DEVELOPMENT OF AN OCTREE BASED GRID COARSENING AND MULTIGRID FLOW SOLUTION

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

BY

EMEL MAHMUTYAZICIOĞLU

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
MECHANICAL ENGINEERING

SEPTEMBER 2010

Approval of the thesis:

DEVELOPMENT OF AN OCTREE BASED GRID COARSENING AND MULTIGRID FLOW SOLUTION

submitted by **EMEL MAHMUTYAZICIOĞLU** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Mechanical Engineering Department, Middle East Technical University** by,

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

Prof. Dr. Süha ORAL _____
Head of Department, **Mechanical Engineering**

Prof. Dr. M. Haluk AKSEL _____
Supervisor, **Mechanical Engineering Dept., METU**

Prof. Dr. İsmail H. TUNCER _____
Co-Supervisor, **Aerospace Engineering Dept., METU**

Examining Committee Members:

Instructor Dr. Tahsin ÇETİNKAYA _____
Mechanical Engineering Dept., METU

Prof. Dr. M. Haluk AKSEL _____
Mechanical Engineering Dept., METU

Prof. Dr. Ünver KAYNAK _____
Mechanical Engineering Dept., TOBB-ETU

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

Asst. Prof. Dr. Cüneyt SERT _____
Mechanical Engineering Dept., METU

Date: 14 / 09 / 2010

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 : Emel MAHMUTYAZICIOĞLU
Signature :

ABSTRACT

DEVELOPMENT OF AN OCTREE BASED GRID COARSENING AND MULTIGRID FLOW SOLUTION

MAHMUTYAZICIOĞLU, Emel
Ph. D., Department of Mechanical Engineering
Supervisor: Prof. Dr. Mehmet Haluk AKSEL
Co-Supervisor: Prof. Dr. İsmail Hakkı TUNCER

September 2010, 167 pages

The multigrid technique is one of the most effective techniques to achieve the reduction of the CPU cost for flow solvers. The multigrid strategy uses the multilevel grids which are the coarsening subsets of fine grid. An explicit solver rapidly reduces the high frequency errors on the computational grids. Since high frequency errors on coarse grids correspond to low frequency errors on fine grids, cycling through the coarse grid levels rapidly reduces the errors ranging from high-to-low frequency. The aim of this study is, therefore, to accelerate SENSE3D solver developed by TUBITAK-SAGE by implementating multigrid concept.

In this work, a novel grid coarsening method suitable for cell-centered hybrid/unstructured grids is developed to provide the cells with high aspect ratio. This new grid coarsening technique relies on the agglomeration of cells based on their distribution on octree data structure. Then, the multigrid strategy is

implemented to the baseline flow solver. During this implementation, the flux calculation along the face loops is modified without changing cell-centered scheme.

The performance of the coarsening algorithm is investigated for all grid types in two and three dimension. The grid coarsening algorithm produces well defined, nested, body fitted coarser grids with aspect ratios of one and the coarse grids have similar characteristics of Cartesian grids. Then, the multigrid flow solutions are obtained at inviscid, laminar and turbulent flows. It is shown that, the convergence accelerations are up to 14 times for inviscid flows and in a range of 4 to 110 fold for turbulent flow solutions.

Keywords: Computational Fluid Dynamic, Multigrid, Agglomeration, Grid Coarsening, Unstructured Grid.

ÖZ

SEKİZDAL VERİ YAPISI İLE ÇÖZÜM AĞI SEYREKLEŞTİRME YÖNTEMİ VE ÇOK KATMANLI AKIŞ ÇÖZÜMLEMELERİNİN GELİŞTİRİLMESİ

MAHMUTYAZICIOĞLU, Emel
Doktora, Makina Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Mehmet Haluk AKSEL
Ortak Tez Yöneticisi: Prof. Dr. İsmail Hakkı TUNCER

Eylül 2010, 167 sayfa

Çok katmanlı çözüm tekniği, akış çözücüleri için çözüm zamanı azaltımında en etkin yöntemlerden biri olarak görülmektedir. Çok katmanlı çözüm tekniği sık çözüm ağından türeyen ardışık seyrekleştirilmiş çözüm ağlarını çalıştırmaktır. Açık uçlu çözücüler yüksek frekanslı hataları hızlı bir şekilde düşürebilmektedir. Sık çözüm ağında düşük frekansta bulunan hatalar, seyrek çözüm ağlarında yüksek frekans hatalara denk geldiğinden, çok katmanlı çözüm yönteminde düşükten yükseğe tüm frekanslardaki hatalar oldukça hızlı bir şekilde düşmektedir. Bu nedenle, bu çalışmanın amacı çoklu çözüm ağı tekniğinin TÜBİTAK-SAGE tarafından geliştirilen SENSE3D akış çözücülerine uygulanması ve bu yazılımların yakınsama hızlarının artırılmasıdır.

Bu çalışmada, üç boyutlu çok katmanlı çözüm uygulamalarında kullanılmak üzere, düzensiz/melez çözüm ağları ve hücre merkezli çözücülere uygun, yüksek en boy oranına sahip hücrelerden oluşan ardışık seyrekleştirilmiş çözüm ağları oluşturulma

yöntemi geliştirilmiştir. Bu yeni seyrekleştirme yöntemi, hücre merkezlerinin sekizdal veri yapısı kullanılarak birleştirilmesi temeline dayanmaktadır. Ardından, çok katmanlı çözüm tekniği temel çözücüye uygulanmıştır. Uygulama sırasında çözücünün hücre merkezli yapısı bozulmadan, akı hesabının kenar veya yüzey döngüsünde olması sağlanmıştır.

Seyrek çözüm ağlarının başarısı iki ve üç boyutlu tüm çözüm ağı tipleri kullanılarak incelenmiştir. Seyrek çözüm ağları incelendiğinde, oluşan hücrelerin geometri özelliğini bozmadığı, en boy oranının yaklaşık 1 olduğu ve seyrekleşme seviyesi yükseldikçe kartezyen tip çözüm ağına sahip olduğu görülmektedir. Ardından, çok katmanlı çözüm tekniğine sahip yazılım kullanılarak ağdasız ve ağdalı çözümler elde edilmiştir. Yapılan çözümlemeler ile, çok katmanlı çözüm yaklaşımının, düşük hızlı ağdasız çözümlemelerde 14 kata kadar, ağdalı çözümlerde ise 4 ila 110 kat arasında hızlanma oranlarına sahip olduğu gösterilmiştir.

Anahtar Kelimeler: Hesaplamalı Akışkanlar Dinamiği, Çok Katmanlı Çözüm, Birleştirme Algoritması, Çözüm Ağı Seyrekleştirme, Düzensiz Çözüm Ağı.

To

Those precious ones whom during my study I’ve been inattentive to

ACKNOWLEDGMENTS

I would like to thank the following people who made possible this thesis. First and the foremost, I would like to express my deepest gratitude to my co-supervisor Prof. Dr. İsmail Hakkı TUNCER for his inspiration and for helpful discussions we have made throughout the study. I would also like to thank my supervisor Prof. Dr. M. Haluk AKSEL for his insight and encouragement thoughout this research.

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. In particular, I am indebted to my colleagues; Osman BAŞOĞLU for his assistance in preparing validation grids, Erdinç Nuri YILDIZ for his helps about visualization of coarse meshes and Ali Emre YILMAZ for our negotiations about turbulent flow solutions.

My studies would not have been possible if along the way my friends did not provide me any moral support. So, my special thanks go to Aysun ÜNDAL, Döne DAYANÇ and Songül BİLGE for their extraordinary support and friendship.

Of course, I'm thankful to my parents, Müşerref - Fahri ÖZDEMİR, my brother and his wife Emrah – Yaprak ÖZDEMİR who offered me unconditional love, support and encourage in my entire academic and non-academic life.

Finally, it will be unfair for me to think that this achievement is only the result of my own effort. My deepest appreciation is expressed to my husband, Gökmen MAHMUTYAZICIOĞLU, for his endless patient, support and sacrifice which made this thesis possible.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF SYMBOLS

C_D	Drag force coefficient
CFL	Courant number
C_l	Rolling Moment Coefficient
C_L	Lift Force Coefficient
C_m	Pitching Moment Coefficient
C_n	Yawing Moment Coefficient
C_N	Normal Force Coefficient
C_p	Pressure Coefficient
C_Y	Side Force Coefficient
L	Length
L	Left State of a Cell
R	Right State of a Cell
M	Mach Number
n	Iteration count
R	Universal Gas Constant
Re	Reynolds Number; $\mathrm{Re} = \frac{\mu \mathrm{VL}}{\rho}$
x, y, z	Cartesian Coordinates
y+	Non-Dimensional Normal Distance
u, v, w	Cartesian Velocities in x, y and z directions

t	Time
T	Temperature

Greek Letters

α	Angle of attack
β	Side slip angle
ν	Kinematic Viscosity
μ	Dynamic Viscosity
ρ	Density

CHAPTER 1

INTRODUCTION

Over the past decades, Computational Fluid Dynamics (CFD) has become a valuable analysis tool for understanding the fluid flow. The use of computers has become an integral part of the design process in the aerospace industry. In this applied aerodynamics context, the discretizations of the Euler and/or Navier-Stokes equations are almost exclusively performed by finite volume methods. The pioneering work of Jameson began this evolution [1, 2, 3, 4]. During the 1980's, upwinding mechanisms were incorporated into finite volume algorithms leading to increased robustness for applications with strong shocks, and perhaps more importantly, to better resolution of viscous layers due to decreased numerical dissipation in these regions [5, 6, 7, 8, 9]. The 1990's saw major advances in the application of finite volume methods to Navier-Stokes simulations, in particular to the Reynolds-Averaged Navier-Stokes (RANS) equations and significant gains were made in the use of unstructured meshes [10, 11, 12].

Solution of flow fields around complex geometries directly addresses to use unstructured grids which do not need any connectivity information like structured grid. Unstructured-grid methodology has emerged as a mature of CFD tool for rapid aerodynamics analysis and design of complex configurations. It offers a substantially reduced turnaround time for CFD solutions due primarily to the ease and speed at which unstructured grids can be generated. Unstructured grids composed of triangular and tetrahedral elements in 2D and 3D. Unstructured meshes are also computationally feasible when the grid generation time concerned. It can be said that the size of the mesh, which directly affects the memory

requirements of the numerical algorithm, can only be minimized with the usage of unstructured grids [13]. Their main drawbacks, however, are the memory overhead associated with storing grid connectivity information and the computer time associated with indirect addressing. In addition, due to the lack of grid structure, it is difficult to implement simple implicit schemes such as approximate factorization, while explicit schemes suffer from slow convergence. In other words, the relaxation schemes in these flow solvers efficiently eliminate high frequency error modes but fail to reduce low frequency errors that hamper the flow convergence [10, 11, 13, 14]. Present-day convergence acceleration methods are mostly based on trying to achieve the optimum balance between speed of convergence and cost of iterations [15, 16].

The empirical verification of Moore's Law (i.e. doubling of computational power at fixed cost every 18 months) over the last two decades has caused some to question the need for improved convergence acceleration techniques, opting instead of concentrating on incorporating additional physics through increased model complexity and/or resolution. Unfortunately, the incorporation of additional physics most often also increases the stiffness of the problem, resulting either in problems which simply cannot be solved by simple solution techniques, or take even longer to solve in spite of the availability of faster hardware. In the "Blue Book" report on scientific computing compiled by the National Science Foundation, a comparison of the enabling hardware advances versus the enabling algorithmic advances, reproduced here in Figure 1.1, serves to illustrate how the two fields have contributed almost equally to the overall advances in near past-day simulation capability [15].

The need for more efficient steady-state solution algorithms takes on even more significance when one considers the trend from steady state Navier-Stokes solvers to unsteady solvers, and design optimization capabilities, which involve the solution of many intermediate steady-state or pseudo steady-state problems for each run.

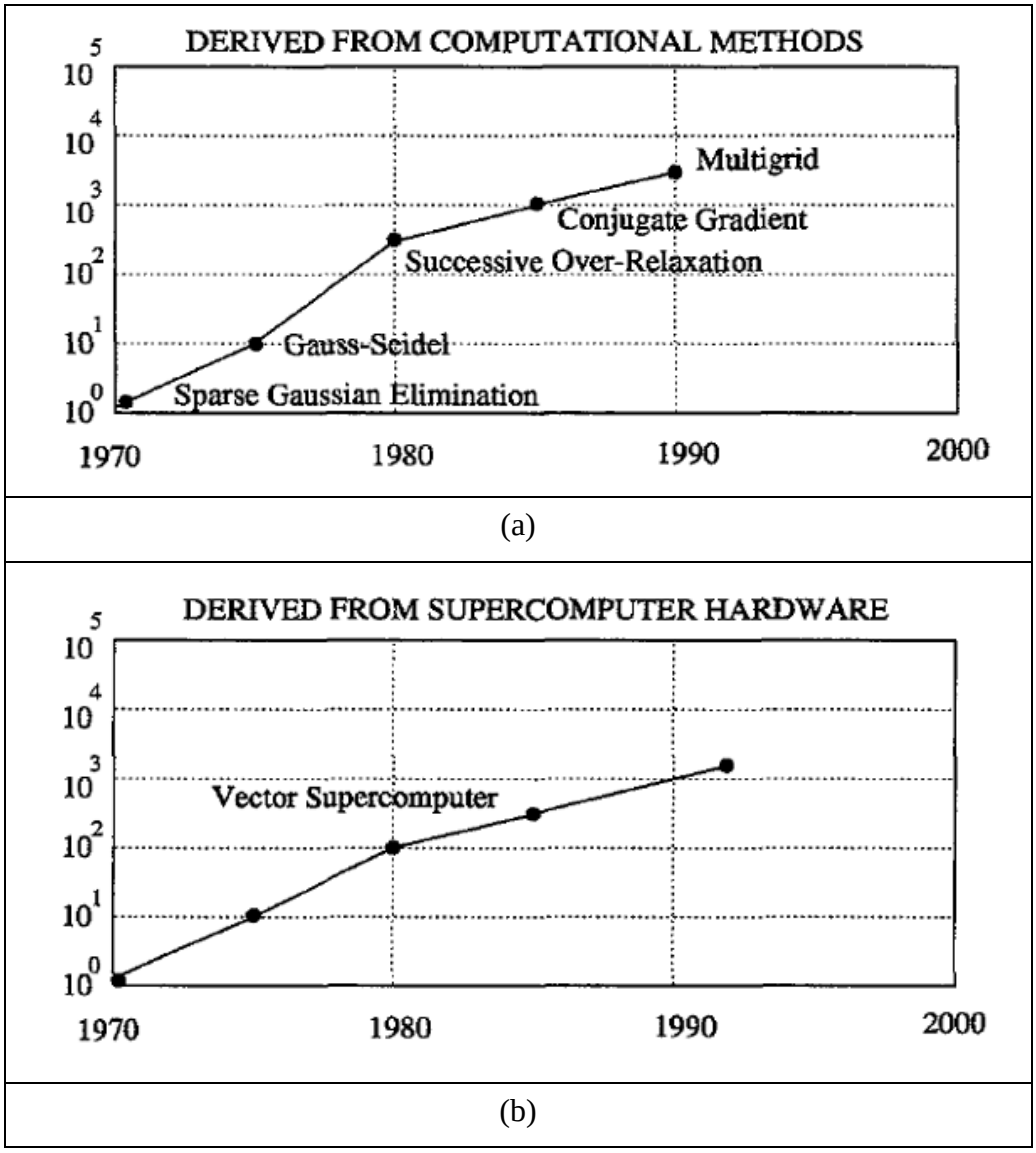


Figure 1.1 Illustration of advances due to algorithmic improvements and hardware

Due to the widespread usage of CFD in the Research & Development departments of industrial outfits, there is need for enhancing any CFD tool used for industrial purposes with techniques capable of reducing the CPU cost of a single computation. According to literature survey about unstructured grids, there are many studies about accelerating the CFD studies, generally by convergence acceleration techniques and parallelization techniques, rather than solution technique directly.

These acceleration techniques were investigated and it was determined that multigrid (MG) is likely to be the most effective technique to achieve this goal [15, 17, 18]. It gives attractive results for convergence and accuracy rates. The basic idea of a multigrid strategy is to accelerate the solution of a set of fine grid equations by computing corrections on a coarser grid. The motivation for this approach comes from an examination of the error of the numerical solution in the frequency domain. Multigrid methods have been successfully utilized by several structured grid Euler/Navier-Stokes solvers where a sequence of optimized coarser meshes can be readily generated from a base fine grid. In the past decade, various multigrid strategies have also been successfully demonstrated for the unstructured grid flow solvers [10, 14, 15, 18, 19, 20, 21, 22, 23, 24, 25, 26].

1.1 Literature Survey

1.1.1 Overview of Multigrid Strategy

The basic idea of a MG strategy is to accelerate the solution of fine grid equations by computing corrections on a coarser grid. High-frequency errors, which involve local variations in the solution, are well annihilated by simple explicit methods. Low-frequency or more global errors are much more insensitive to the application of explicit methods. This is natural, considering the local nature of the information employed in explicit schemes. In fact, the convergence rate of explicit schemes usually consists of a rather rapid initial residual reduction phase, which gradually develops into a much slower residual reduction phase, corresponding to a situation where all high-frequency errors have been eliminated and low frequency errors dominate, as shown in Figure 1.2.

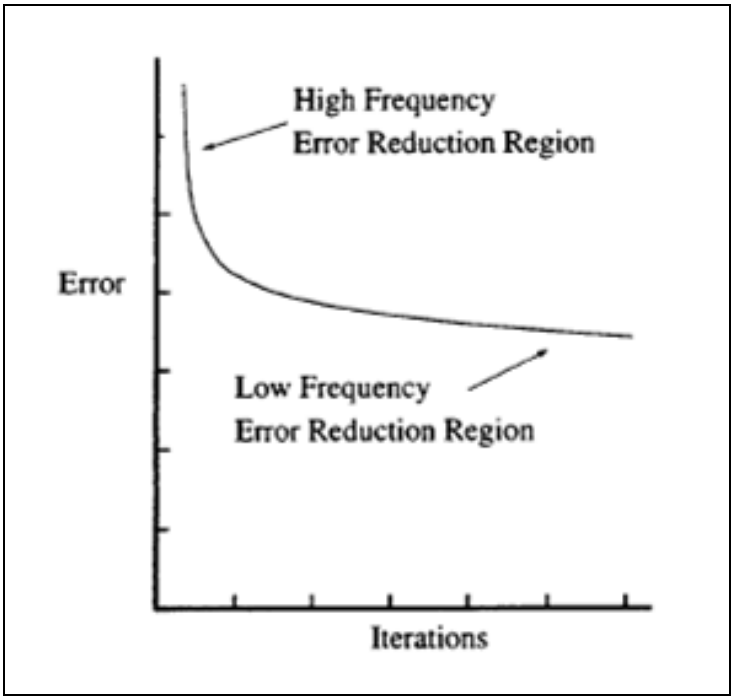


Figure 1.2 Typical convergence characteristics of an explicit scheme

MG strategies capitalize on this rapid initial error reduction property of explicit schemes. Typically, a MG scheme begins by eliminating the high-frequency errors associated with an initial solution on the fine grid, using an explicit scheme. Once this has been achieved, further fine grid iterations would result in convergence degradation. Therefore, the solution is transferred to a coarser grid. On this grid, the low-frequency errors of the fine grid manifest themselves as high-frequency errors, and are thus eliminated efficiently using the same explicit scheme. The coarse-grid corrections computed in this manner are interpolated back to the fine grid in order to update the solution. This procedure can be applied recursively on a sequence of coarser and coarser grids, where each grid-level is responsible for eliminating a particular frequency bandwidth of errors. MG strategies are generally considered as convergence acceleration techniques, rather than solution methods themselves. In fact, they may be applied to any existing relaxation technique, explicit or implicit. The success of the overall solution strategy depends on a close matching between

the bandwidth of errors in order to represent the entire error frequency range. These errors can be efficiently smoothed on a given grid using the particular chosen relaxation strategy, with a careful construction of a sequence of coarse grids.

MG methods have been successfully utilized by several structured grid Euler/Navier-Stokes solvers where a sequence of optimized coarser meshes can be readily generated from a base fine grid. They may also be employed to accelerate the solution of the full non-linear equation set, or they may be used to operate on the linear system which arises at each time-step in the implicit scheme of equations. While applying multigrid to the solution of the linear system in an implicit scheme affords certain advantages, and has been demonstrated successfully, it forfeits one of the principle advantages of the multigrid method, which is the low memory overheads required [14].

1.1.2 Background on Multigrid Strategy

The concept of using multiple grids as a means to accelerate convergence was first proposed in 1964 by Federenko [27] for a Poisson-type problem on a rectangular grid. It was not until 1977 when Brandt [28] presented his seminar paper that MG became practical. MG convergence was studied for finite element systems in 1977 by Nicolaides [29], and proven for certain PDEs in 1978 by Hackbusch [30]. 1981 marked the appearance of a finite-volume solver by Jameson, Schmidt, and Turkel [4]. The solver computed the Euler equations using an explicit five-stage Runge-Kutta time-stepping scheme. The spatial discretization was a second-order finite-volume technique, and was used in combination with an artificial dissipation scheme that allowed for the accurate solution of shock waves in transonic flow. A novel MG scheme for the solution of the Euler equations was proposed by Ni [31] in 1982. Using structured grids, Ni solved the transonic flow over a bump having a maximum height of 10 % of the chord in a channel of height one chord. Ni's scheme uses Lax-Wendroff time-stepping and is second-order accurate except in the

neighborhood of shocks. Jameson [1, 32] presented a multigrid extension of his 1981 Euler code that significantly improved the convergence rate of the solver. Jameson and Mavriplis introduced an algorithm that utilized unstructured grids [33]. This work was continued by Mavriplis [34], utilizing a sequence of non-nested, unstructured grids. This research demonstrated the feasibility of using unrelated meshes to produce results that greatly improved the convergence rate over that of single-grid, explicit solvers and was competitive with MG applied to structured grids. The use of MG on unstructured grids, including extensions to 3D, was also explored by Peraire et al. [35], Mavriplis [14], Morano and Dervieux [36], Riemslagh and Dick [37], Ollivier-Gooch [38], Lassaline [26], Zuliani [39] and Fidkowski [40]. In addition to the added complexity of intergrid operators, a potential area of difficulty is the generation of a sequence of coarse grids for unstructured fine grids.

1.1.3 Basic Principles of Multigrid Strategy

The basics of MG methods for both linear/nonlinear equations with details of common MG cycles and intergrid transfer operators are given in this section. These are the basic principles since they do not depend on the particular set of equations being solved, the discretization and types of grids employed, or the dimensionality of the problem.

1.1.3.1 Linear Systems (MG Correction Scheme)

A system of linear equations can be written as:

$$L(u) = f \tag{1.1}$$

where L is a linear operator, u is the solution vector, and f is a forcing function.

The discrete approximation of the system on a grid characterized by spacing h is written as

$$L_h(u_h) = f_h \tag{1.2}$$

where u_h is the converged solution to the discrete system.

The current estimate of the solution u_h is denoted as $\bar{u}_h$, which is obtained by approximate solution techniques such as iterative technique. Since $\bar{u}_h$ does not satisfy the above equation exactly, the error can be defined as:

$$v_h \equiv \bar{u}_h - u_h \tag{1.3}$$

Now, Equation (1.2) can be written as:

$$L_h(v_h + \bar{u}_h) = f_h \tag{1.4}$$

which, since L is a linear operator, can be written as:

$$L_h(\bar{u}_h) + L_h(v_h) = f_h \tag{1.5}$$

The error v_h can be represented on a coarser grid characterized by spacing H or 2h provided that it is sufficiently smooth to prevent aliasing of high-frequency components on the coarse grid. An approximation to v_h can be calculated on the coarse grid as:

$$L_H v_H = I_h^H (f_h - L_h \bar{u}_h) \tag{1.6}$$

where I_h^H is referred to as the restriction operator, which transfers quantities from the fine grid to the coarse grid. The subscripts h and H show the grid characterized by spacing of fine and coarse grids respectively. The implementation of this operator is described in the following section. It is seen that $f_h - L_h \bar{u}_h$ is the

residual on the fine grid. Defining forcing function as $f_H = I_h^H\left(f_h - L_h\bar{u}_h\right)$, it is possible to write

$$L_H v_H = f_H \tag{1.7}$$

Once v_H is obtained, the fine grid level can be corrected using

$$\bar{u}_h^{new} = \bar{u}_h^{old} + I_H^h v_H \tag{1.8}$$

where I_H^h, the prolongation operator, which represents the interpolation of the coarse grid corrections v_H to the fine grid. Details of this operator are presented in a later section.

Low-frequency error components can be efficiently eliminated on coarse grids at a fraction of the cost of a fine grid calculation. Eliminating these error components on the fine grid is very costly, as many more relaxation cycles are required than would be on the coarse grid. In addition, this process can be performed recursively on successively coarser grids with each coarse grid being used to compute a correction to the next higher grid level.

1.1.3.2 Non-Linear Systems (Full Approximation Storage Scheme)

For systems of nonlinear equations, the step taken between Equations (1.4) and (1.8) in the previous section cannot be performed, so a different formulation must be used. Followings are the description of the Full Approximation Storage (FAS) scheme [41].

$L_h\bar{u}_h$ can be substracted from both sides of Equation (1.4) to obtain a residual at the right hand side as

$$L_h\left(v_h + \bar{u}_h\right) - L_h\bar{u}_h = f_h - L_h\bar{u}_h = -r_h \tag{1.9}$$

for the coarse grid. Then, the above equation becomes:

$$L_H\left(I_h^H \bar{u}_h + v_H\right) - L_H\left(I_h^H \bar{u}_h\right) = I_h^H\left(f_h - L_h \bar{u}_h\right) = -I_h^H r_h \tag{1.10}$$

The coarse level grid solution, $\bar{u}_H$, can now be defined as:

$$L_H \bar{u}_H = f_H \tag{1.11}$$

with

$$\bar{u}_H = \bar{I}_h^H \bar{u}_h + v_H \tag{1.12}$$

By rearranging these terms and introducing a new coarse grid variable called “the coarse grid forcing function” (sometimes called the *defect correction*) as:

$$f_H = I_h^H\left(f_h - L_h \bar{u}_h\right) + L_H\left(I_h^H \bar{u}_h\right) = L_H\left(I_h^H \bar{u}_h\right) - I_h^H r_h \tag{1.13}$$

Once $\bar{u}_H$ is calculated, the fine grid solution is updated according to the following relaxation.

$$\bar{u}_h^{new} = \bar{u}_h^{old} + I_H^h\left[u_H - I_h^H \bar{u}_h^{old}\right] \tag{1.14}$$

The presence of the defect-correction term on the right-hand side ensures that the fine grid problem is represented by the coarse grid discretization, and that both coarse and fine grid equations converge to the same solution. This can be seen by considering the case where fine grid equations have been solved exactly. In this situation, the fine grid residuals all vanish, as does their interpolated result on the coarse grid. The ability to directly handle non-linear problems is one of the great advantages of MG algorithms.

1.1.3.3 Intergrid Transfer Operators

The restriction of the solution from a fine grid to a coarser grid and the prolongation of the correction from a coarser grid to a finer grid both utilize bilinear interpolation. One of the key elements for the success of such a method is the development of efficient transfer mechanisms between grids. Since the MG algorithm makes use of finer cells, accurate state values are needed for coarse cells [25].

For computational fluid dynamics problems, the most common choices are either injection or some variant of linear interpolation. Injection corresponds to the interpolation operator which preserves a constant function exactly. As an example, the value of a coarse grid cell would be assigned to all constituent fine grid cells which are contained inside the coarse grid cell by the injection operator. MG methods for structured grid often employ bilinear (in two dimensions) and trilinear (in three dimensions) inter-grid transfer operators. The simple piecewise linear interpolation is easily implemented to MG methods for unstructured grid based on triangular elements in two dimensions, and tetrahedral elements in three dimensions, using the linear finite-element shape functions associated with these elements. Piecewise linear interpolation operators preserve linear functions exactly.

The accuracy of the restriction and prolongation operators must be sufficient to avoid introduction of excessive errors to the solutions, which can in turn have a detrimental effect on convergence efficiency.

1.1.3.4 Cycling Strategies

A particular implementation of recursive coarse grid correction scheme is referred as a multigrid cycle. Cycling strategies refer to techniques employed to determine when to switch from one grid to the next, rather than to how to win a race on two

wheels. These can be divided into two basic approaches: adaptive and fixed cycling strategies. Adaptive cycling methods involve the monitoring of the numerical convergence process. When it is determined that the high-frequency errors on the current grid have been effectively eliminated, usually by observing a sharp slowdown in the convergence rate, the jump to a coarser grid is triggered. Although adaptive cycling strategies may appear more desirable, practical considerations such as simplicity and robustness usually result in the use of fixed cycling strategies, where a fixed pattern of coarse and fine grid iterations is prescribed.

The most common cycling patterns are V-cycle, W-cycle and full MG strategy. The choice of a particular cycling strategy must necessarily consider the complexity of the various grid levels.

1.1.3.4.1 V-cycle

The MG V-cycle, which is the most popular cycling strategy in the literature, begins on the finest grid of the sequence, where one relaxation or time-step is performed. The solution and residuals are then interpolated to the next coarser grid, where another time-step is performed. This procedure is repeated on each coarser grid until the coarsest grid of the sequence is reached seen in Figure 1.3. Then refinement phase starts and the coarse grid corrections are prolongated back to each successively finer grid. At the classical V-cycle strategy, single or multiple time-steps on each grid level is performed. This refinement procedure is repeated until the finest grid of the sequence is reached. At the particular variant of the V-cycle is sometimes known as a saw-tooth cycle (Figure 1.3), the coarse grid corrections are prolongated back to each successively finer grid until the finest grid is reached. There is no time-stepping on the coarse-to-fine phase of the cycle. It has been employed extensively for computational fluid dynamics problems.

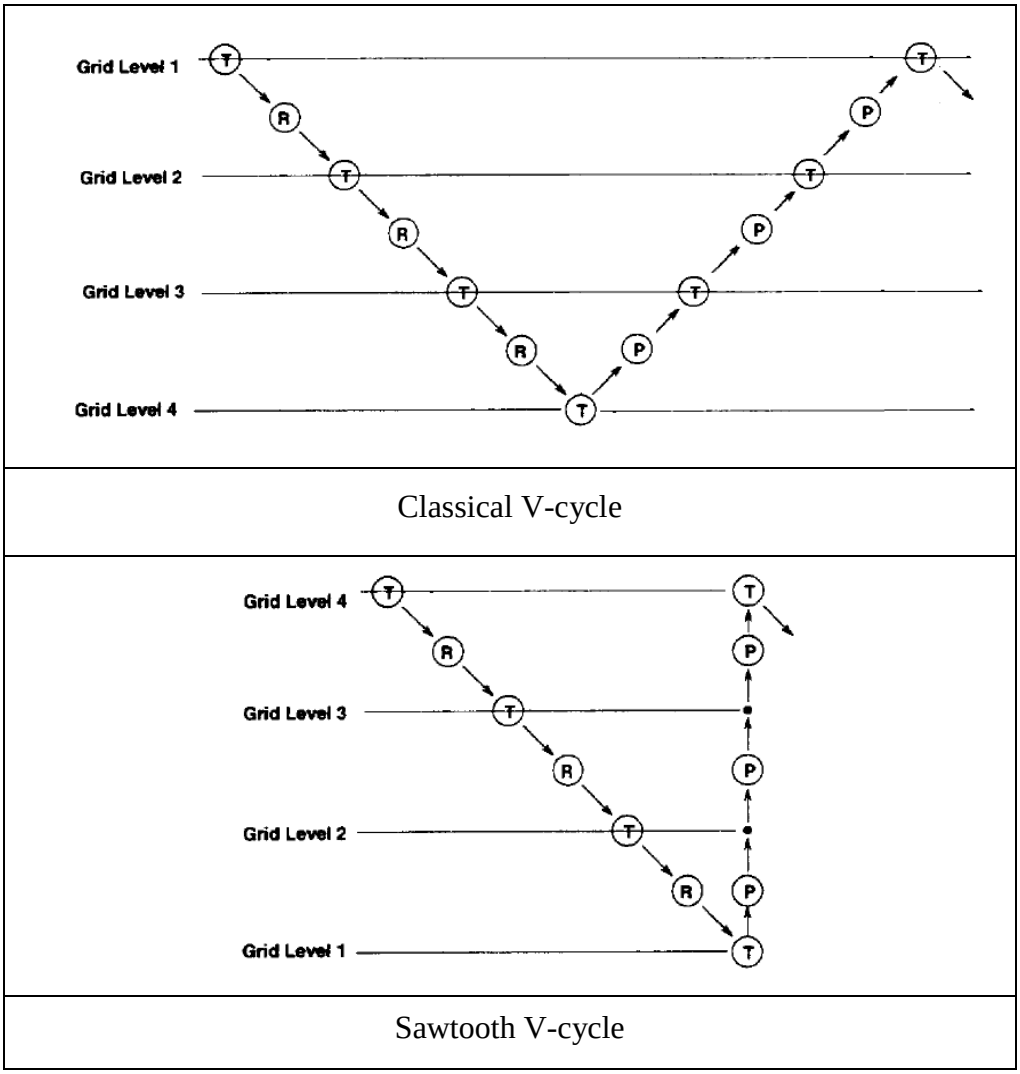


Figure 1.3 MG V-Cycle (T=time step, R=restriction, P=prolongation)

1.1.3.4.2 W-cycle

The W-cycle is the second common recursive strategy which weights coarse grids more heavily, as shown in Figure 1.4. Like V-cycle, it begins on the finest grid of the sequence, where one relaxation or time-step is performed, goes to the highest level coarse grid. Then refinement phase starts and the coarse grid corrections are prolongated back to each successively finer grid. However, different than the V-

cycle appearances, it again goes to the high level coarse level before reaching the root (fine) grid. The use of W-cycles is often found to be more efficient overall, and more robust than V-cycles.

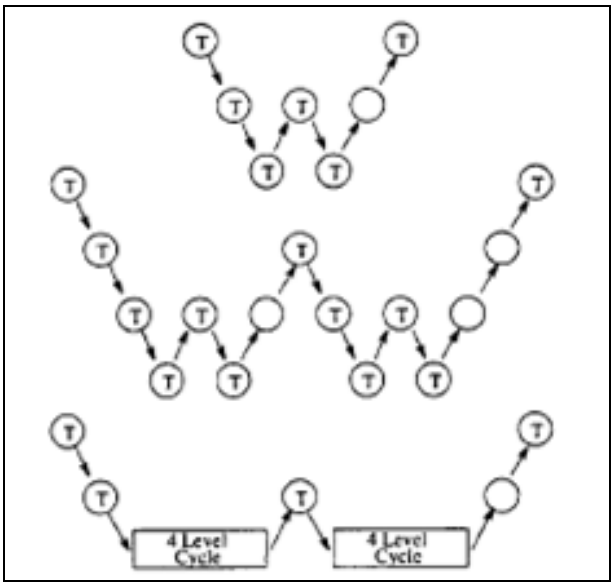
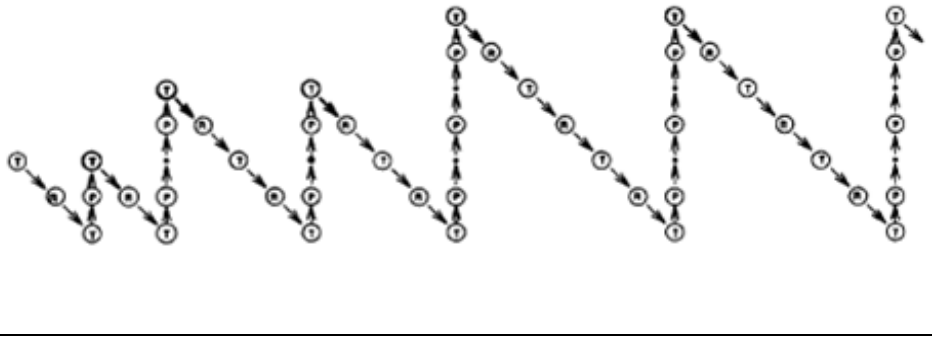


Figure 1.4 MG W-cycle (T=time step)

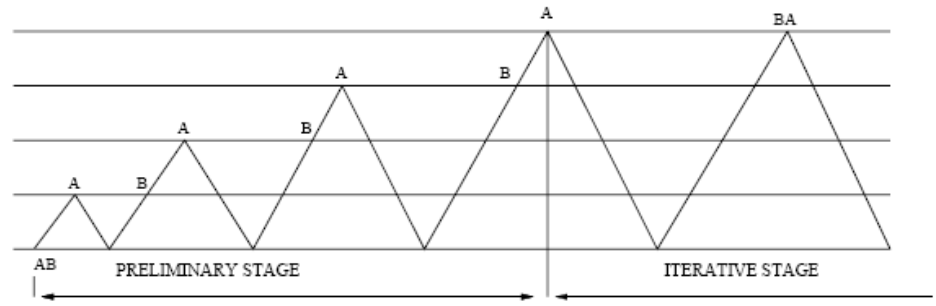
1.1.3.4.3 Full Multigrid (FMG)

The combination of mesh sequencing with a MG method (where the solution on the current grid is initiated from a previously computed solution on a coarser grid) results in a strategy known as FMG procedure. Beginning with an initial sequence of grids, the solution on the finest grid of the sequence is obtained at the preliminary stage by again using MG procedure. This sequence starts at fine or coarse grid levels as seen in Figure 1.5-b and c [15-41]. The procedure can be repeated, each time adding a new finer grid to the sequence, until the desired solution accuracy has been achieved, or the finest available mesh has been reached. After preliminary

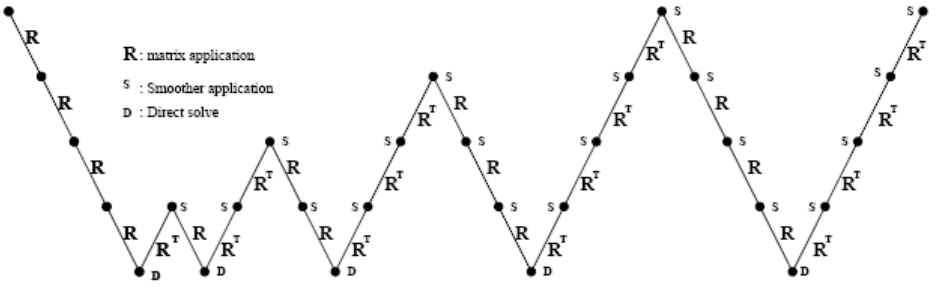
stage, it continues the iterative stage using saw-tooth or classical V-cycle depicted in Figure 1.5-a and b.



