vortex in the Near Field* AIAA Journal, 33(9) (1995), pp. 1561-1568.

[27] Curtiss, C.F.; Hirshfelder, J.O.: *Integration of Stiff Equations.* Proceedings of the NAtional Academy of Sciences of the USA, Vol.38, 1952.

[28] *ANSYS FLUENT User's Guide,* version 13, ANSYS Inc., Canonsburg, 2010.

[29] *CFL3D Users Manual,* version 5.0, BaNANePOS, Inc., Virginia, 1998.

[30] Lallemand, M.H.; Steve, H.; Dervieux, A.: *Unstructured Multigridding by Volume Agglomeration: Current Status*. Computers and Fluids, 21 (1992), pp. 397-433.

[31] Koobos, B.; Lallemand, M.H.; Dervieux, A.: *Unstructured Volume Agglomeration Multigrid: Solution of the Poisson Equation*. International Journal for Numerical Methods in Fluids, 18 (1994), pp. 27-42.

[32] Mahmutyazıcıoğlu, E.: *Development of an Octree Based Grid Coarsening and Multigrid Flow Solution*. Phd. Thesis, Middle East Technical University, 2010.

[33] Patel, V.C.; Rodi, W.; Scheuerer, G.: *Turbulence Models for Near-Wall and Low-Reynolds Number Flows: A Review.* AIAA Journal, 23 (1985), pp. 1308-1319.