(a) (T=Time Step, R=Restriction, P=Prolongation)



(b)



(c) (s=Smoothing, D=Direct Solving, R=Restriction, R^T=Prolongation)

Figure 1.5 FMG strategy

1.1.4 Coarse Grid Construction Techniques for Unstructured Grids

The previous section described the basic principles of generic multigrid methods without regards to the types of grids on which these methods are to be applied. The main difficulty with unstructured multigrid methods is due to the construction of the coarse grid levels for the solution of the fine-grid equations. For structured mesh multigrid methods, a coarse mesh can be derived from a given fine mesh by omitting every second point in each coordinate direction. Recursive application of this procedure results in a sequence of coarse meshes where the complexity of the meshes decreases by a factor of 4:1 in two dimensions and 8:1 in three dimensions, for each successively coarser level.

For unstructured meshes, such techniques are no longer feasible. Due to the lack of mesh structure, simple coarsening strategies do not result in consistent coarse grid meshes. A variety of techniques have been proposed for unstructured multigrid coarse mesh constructions. These vary from methods which attempt to reproduce the nested property of structured mesh multigrid methods, to techniques which permit the use of arbitrary (triangular or nontriangular) coarse meshes to algebraic methods which never consider the construction of coarse meshes altogether [17]. In general, all methods are capable of delivering similar efficiencies and the issues involved in choosing a particular method include ease of implementation, degree of automation, and robustness for highly complex geometries. With fully unstructured meshes, where each point can have an arbitrary number of neighbors and the elements are non-uniform, the problem is much more difficult, particularly in three dimensions.

To date, six main and the most popular approaches towards grid coarsening on unstructured meshes have become prevalent. The first approach for obtaining coarse grid levels begins with the coarse mesh and generates finer nested levels by ***subdividing the coarse grid*** cells. The other approach is non-nested approach with

overset grids. In this approach, coarse grid levels are generated independently from the finer levels using any grid generator. A more automated technique operates on a fine grid by selecting a point to create the coarse grid using Delaunay ***triangulation algorithm***. For complex geometries, it is often difficult to generate a coarse grid which preserves the original geometry. An alternative method which keeps away from this problem is the ***agglomeration approach***. This method agglomerates the cells without creating any new edge or face. For complex geometries ***adaptive grid methods*** can also be used. These algorithms detect the regions that have prominent flow features and increase the grid resolution locally in such areas. Another method which avoids the generation of coarse grid is the ***algebraic multigrid approach***. This method operates on the matrix rather than on the grid. In this section, the brief theories of mentioned coarsening strategies will be presented.

1.1.4.1 Nested-Mesh Subdivision Method

One of the simplest unstructured mesh multigrid strategies is to generate a sequence of finer meshes from an initial coarse mesh by recursively subdividing the cells of the mesh [41, 43, 44], either globally, or adaptively. This results in a fully nested sequence of grids, as shown in Figure 1.6, and enables a particularly simple construction of the inter-grid transfer operators. For example, in the context of a vertex scheme, the values at the vertices which are common to coarse and fine grids are simply transferred by injection. Similarly, the newly introduced fine grid points always lie midway along a coarse grid edge, and thus the values at these points may be transferred by averaging the two values at the end points of the coarse grid containing the edge, which corresponds to linear interpolation.

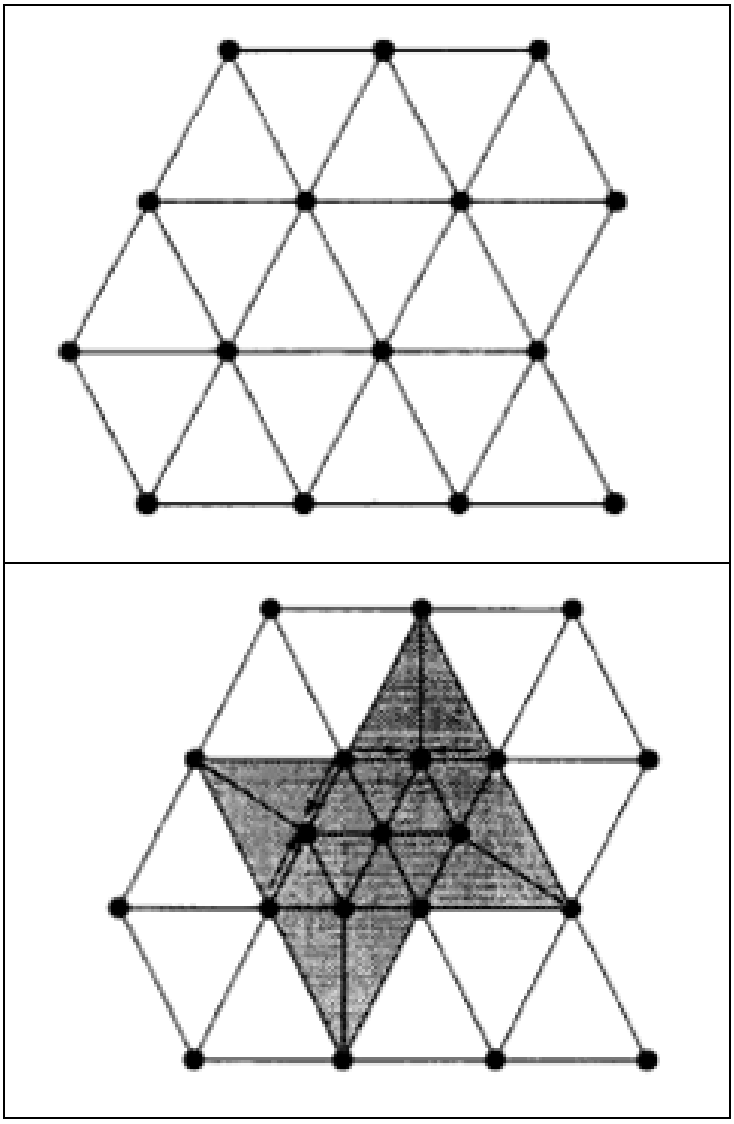


Figure 1.6 Illustration of the nested mesh subdivision construction

For a cell-centered scheme, volume weighted restriction is easily achieved by identifying the fine grid constituent cells of each coarse grid cell, and summing their weighted values. Another advantage of this approach is that it can be easily automated. This method has a somewhat inverted nature, i.e., it begins with a coarse mesh and subsequently generates finer meshes, whereas most multigrid methods begin with the finest mesh and construct coarser levels. There are several

disadvantages associated with such a strategy. The most obvious one is the lack of flexibility in handling problems on a specified fine grid of unknown origin. In fact, this approach requires a tight coupling between the grid generation and the multigrid solution strategies, and, thus has often been implemented in the context of adaptive meshing problems. The other difficulties are somewhat more subtle, but are interrelated. They concern with the ability of the coarsest initial grid to provide efficient convergence properties for the multigrid algorithm, and the quality of the resulting fine grid. In a multigrid process, the coarsest grid of the sequence determines the convergence rate of the algorithm, while the finest grid determines the accuracy of the solution. The present multigrid strategy places conflicting demands on the coarse mesh construction. On the other hand, a very coarse mesh is desired, since this enables a rapid multigrid convergence. However, the use of very coarse initial mesh may result in poor quality fine meshes, particularly when using simple subdivision refinement techniques. This, in turn, has a detrimental effect on the solution accuracy.

1.1.4.2 Overset Meshes Method

An alternate approach to unstructured MG methods is to generate a sequence of completely independent coarse and fine meshes, and use linear interpolation to transfer variables back and forth between the various meshes of the sequence, within a MG cycle [34, 35, 21, 36, 37, 45]. The meshes may be generated using any grid generation technique, will generally be non-nested, and may not even contain any common points, as shown in Figure 1.7. An essential step in the construction of the inter-grid transfer operators is the determination of the enclosing triangle on one grid for each vertex of the other grid. A native implementation of this operation consists of checking every triangle on the first grid for each vertex of the second grid. The other requirement is that they conform to the same domain boundaries.

This technique is more flexible than the nested subdivision approach, since the fine and coarse meshes are not constrained and may be optimized independently for accuracy and speed of convergence, respectively. Furthermore, this approach can be applied to a problem with a pre-specified fine mesh. On the other hand, the construction of the inter-grid transfer operators becomes more involved.

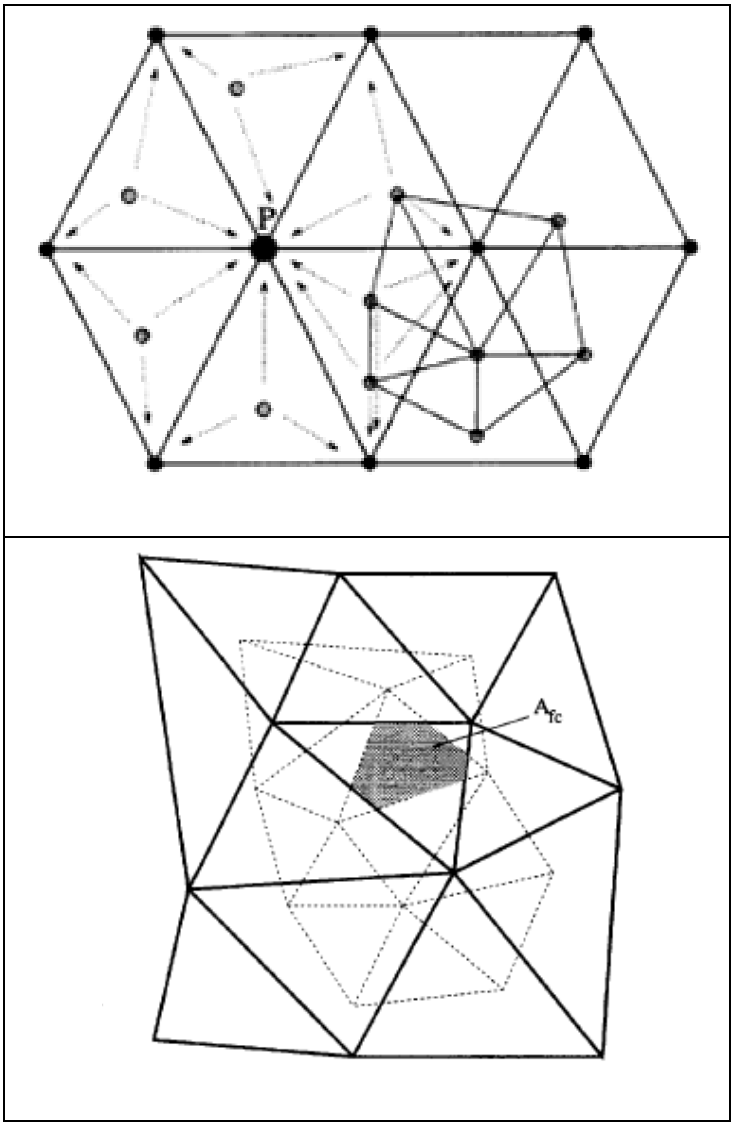


Figure 1.7 Illustration of conservative residual restriction for overset meshes

1.1.4.3 Re-triangulation Method

For automated coarse mesh production, the simple method is the removal of selected fine grid vertices and the re-triangulation of the remaining grid points. The re-triangulation procedure may be accomplished as a global operation, by regenerating the triangulation of the remaining coarse grid points, or incrementally, by removing each selected point sequentially and locally reconfiguring the mesh connectivity. For example, a reverse Delaunay point-insertion may be utilized in two-dimensions to remove mesh points [46, 47]. These techniques result in vertex-nested meshes, where the coarse grid vertices form a subset of the fine grid vertices, as shown in Figure 1.8. The triangulations themselves are not necessarily nested, since the connectivity of the coarse mesh need not be related to that of the fine mesh.

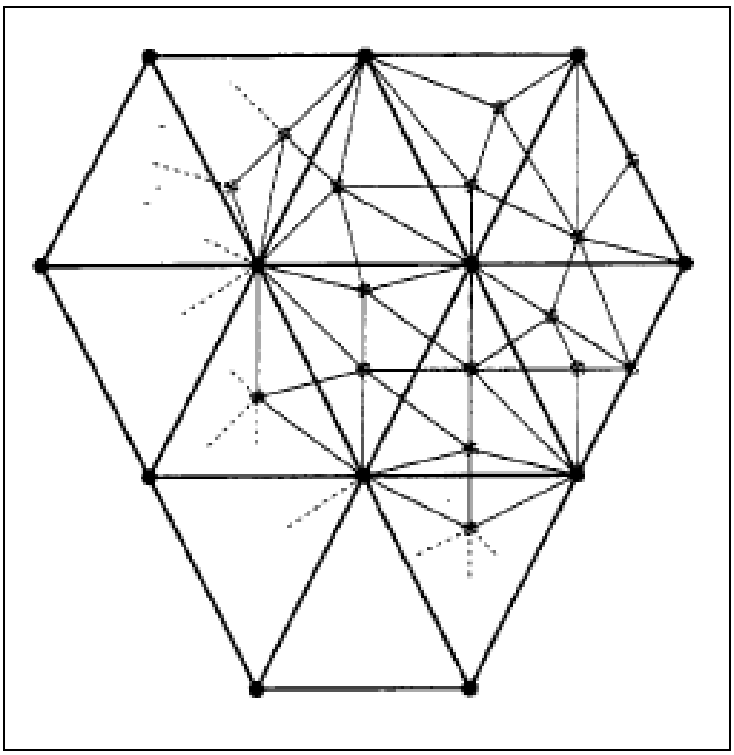


Figure 1.8 Illustration of vertex-nested coarse and fine mesh re-triangulation

Although the vertex-nested property may be employed to simplify the construction of the inter-grid transfer operators (i.e., for example the $\bar{I}_h^H$ operator reduces to simple injection), the construction techniques discussed in the previous section for overset-mesh MG methods are equally applicable in this case.

The point-removal procedure of automated coarsening strategies can be configured to generate "optimal" or near-optimal coarse meshes. This, of course, assumes some definition of optimal coarsening. A common strategy is to attempt to reproduce the coarsening characteristics encountered in structured mesh MG methods. Thus, coarse meshes which contain approximately half the resolution of the originating fine mesh in each coordinate direction throughout the entire domain are generally sought [14].

On the other hand, there are some practical difficulties in constructing coarse mesh levels for unstructured mesh MG algorithms; since, they do not address the issue of the robustness of the coarse grid constructions. For example, it may often be found that an automated coarsening procedure has removed one or several boundary mesh points which critically define the geometry, and the resulting changes in the geometry between grid levels produces a slowdown or failure of the MG algorithm. In fact, the triangulation of a coarse point-set about a complex geometry can prove to be a difficult task. For certain problems, the uniform coarsening characteristics of maximal independent sets which minimize structured mesh MG methods may be far from optimal. This is particularly true for problems with large disparities in length scales and anisotropic problems.

1.1.4.4 Agglomeration Method

The object of agglomeration or sometime called volume weighted coarsening technique is to derive a sets of coarse grids from a given fine grid and is based on a neighborhood relation [48, 49, 50]. The coarse grids are constructed in two steps by

volume agglomeration. In the first step, all the fine grid cells attached to the body surface or a far-field boundary are identified and merged with its neighboring (only those cells are eligible that are not already assigned to a previous coarser cell) cells to form a new coarser cell. After all of these prioritized boundary cells are assigned to a coarser cell, an unassigned fine grid cell on the agglomeration front is picked in a random order and merged with its eligible neighbors to form a new coarser cell. The procedure is repeated until all the fine grid cells are assigned to a coarser parent cell. In the second step, a set of fine grid faces at the interface of a given pair of coarser cells is used to render a single resultant face [15]. This step reduces the number of faces in a coarse grid, which has a direct bearing on the computational efficiency of the agglomeration MG technique.

Agglomeration methods are control-volume-based methods, and can thus be applied to either cell centered or vertex-based schemes. For cell-centered schemes, the control-volumes, themselves, are taken as the triangles themselves, whereas for a vertex-based scheme the control volumes are taken as the cells defined by the dual mesh formed by drawing the triangle median segments, as shown in Figure 1.9 [15].

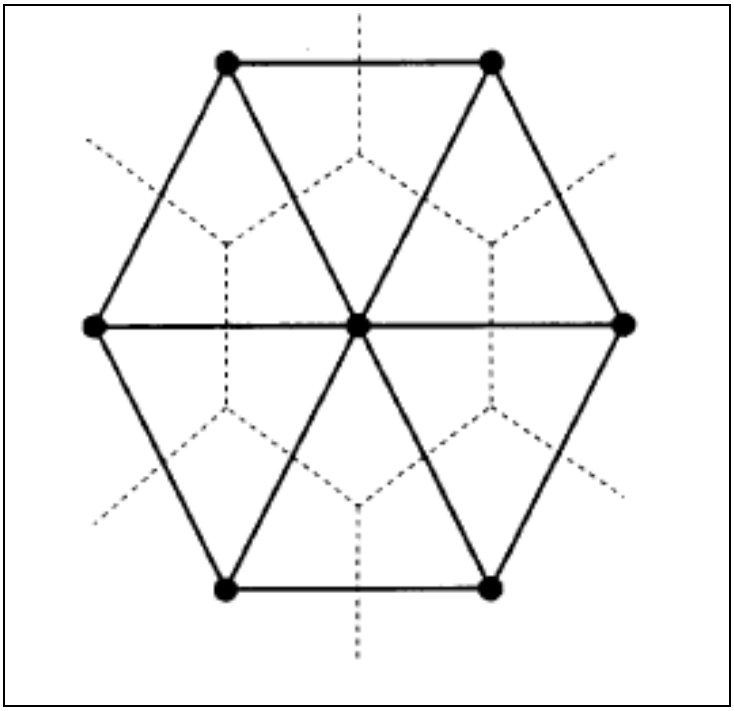


Figure 1.9 Median dual control volume for a triangular mesh in agglomeration

In other words, the idea of the agglomeration method is to fuse together or agglomerate neighboring fine grid control volumes, creating a smaller set of larger polygonal (or polyhedral in 3D) control volumes. This process can be performed recursively, thus generating an entire sequence of coarse agglomerated meshes. The degree of the coarse agglomerated polygons increases on each coarser mesh level, but they always conform exactly to the original fine grid boundaries. Figure 1.10 shows the agglomeration method on a vertex-based scheme.

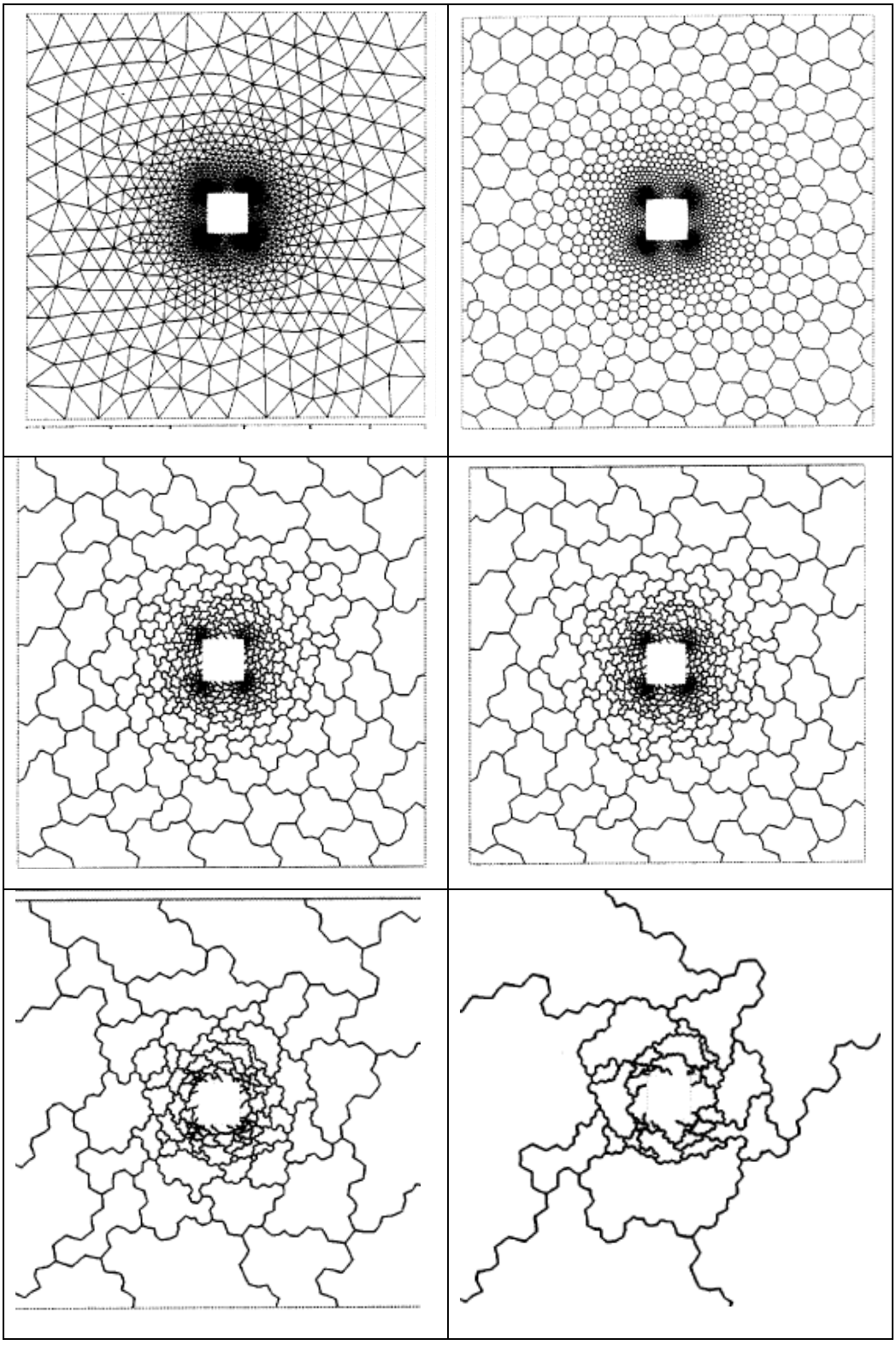


Figure 1.10 Original fine mesh, its dual mesh and coarse agglomerated mesh levels

The most important issue in agglomeration method is to decide the neighbor to agglomerate and, consequently, to define the points for removal. The techniques employed for creating the coarse agglomerated grids are similar to the automated coarsening strategies described in the previous section. In fact, there is a duality between agglomeration of control volumes and point removal. If each agglomerated control volume is thought of as consisting of its seed point, i.e., the point corresponding to the control volume from which the agglomeration process was initiated, and its agglomerated control volumes (or corresponding points), as shown in Figure 1.11, then the seed point corresponds to a point which is retained for the coarse grid in the point removal procedure, and the agglomerated points correspond to the deleted points.

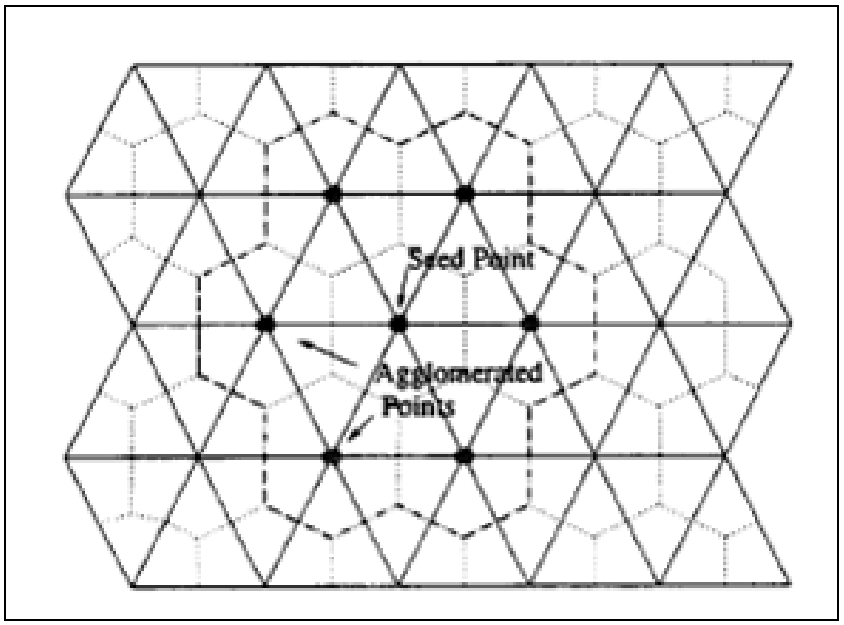


Figure 1.11 Illustration of seed point and agglomeration coarse grid construction strategy

1.1.4.5 Adaptive Multigrid Meshes Method

Adaptive grid methods have evolved as an efficient tool to obtain numerical solutions without a priori knowledge of the nature and the resolution of the grid necessary to efficiently capture the flow features. These algorithms detect the regions that have prominent flow features and increase the grid resolution locally in such areas. Furthermore, they coarsen the grid by deleting the cells over the regions where flow features no longer exist. In MG applications, agglomeration type coarsening algorithm is generally used [51, 52]. The data structures needed for the implementation of adaptive algorithms on 3D unstructured grids are quite complicated and this has been a challenging topic in itself and until recently not many such schemes existed. Significant progress has been made during the implementation of adaptive schemes for tetrahedral grids.

The coarser grids are used to propagate changes of the fine grid solution in time properly and rapidly throughout the flow domain, thus accelerating the convergence to the steady state, while at the same time maintaining the low truncation error on the fine grid. Adaptive grid algorithms generally use methods, which employ special upwind-like smoothing operators for shock-capturing and background smoothing. The adaptive grids are created by the division of tetrahedral cells [53].

1.1.4.6 Algebraic Multigrid Method

Algebraic MG methods are methods that enable the efficient solution of systems of algebraic equations, which are not necessarily derived from the spatial discretization of a partial differential equation [54]. In fact, the notion of a grid, of linear interpolation in space, and spatial smoothness are not always possible in this context. Thus, algebraic MG methods require the redefinition of such concepts in

the context of algebraic rather than geometric quantities, in order to make use of traditional MG principles. The algebraic formula is given as:

$$x^{new} = Gx^{old} + I_H^h \left[(A_c)^{-1} b_c - x_c^{old} \right] \tag{1.15}$$

where G represents the fine grid smoother, and it is assumed that the coarse grid matrix A_c may be easily inverted. The above sequence of operators represents a MG cycle having two-grids. This is described here for simplicity and in practice, a multiple level cycle may be defined by recursive application of the above two-grid procedure. A standard algebraic MG construction is to take the restriction operator as the transpose of the prolongation operator:

$$I_H^h = \left(I_h^H \right)^T \tag{1.16}$$

as was done in the overset-mesh MG algorithm, and to use the Galerkin coarse grid operator construction to define the coarse level matrix, A_c:

$$A_c = \left(I_H^h \right)^T A I_H^h \tag{1.17}$$

Once these steps are taken, the complete algebraic MG algorithm is determined solely by the definition of the prolongation operator and the set of coarse level variables. Since geometric information is not available, the coarse level variable sets must be determined from the algebraic information contained in the matrix A. To do this, the graph of the matrix A can be used. The graph of a sparse matrix is defined as the graph which is obtained by drawing an edge between the two vertices which correspond to the row and column number of each non-zero entry in the matrix. An algorithm which generates a maximal independent set of this graph may be utilized to construct a coarse level subset of variables, just as in the agglomeration or automated coarsening approaches for geometric MG. Algebraic MG, however, adds an extra degree of sophistication to the process, by considering the magnitude of the non-zero matrix entries. Coarsening is performed

preferentially along edges associated with large matrix entries, since this represents neighboring equations which are strongly coupled, and which will thus have similar errors (i.e., the error distribution will be smooth in that direction).

One of the drawbacks of algebraic MG methods is the complexity of their construction. The prolongation operator is not only used to transfer corrections from coarser grids to finer ones, but also contributes to the construction of the coarse grid operator. Thus, a prolongation operator with large or widely varying stencils may result in considerably complex coarse grid operators. In fact, the coarse grid operator is usually much denser (contains relatively more non-zero elements) than the original fine grid operator, which results in increased coarse grid complexities for the MG cycle. Thus, the construction of algebraic MG methods necessarily involves a trade-off between accuracy of the operators and complexity of the coarse grids.

1.2 Multigrid Adaptation: Motivation

Multigrid adaptation, as it is used in many different areas like CFD, acoustic, finite element problems, are subject to speed any kind of platform and programming language without losing accuracy and the performance requirements.

The Aerodynamic Design Team in TÜBİTAK-SAGE works with CFD tools to make the final design of various air vehicles due to lack of a high speed wind tunnel in Turkey. Besides the accuracy of the flow solver, the aerodynamic characteristics of a newly designed air vehicle should be investigated to cover the overall flight envelope. For this reason, a huge number of solutions is required to cover whole domain. However, in general, it is not possible to obtain such large number of solutions by using CFD tools, due to computational and time limitations of the project. The main motivation in this study is, therefore, to accelerate SAGE Euler / Navier Strokes Equation Solver (SENSE2D-SENSE3D) that is developed by

TUBITAK-SAGE by implementing MG capability. In this context, a sufficient number of solutions can be obtained to generate a safe aerodynamic database for the projects.

1.3 Scope of the Research

The first objective of the thesis is to develop an automated grid coarsening technique suitable for cell-centered hybrid/unstructured grids. The second objective is to modify the baseline flow solvers, SENSE2D-SENSE3D, by making flux calculation along the edge/face loops rather cells without changing cell-centered scheme. Doing so, edge/face based solution algorithm can easily accommodate complex cell structures with large number of edges in coarse MG levels. Final objective is to implement the MG routines, flow variable and residual transfer operators and cycling strategies to the baseline solver to finalize the MG adaptation.

1.4 Organization of Thesis

The thesis comprises six chapters. In Chapter 1, the idea, background and basics of MG strategy with linear and nonlinear correction schemes, intergrid transfer operators, cycling strategies are presented.

In Chapter 2, the overview of mesh construction techniques are summarized with advantages and disadvantages and the agglomeration method is chosen for utilization in this thesis. For agglomeration method, the algorithms for point selection are investigated. Then, quadtree / octree data structure approach and a new grid coarsening method, which is based on representation of the grid cells in a quadtree/octree hierarchical data structure, are presented for 2 and 3 dimensions with algorithms and flowcharts.

In Chapter 3, Euler Navier Stokes Equation flow solvers, SENSE2D and SENSE3D are explained as the baseline solver. Then, the modifications and validations on baseline solver about flux calculation and MG adaptation are presented. Finally, all MG algorithms (FAS application, cycling adaptation and intergrid transfer operators between grids algorithms) are presented with flowcharts.

In Chapter 4, the grid coarsening strategy developed and MG adaptation on baseline flow solver are investigated about performance on inviscid, laminar and turbulent flow solution in 2D and 3D cases. At the test cases, the coarsening applications are presented to show the success of newly generated grid coarsening technique on variety of grid types. Then, the effects of MG variables on MG convergence acceleration are investigated. Discussion on the results of computations and the performance of the developed solver are also stated in this chapter.

In Chapter 5, the conclusions emerging from the present work are discussed. We first discussed the efficiency and mesh dependency of the automated grid coarsening method based on quadtree and octree data structure hierarchy. Then we evaluated the acceleration of baseline code on inviscid, laminar and turbulent flow solutions with implementing of MG strategy. Finally, some future work recommendations are made.

CHAPTER 2

GRID COARSENING

In this chapter, firstly, the reason for choosing the agglomeration coarsening method in this thesis is explained and the studies in the literature about the point removal algorithms used in this method are summarized. Then a new automated grid coarsening technique suitable for cell-centered based hybrid/unstructured grid developed in this study is presented.

2.1 Overview of Agglomeration Coarsening Method

The technical specifications of six main and the most popular approaches of grid coarsening are summarized in literature survey presented in Section 1.1.4. All coarsening techniques are compared according to automation capability, implementation simplicity, nested property, time efficiency, accuracy and geometric conservation in Table 2.1. Since it is a widely used method due to being fully nested, easily automated, no geometry loss and high solution accuracy; the agglomeration coarsening approach is the most powerful technique. In an agglomeration method, grid cells are fused together to form a smaller set of larger polygonal (or polyhedral in three dimensions) control volumes. For this reason, it satisfies the requirement of this study and is decided to implement to baseline cell-centered code in this thesis.

Table 2.1 Advantages and disadvantages of coarsening methods approach

Methods	Advantages	Disadvantages
Cell Subdivision	• Fully nested • Simply transferred by injection • Transferred by simply averaging • Volume weighted restriction is easily achieved • Easily automated	• Tight coupling between the grid generation and the multigrid solution • Geometry loss • Poor quality in fine meshes • Solution accuracy loss
Overset	• Flexible and can be used with any grid generation technique • Mesh optimization for each level • Applicable with a pre-specified fine mesh • Transferred by linear interpolation	• More CPU time is required • Hard construction of the inter-grid transfer operators • To reduce the CPU time, graph-traversal type algorithms are generally used for vertex-based solver • All intersecting cell areas, addresses of coarse and fine grid cells should be determined for cell based solver • Non automated
Re-triangulation	• Easily automated • Vertex-nested meshes	• The point-removal procedure is important • Poor robustness of the coarse grid • Geometry loss • Triangulation of a coarse point-set about a complex geometry is very difficult and complex

Table 2.1 Advantages and disadvantages of coarsening methods approach (continued)

Methods	Advantages	Disadvantages
Agglomeration	• Fully nested • Volume weighted restriction is easily achieved • Easily automated • No geometry loss • High solution accuracy	• The point-removal procedure is important • Linear interpolation prolongation operators are not easily constructed
Adaptive	• No inter-grid transfer algorithms • No geometry loss • High solution accuracy • Automated	• Hard to implement • Difficult to automate • The data structures are quite complicated • Both coarsening and point insertion methods used
Algebraic	• No geometric requirement • Automated • The solution is efficient	• Complexity • Considerably complex coarse grid operators • Involves a trade-off between accuracy of the operators and complexity of the coarse grids

For structured or Cartesian grids, a coarse grid can easily be derived from a given fine grid by omitting every other point in each coordinate direction. A recursive application of this procedure results in a sequence of coarse grids. The main difficulty of agglomeration approach with unstructured MG methods is the selection of the cells to be agglomerated (or sometimes called point removal) so that the new cells formed can acceptable aspect ratios.

The different algorithms are presented about selection of the cells (or sometimes called point removal) to be agglomerated in the literature. More popular point removal procedure which is global coarsening algorithm is “Greedy type Frontal Algorithm” by Mavriplis. The algorithm starts by selecting a starting vertex and listing all of its neighbors. It continues by choosing a suitable control volume from the list so that the aspect ratio of coarse cell is maximized. After that, the list of the new cell information is updated and all vertices have been agglomerated until whole domain is processed [34]. About this subject, many published studies can be found. In the following part, agglomeration coarsening studies or point removal procedures is summarized. Francescatto and Dervieux [55] propose a directional semi-coarsening strategy based on Poisson’s equation using directional coarsening in a structured grid domain. Their algorithm relies on two mechanisms. Firstly, the ‘local metrics’ are identified, i.e. the stretching direction and strength. Secondly, agglomeration is adapted to local metrics. In order to build the local metrics, algebraic MG idea is inspired. Ollivier-Gooch [56] presented a new approach to the generation of coarse triangular and tetrahedral meshes that always produce a valid coarse mesh at each level, regardless of the fine mesh input and the number of coarse meshes generated. In this algorithm, an apex represents a boundary vertex at which a sharp corner is formed and it is always included in the coarse mesh. A fold is a line on the surface of a three-dimensional object where the surface normal is discontinuous such as the trailing edge of a wing. For isotropic surface meshes, every second fold vertex is retained. All fold vertices are retained, and every second vertex along closely-spaced lines leaving the fold is also retained. A maximal

independent set (MIS) of the remaining surface vertices is included in the coarse mesh. Pseudo-structured interior mesh fragments are coarsened in much the same way as the pseudo-structured surface mesh fragments. Finally, an MIS of the remaining interior vertices is selected for inclusion in the coarse mesh. Ahlawat, Johnson and Vanka [57] used a vertex based agglomeration algorithm that agglomerates cells around a vertex. A vertex front moves inward from boundaries. The algorithm parameterizes the cell fusion rate, which is the number of fine grid cells that get fused into a coarse cell. It is equipped with automatic detection of stretched grids, where it performs directional agglomeration and it has an algorithm for improving grid quality by filling up sharp folds (hills and valleys) on coarse cells. Waltz and Löhner [23] generated an algorithm, termed Dynamic Graph Reduction with Swapping (DGRS), which is able to produce nested coarse grids suitable for unstructured MG applications. The grid coarsening procedure can be broken down into two basic parts: point selection and element reconnection. In the point selection algorithm, the vertices take a hierarchy according to boundaries while in the element reconnection algorithm, the dynamic reconnection, i.e. reconnection during the point marking procedure, is found to be an efficient approach. Chan and Zikatanov [58].considered a new and rather simple technique for defining nested coarse spaces and the corresponding interpolation operators based on the graph-theoretical approach The goal is to construct a coarse grid using only the combinatorial (not the geometrical properties) of the graph of the underlying fine grid. This coarse grid is formed by groups of elements and called agglomerated macro elements. Okamoto, Nakahashi, Obayashi [59] propose a new agglomeration algorithm to generate coarse grids for MG methods on unstructured and hybrid grids. The algorithm, which is called a global coarsening algorithm, is based on the edge coloring of the grids so that it can agglomerate any type of grids. The edges marked according to the aspect ratio, which is calculated by using the control volume comprising the two control volumes sharing each edge. Two control volumes sharing the edge that give maximum aspect ratio are agglomerated into a coarse control volume.

2.2 The New Grid Coarsening Method Based on Quadtree / Octree Data Structure

In this study, a new automated grid coarsening technique suitable for cell-centered based hybrid/unstructured grid is developed for the MG implementation. The aim is to group the finest mesh cells such a way that new generated levels and corresponding cells have good aspect ratio. The new grid coarsening technique relies on the agglomeration of hybrid/unstructured cells based on their distribution on a quadtree and octree data structure for 2D and 3D applications, respectively. This agglomeration strategy or point removal algorithm can be defined as globally coarsening method by merging cells according to parent quadrant/octant or sub-groups.

2.2.1 Overview of Quadtree / Octree Approach

Hierarchical data structures are becoming increasingly important representation techniques in the area of computer graphics, image processing, computational geometry, geographic information systems, and robotics. They are based on the principle of recursive decomposition method. One such data structure is the quadtree or octree [60]. The term quadtree or octree is used to describe a class of hierarchical data structures whose common property is that they are based on the principle of recursive decomposition of space. The most investigated quadtree / octree approach for region representation is based on the successive subdivision of the image array into four equal-sized quadrants / eight equal-sized octants. If the array does not consist entirely of 1's or entirely of 0's (i.e., the region does not cover the entire array), it is then subdivided into child quadrants or octants until blocks are obtained (possibly single pixels) that consist entirely of 1's or entirely of 0's; that is, each block is entirely contained in the region or entirely disjoint from it.

Thus the region quadtree can be characterized as a variable resolution data structure. For example, consider the region shown in Figure 2.1-a, which is represented by the 2^3 by 2^3 binary array in Figure 2.1-b. Observe that the 1's correspond to picture elements (termed pixels) that are in the region and the 0's correspond to picture elements that are outside the region. The resulting blocks for the array of Figure 2.1-b are shown in Figure 2.1-c. This process is represented by a tree of degree 4 (i.e., each nonleaf node has four children). The root node corresponds to the entire array. Each child of a node represents a quadrant (labeled in order NW, NE, SW, SE) of the region is represented by that node. The leaf nodes of the tree correspond to those blocks for which no further subdivision is necessary. A leaf node is said to be BLACK or WHITE, depending on whether its corresponding block is entirely inside or entirely outside the represented region. All nonleaf nodes are said to be GRAY. The quadtree representation for Figure 2.1-c is shown in Figure 2.1-d.

The octants and octal tree can be also characterized as given in Figure 2.2. It is observed that this process is represented by a tree of degree 8 (i.e., each nonleaf node has eight children).

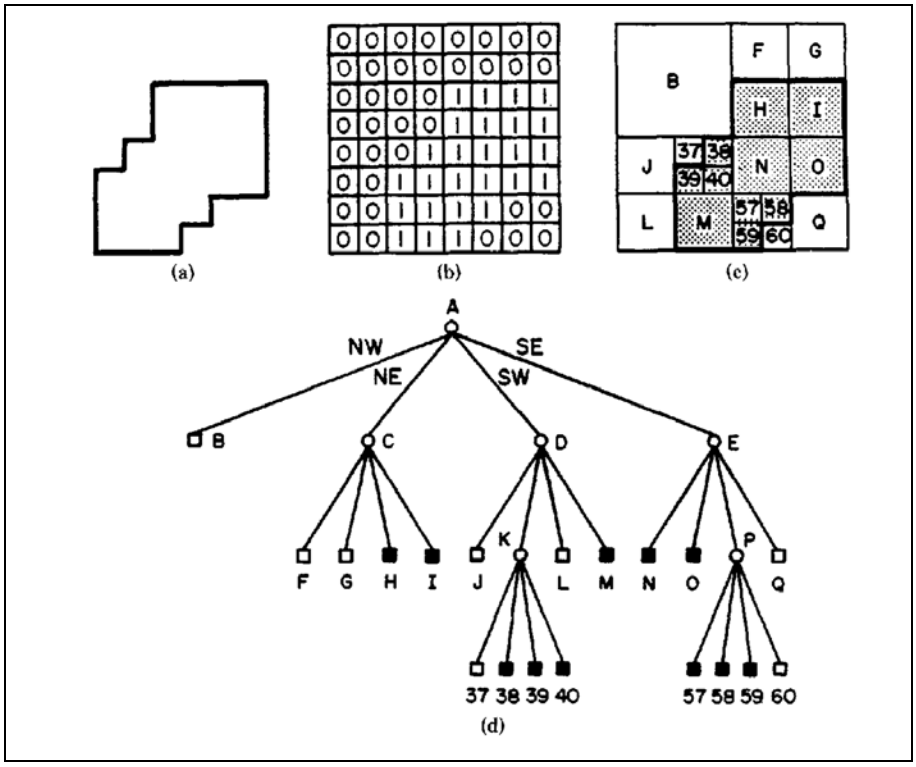


Figure 2.1 A region, its binary array, its maximum block and the corresponding quadtree [60]

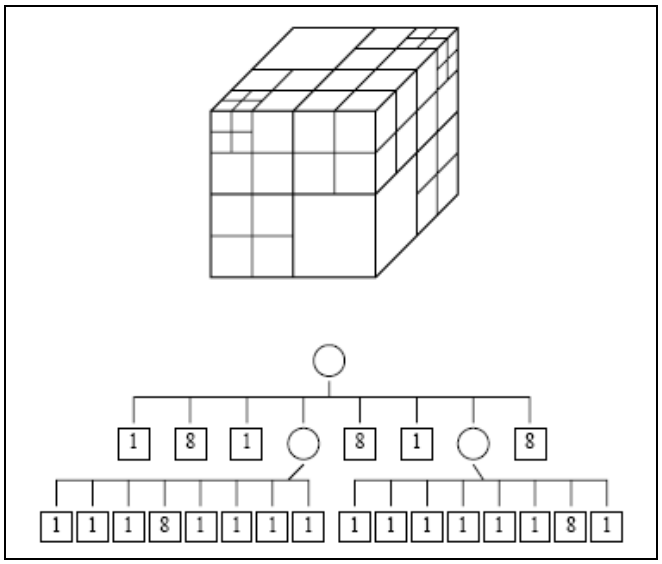


Figure 2.2 Cells octree example [60]

2.2.2 The Algorithms of Grid Coarsening Method Based on Quadtree / Octree Data Structure

The grid coarsening algorithm can be divided in two steps. The first step is to form the quadtree or octree data structure hierarchy of the input grid file. Then the grouped fine meshes are agglomerated to create the coarse grid mesh. To obtain the higher coarse grid level, the parent/child structure, which are formed during the data structure hierarchy, is used.

2.2.2.1 Forming Quadtree / Octree Data Structure Algorithm

The quadrant or octant cells are created as imaginary cells over the cell domain such a way that each quadrant covers maximum of four; each octant covers maximum of eight cell center points. In other words, the quadtree / octree approaches are used for grouping the finest mesh cells so that new generated levels and corresponding cells have good aspect ratio. A sample quadtree structure is presented in Figure 2.3.

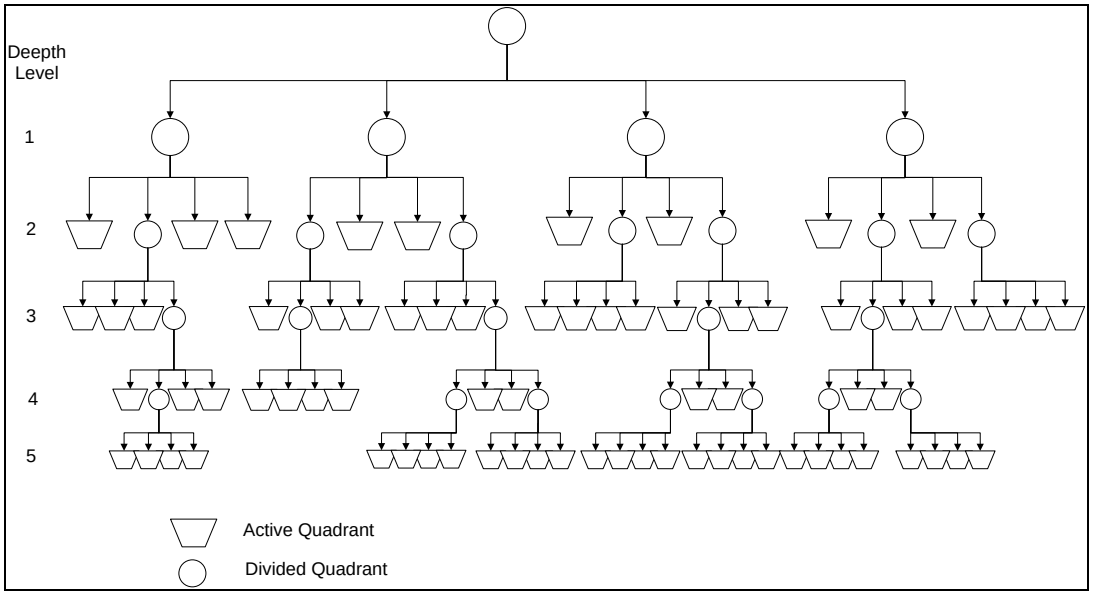


Figure 2.3 A sample quadtree structure

The flowchart of quadtree / octree data structure algorithm (subroutine FORMQUADTREE or FORMOCTREE) is presented in Figure 2.4 and formation the data structure algorithm for coarsening is described as follows:

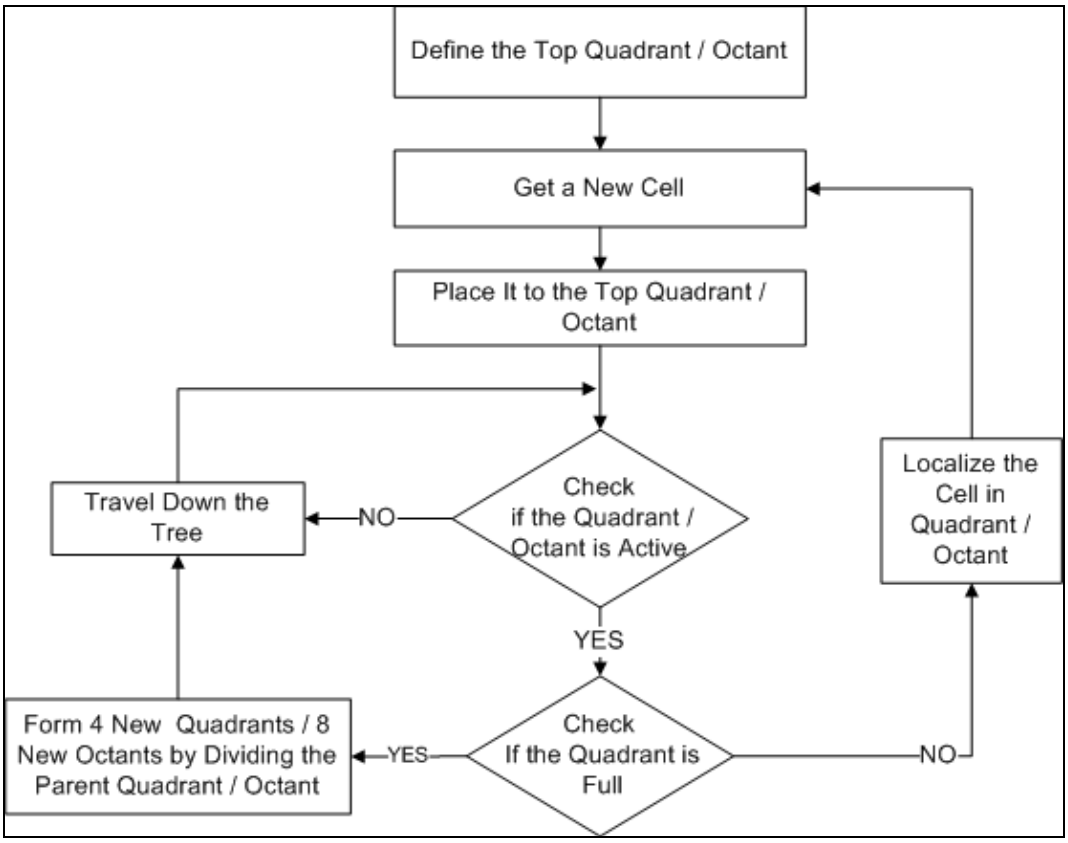


Figure 2.4 The flowchart of forming quadtree/octree

1. The limits of the domain are found, the maximum length of domain and cell center nodes are defined. The schematic view of the cell centered nodes around an ellipse in 2D domain is presented in Figure 2.5.

2. The edge length of the largest quadrant/octant is defined as the maximum length of domain.

3. One cell center node point is selected. Starting from the largest quadrant / octant, the appropriate quadrant / octant where the cell center is located is found. The schematic view of quadrants and the cell center node distribution on quadrants is given in Figure 2.6.

a) If the quadrant/octant is deactive (has divided already) (subroutine GODOWN);

i. The quadrant / octant is divided into 4 or 8 (North-east, north-west, south-east, south-west and upper and lower position in 3D)

ii. The child quadrant/octant is found according to position of the cell center node in processing

iii. The limits of the domain are changed according to the position of the cell center point

iv. Returns the decision without changing cell-center node in processing

b) If quadrant/octant is full (subroutine NEWQUAD);

i. The quadrants/octants are deactiveated.

ii. The new quadrants/octants are generated.

iii. The cell centers, which are allocated before according to position of cell center location, are distributed (subroutine FINDPOS)

iv. Subroutine GODOWN is called with the cell center node in process

c) If quadrant/octant is active (the number of cell center of quadrant is less than 4 or octant is less than 8);

i. The number of cell center nodes is increased by one.

ii. The cell-center node in process is allocated to the quadrant or octant.

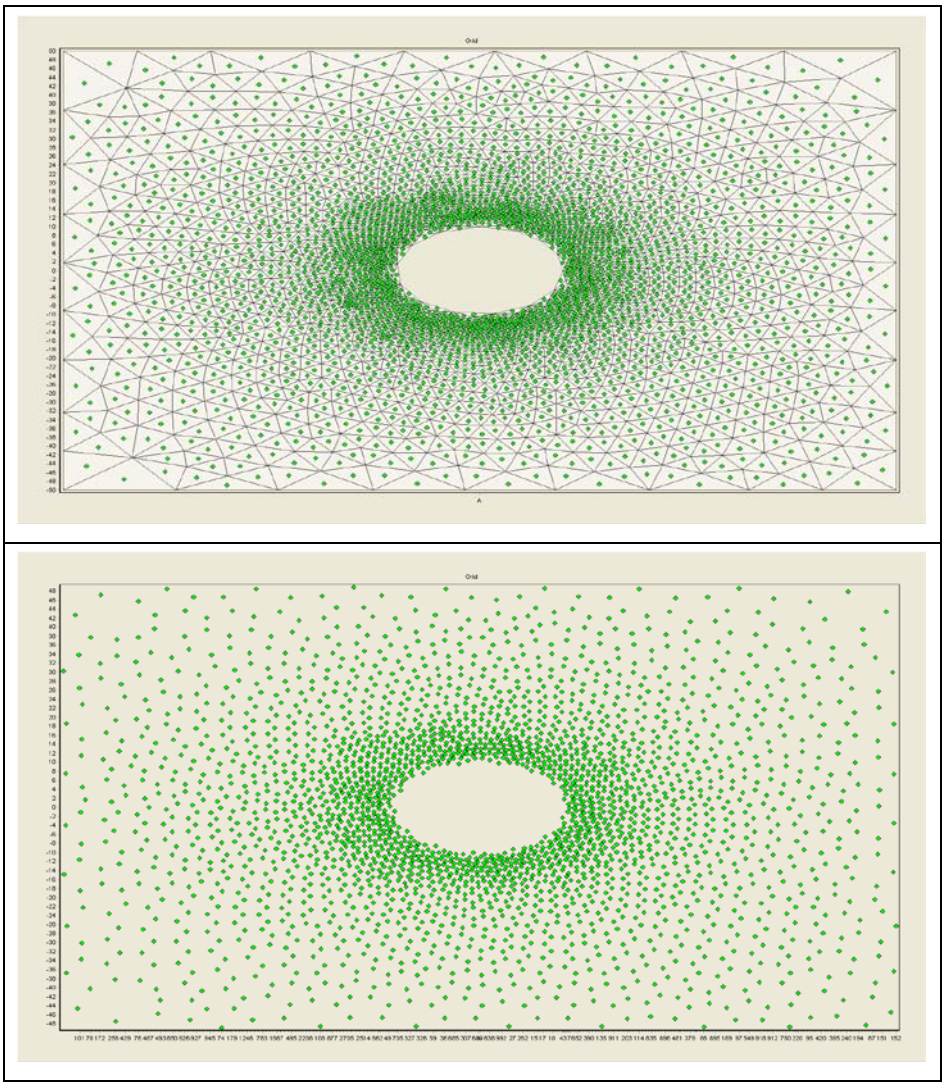


Figure 2.5 The schematic views of the cell center nodes

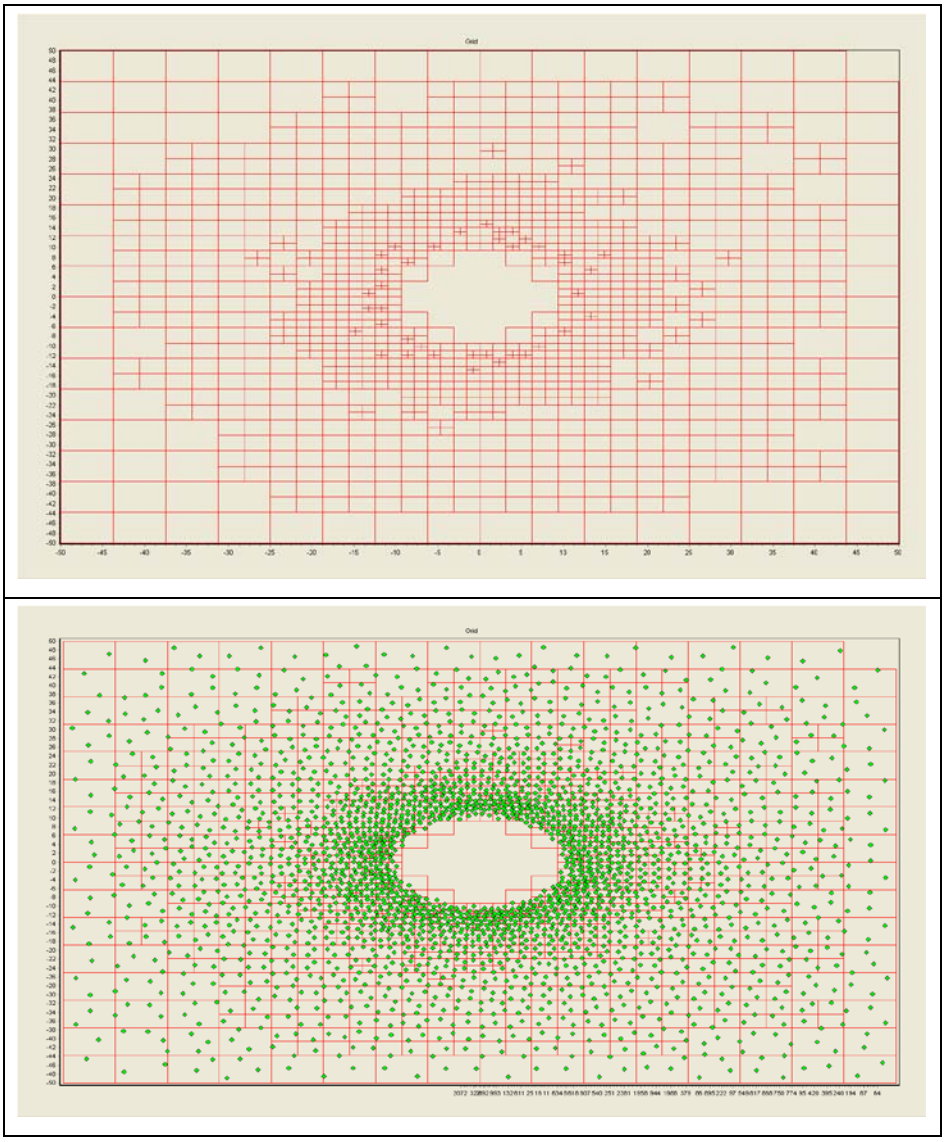


Figure 2.6 Schematic view of quadrants and the distribution of cell center nodes on quadrants

2.2.2.2 Agglomeration Algorithm

During grouping of the finest mesh cells according to quadtree / octree data structure, the quadrant / octant deepness level and parent-child relationship are also stored for creating coarse grid levels. The second coarse level is generated by

agglomerating the cells belongs the same active quadrant / octant which is located at the end of the tip of the tree. The higher coarse grid levels are generated according to deepness level of quadrant. The child cells of the quadrant which is stated at defined deepness level are agglomerated and the coarse meshes are created for the coarse level grids.

The flowchart of coarsening algorithm (subroutine COARSENING) is presented in Figure 2.7 and the coarsening algorithm according to parent/child hierarchy can be described as follows:

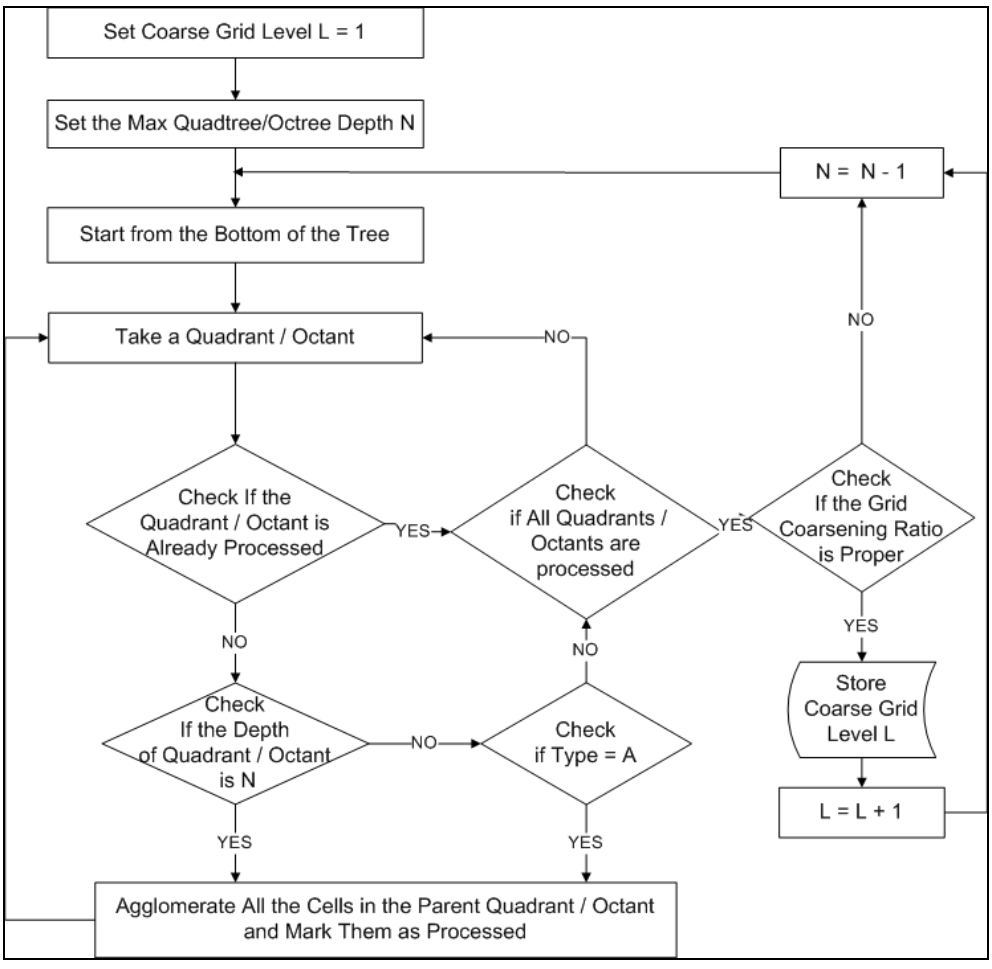


Figure 2.7 The flowchart for the coarsening algorithm

1. The maximum / minimum deepness level of active quadrants is found and the quadrants are ordered from the highest to lowest deepness level.

2. The delta of deepness level from the tip of the branch according to coarse grid level is defined.

3. One quadrant/octant is considered. If this quadrant/octant is active and is not proceedede before, it is evaluated. (subroutine CELLGROUP);

a) If delta of deepness level is 0;

   i. The quadrant/octant which is in process is indicated by a flag.

   ii. The quadrant/octant is dropped from the quadrant/octant pool.

   iii. The active or deactive quadrant/octants are checked. If quadrant/octant is active, the child quadrants/octants are dropped to the pool and the original quadrant/octant are erased until all quadrants are active. Then the cells from all active quadrants/octants are collected. (subroutine COLLECT)

b) If delta of deepness level is not 0;

   i. The parent of the quadrant is found according to delta of the deepness level

   ii. The parent quadrant/octant is checked whether it is preceded before or not. If it is already proceed, the flag is put to show that quadrant/octant is not in process.

iii. If the parent is not proceeded, the deepness level of childs belongs to the parent quadrant/octant is checked. If the deepness level of all children higher that the quadrants that is in process, the delta deepness level is decreased and the process is repeated from finding the parent. If the deepness level of all childs is equal or less then the quadrants that is in process, COLLECT subroutine is repeated.

4. If the quadrant/octant is in process according to flag, the cells are designated (subroutine DESIGNATION) with taking the first cell. The cells that are grouped is checked whether that is irregular (no common edge / face with the other cells) or not.

a) If the cell is irregular, the sub-groups of cells which are regular inside are created (subroutine REARRANGE). Then DESIGNATION process is repeated the with the sub-groups.

b) If the cell is not regular, the common edges/faces are erased and the left and right cell number of active edges/faces are defined (Subroutine ARRANGE).

5. The ratio between the size of candidate coarse grid level and the previous (stored) coarse grid level is checked.

a) If the ratio is less than the desired ratio, the candidate coarse grid level is accepted and, the coarse mesh properties are calculated and stored (subroutine POST).

b) If ratio is greater than the desired ratio, the delta of deepness level is increased and CELLGROUP subroutine is repeated.

6. If the coarse grid level is prepared, the coarsening level and the delta of deepness level are increased, and CELLGROUP subroutine is repeated.

To demonstrate the octree data structure, an unstructured grid over the cube is prepared. The volume and surface meshes are shown in Figure 2.8. This grid consists of 17,047 nodes and 97,451 cells. 1,660 cells lie on the cube surface. The four agglomerated coarse grid levels are derived by octree based agglomeration coarsening algorithm and they contain 37,322 – 14,374 – 4,676 and 647 cells from second to fifth grid levels, respectively.

The volume cell bounds on the faces of cube are demonstrated (red color) at each coarse level in Figure 2.9. It is obviously seen that, the octree data structure is formed and the octree based coarser grids have characteristics similar to Cartesian grids with good aspect ratios. The cubic volume becomes larger at higher levels of coarsening similar to Cartesian volume meshes.

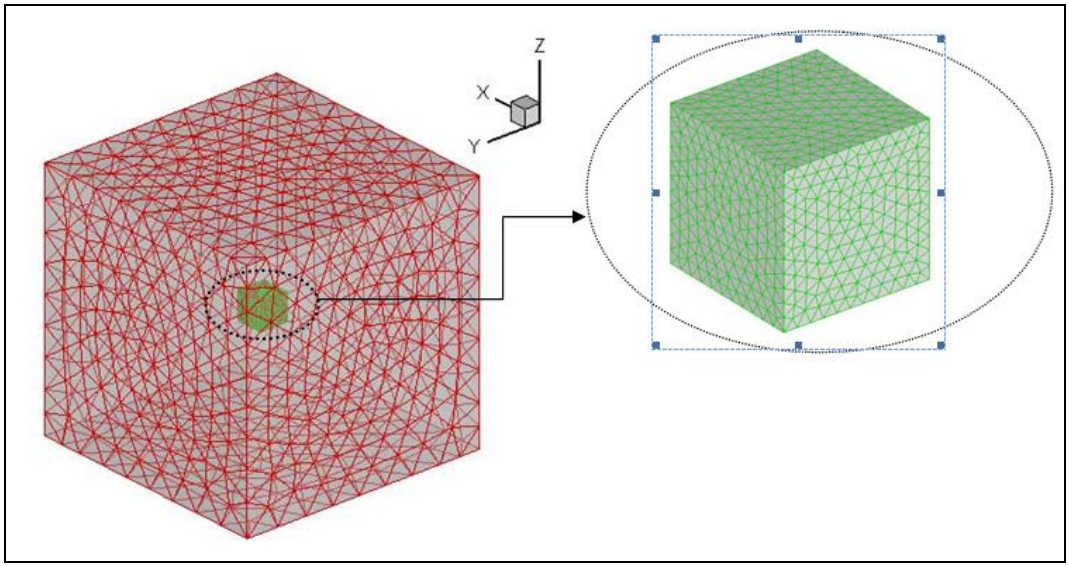


Figure 2.8 Volume and surface meshes around cube

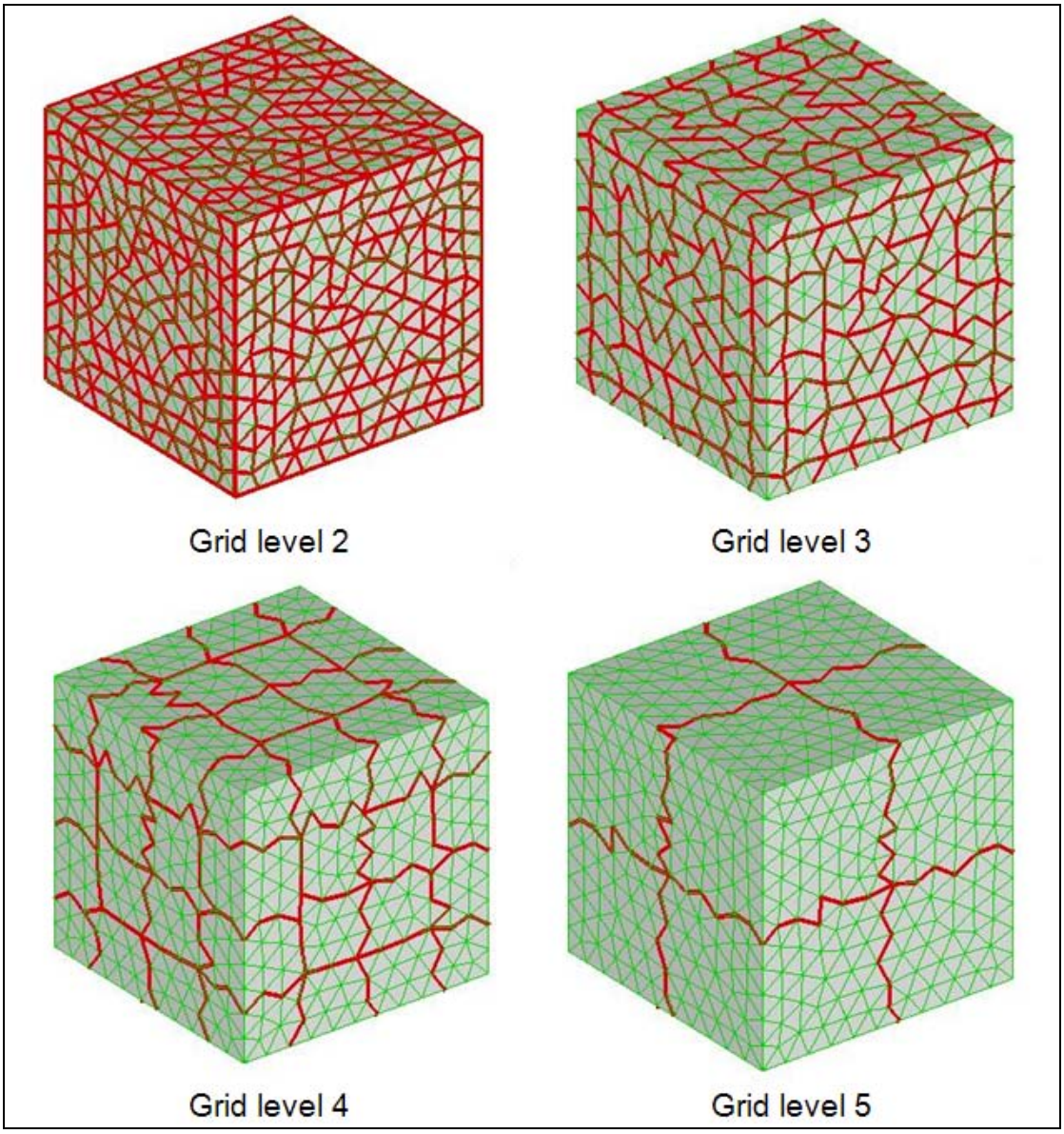


Figure 2.9 The coarse level grids on surface of cube.

CHAPTER 3

MULTIGRID IMPLEMENTATION

In this chapter, two dimensional and three dimensional Euler/Navier Stokes Equation flow solvers, SENSE2D and SENSE3D developed by TUBITAK-SAGE are explained as the baseline solver. After briefing the properties of solver, the modifications for the flux calculation and MG adaptation are presented with validation of modification. In the last section, the algorithms of all MG routines (FAS application, cycling adaptation and transfer operators between grids algorithms) with their flowcharts are explained.

3.1 The Baseline Euler / Navier Stokes Equation Flow Solver

The viscous flow solver, SAGE Euler / Navier Stokes Equation Solver (SENSE) which is a computational fluid dynamics solver developed by TÜBİTAK-SAGE, is taken as the baseline flow solver in this thesis. It is a hybrid / unstructured finite volume method (FVM) solver. Flow variables are stored at cell centers and second order Roe's upwind flux computations are employed. The time dependent equations are solved explicitly using the third order Runge-Kutta method with variable time-stepping. The methods used at the two dimensional version called SENSE2D and three dimensional version called SENSE3D are summarized in the following sections [13].

3.1.1 Numerical Discretization Technique

SENSE2D/3D solver is formulated by using Finite Volume Method (FVM) that is based on the physical concept of using macroscopic control volumes to numerically solve the conservation laws of fluid motion. The use of integral form of the governing equations is the basis of FVM. The direct discretization of the conservation laws in integral form ensures that the mass, momentum and energy are conserved over discrete control volumes. FVM takes full advantage of an arbitrary mesh, where a large number of alternatives are available for the definition of the control volumes for conservation laws. Its success is based not only on its relative simplicity as compared to Finite Difference Method (FDM) and Finite Element Method (FEM) approximations, but also on its flexibility and ability to unite ideas from FEM with those from FDM.

There are mainly two approaches for the approximation of mass, momentum, energy fluxes over the surface of control volumes in computational domain: cell vertex and cell centered schemes.

In the cell centered formulation which is used in SENSE2D and SENSE3D solvers, the flow properties are directly calculated at the center of the computational cell which itself is the control volume for finite volume discretization, Figure 3.1. This eliminates the need for the control volume generation affords. The most important disadvantages of the cell centered formulation is the requirement for finite element approximations to distribute the variables to the nodes, which may bring additional numerical errors to the results.

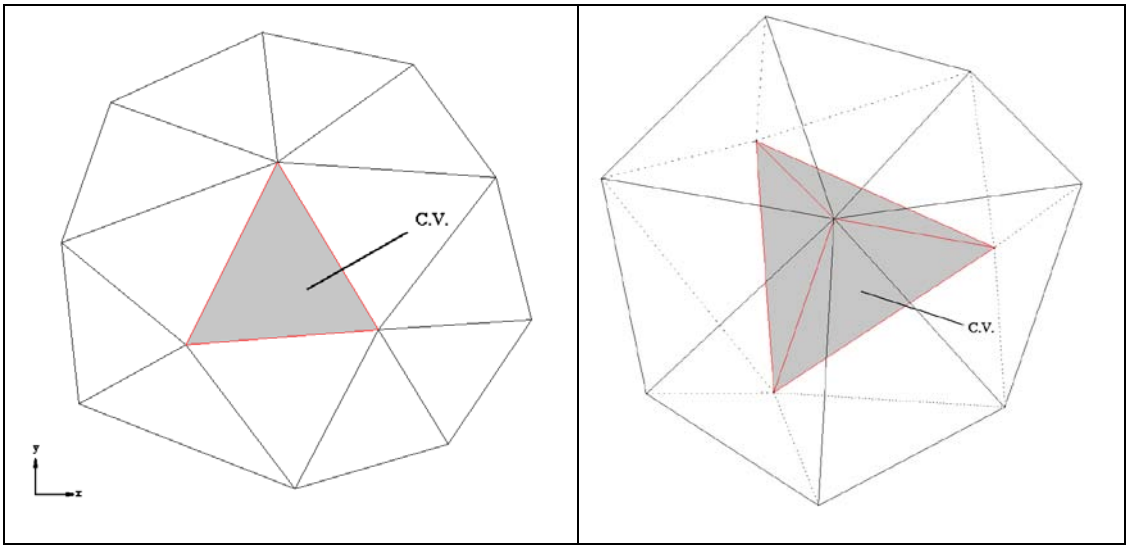


Figure 3.1 2-D and 3-D cell centered median dual cells

3.1.2 Numerical Scheme

SENSE2D/3D solver is based on upwind differencing which utilizes the propagation of information within a mesh in accordance with the theory of characteristics in constructing type-dependent differencing for components of the information traveling in opposite directions in a separate and stable manner. There is no need of scalar artificial dissipation formulas in upwind methods which are necessary for second order central schemes to damp odd-even oscillations generated especially in the vicinity of discontinuities. By using high order upwind methods, shocks and expansion waves that are observed in high speed compressible flows can be detected in a very sensitive and accurate manner. Although this approach is more difficult than central differencing in computational sense, it brings the advantages of being more robust, having high convergence speed and requiring less user interaction.

In the flux-difference splitting schemes, Riemann problem on the cell faces are solved locally. The conservative variables are taken as piecewise constant over the

cells at each time step and time evolution is obtained by the solution of Riemann problem at the cell faces. By this way, exact contributions of local Euler equations are introduced to the numerical schemes which make sense in physical point of view. Another important advantage of upwind schemes was that with the flux-difference splitting scheme of Roe, the resolution of boundary layer details typically requires only half as many points as with a central differencing code.

There exists basically two types of time stepping algorithms used both for integrating governing flow equations in time to obtain steady state solution and for unsteady applications: explicit and implicit time stepping algorithms. Although implicit algorithms offer more stable and faster results, they have the shortcoming of large amount of memory usage. Also, the implementation of implicit time stepping algorithms especially for viscous flows is quite complicated. Most commonly used method of time discretization technique which is explicit in nature and of a high order of accuracy is the Runge-Kutta method. It achieves the accuracy of a Taylor series approach without any need for the evaluation of higher order derivatives. Explicit Runge-Kutta method is among the oldest and best-understood schemes in the numerical analysis. The simplicity of explicit Runge-Kutta formula lies in its self-contained, one-step nature.

3.1.3 Computational Grid

In order to eliminate the difficulty in generating high quality unstructured viscous meshes within the boundary layer with the available grid generation tools, hybrid grids can be used. Hybrid grids offer usage of structured high quality grids in the vicinity of boundaries and usage of unstructured grids where dense mesh is not required. By so, it is possible to obtain a computational mesh which is dense enough to observe the boundary layer and which is small in terms of number of elements, i.e. less memory usage and higher computational convergence rate. In Figure 3.2,

high quality viscous structured mesh near a wall boundary with smoothly growing unstructured mesh up to the far field is presented.

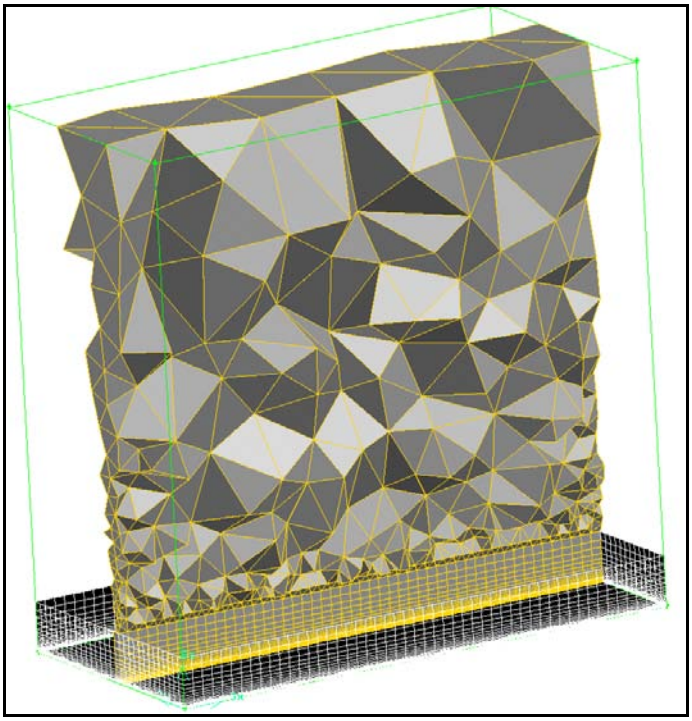


Figure 3.2 Hybrid/unstructured grid

3.2 Baseline Solver Modifications

The theory behind the baseline solver is explained in Section 3.1. For easily adaptation of MG application, it is decided to change SENSE solver flux calculation algorithm by changing the cell based loops to edge/face based loops without changing the cell centered scheme. By doing so, due to the nested coarse grid levels, which are obtained by using the agglomeration coarsening method, the information can be easily transferred between grid levels. The second advantage is that, following the edges/faces instead of cells needs less memory and it is time

consuming by making calculation once at each edges/faces instead of two from both sides of cells. For 2D and 3D applications, all routines are updated according to edge/face loop and cell information is supplied by keeping only left and right neighbor cells of each edge/face.

The second modification on the baseline solver is implementation of coarsening and MG routines. In this modification, required variables are updated to carry the information with the level variable.

During these modification studies, the following changes are made in both 2D and 3D solvers:

- The input file is modified to render more user friendly (An input data file format is updated and a sample file is given in APPENDIX A).
- New default grid data file named grid_level_1.dat is added to obtain the parallelism with the coarse grid levels about edge/face numbering.
- The boundary condition code number is revised to put in a sequence with negative numbering.
- The output information file is prepared for configuration management (A sample output info file format is given in APPENDIX B).
- The residual calculation for density, x velocity, y velocity, energy and turbulent viscosity are added.
- The mesh conversion algorithm is implemented from the generic format (*.neu) to default formats is implemented.
- A mesh connectivity output file is prepared to speed up the code when same grid is used for different flow parameters

3.2.1 Modifications about Flux Calculation

The executable that has been named as “MASTER” is first updated by changing the cell based loops to edge/face based loops. MASTER performs the following jobs in sequence. The flowchart of the solver is also presented in Figure 3.3.

1. The size of the computational mesh is read and the memories to the arrays are allocated (subroutine GETSIZE).

2. The necessary input file which consists of properties about the solver, flow, grid, multigrid, parameters and files for initialization and finally iteration informations is loaded (subroutine CONFIG).

3. The computational mesh is loaded (subroutine GRID):

   a) The existence of the input file is checked. If default input file (grid_level_1.dat) exists, the mesh and connectivity are read. (subroutine READGRID). If it does not exist, the generic input file is converted (mesh.neu) to grid_level_1.dat and prepares the mesh and connectivity (subroutine MESHCONVERSION).

   b) The boundary condition inputs are checked and the inputs of boundary conditions are updated.

   c) The neighbors of each of cell are found (subroutine FINDNEIGH).

   d) The numbers of each of the computational edges / faces are designates and the right and left cell numbers of each edges/faces are stored (subroutine FINDNEIGH).

4. The geometric properties of edge/faces (length/area, sinus and cosine value, midpoint location) and cell areas/volumes are calculated (subroutine SETGEOM).

5. The computational mesh with side slip angle β and angle of attack α is rotated to make the global *x-axis* coincident with the axis of trajectory if α or β is not equal 0° (subroutine TROTATE).

6. The computational domain is initialized with free stream conditions or from the result of previous calculations (subroutine INIT).

7. The wall distances of each cell of domain are calculated for the Spalart-Allmaras turbulence model (subroutine WALLDIST).

8. The flow solution is iterated for one step of iteration (subroutine STEP).

a) The pressure values from conservative variables are calculated (subroutine PRESSURE).

b) The local or global time steps for each cell are calculated at each iteration steps (subroutine CALDTL).

c) The third order Runge-Kutta time stepping algorithm is started (subroutine RK3). Each of the following three steps of this algorithm is applied to every computational cell:

i. The values of conservative variables are evaluated at cell edges / faces, i.e. defines the left and right states at cell faces (subroutine QFACE).

ii. The boundary conditions to right state of the cell edges / faces are intoduced (subroutine BC).

iii. The flux differences for edge/ face fluxes based on edge/face loop are computed (subroutine ROE).

iv. The viscous fluxes at each edge / face based on edge / face loop are computed (subroutine VISCOUS).

v. The fluxes for each cell based on edge / face loop are calculated.

vi. Implicit residual smoothing is applied (subroutine SMOOTH).

vii. The pressure values from conservative variables are calculated (subroutine PRESSURE).

d) The next third order Runge-Kutta time stepping algorithm are started for the calculation of the turbulent viscosity after one full step Runge-Kutta time stepping algorithm is finished for conservative variables (subroutine SPALART).

e) The screen output or saving frequency is checked and if required, the residual for each partition are calculated (subroutine RESCALCULATION). The aerodynamic coefficients are evaluated by integrating the pressure values over wall surfaces (subroutine LOADS).

f) The information file of the solution includes date, solver properties, flow and grid properties, MG properties, results with computational and CPU time, number of iteration or cycle, aerodynamic loads, logarithmic residuals are prepared (subroutine OUT).

9. These steps are repeated until maximum number of iteration is reached or a “stop” command comes from the “Master”.

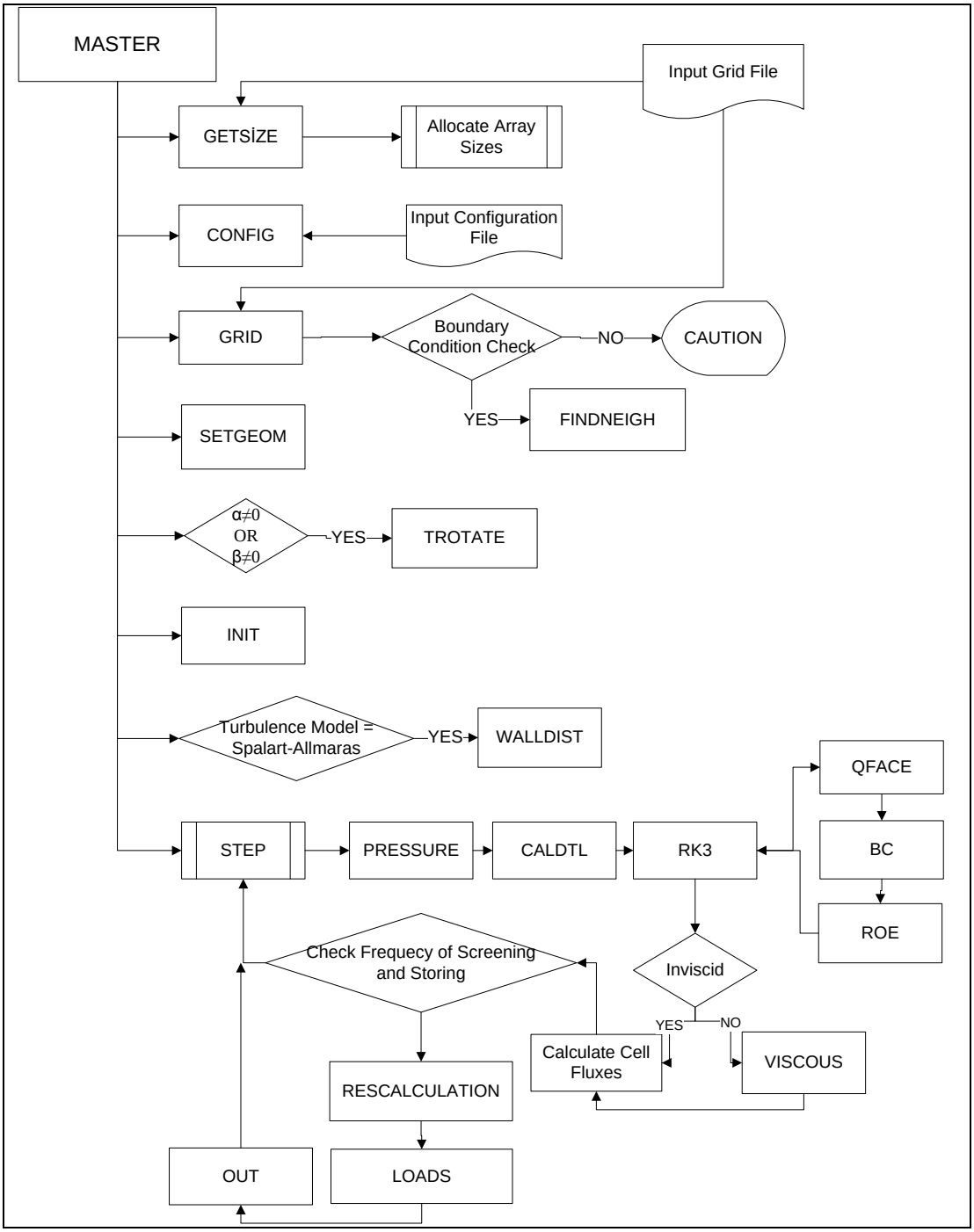


Figure 3.3 Algorithm for the baseline solver

3.2.2 Validation of Updated Baseline Solver

To validate the new method in 2D, an unstructured grid over NACA0012 airfoil with 9421 nodes and 18390 triangles generated for SENSE-2D solver as seen in Figure 3.4. Both versions, the original solver called cell based loop during flux calculation and updated solver called edge based loop during flux calculation, are solved at a Mach number of 0.1, an angle of attack of 3° and a Reynolds number of 100,000. The residual is actually represented by the flux around the boundary of control volume and is therefore related to the conservative variables. For this reason, L2norm residual (which shows the sum of flux calculation of all variables) is used for comparing the versions of the solvers. The results for the inviscid flow are obtained by using first order discretization and residual history is presented in Figure 3.5. It is seen that both 2D solvers are exactly equivalent to each other.

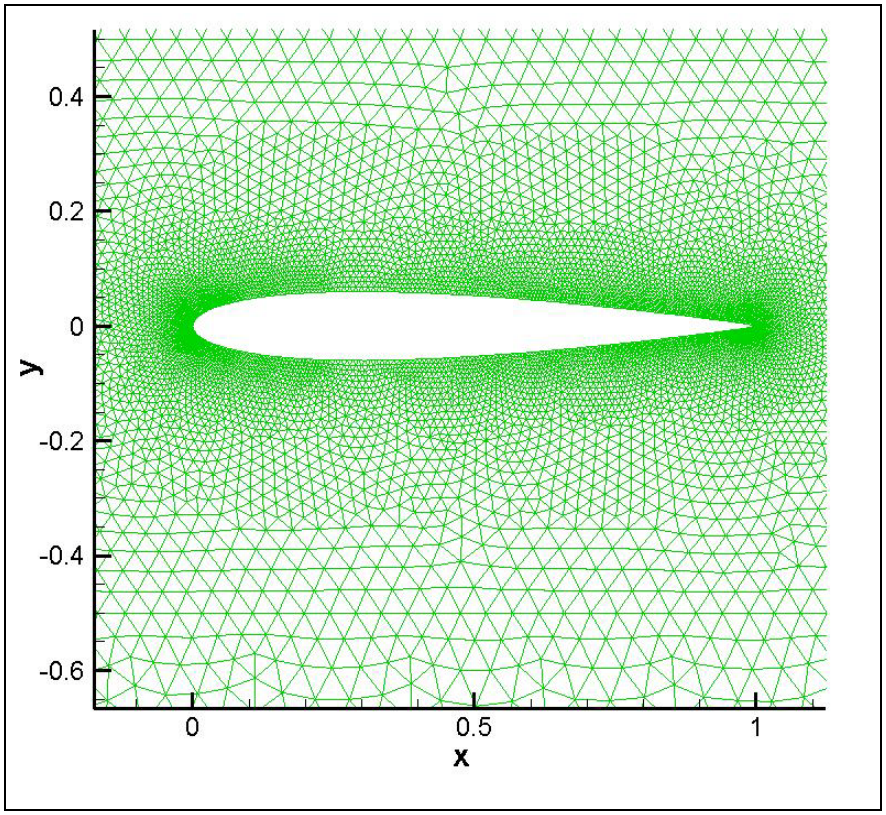


Figure 3.4 2D unstructured grid over a NACA0012 airfoil

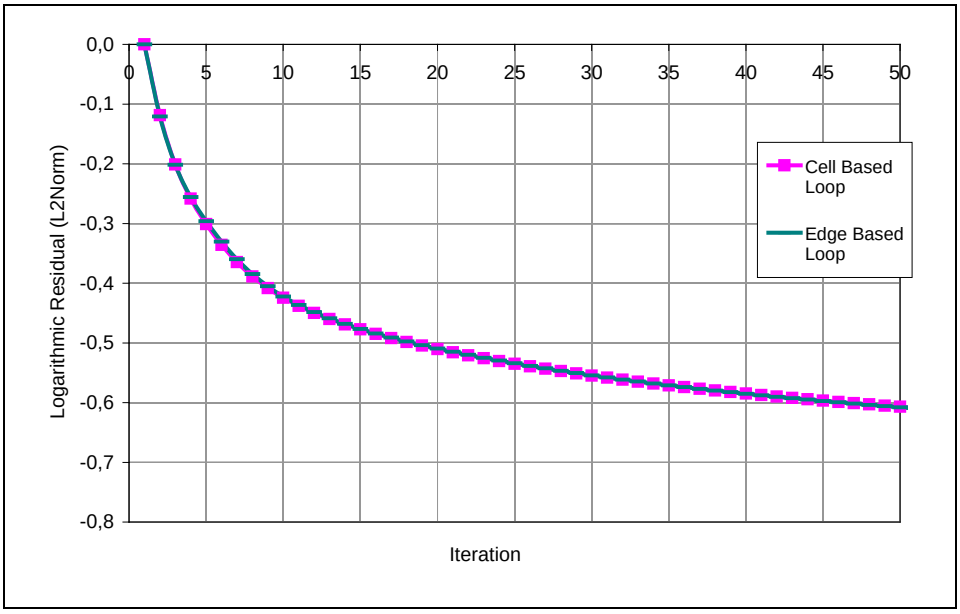


Figure 3.5 Residual versus the number of iterations for 2D validation

For SENSE-3D solver, an unstructured grid around an ONERA M6 wing (shown in Figure 3.6) having 255,156 nodes and 1,391,537 cells is used to validate the new method. Both versions, the original code called cell based loop during flux calculation and updated code called faced based loop during flux calculation, are solved for a Mach number of 0.5. The results for the inviscid flow are obtained by using first order discretization and residual history is presented in Figure 3.7. It is also seen that both 3D solvers give exactly equivalent results.

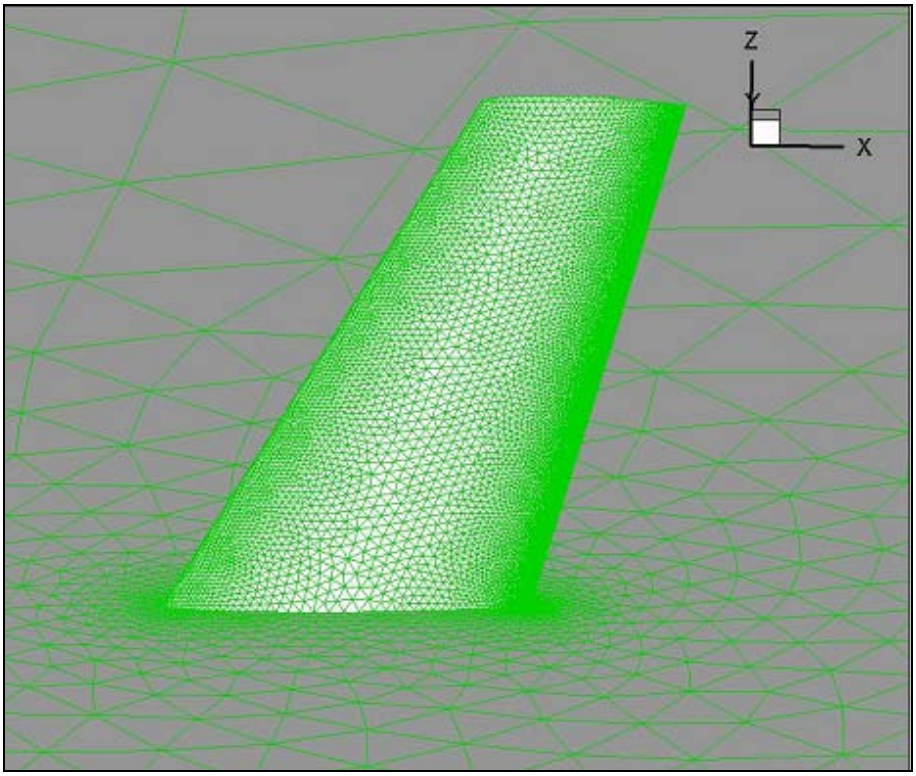


Figure 3.6 3D unstructured grid over an ONERA M6 wing

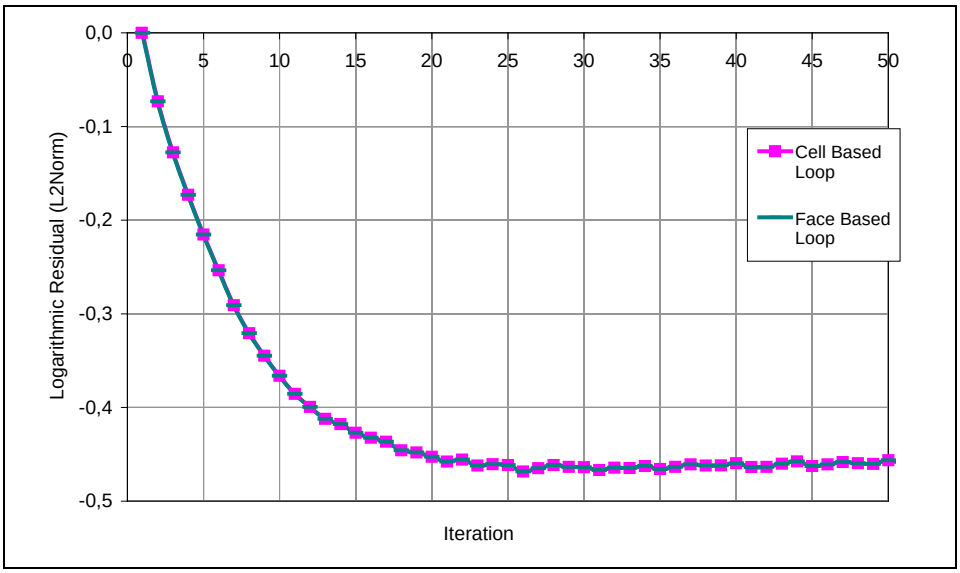


Figure 3.7 Residual versus the number of iterations for 3D validation

3.2.3 Modifications about Multigrid Adaptation

The main progam of the baseline code is modified according to MG implementation. The grid coarsening algorithms that are FORMQUADTREE for 2D and FORMOCTREE for 3D presented in Figure 2.4 and COARSENING presented in Figure 2.7 are implemented into the main program. These algorithms are coded in a compact form and called if MG is active as indicated by red color in Figure 3.8. FORMQUADTREE or FORMOCTREE subroutines are called after reading grid properties and connectivity information. COARSENING subroutine is called after calculating the geometric properties of edge/faces and cell areas/volumes. The coarse level grid properties are calculated in this subroutine and the deactive edges/faces are signed as “0” for responding coarse level. The information about the left and right cell numbers of each active edge / face is stored at each coarse level.

STEP subroutine is directly called if MG is deactive. If MG is active, MGRID subroutine which will be defined as the following section is called by the STEP subroutine with the level of coarsening information.

Finally, the other subroutines are modified to process the routines with checking that the edge/face is active or deactive. If the edge/face is deactive, the process is skipped and an active edge/face is sought. In addition, some arrays are updated to store the information with the responding coarsening level.

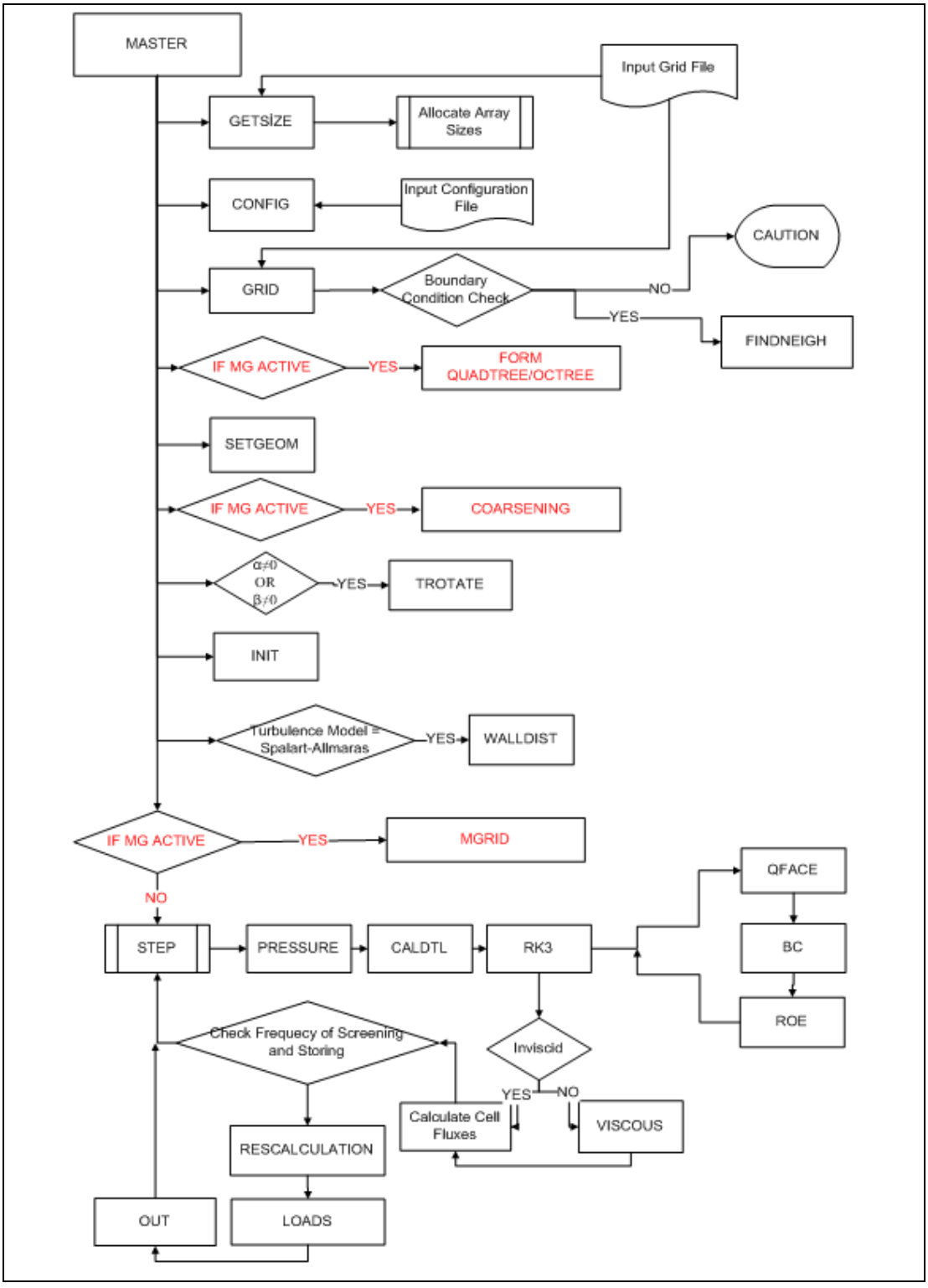


Figure 3.8 The updated algorithm for the baseline solver

3.3 Multigrid Algorithms

For MG implementation, FAS scheme for solution, cycling strategies, (sawtooth, V-cycle, W-cycle and FMG) transfer operators between grids, (restriction and prolongation) and algebraic smoothers are prepared to implement MG capability to baseline code.

3.3.1 FAS Algorithm

In the FAS scheme, all coarse grid levels are solved exactly at the same free flight conditions using the same numerical discretization methods, CFL number and boundary conditions as the fine grid solution. As an example, 3-level coarsening of FAS concept explained at Section 1.1.3.2 is applied recursively and presented in Table 3.1.

Table 3.1 Application of FAS to three grid levels

Level	1	2	3
Equation to be Solved	$L_1\bar{u}_1 = r_1$	$L_2\bar{u}_2 - f_2 = r_2$	$L_3\bar{u}_3 - f_3 = r_3$
Restricted Conservative Variable From Finer Grid	-	$\bar{u}_2^{old} = \bar{I}_1^2\bar{u}_1$	$\bar{u}_3^{old} = \bar{I}_2^3\bar{u}_2$
Restricted Residual From Finer Grid	-	$I_1^2(f_1 - r_1)$	$I_2^3(f_2 - r_2)$
Forcing Function	$f_1 = 0$	$f_2 = L_2\bar{u}_2^{old} - I_1^2 r_1$	$f_3 = L_3\bar{u}_3^{old} + I_2^3(f_2 - r_2)$
Conservative Variable After Iteration	$\bar{u}_1$	$\bar{u}_2$	$\bar{u}_3$
Residual After Iteration	r_1	r_2	r_3
Conservative Variable After Prolongation	$\bar{u}_1^{new} = \bar{u}_1 + I_2^1(\bar{u}_2^{new} - \bar{I}_1^2\bar{u}_1)$	$\bar{u}_2^{new} = \bar{u}_2 + I_3^2(\bar{u}_3 - \bar{I}_2^3\bar{u}_2)$	-

3.3.2 Cycling Algorithms

Due to the simplicity and robustness, the fixed cycling strategy having a fixed pattern of coarse and fine grids is preferred in this thesis. All common cycling patterns, V-cycle, W-cycle and FMG, are adapted to the solver.

To operate the cycling routines, a compact executable program, called “MGRID”, is prepared and adapted to baseline flow solver as shown Figure 3.3. MGRID master routine aims the arrange level of solution according to cycling strategy and uses the baseline solver subroutine for both fine and coarse level iteration. This subroutine performs as the following sequence and its flowchart is presented in Figure 3.9.

1. The cycle logic is arranged for all kind of cycling strategies that are V-cycle, W-cycle and preliminary stage of FMG. The logic of cycling is modeled as 123454321 for V-cycle defined in Figure 1.3, 1234543454321 for W-cycle defined in Figure 1.4 with each digit showing the coarsening level. 1 shows fine and 5 shows the 5th coarse grid level. FMG uses V-cycle definition for the main part defined in Figure 1.5 (subroutine CYCLEDEFINITION).

2. The preliminary stage of FMG is iterated if the desired cycling strategy is FMG (subroutine PRECYCLE):

   a) The repeat number of each V-cycles like 454, 34543 which constitute the preliminary stage are defined according to cycle definition.

   b) The coarse grid level number is defined starting from the initial V-cycle definition and iteration step or transfer operators are addressed.

   c) V-cycle that defined initially is iterated and V-cycle type is updated by adding one level of finer grid.

3. MG cycles are applied according to cycling definition (subroutine MGCYCLE):

a) The coarse grid level is defined from cycle definition and it is compared with the prelevel that is already completed the process.

b) If grid level is greater that the prelevel (towards to coarse direction),

i. The flow parameters and residuals are restricted (subroutine RESTRICTION).

ii. The defined number of iteration is made and the solution is obtained (subroutine STEP).

iii. The prelevel is defined as level and the next grid level number is chosen according to cycle definition.

c) If level is less that the prelevel (towards to fine direction),

i. The calculated errors for correction are prolonged according to next finer grid level (subroutine PROLONGATION).

ii. The prelevel is defined level and the next grid level number is chosen according to cycle definition.

4. The screen output or saving frequency is checked and if required, the residual for each partition are calculated (subroutine RESCALCULATION) and the aerodynamic coefficients are evaluated by integrating the pressure values over wall surfaces (subroutine LOADS).

5. The info file of the solution includes date, solver properties, flow and grid properties, MG properties, results with computational and CPU time,

number of iteration or cycle, aerodynamic loads, logarithmic residuals are prepared (subroutine OUT).

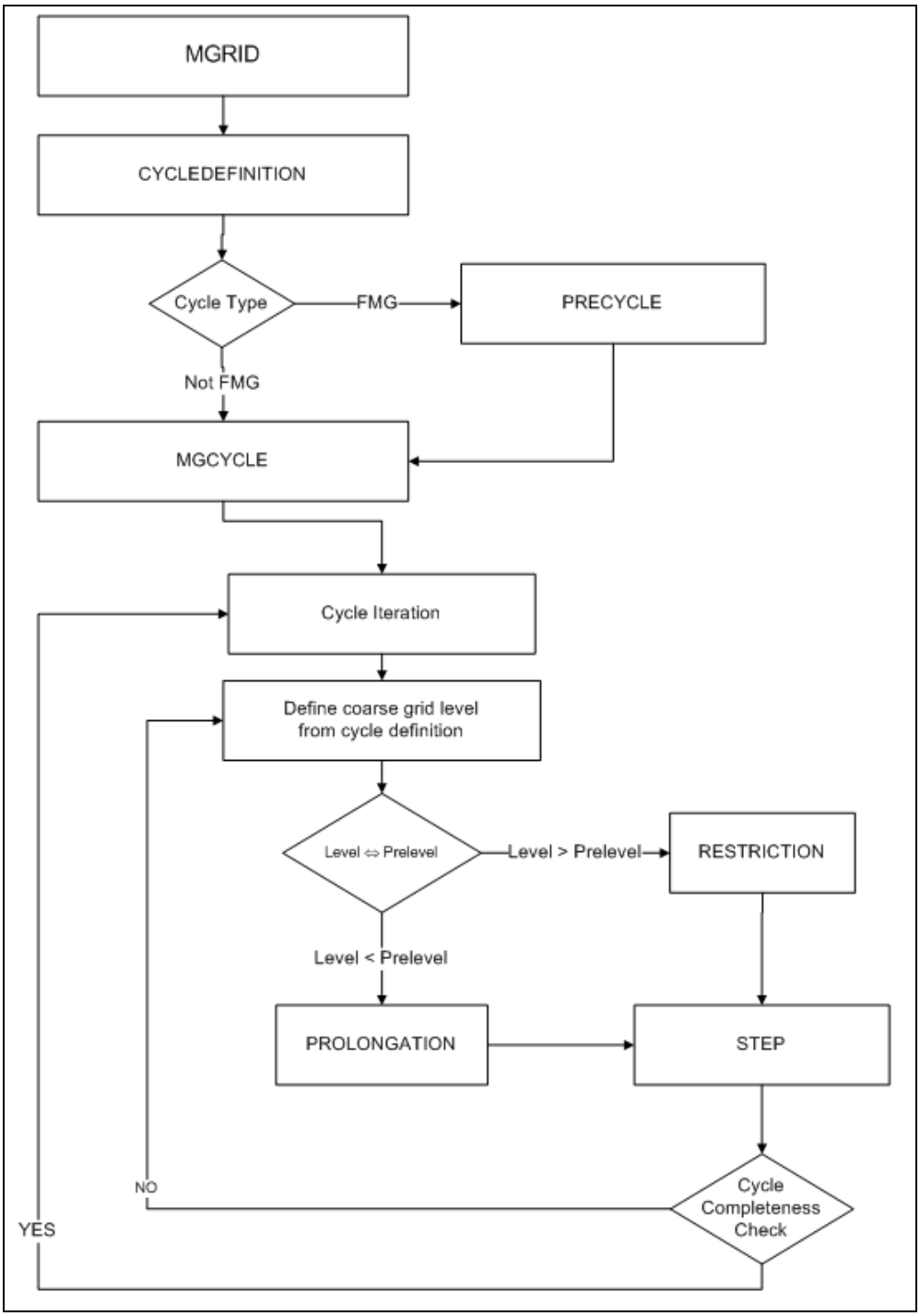


Figure 3.9 Algorithm for the MGRID subroutine

3.3.3 Intergrid Transfer Operators Algorithms

For 2D application, the area weighting rule and for 3D application, the volume weighting rule is used to transfer flow variables from fine to coarse meshes, $\bar{I}_h^H$

$$\bar{I}_h^H(u_h) = \frac{\sum A_{fine} u_h}{\sum A_{coarse}} \tag{3.1}$$

$$\bar{I}_h^H(u_h) = \frac{\sum V_{fine} u_h}{\sum V_{coarse}} \tag{3.2}$$

The residual is actually the fluxes around the boundary of control volume and is therefore related to the time rate of change of conserved variables. In order for this rate of change to be the same for all grids, it is necessary that the residual transfer be conservative, that is, that the sums of the residual on the fine and coarse grids be equal. For this reason, collection operator I_H^h does not use area/volume weighting, but rafter just sums the residuals of finer meshes. These two collection operators make up a process often called the restriction from one grid to a coarser grid.

The restriction and prolongation algorithms perform as the following sequences respectively.

The conservative variables and residuals are restricted from fine to coarse grid levels (subroutine RESTRICTION):

a) The conservative variables of prelevels are appointed.

b) The previous (finer) residual $r_h = (f_h - r_h)$ is updated as forcing function values.

c) The restricted new level conservative variables from finer meshes that formed coarse mesh are generated by area/volume averaging.

d) The restricted new level residuals from finer meshes that formed coarse mesh are generated by summation.

e) One iteration of flux calculation is made by restricted values (subroutine FLUX).

f) The forcing function of coarse meshes for the new coarse level is prepared and the forcing function and the conservative variables are stored as original forcing function with level information.

The conservative variables from coarser to finer grid levels are prolonged (subroutine PROLONGATION):

a) The errors of conservative variables which are originally stored during restriction are found (coarser level) $(\overline{u}_H^{new} - \bar{I}_h^H \overline{u}_h^{old})$.

b) The error values for the fine meshes that formed the coarse mesh are designated by direct injection the error of coarse mesh.

c) If MG cycling type is sawtooth, the error smoothing is done.

d) The conservative variables of new coarse level are updated by adding error values to conservative variables.

e) If MG cycling type is different then sawtooth type, the desired number of iteration is done (subroutine STEP).

CHAPTER 4

RESULTS AND DISCUSSION

In this chapter, the developed grid coarsening algorithm is applied to a variety of grid structures to assess its performance and roboustness. The quadtree based grid coarsening algorithm is used for two dimensional, octree based grid coarsening algorithm is used for three dimensional grids. The performance of the multigrid flow solvers, SENSE2D and SENSE3D which are modified, are investigated with six validation and verification test cases. The time dependent equations are solved explicitly using the third order Runge-Kutta method with variable time-stepping. The solution algorithm proceeds by flux calculation on active edges.

Solutions of the validation cases are carried out on a HP Z600 Workstation. The workstation is based on Intel Xeon X5570 2.93GHz 8MB 1333 FSB Quad Core Processor with 12 GB 1333 MHz DDR3 ECC Registered RAM running under Scientific Linux 5.3 operating system. X5570 series are using hafnium-based Intel® 45nm hi-k metal gate silicon technology and Nehalem.

The flow cases studied in this thesis are listed at Table 4.1.

Table 4.1 List of test cases

Case Number	Dimension	
Case 1	2D	An inviscid flow solution over NACA0012 airfoil
Case 2	2D	A laminar flow solution over NACA0012 airfoil
Case 3	2D	A turbulent flow solution over RAE2822 airfoil
Case 4	3D	An inviscid flow solution over ONERA M6 wing
Case 5	3D	A laminar flow solution over a wing with NACA0012 airfoil
Case 6	3D	A laminar flow solution over a wing with RAE2822 airfoil

The MG adaptation is first presented for an inviscid 2D solution over a NACA 0012 airfoil section with an unstructured grid. In the first part of this validation case, the coarse grids, which are generated by the automated quadtree based grid coarsening algorithm are presented. Then the efficiency of MG strategy on inviscid flow with different cycling strategies are investigated with convergence speeds. In the solution process, V-cycle, W-cycle and FMG multigrid strategies are applied and the normalized continuity residual and convergence of the aerodynamic coefficients are investigated with respect to CPU time. In the second part, MG efficiency is investigated for different angle of attacks at a Mach number of 0.15 and different Mach numbers at angle of attack of 3°. The density residuals convergence histories are demonstrated for the flow parameters.

The second validation case, a hybrid/unstructured viscous type 2D grid over an NACA0012 airfoil section is taken as a root grid and the laminar solutions are obtained at low Reynolds numbers. Then, the MG efficiencies with respect to the

grid level number, the coarsening ratio and the iteration count at coarse level solutions are investigated.

In the third validation case, the automated grid coarsening algorithm is applied to a hybrid/unstructured viscous type grid over an RAE2822 airfoil section. Then viscous flow solution with one equation Spalart Allmaras turbulence model at a transonic Mach number is obtained by using a single grid and V-cycle MG solution.

The fourth validation case that is the first validation case of 3D MG adaptation is presented on an inviscid 3D solution with an unstructured grid over well known validation geometry, ONERA M6. In the first part of this test case, the coarse grids which are generated by the automated octree based grid coarsening algorithm are presented. Then, MG efficiency with different cycling strategies are investigated based on convergence speed ups. In the solution process, V-cycle, W-cycle and FMG multigrid strategies are applied and the reduction of normalized density residual and the convergence of aerodynamic coefficients are investigated with respect to CPU time. In addition, the MG solution efficiencies based on the iteration count at coarse level solutions are investigated.

In a fifth validation case, flow solution for a NACA0012 airfoil on a 3D structured type grid is presented. First, the coarsening algorithm is applied to this structured grid and the coarse grid levels are presented at surface and symmetry plane. Then, the convergence of variations is investigated with convergence speeds on laminar flow at low Reynolds number. In this test case, the MG efficiencies about the iteration count at coarse level solutions are again investigated in the third part to see the effect of solver type. Finally, MG efficiencies which are depent on the grid level number and solver type used at coarse grid levels are presented.

Finally, the automated grid coarsening algorithm are applied a hybrid/unstructured viscous type grid over a 0.2 chord wing with RAE2822 airfoil section. Then laminar

flow solution at transonic Mach number is obtained by single grid and FMG solution.

4.1 Case 1: An Inviscid Flow Solution over NACA0012 Airfoil

4.1.1 Grid Coarsening

Validation of the grid coarsening algorithm is performed on a 2D unstructured grid for NACA0012 airfoil with 9421 nodes and 18390 triangles shown in Figure 4.1. An automated quadtree based grid coarsening algorithm is implemented to unstructured grid around a NACA0012 airfoil and quadtree structure is presented in Figure 4.2.

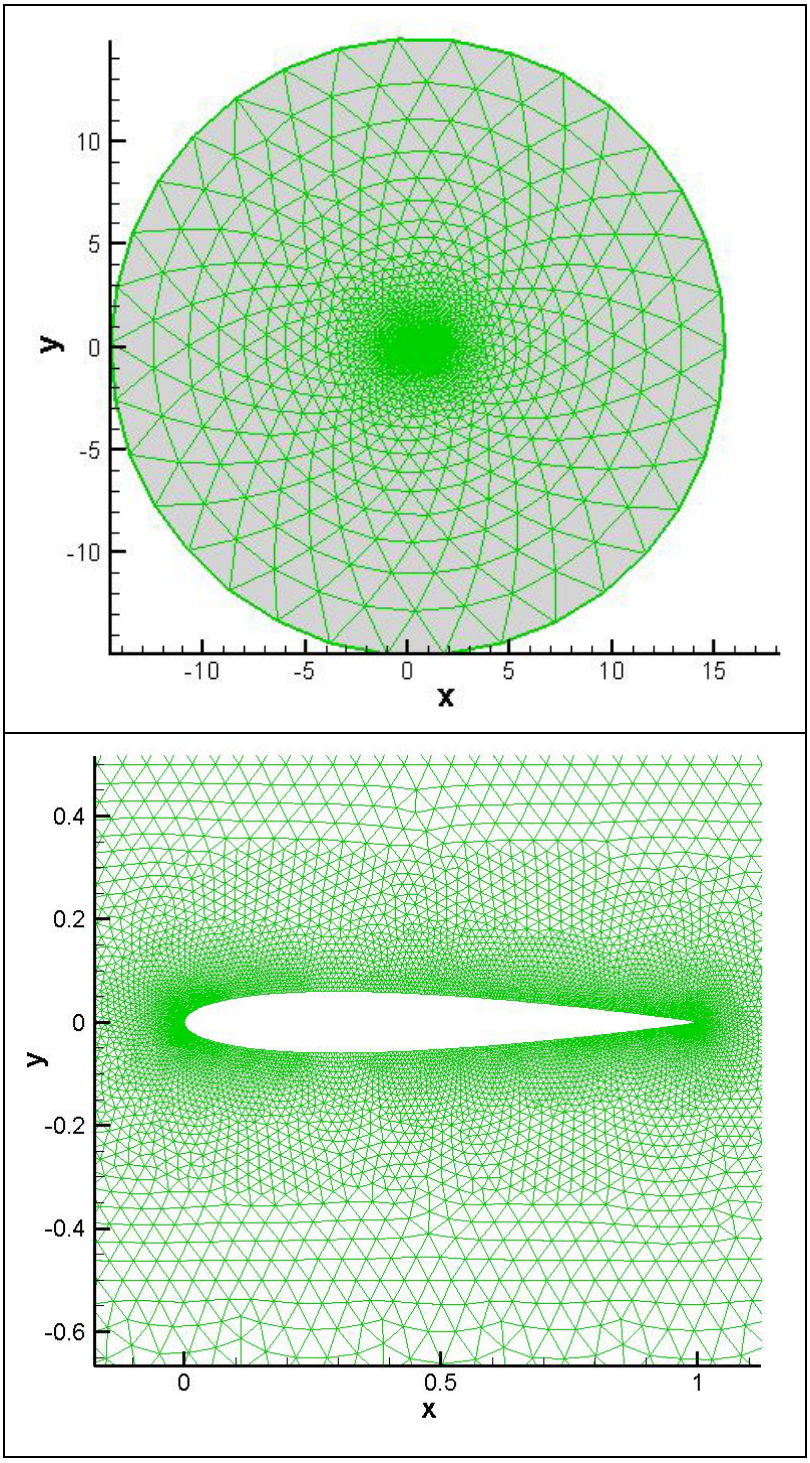


Figure 4.1 Fine grid for case 1

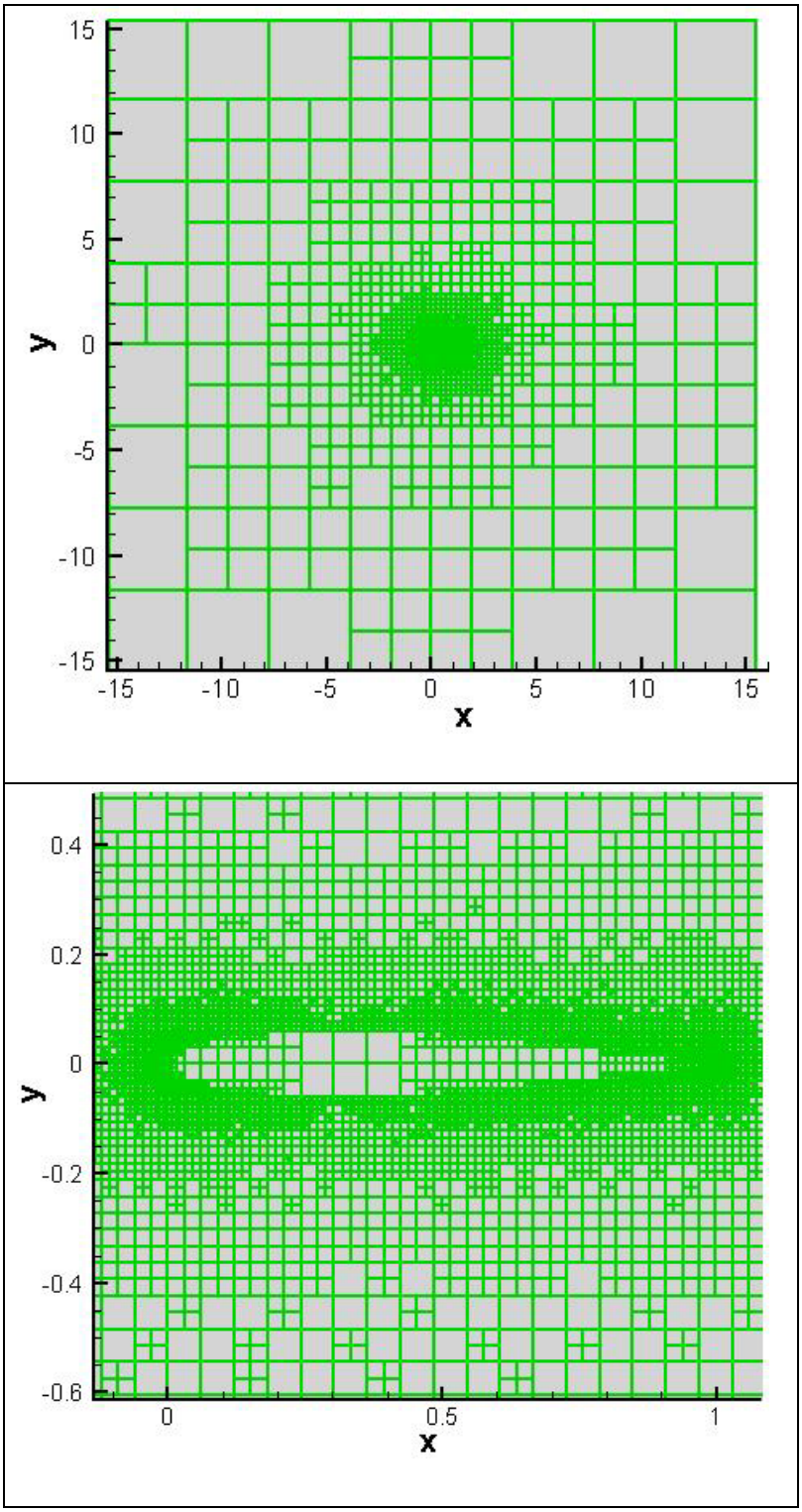


Figure 4.2 Quadtree structure for case 1

A sequence of four coarse hybrid grids is generated with a maximum coarsening ratio of about 40 %. The connectivity information between grids is obtained from the data structure. The fine grid and coarser grids for MG levels with 3204, 1118, 441 and 156 cells from second to fifth grid levels are presented in Figure 4.3.

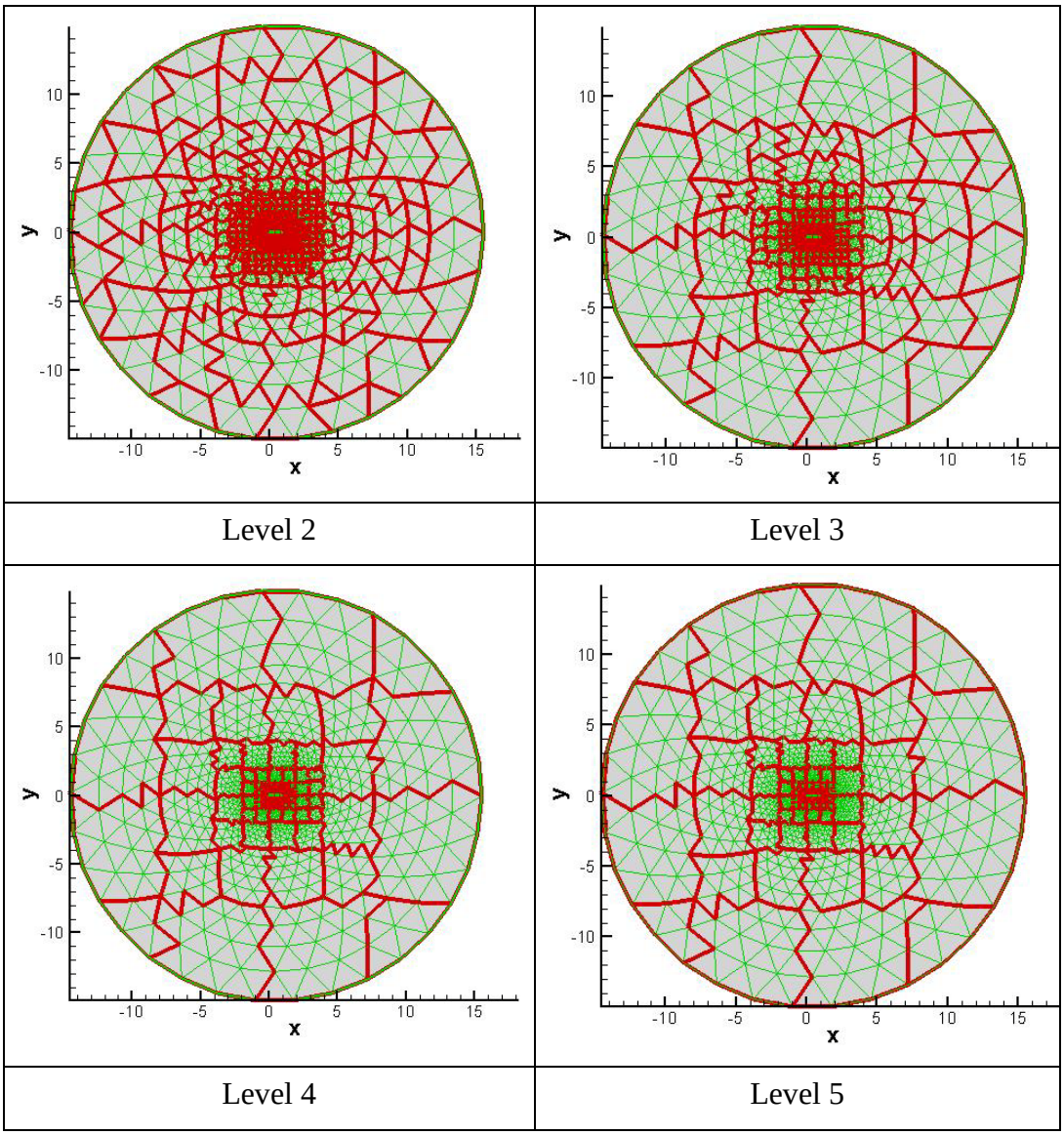


Figure 4.3 Coarse grid levels for case 1

The coarsening behavior around the airfoil is also seen from the close-up view shown in Figure 4.4. When the coarse grid levels are investigated, the coarse meshes have high quality cells with aspect ratios of about unity like Cartesian type grids. The coarsening ratio effect is seen around the airfoil very clearly. In addition, the coarse level grids keep the original anisotropic grid distribution at all coarse levels.

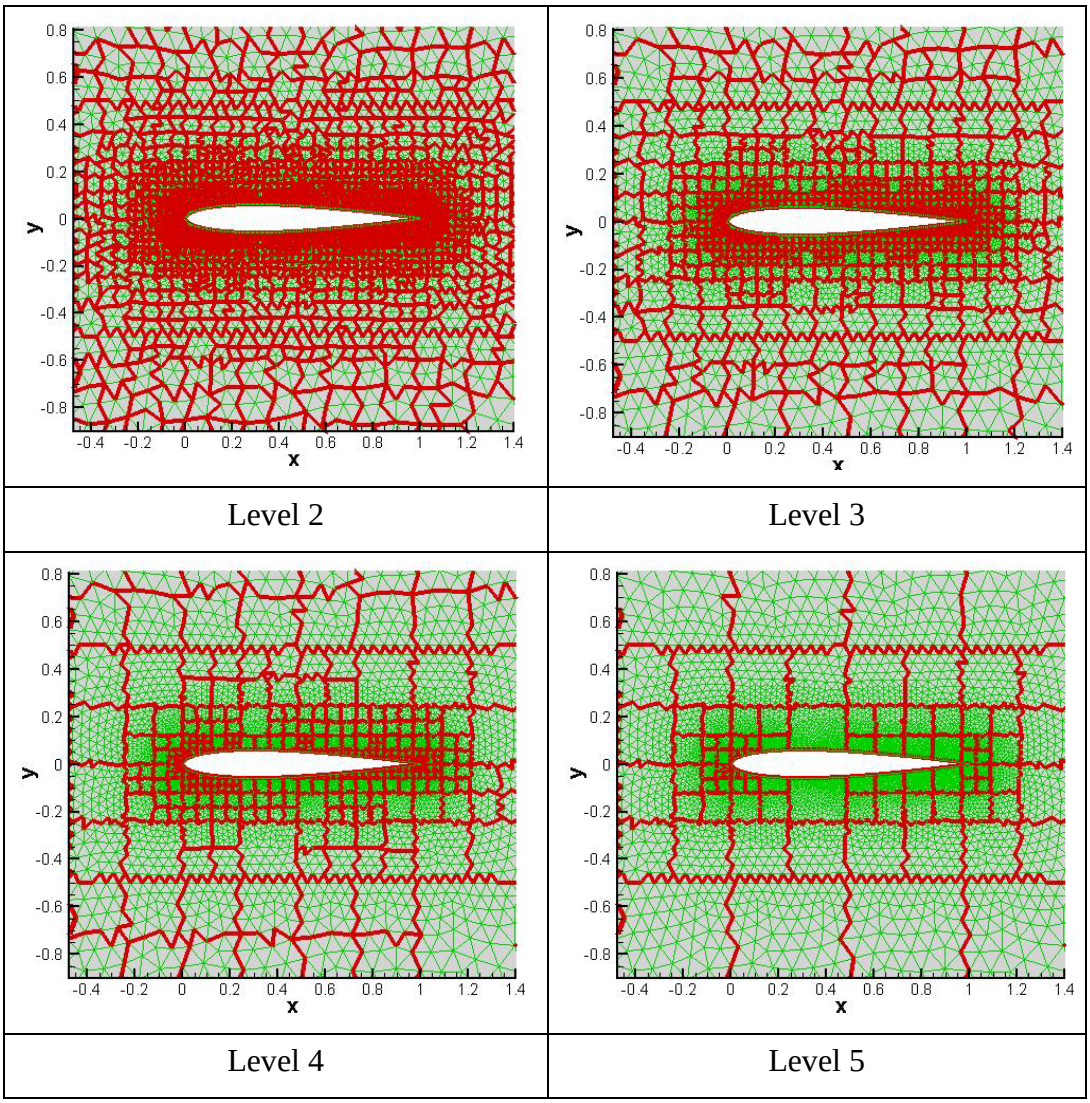


Figure 4.4 Coarse grid levels (close-up view) for case 1

4.1.2 MG Efficiencies on Inviscid Solution

The single grid solution at a low Mach number of 0.15 and at an angle of attacks of 3° is obtained using the first order flux computations with a CFL number of 0.9. The same solution is then obtained by applying V-cycle, W cycle and FMG algorithms with 20 time steps at the fine grid level and 10 time steps at coarse grid levels. The density distribution around the airfoil obtained from the single grid is presented in Figure 4.5.Then, the distribution of the pressure coefficient computed by the single grid and V-cycle MG, which reached the exactly same results, is given in Figure 4.6.

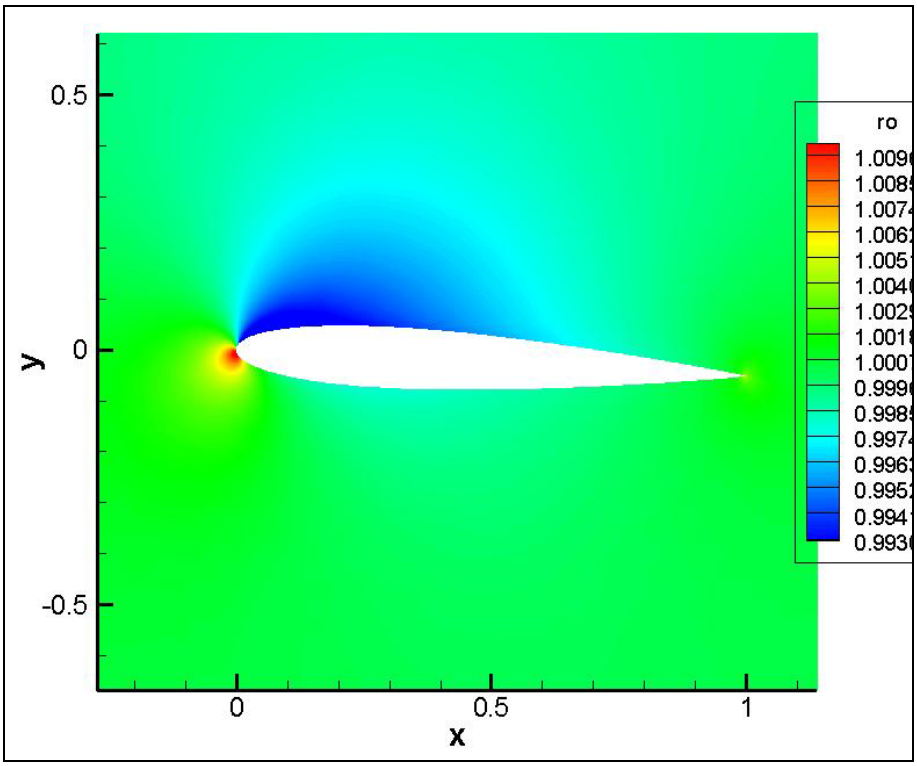


Figure 4.5 The density distribution around NACA0012 airfoil for case 1

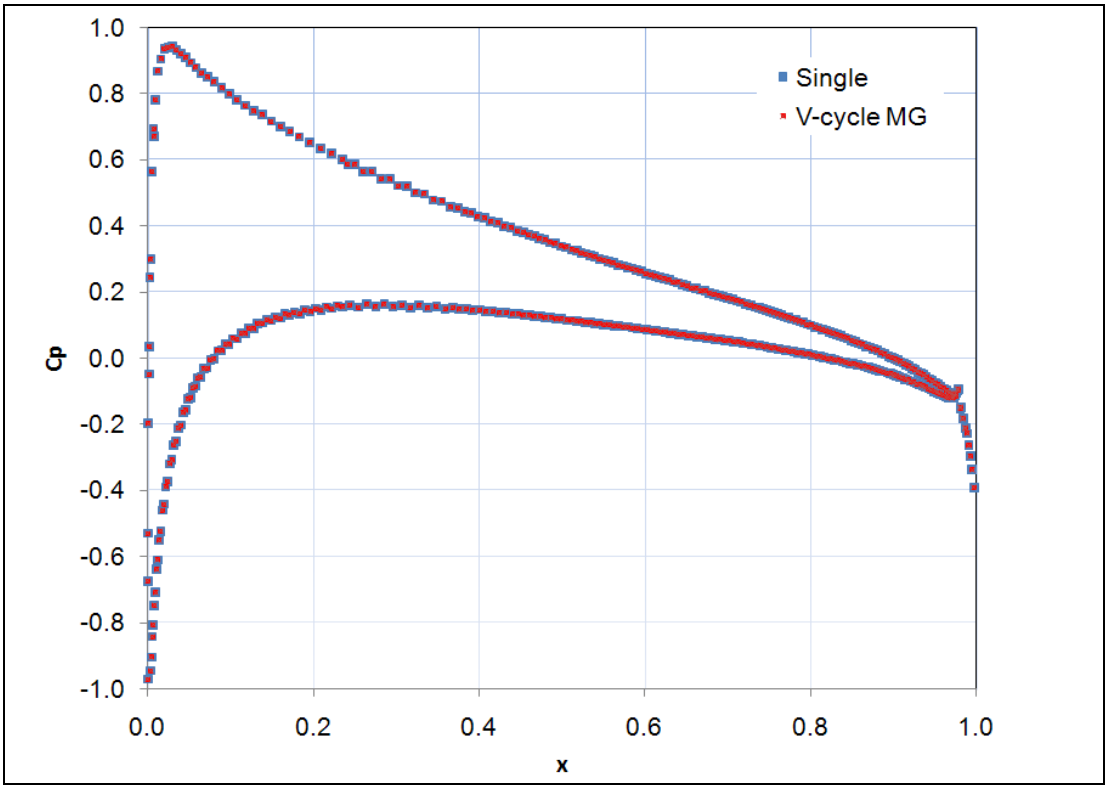


Figure 4.6 Distribution of the pressure coefficient around NACA0012 airfoil for case 1

The convergence histories of the single grid, V-cycle, W cycle and FMG cycle solutions in terms of the variation of the normalized density residual are presented in Figure 4.7. It is seen that, all multigrid solutions exhibit nearly the same converge rate as expected and they are approximately 10 times faster than the single grid solution without the MG. Such convergence acceleration is in agreement with the findings in literature [61]

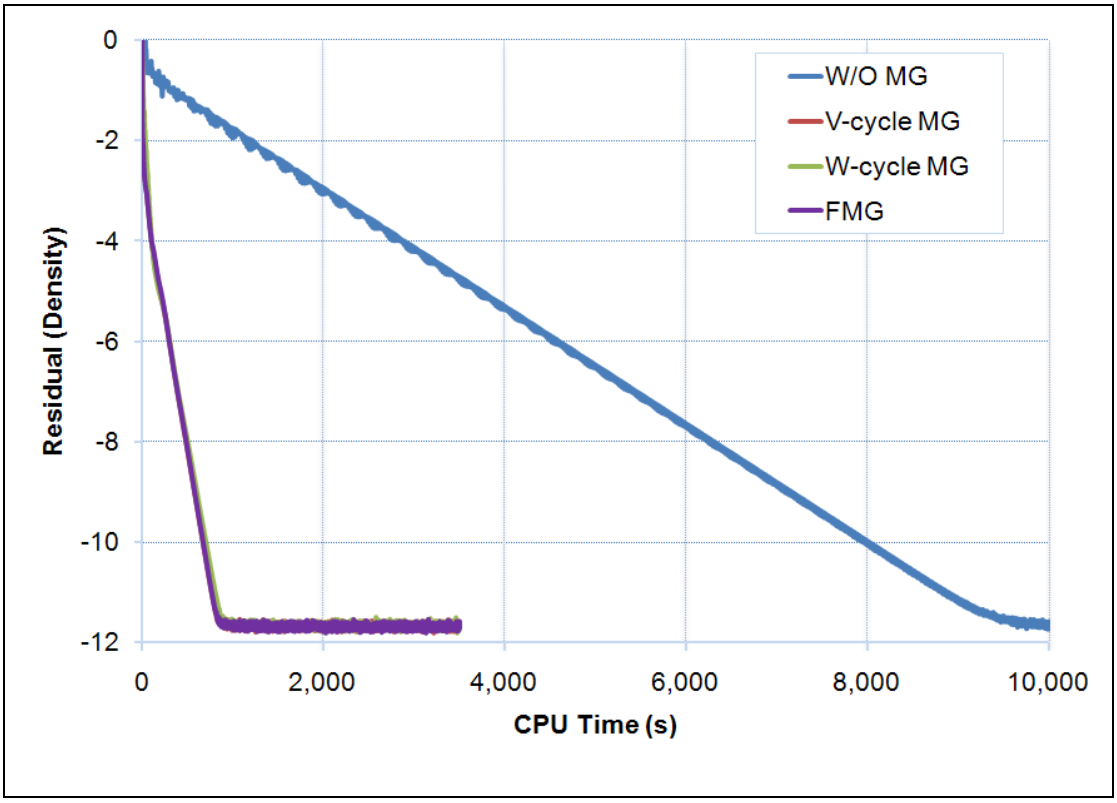


Figure 4.7 Residual histories of single and MG solutions for case 1

The MG cycling strategies only differ during the initial iterations as observed from Figure 4.8. FMG uses a classical V-cycle after the preliminary stage. Therefore classical V-cycle and FMG cycling strategies give nearly same residuals after 50 seconds. W-cycle uses the coarse level grids more heavily due to the logic of cycling. For this reason, the initial convergence is slower than the other cycling strategies. Finally,FMG gives the fastest convergence due to the preliminary stage due to having preliminary stage of the strategy.

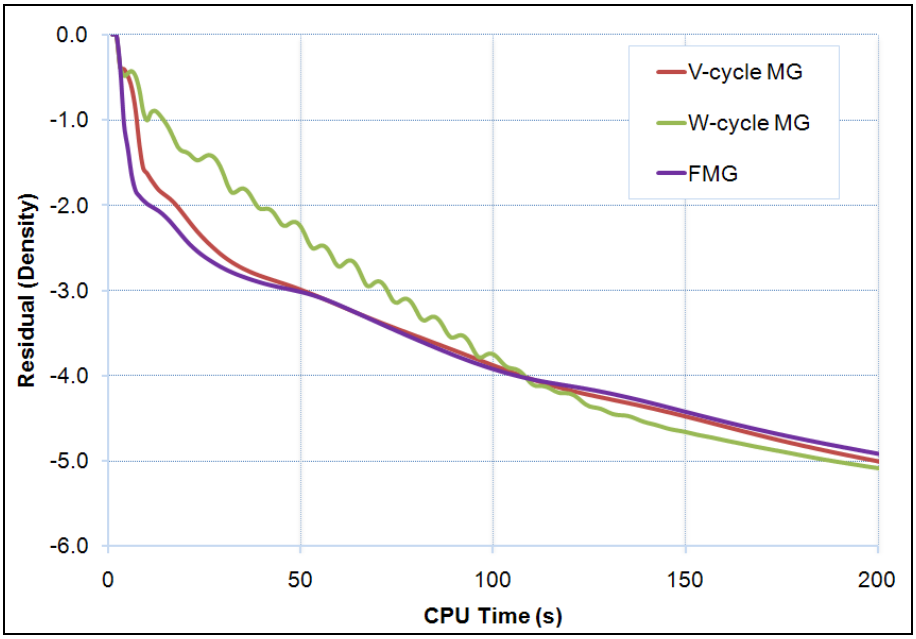


Figure 4.8 Residual histories of MG solutions at initial stages for case 1

Then, the cycling strategies are investigated according to the convergence of aerodynamic loads in Figure 4.9 for the solutions with and without MG. Like residual convergence characteristics, the histories of the convergence of aerodynamic loads are similar and reach the same value after a few number of iterations.

The convergence of aerodynamic loads for all MG cycles is investigated again at the initial stage of the solution in detail. Within the first few seconds, FMG reaches the final value of aerodynamic loads very rapidly without any oscillations as shown in Figure 4.10. V-cycle and W-cycle converge the same value all about the same CPU time.

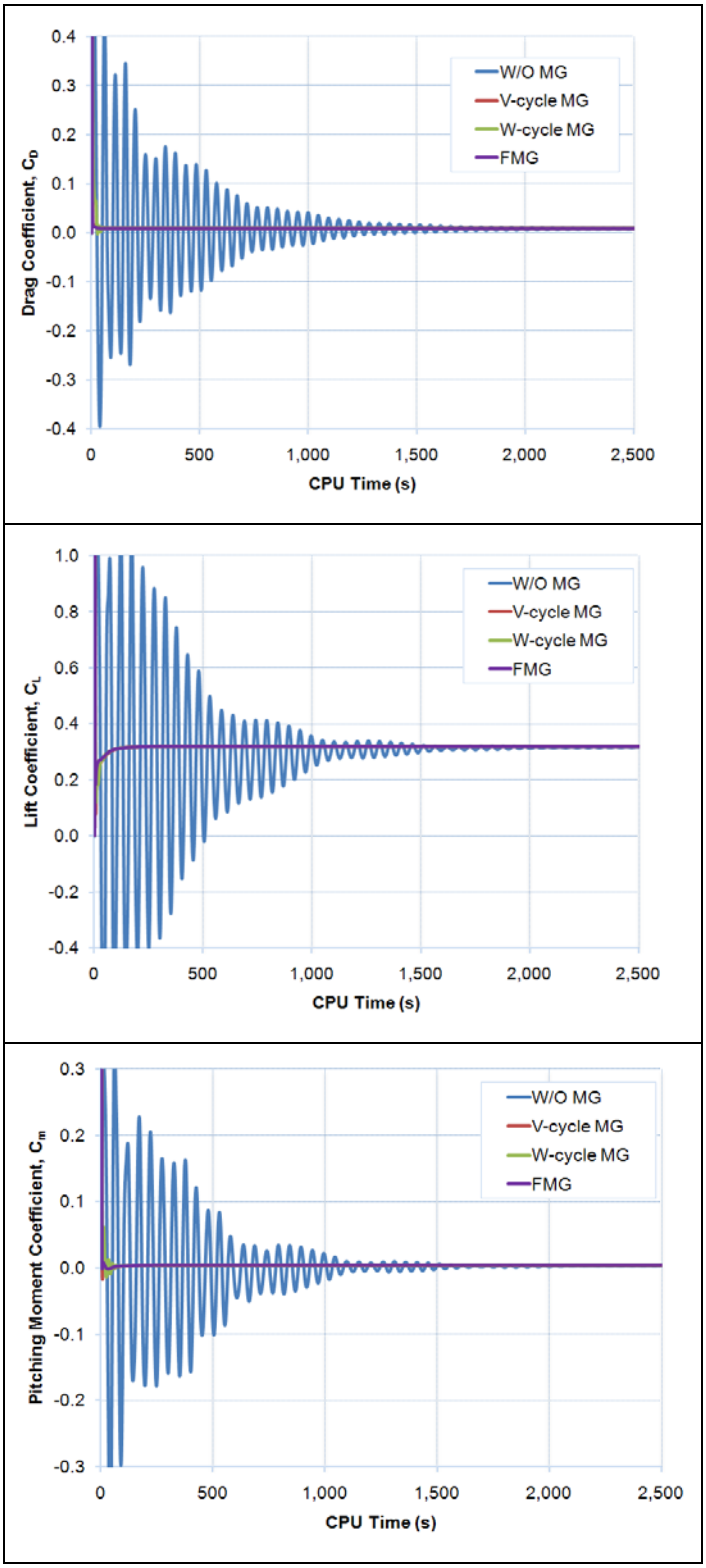


Figure 4.9 Convergence histories of aerodynamic coefficients for case 1

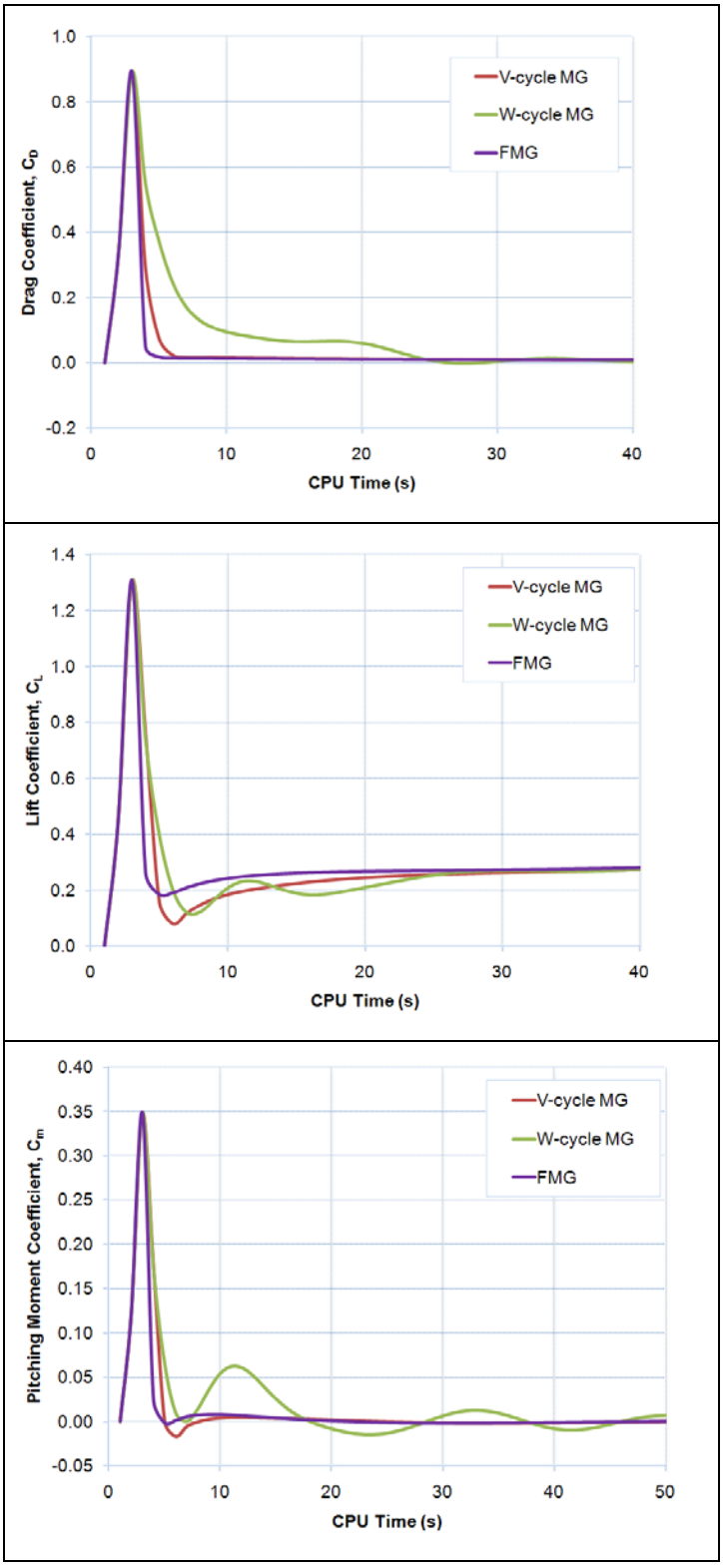


Figure 4.10 Convergence of aerodynamic coefficients at the initial stage for case 1

The runtime needed to reach 20 %, 10 %, 5 %, 1 % and 0.1 % error bands of aerodynamic force coefficients with all cycling types and fine grid solution is reported and the lifting and drag force acceleration ratios to reach the above error bands with respect to single grid in terms of CPU time (clock time) are calculated. The nondimensional CPU time according to time needed for single grid and speed up ratios are presented for drag and lift force coefficients in Figure 4.11 and Figure 4.12 respectively.

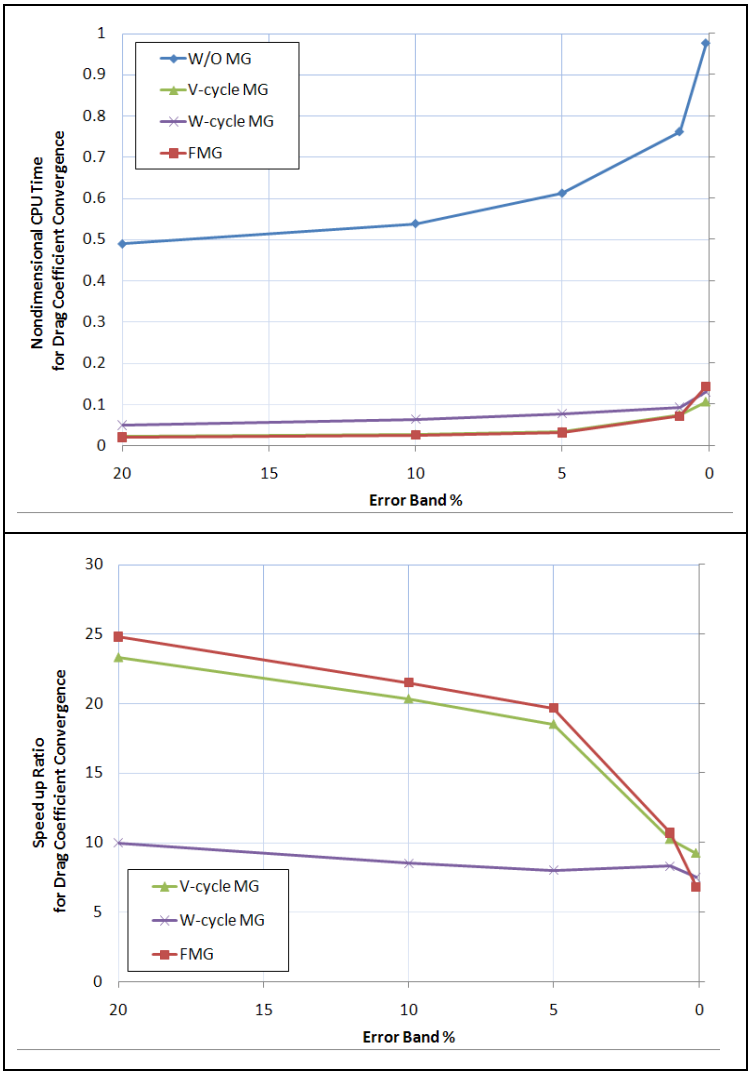


Figure 4.11 CPU time and speed up ratio for the drag coefficient convergences to reach 20 %, 10 %, 5 %, 1 % and 0.1 % error band for case 1

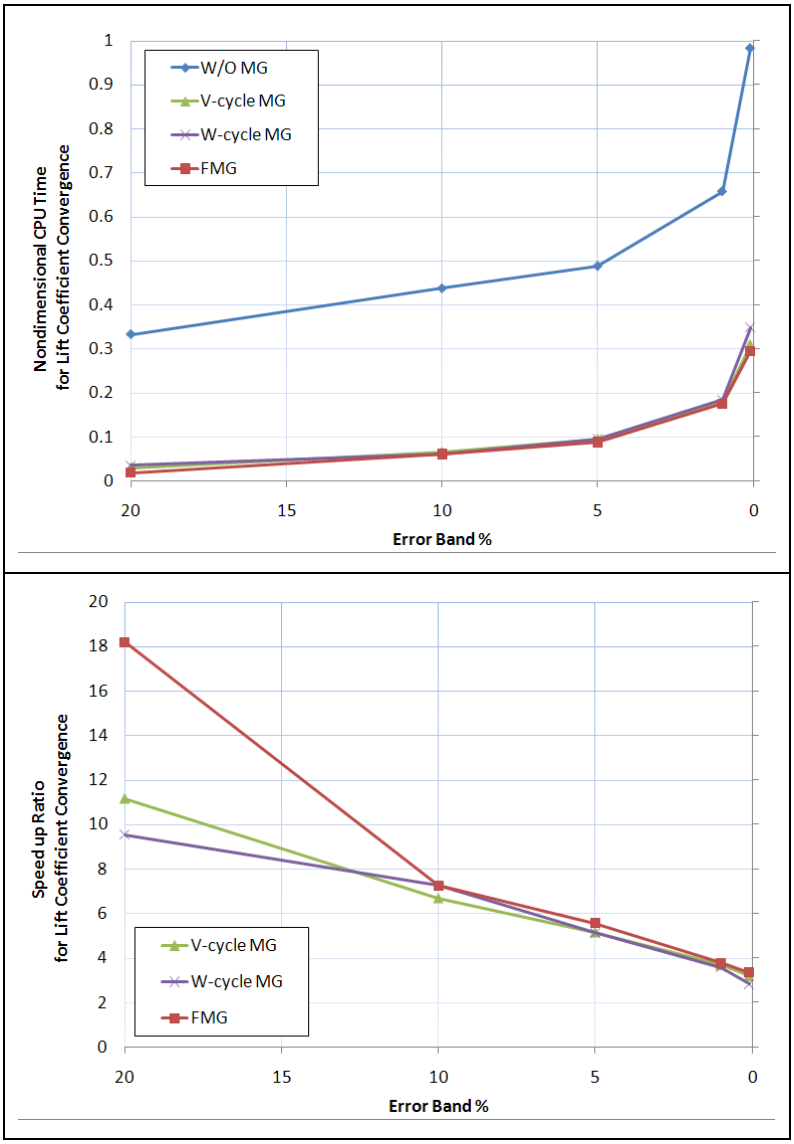


Figure 4.12 Plot of CPU time and speed up ratio for the lift coefficient convergences to reach 20 %, 10 %, 5 %, 1% and 0.1 % error band for case 1

It is be observed that it is enourmously fast to reach especially 20 % error bands for the force coefficients for all MG cycling strategies. It is found that to reach 5 % error band, the drag coefficients for V-cycle and FMG solutions have approximately 20 times, W-cycle solution has 9 times, the lift coefficients for all MG solutions have approximately 5 times faster convergence ratio than the baseline solution.

4.1.3 Dependence of Flow Parameters for the Inviscid Solution

4.1.3.1 Mach Number Dependency

MG efficiencies at low subsonic, subsonic and transonic flow solutions with Mach number of 0.15 , 0.3, 0.45, 0.6 and 0.75 at 3° angle of attack and Reynolds number of 3,000,000 are investigated using the V-cycle strategy. The unstructured fine grid over NACA0012 airfoil with 18390 cells and its coarse level grids are used.

The residual histories for all cases with and without MG algorithm are presented in Figure 4.13. It is seen that MG solutions converge to the same residual value of their fine grid solutions. Except the low subsonic case, having a Mach number of 0.15, all MG solutions converge at the same time. Therefore, MG solutions are more efficient in terms of the residual convergence, as the Mach number is decreased. Although, MG solution at low Mach numbers is slightly different than the other solutions, it is still very efficient when compared to the single grid solution.

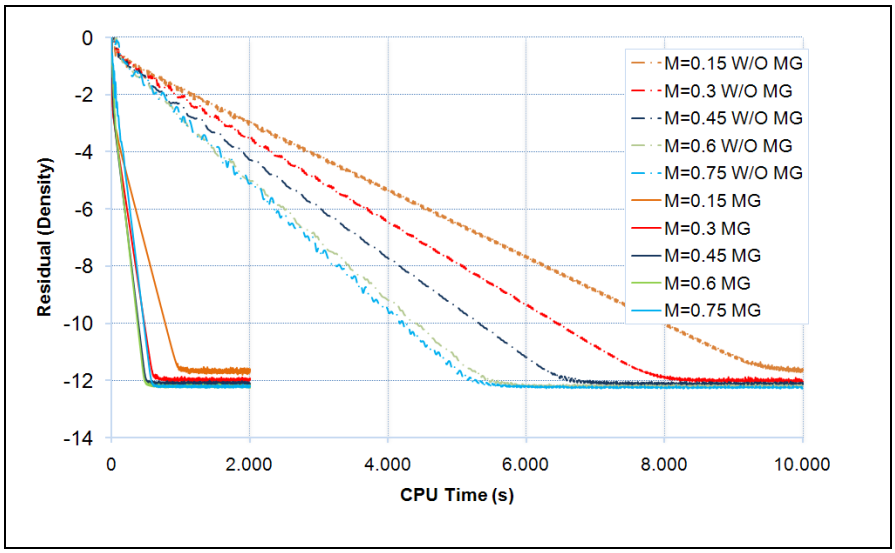


Figure 4.13 Residual histories of single grid and MG solutions for a variety of Mach numbers for case 1

4.1.3.2 Angle of Attack Dependency

A set of angle of attacks, 0°, 3°, 6°, 9° and 12° are studied at the same flow conditions of Case 1. Like Case 1, the unstructured grid over NACA0012 airfoil with 18390 cells and its coarse grid levels are used in this part. For comparison purposes, V-cycle MG algorithm is applied to these inviscid cases.

The residual histories for all the cases with and without MG algorithm are presented in Figure 4.14. It is seen that, the convergence rates are approximately the same and do not depend on the angle of attack. In addition, the MG solution at 0° angle of attack damps the oscillations similar to fine grid solution. The convergence histories of aerodynamic loads also show similar characteristics at an angle of attack of 3°.

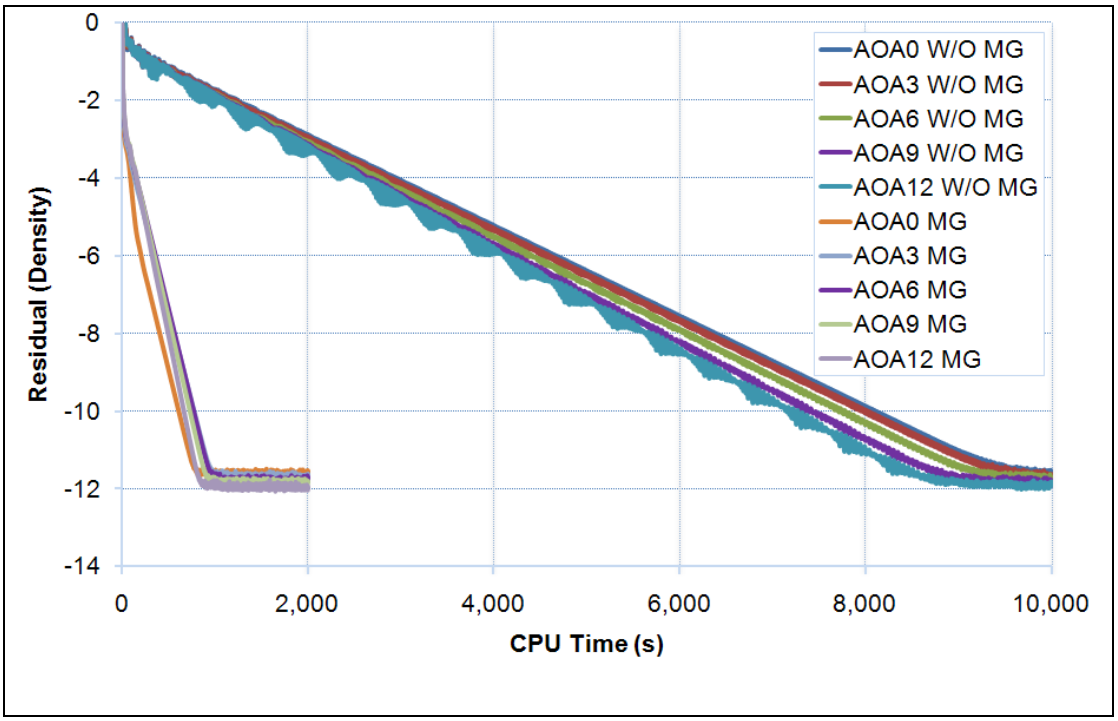


Figure 4.14 Residual histories for the single grid and MG solutions for variety of angle of attacks for case 1

4.2 Case2: A Laminar Flow Solution over NACA0012 Airfoil

4.2.1 Grid Coarsening

The developed grid coarsening algorithm is next implemented on a 2D hybrid/unstructured grid containing 23672 nodes and 33110 cells with approximately boundary layer thickness of 0.08 m for 1 m chord length, as shown in Figure 4.15. A sequence of four coarse level grids is generated with a maximum coarsening ratio of about 40%. The coarse level grids for MG application have 5264, 1745, 636 and 237 cells.

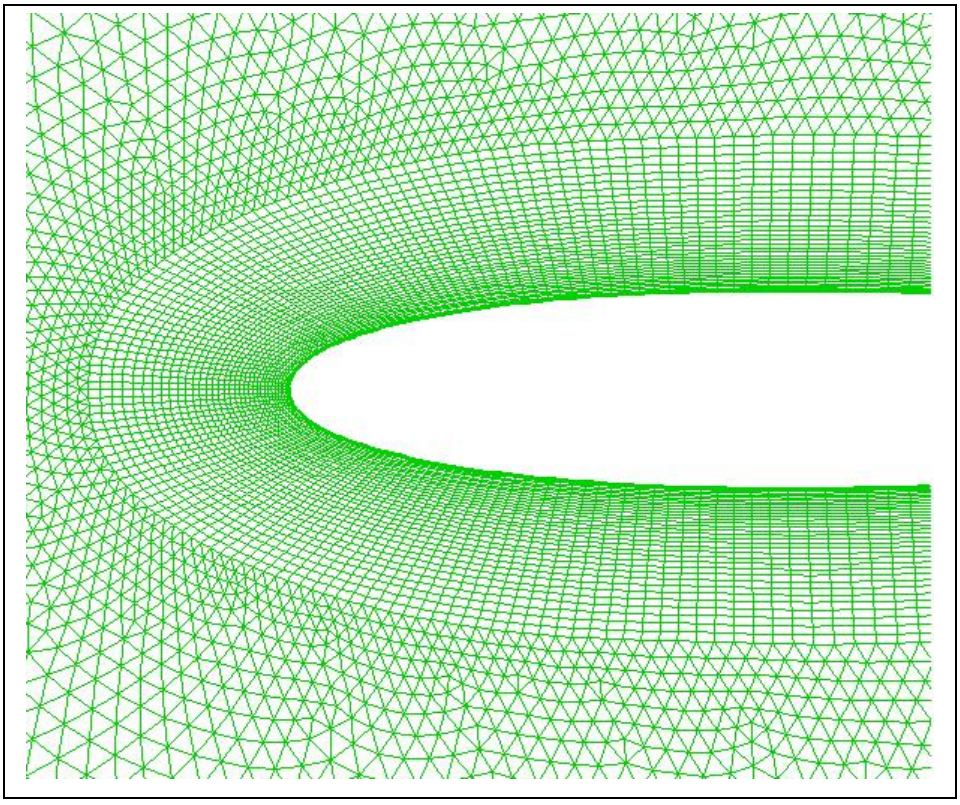
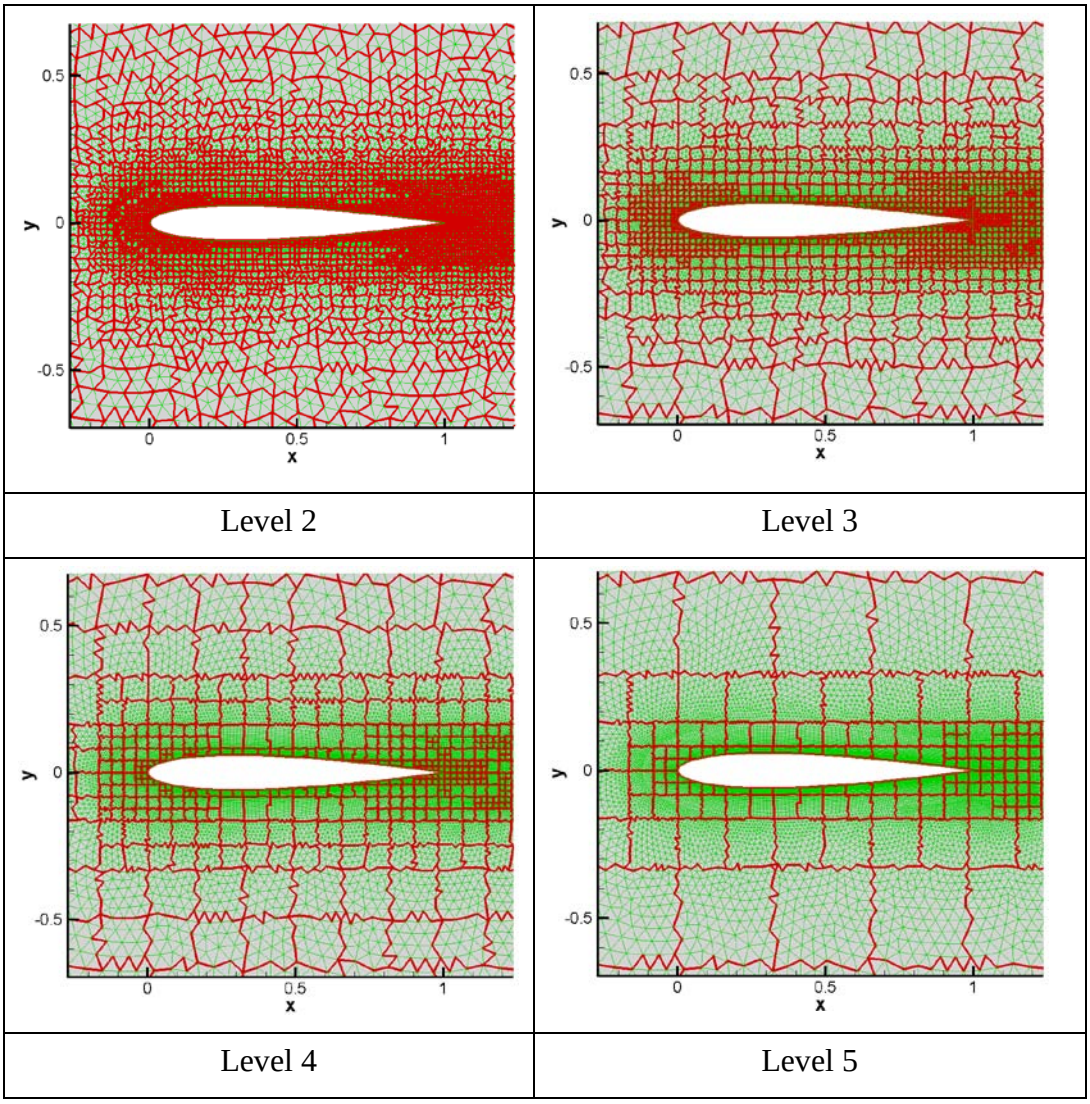


Figure 4.15 The boundary layer view for case 2

The coarsening behavior around the airfoil is seen from the close-up view shown in Figure 4.16. Like Case 1, the coarse grid levels have high quality cells with aspect ratios of about unity similar to a Cartesian type grid. The coarse level grids keep the original anisotropic grid distribution at all coarse levels.



Level 2	Level 3
Level 4	Level 5

Figure 4.16 Coarse grid levels (close-up view) for case 2

4.2.2 MG Efficiencies on Laminar Solution

This validation case is for laminar flow over NACA0012 airfoil similar to Case 1 at a low Reynolds number of 10000, a Mach number of 0.1 and an angle of attack of 3°. The solution using single grid is obtained using the first order flux computations with a CFL number of 0.9. Then, the same solution is obtained by applying FMG algorithms with 20 time steps at the fine grid level and 10 time steps at coarse grid levels. The flow solutions at coarse grids for suppling the correction term to the fine grid solution are obtained by using the inviscid solver. The velocity distribution and boundary layer velocity profile, which are similar in the single grid and FMG solutions, is presented in Figure 4.17.

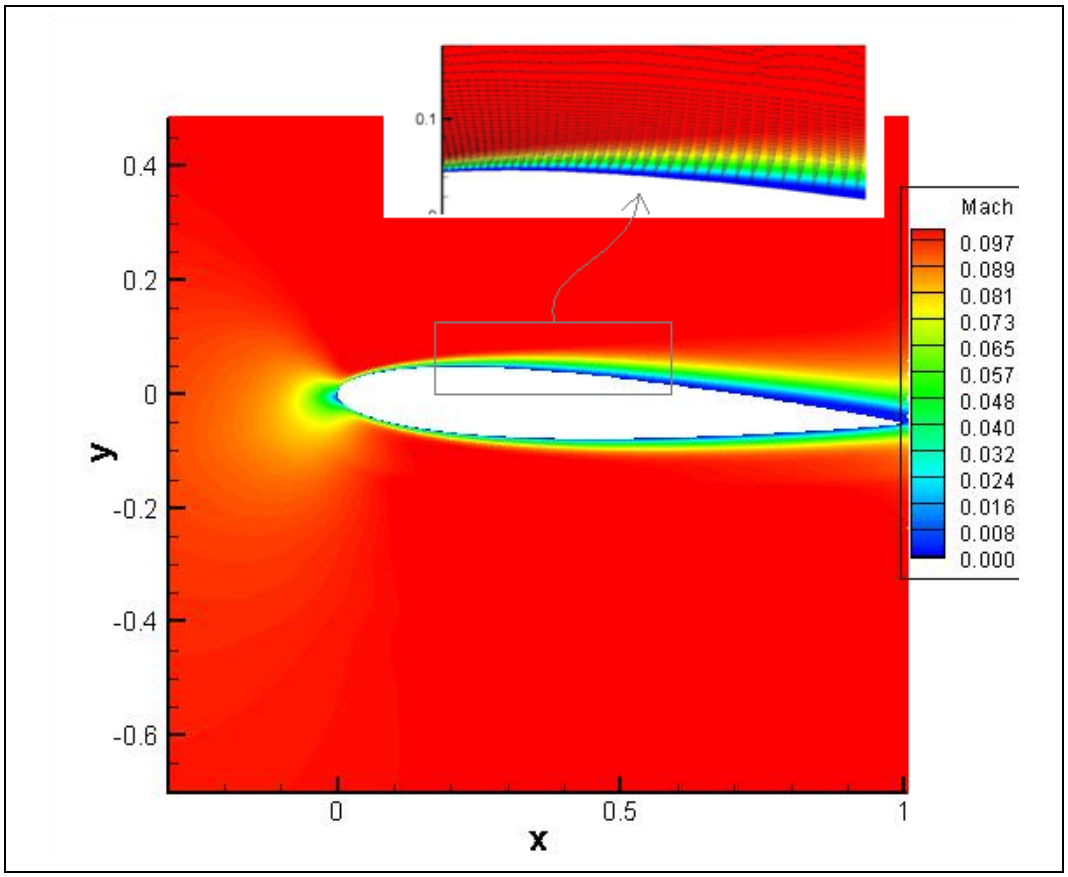
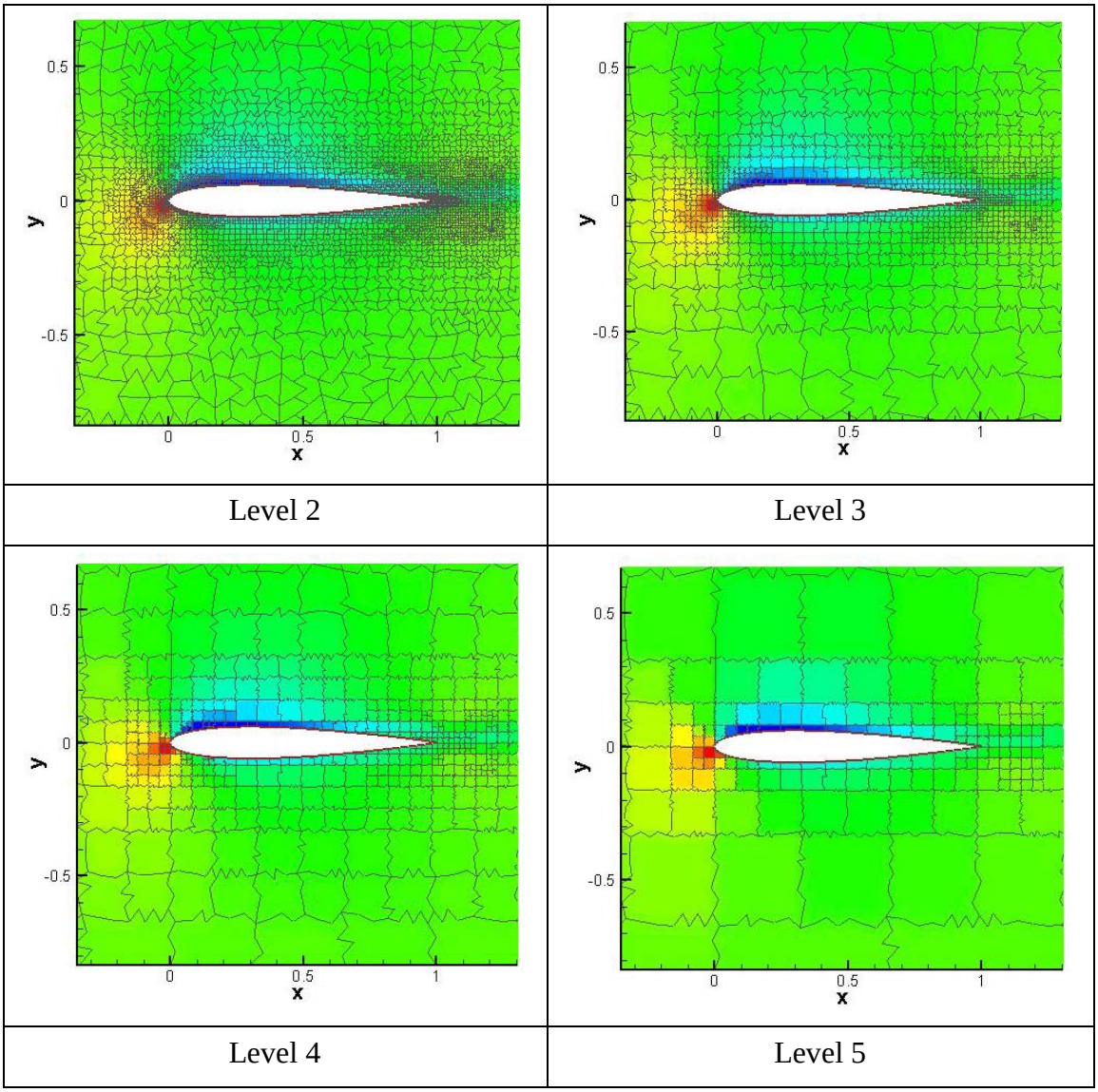


Figure 4.17 The Mach number distribution and boundary layer velocity profile for case 2

The Mach number distribution in FMG solution at the coarse grid levels are demonstrated in Figure 4.18. In addition, Mach distribution on coarse grid levels is very regular since the original anisotropic grid distribution is kept at all levels.



Level 2	Level 3
Level 4	Level 5

Figure 4.18 The grid (close up view) and Mach number distribution at coarse level grids for case 2

The convergence histories of the solutions in terms of the variation of the normalized density residual are shown in Figure 4.19 and compared with the convergence history of the single grid solution. As in the case of inviscid solutions, the MG adaptation on laminar solution has very fast convergence capability with respect to the single grid solution. A similar convergence rate is observed in the variation of drag and lift force coefficients when the solutions with and without MG are compared as shown in Figure 4.20.

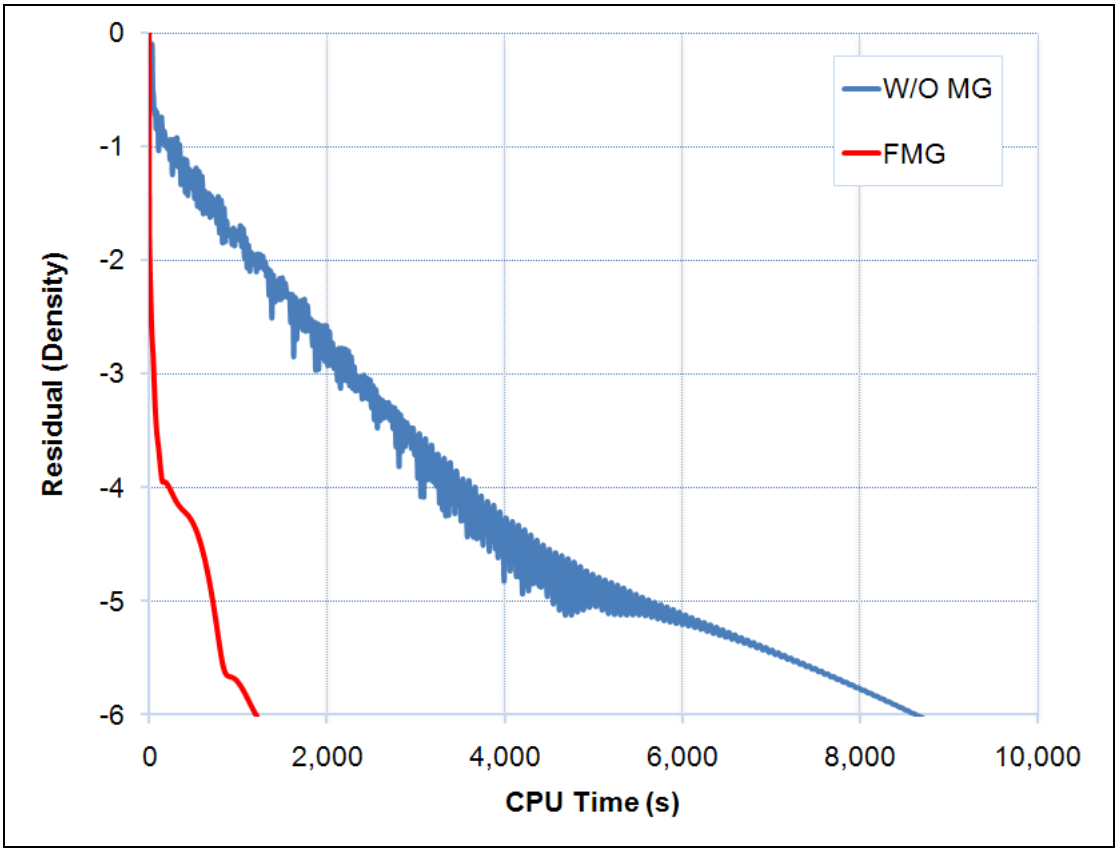


Figure 4.19 Residual histories of single and MG solutions for case 2

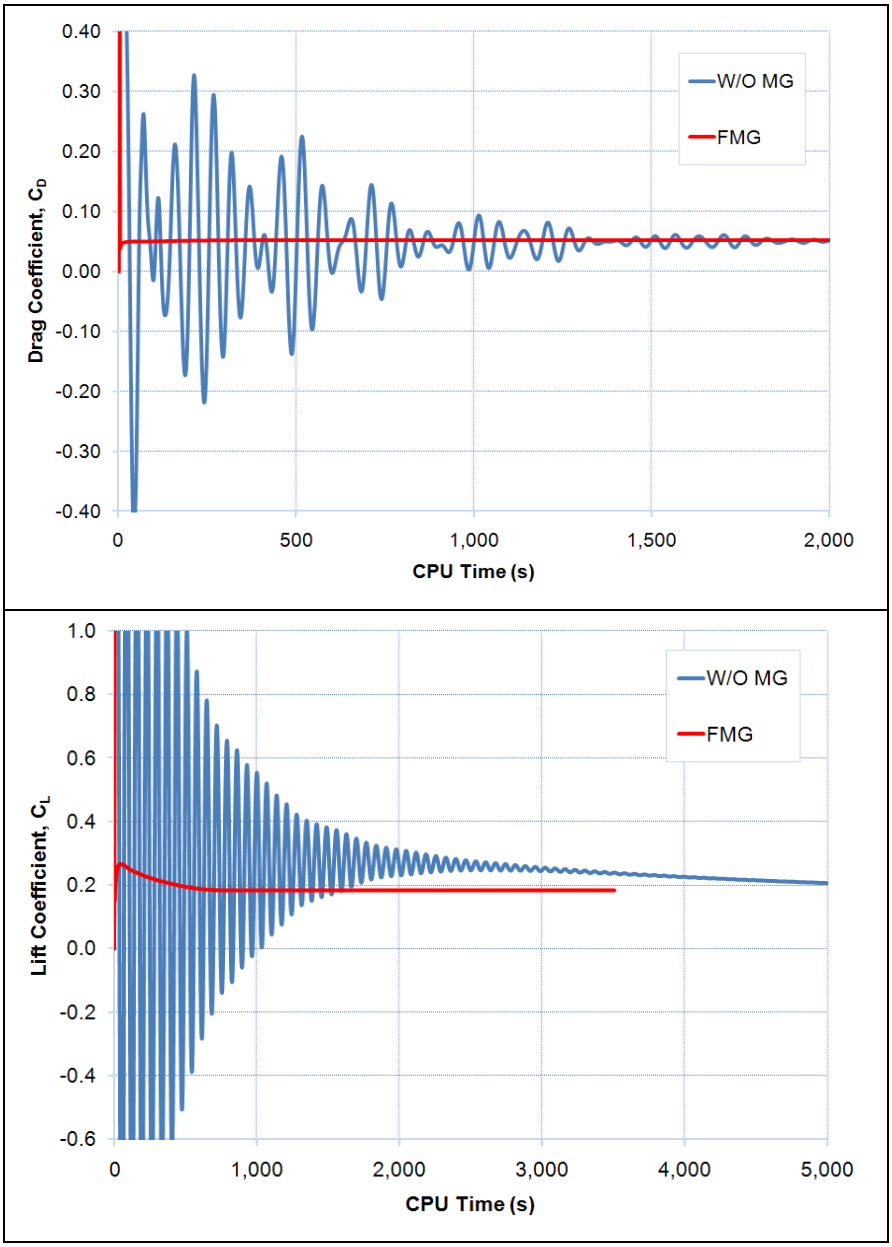


Figure 4.20 Convergence histories of drag and lift force coefficients for case 2

The speed-up in the convergence of the drag and lift coefficients in fine grid level without MG and FMG in terms of iteration step and in terms of CPU time (clock time) is shown in Figure 4.21. Solution based on FMG is taken as a reference and CPU times to reach 20 %, 10 %, 5 %, 1 % and 0.1 % error bands are recorded. It is

seen that for viscous case, the single grid solution converges very slowly and, therefore, MG solution acceleration rate is very efficient. Due to the initial convergence speed of FMG, drag force coefficient reaches to 20 % error band, 97 times faster than the single grid solution. To reach 5 % error band, the drag and lift coefficients in FMG solutions have approximately 25 and 4 times faster convergence ratio than the single grid solution, respectively.

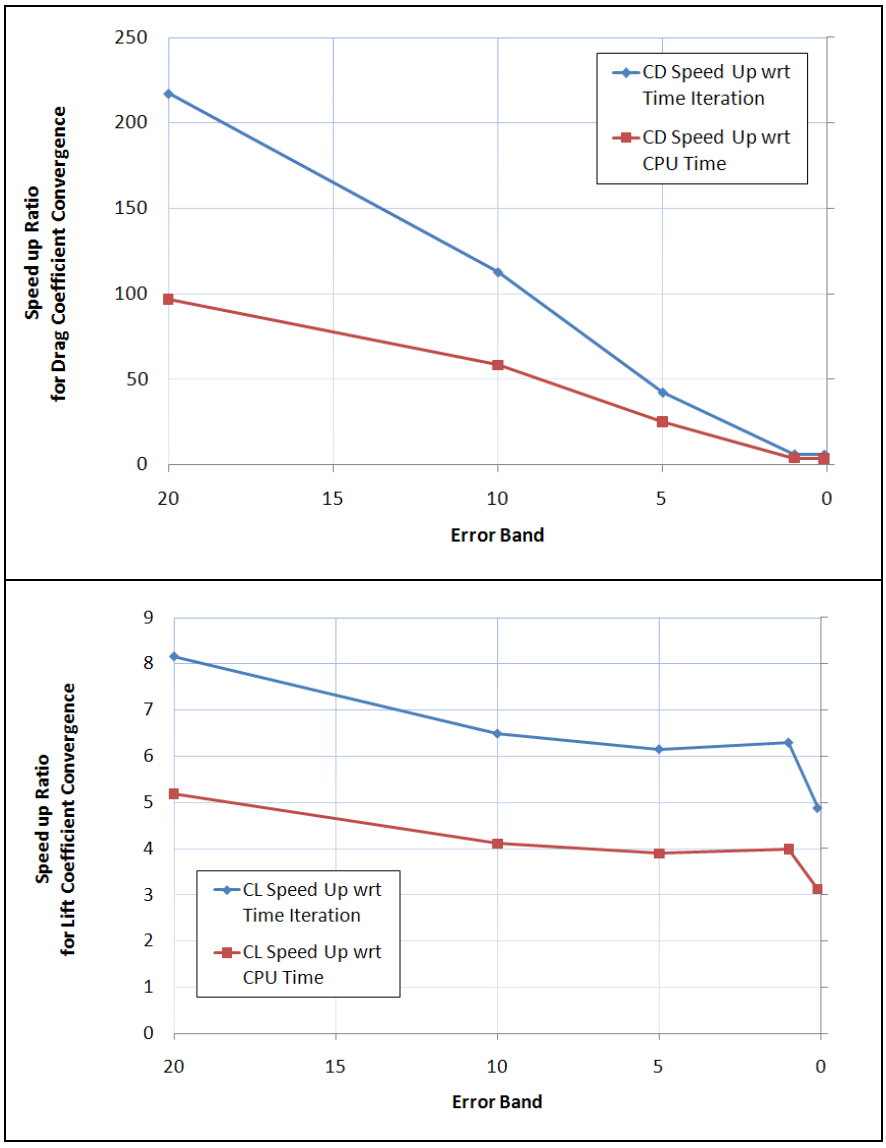


Figure 4.21 Drag and lift coefficient convergence ratios with respect to time iteration step and CPU time for case 2

4.2.3 Effect of Coarsening Ratio between Grid Levels

In this part, the effect of coarsening ratio between coarse grid levels in MG applications is investigated. Three set of coarse grid levels are generated with the coarsening ratios of 25 %, 40 % and 55 %. The grid size of coarse grid levels is tabulated in Table 4.2. The highest level coarse grids are shown in Figure 4.24, Figure 4.23 and Figure 4.24.

Table 4.2 The grid size of coarse grid levels for coarsening ratios of 25 %, 40 % and 55 %

Level	Coarsening Ratio		
	Maximum 25%	Maximum 40%	Maximum 55%
2	5264	5264	15675
3	636	1745	5264
4	148	636	1745
5	29	237	636

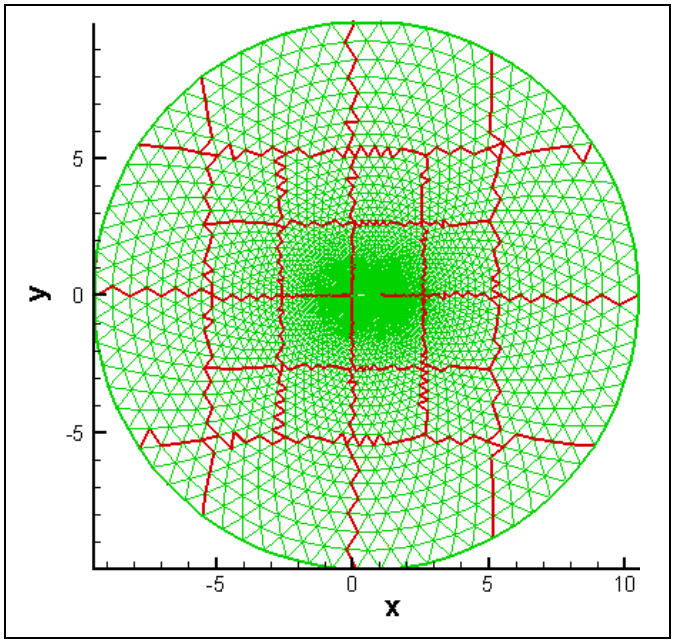


Figure 4.22 The highest coarse grid level having coarsening ratios of 25 % for case 2

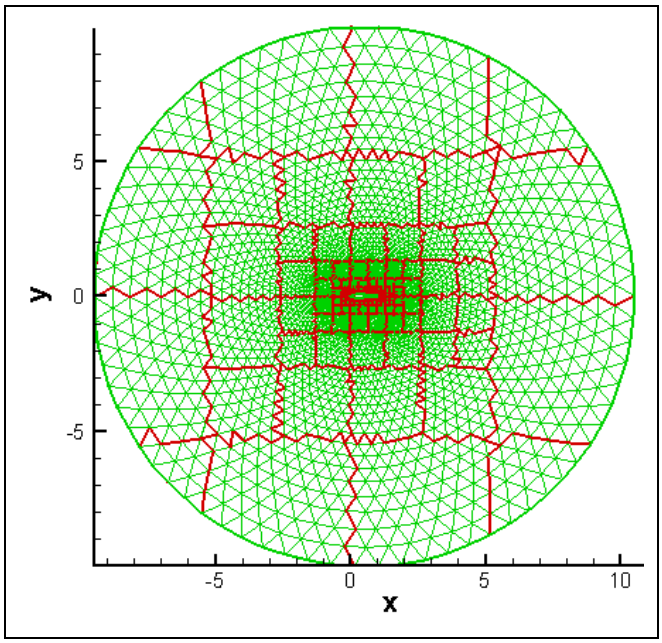


Figure 4.23 The highest coarse grid level having coarsening ratios of 40 % for case 2

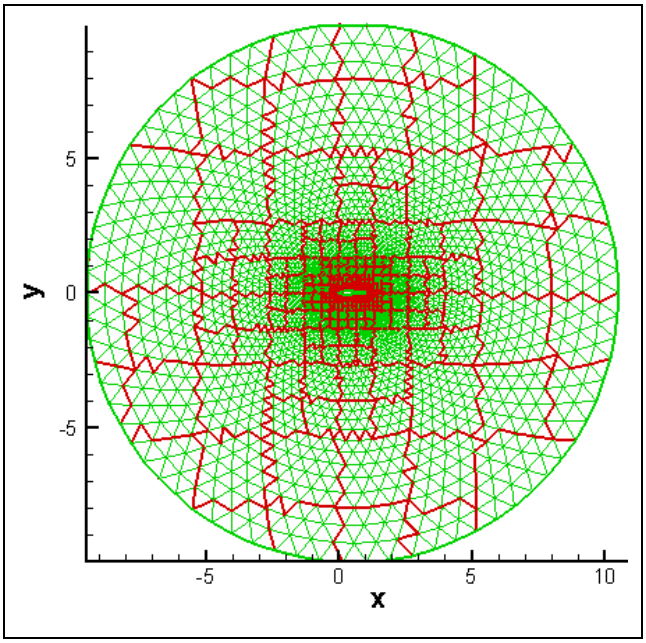


Figure 4.24 The highest coarse grid level having coarsening ratios of 55 % for case 2

The laminar flow solutions with different coarsening ratios using V-cycle MG at a low Reynolds number of 10000, a Mach number of 0.1 and an angle of attack of 3° are obtained. The convergence histories of the solutions in terms of the variation of the normalized density residual are shown in Figure 4.25. It is seen that all coarse grid levels with different coarsening ratio give approximately the same convergence histories. However, as the coarsening ratio level is increased, load convergence histories show oscillatory characteristics similar to the single grid solution as seen in Figure 4.26. In addition, coarsening ratio of 40% gives the most rapid initial load convergence without any oscillation.

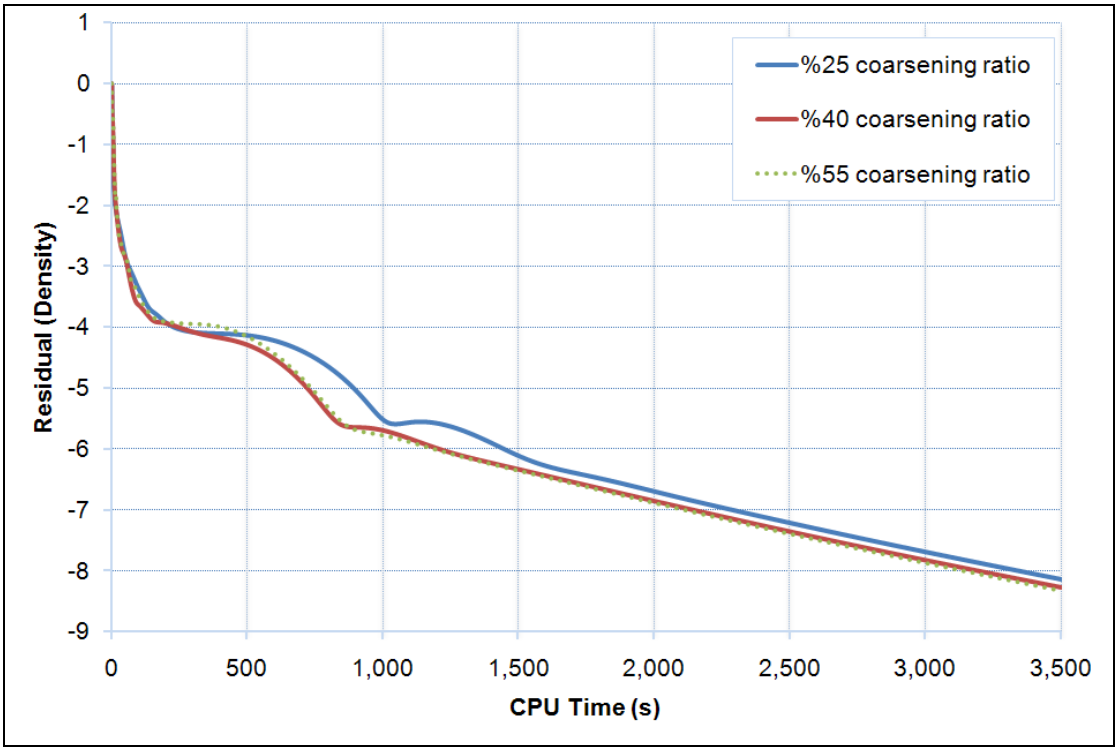


Figure 4.25 Residual histories of MG solutions with coarsening ratios of 25 %, 40 % and 55 % for case 2

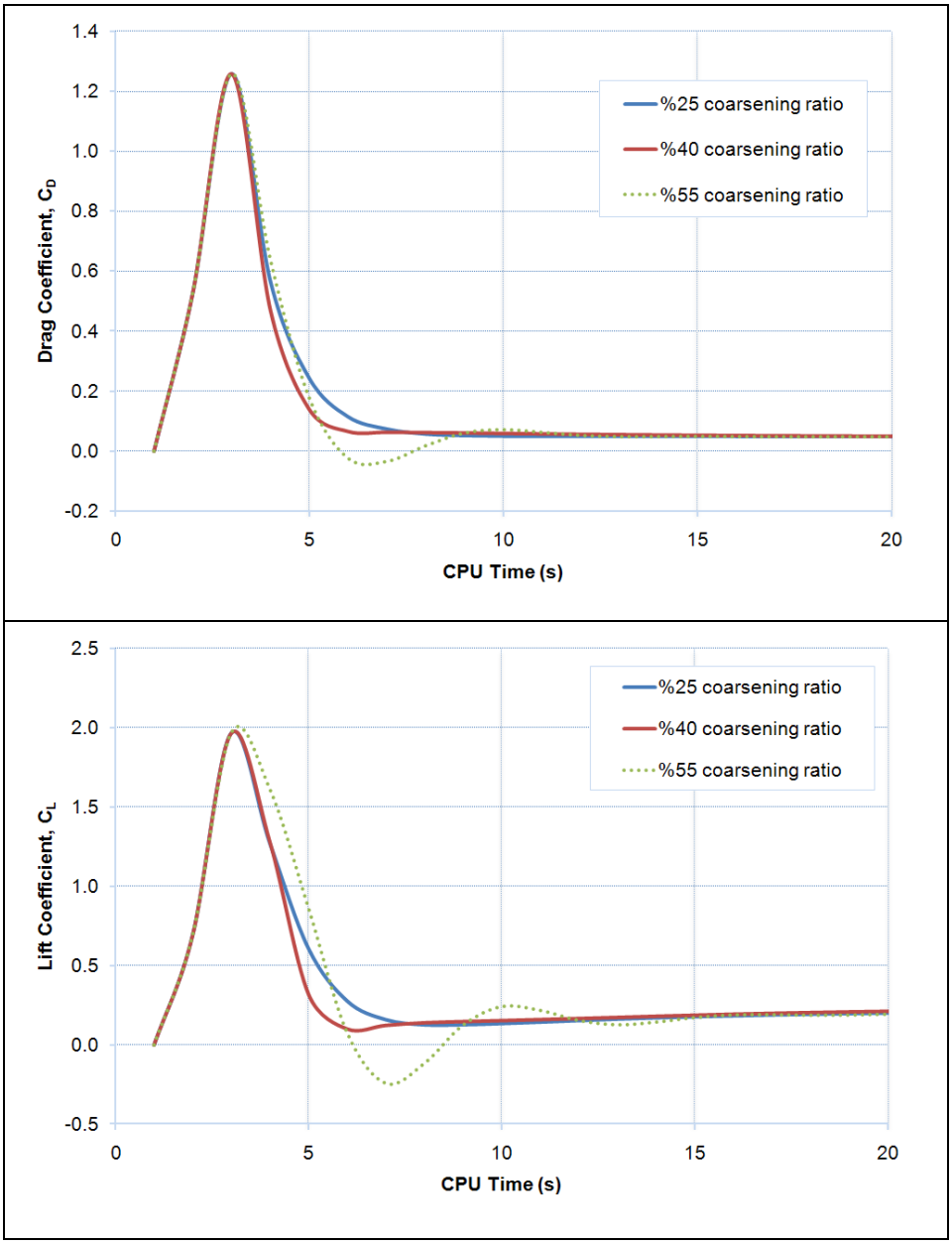


Figure 4.26 The convergence histories of drag and lift force coefficients for grid coarsening ratios of 25 %, 40 % and 55 % for case 2

4.2.4 Effect of Level Number on MG Efficiency

The efficiency of coarse grid level number on MG applications is investigated in this section by using the coarse grid levels with the coarsening ratio of %40. The laminar flow solutions are obtained with 2, 3, 4 and 5 grid levels using V-cycle MG at a low Reynolds number of 10000, a Mach number of 0.1 and an angle of attack of 3°. The convergence histories of the solutions in terms of the variation of the normalized density residual and the load convergence histories are shown in Figure 4.27 and Figure 4.28 respectively. It is seen that MG solutions with different number of coarse grid levels converge approximately at the same CPU time. 4-level and 5-level MG solutions give similar characteristics in both residual and load convergence histories except during the initial iterations. Finally, using high number of coarser grid levels prevents the oscillatory characteristics coming from fine grid solution.

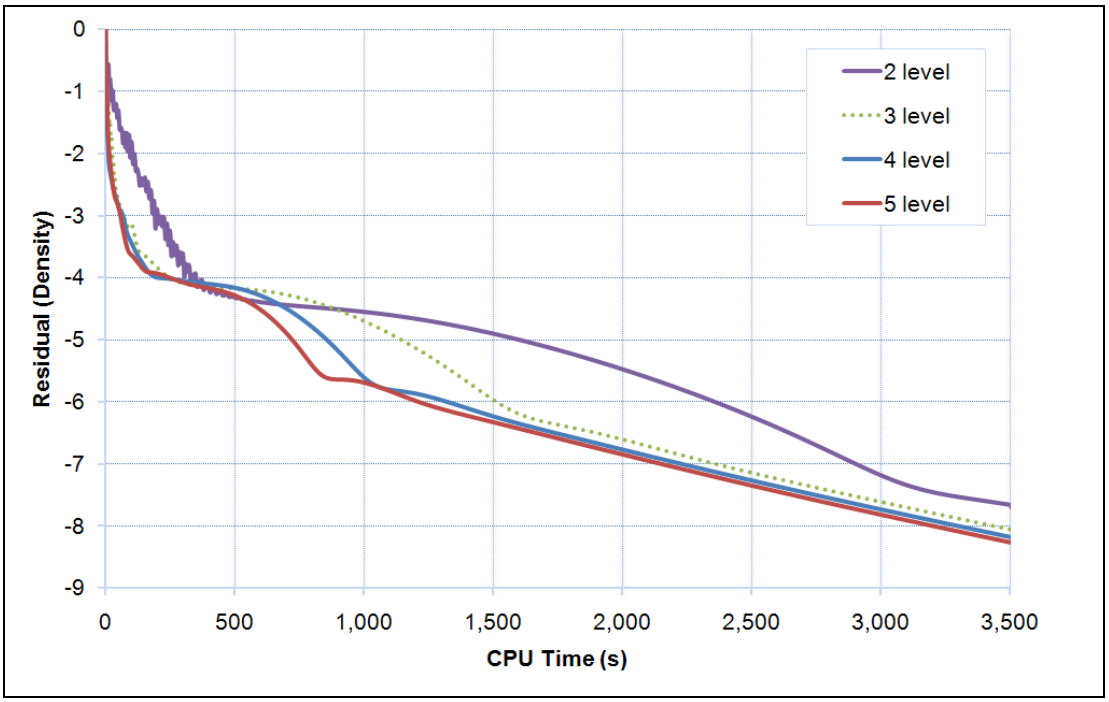


Figure 4.27 Residual histories of V-cycle MG solutions with different number of coarse grid levels for case 2

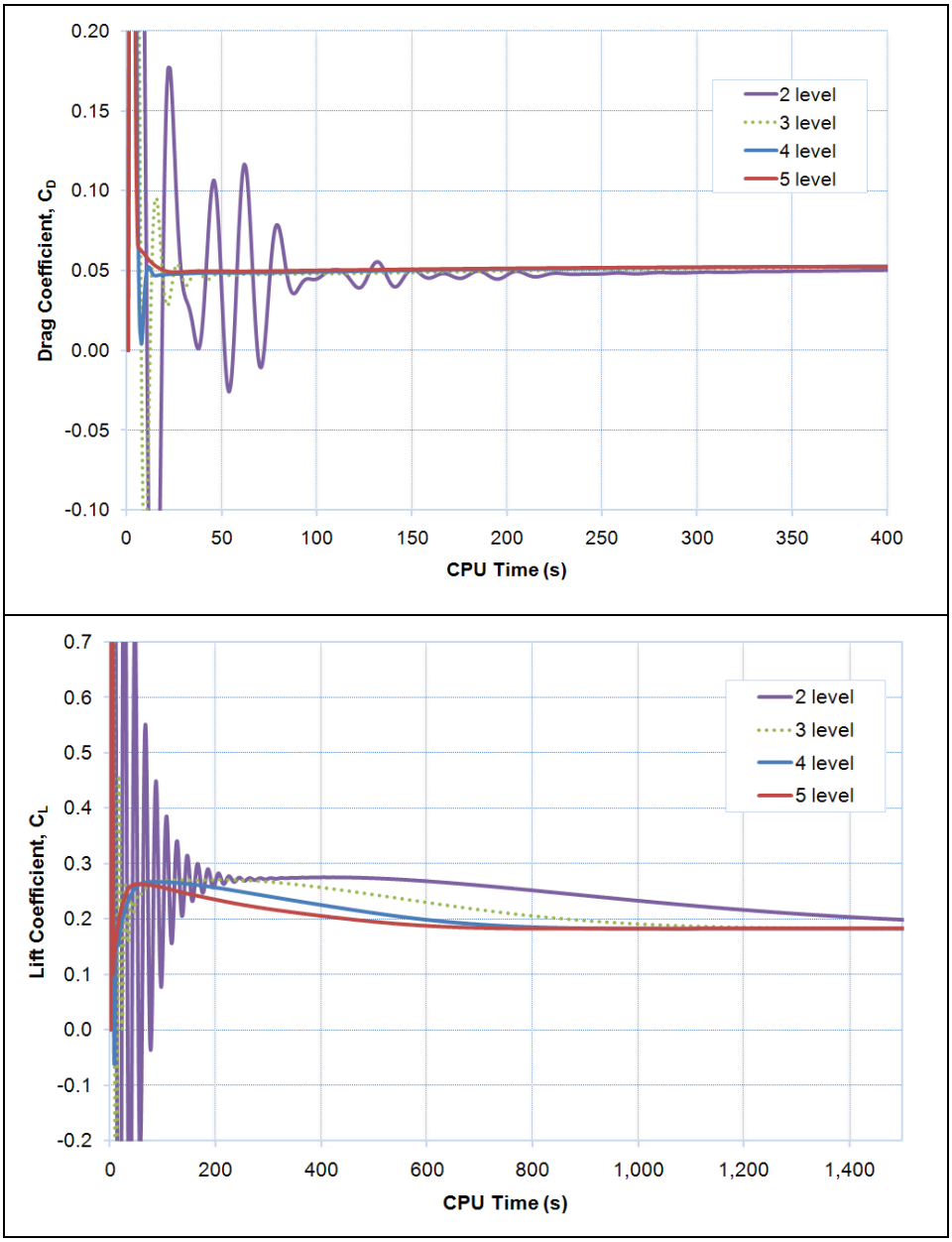


Figure 4.28 The convergence histories of drag and lift force coefficients with different number of coarse grid levels for case 2

4.2.5 Effect of Time Step Number at Coarse Grid Levels on MG Efficiency

The efficiency of iteration count at coarse grid levels is investigated in this section. The alternatives are investigated in two groups. In the first group, the iteration count is kept the same for fine and coarse grid levels. In the second group, the iteration count is the same in the coarse level grids, but, it is doubled, twice iteration count at the fine grid level. The iteration count alternatives are chosen as 5, 10, 20, 30 and 40 iteration counts at coarse levels.

The laminar flow solutions are obtained for all cases like previous sections using V-cycle MG. The MG solutions with iteration counts of 30 and 40 in the first group (i.e. 30 equal time step at fine and coarse grid levels) did not converge. Therefore these alternatives are eliminated. The convergence histories of the solutions in terms of the variation of the normalized density residual are presented in Figure 4.29.

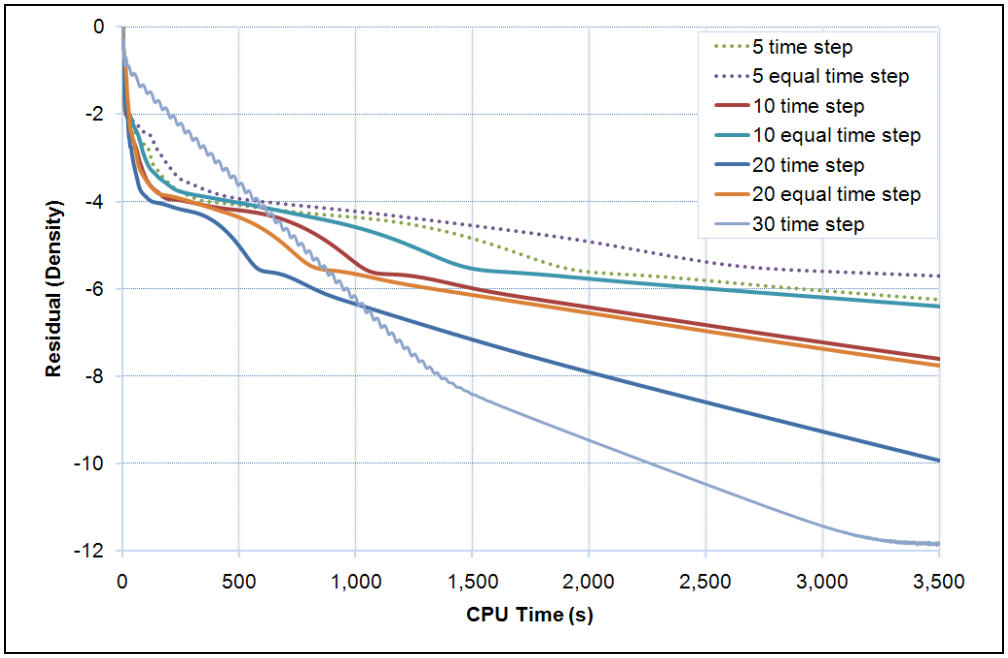


Figure 4.29 Residual histories of MG solutions for a variety of iteration count at coarse grid levels for case 2

It is seen that, the second group solutions (with the number of iterations being doubled at the fine grid level) are more preferable as long as the convergence of the density residual is considered. The initial responses of MG solution with 20 iterations at coarse grid levels and 40 iterations at the fine grid level is the most efficient when compared to the other alternatives. Although 30 time step alternative converges (reaches normalized density residual of -12) more rapidly, it has oscillatory characteristics and seems to be less robust.

Then, the load convergence histories for all alternatives are investigated and the convergence histories of drag and lift forces are shown in Figure 4.30 by focusing their initial behavior. It is seen that 30 iterations show different character than the other alternatives especially for the convergence of the drag force. Solutions with 10 and 20 iterations show approximately the same character for the convergence of the drag force. However, for the lift force, MG solution with 20 time iterations accomodates oscillations over the initial part of convergence histories. The most efficient iteration count for MG efficiency is found as 10 time steps at coarse grid levels.

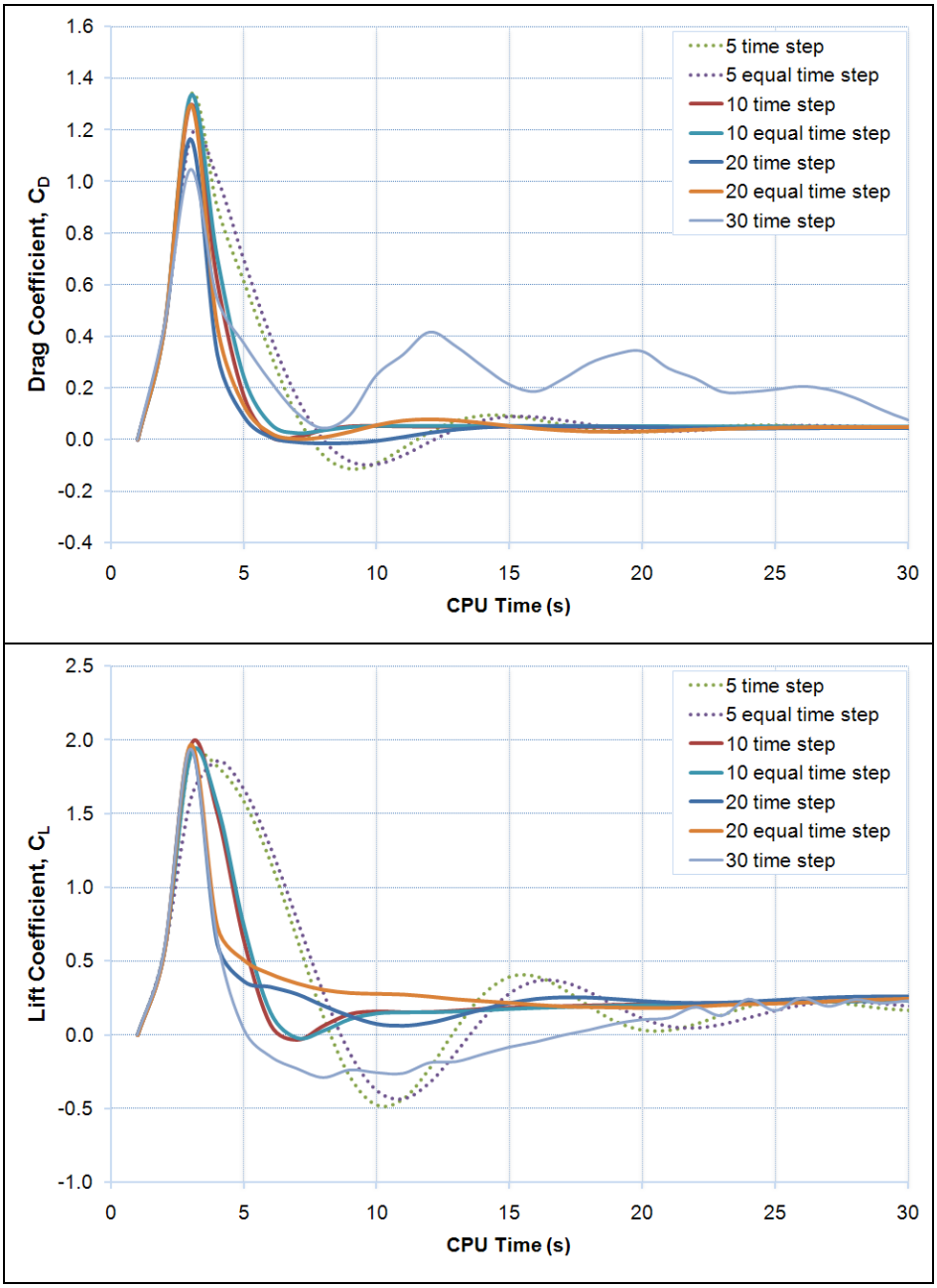


Figure 4.30 The convergence histories of drag and lift force coefficients with a variety of iteration counts at coarse grid levels for case 2

4.3 Case3: A Turbulent Flow Solution over RAE2822 Airfoil

4.3.1 Grid Coarsening

The final 2D test case is a hybrid/unstructured grid over RAE2822 airfoil containing 89914 nodes and 130336 cells having a y+ value of 1, as shown in Figure 4.31. A sequence of four coarse hybrid grids around a RAE2822 airfoil is generated with again maximum coarsening ratio of about 40 %. The coarse level grids for the MG application have 23638, 5323, 1853 and 679 cells. The highest level coarse grid is demonstrated in Figure 4.32.

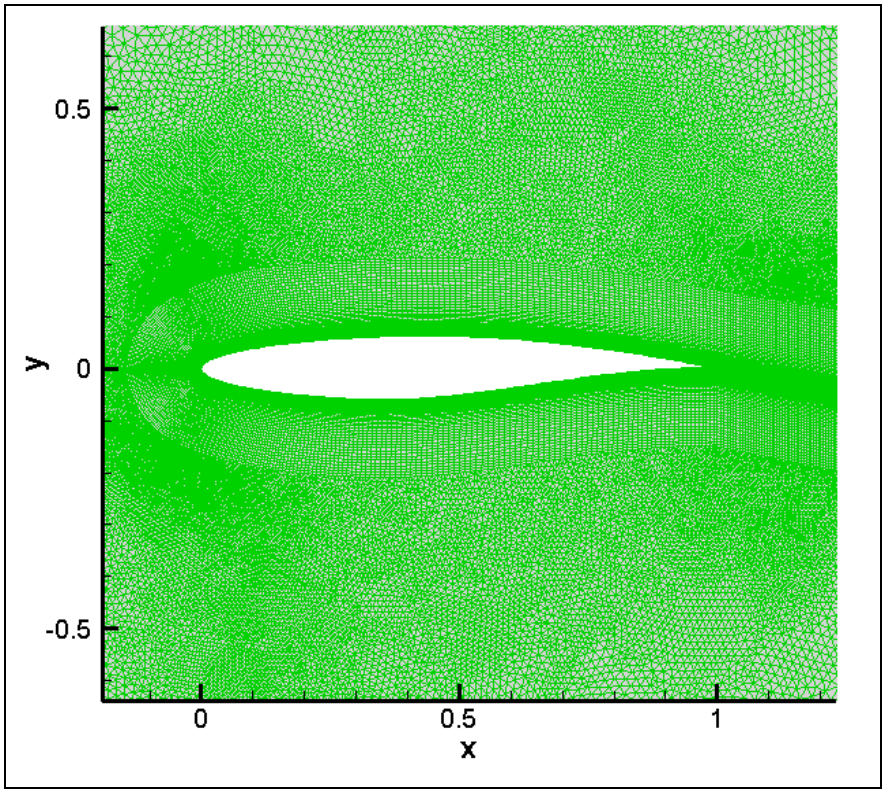


Figure 4.31 The hybrid / unstructured grid for case 3.

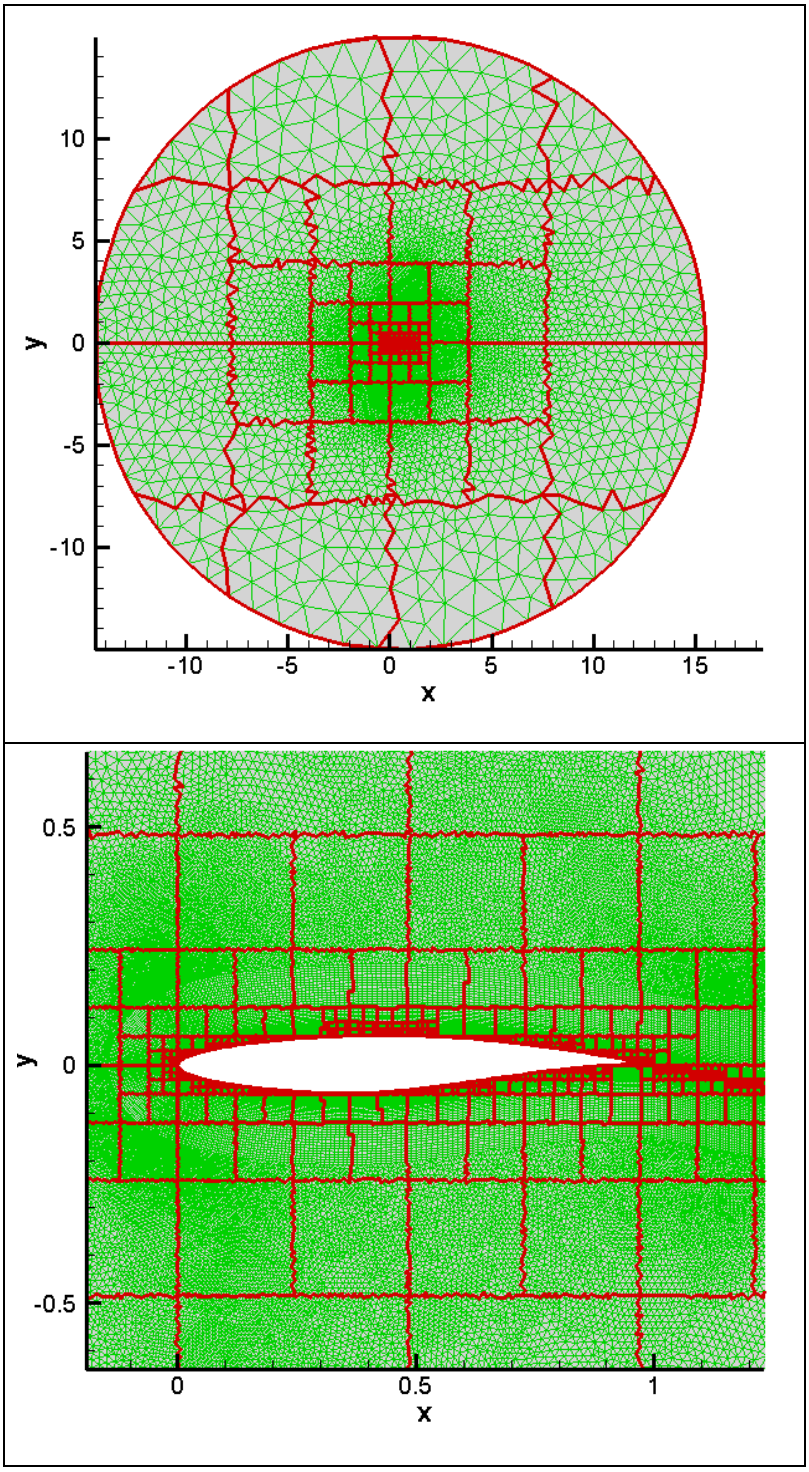


Figure 4.32 The highest level coarse grid for case 3

Like the case studies in the previous sections, the coarse meshes have high quality cells with aspect ratios of about unity similar to a Cartesian type grid.

4.3.2 MG Efficiencies on Turbulent Flow Solution

The flow field around the RAE 2822 airfoil has been computed for a free stream Mach number of 0.729, an angle of attack of 2.31° and Reynolds number of 6.5 million. The single grid and FMG solutions are obtained using first order flux computations with a CFL number of 0.1 using one-equation Spalart-Allmaras turbulence model. The pressure distribution around RAE2822 airfoil is presented in Figure 4.33 and compared with the experimental values taken from validation cases given at www.cfd-online.org web site. It is seen that, the first order explicit solution could not capture the shock location very accurately, which may be attributed to the fully turbulent first order solution and the performance of the one-equation turbulence model for transonic flows.

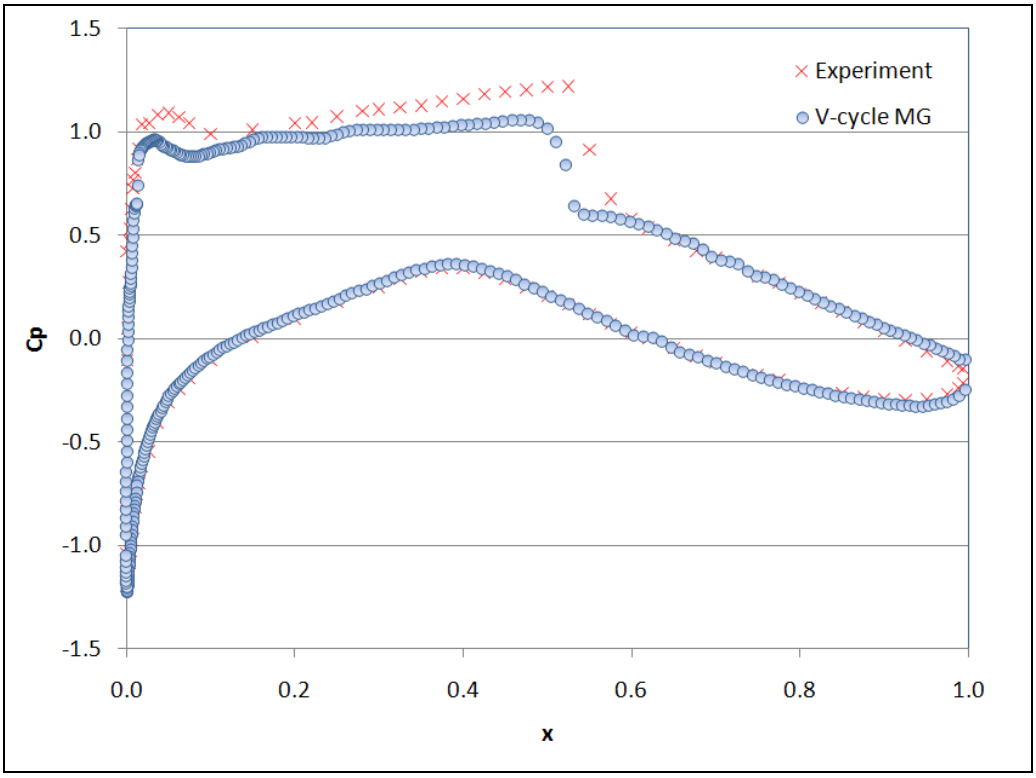


Figure 4.33 Cp distributions around RAE2822 airfoil for case 3

Then, the same solution is obtained by applying V-cycle MG algorithm with 20 time steps at fine grid level and 10 time steps at coarse grid levels. The flow solutions at coarse grids for supplying the correction term to the fine grid solution are obtained by using inviscid solver.

The convergence histories of the transonic flow solutions in terms of the variation of the normalized density residual are shown in Figure 4.34 and compared with the convergence history of the single grid solution. It is apparent that V-cycle MG solution when compared to baseline solution converges faster at least by a factor of 5.

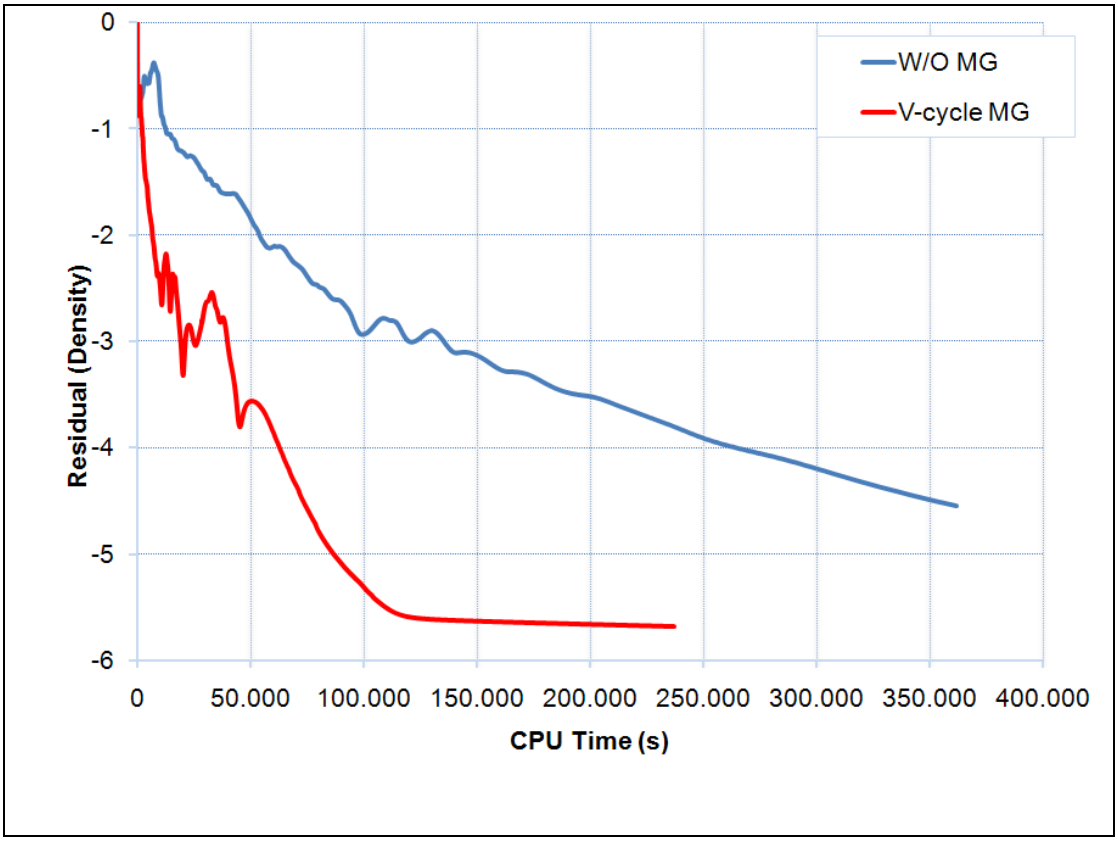


Figure 4.34 Residual histories of the single grid and MG solutions for case 3

A similar convergence rate is observed in the variation of drag and lift force coefficients when the solutions with and without MG are compared as shown in Figure 4.35.

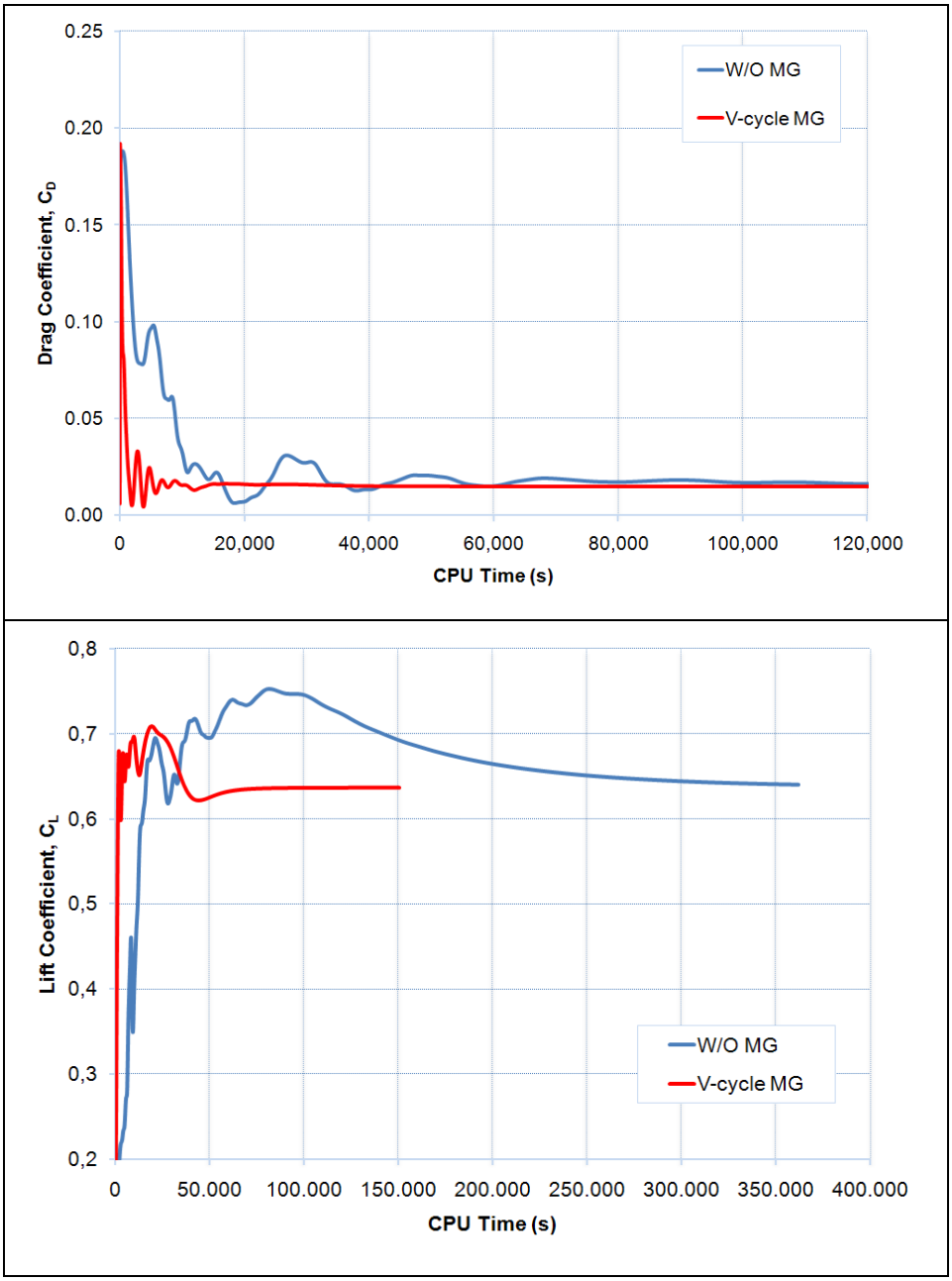


Figure 4.35 The convergence histories of drag and lift force coefficients for case 3

Solution based on FMG is taken as a reference value and CPU times to reach 20 %, 10 %, 5 %, 1 % and 0.1 % error bands are recorded. The lift force coefficient using single grid did not converge exactly after 850000 time iteration. The speed-up ratios

for drag and lift coefficients obtained by V-cycle MG and FMG are shown in Figure 4.36. To reach 5% error band, the drag coefficient has approximately 4 times, the lift coefficient has approximately 6 times faster convergence ratio than the single grid solution in both V-cycle MG and FMG solutions.

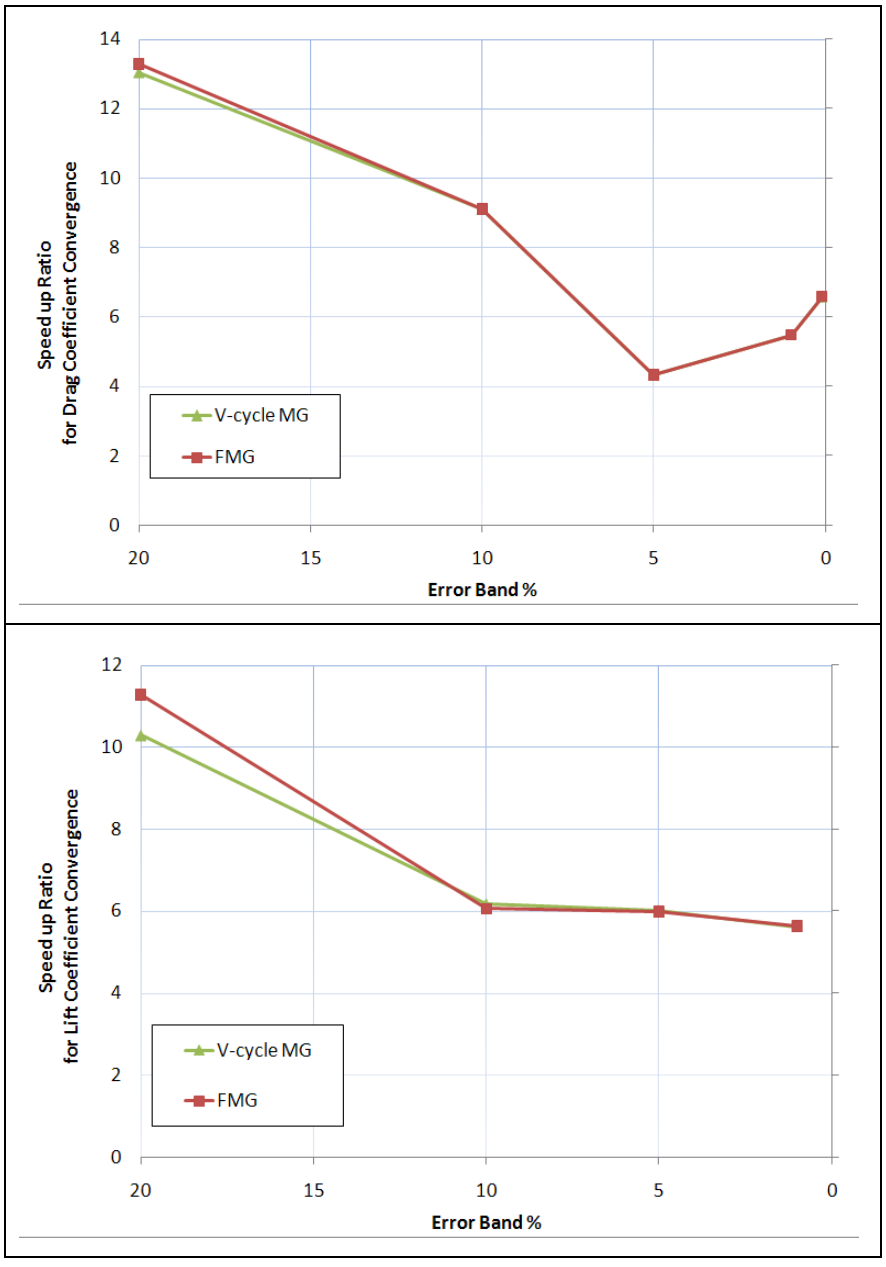


Figure 4.36 Drag and lift coefficient convergence ratios with respect to time iterations and CPU time for case 3

4.4 Case4: An Inviscid Flow Solution over ONERA M6 Wing

4.4.1 Grid Coarsening

A very well known model, ONERA M6 wing, is used in many papers as a test case. Pandya and Frink also used ONERA M6 wing geometry to obtain coarse level grids for cell centered based tetrahedral unstructured grids using their volume agglomeration technique [18]. This technique is described in two steps. In the first step, all the fine grid cells attached to the body surface or a far-field boundary are identified and merged with its neighboring cells to form a new coarser cell. After all of these prioritized boundary cells are assigned to a coarser cell, an unassigned fine grid cell on the agglomeration front is picked in a random order and merged with its eligible neighbors to form a new coarser cell. The generated coarse level grids are presented Figure 4.37.

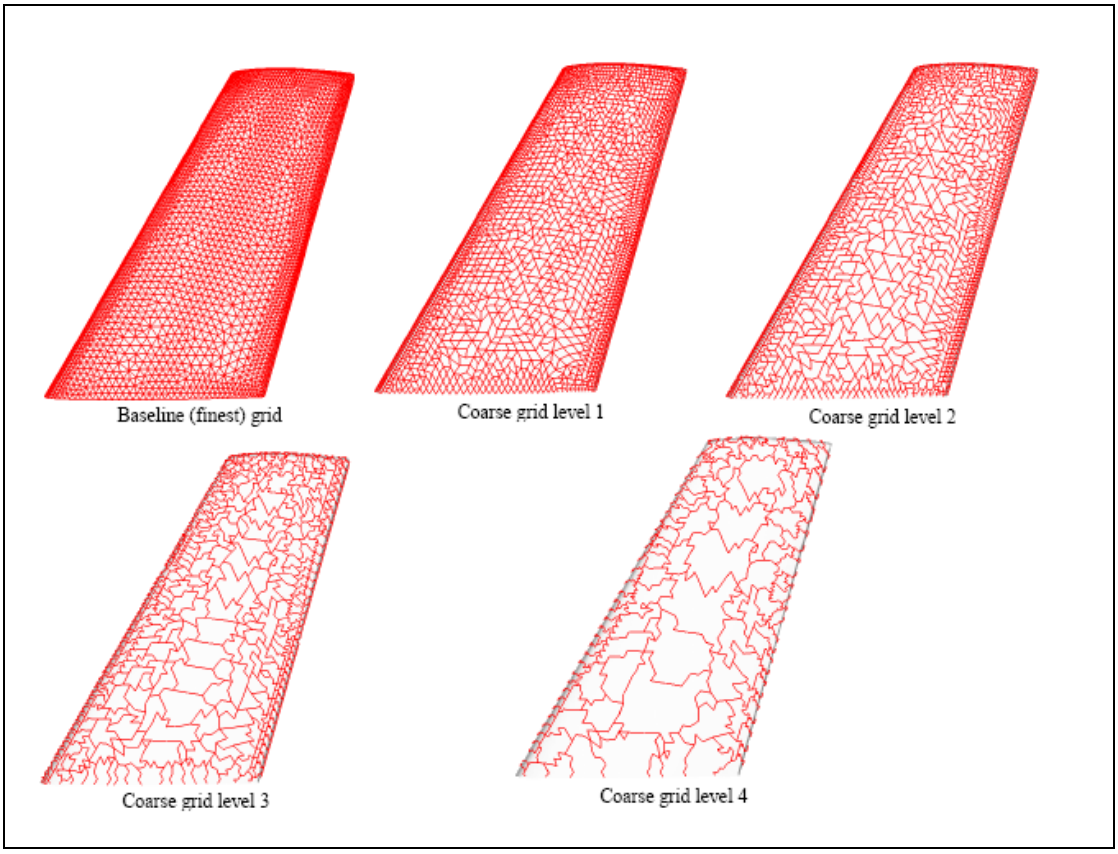


Figure 4.37 Coarse level grids generated by Pandya and Frink for ONERA M6 wing [18]

An automated octree based grid coarsening algorithm developed in this study is used for coarsening the unstructured grid around the same geometry, ONERA M6 wing. The fine grid level, shown in Figure 4.38, contains 255,156 nodes and 1,391,537 cells.

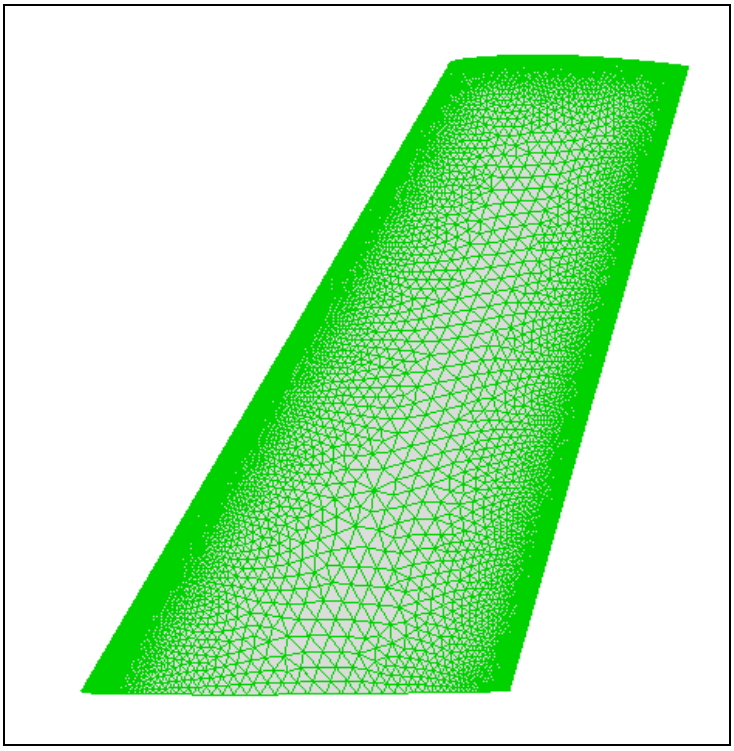


Figure 4.38 Fine grid for case 4

A sequence of four coarser grids is generated with a maximum coarsening ratio of 40 % between the coarse grid levels having 546424, 99851, 31065 and 11456 cells from second to fifth grid levels as shown in Figure 4.39. The connectivity information between grids is obtained from the data tree via this parent / child relationship. The coarser grids for MG levels are compared with the Pandya and Frink's study and it is seen that especially at high level coarse grids, which are generated in this study keep the ratios of mesh sizes at fine grid level by grouping the cells according to location of the octree deepness level. Doing so, the large sized fine meshes become large sized coarse meshes and the number of fine mesh at each coarse meshes are approximately the same. Finally, the octree based coarser grids over the wing have again good aspect ratios and it produced better quality meshes at coarse grid levels with respect to ones that are generated by using a globally coarsening algorithm.

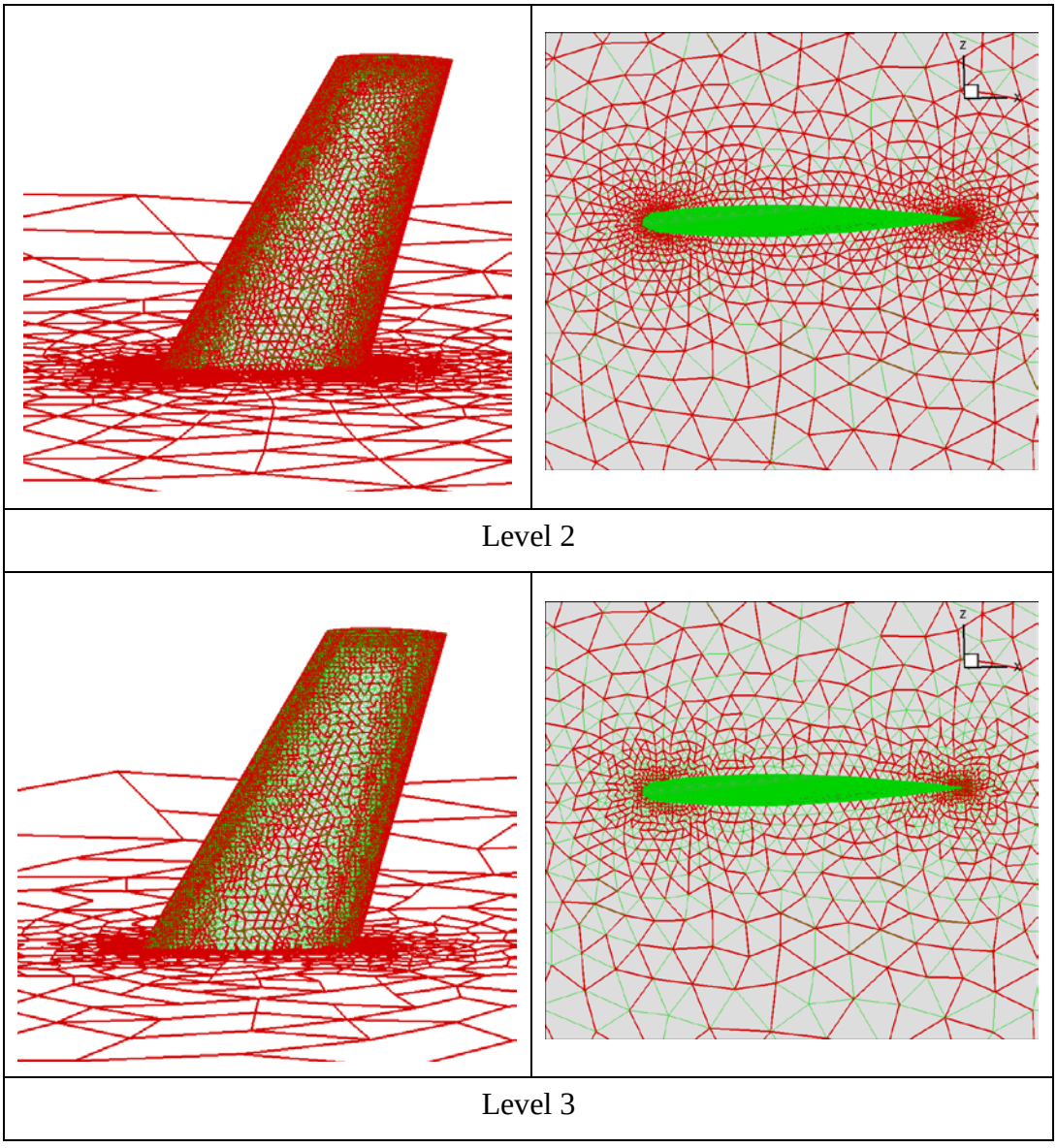
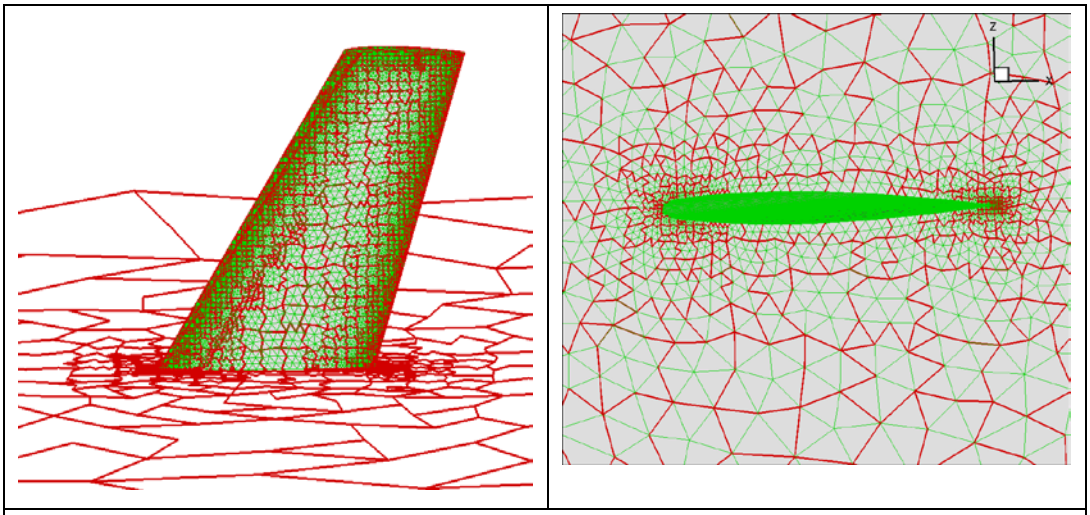
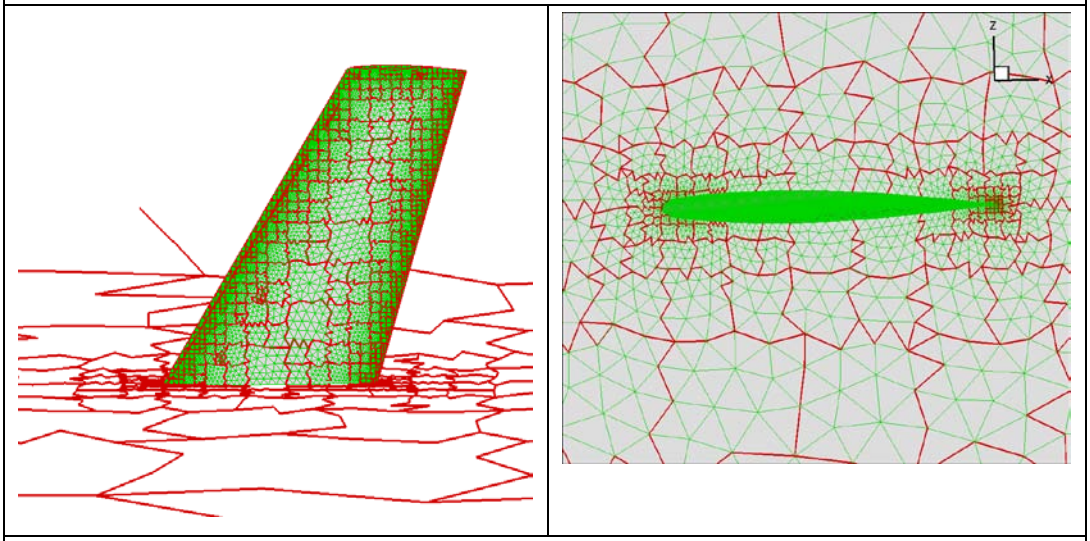


Figure 4.39 Coarse grid levels for case 4



Level 4



Level 5

Figure 4.37 Coarse grid levels for case 4 (continued)

4.4.2 Effect of MG Efficiencies on Inviscid Solution at Subsonic Mach Numbers

The explicit single grid solution at a Mach number of 0.3 and an angle of attack of 0° is obtained by using inviscid, first order flux computations with a CFL number of

0.2. The same inviscid solutions are also obtained using V-cycle, W-cycle and FMG with 20 equal time steps at all grid levels. The Mach number distribution on the wing and on the tip of the wing computed by the single grid is presented in Figure 4.40 and on the symmetry axis in Figure 4.41.

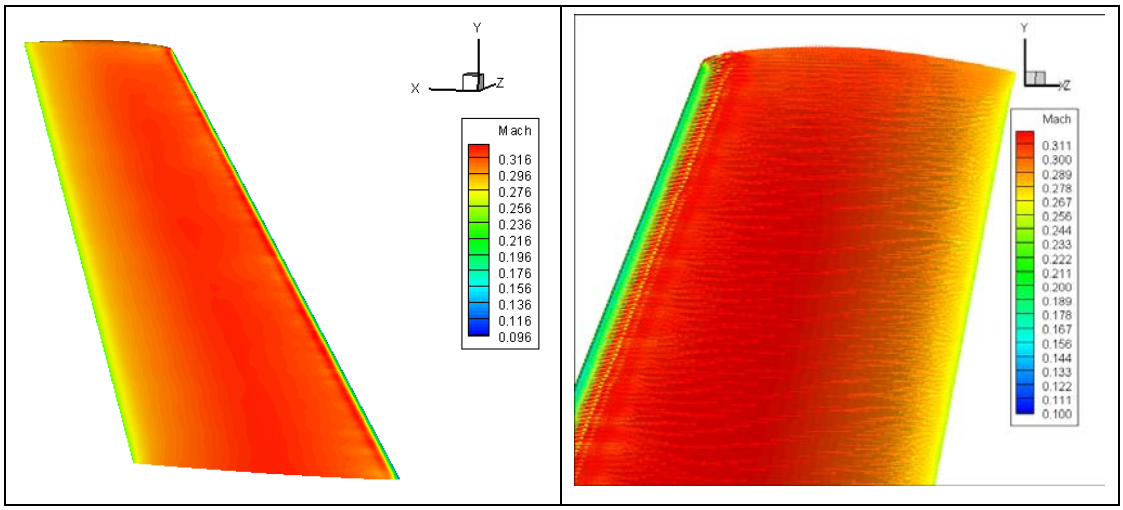


Figure 4.40 Mach number distribution on the ONERA M6 wing for case 4

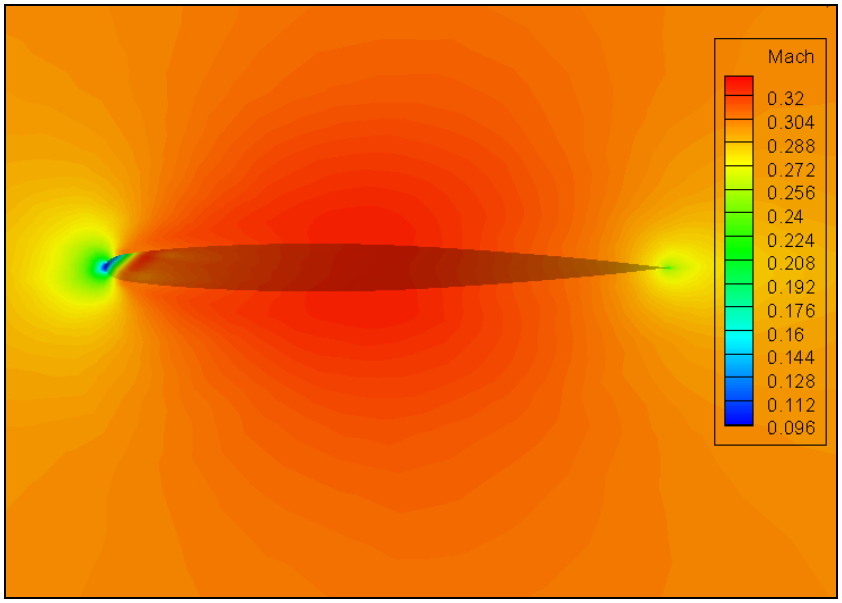


Figure 4.41 Mach number distribution on symmetry axis for case 4

The convergence histories of single grid, V-cycle, W cycle and FMG cycle solutions in terms of the variation of the normalized density residual are presented in Figure 4.42. It is seen that, all multigrid solutions exhibit approximately the same converge rate as expected and converge very rapidly. MG solutions are approximately 5 times faster than the single grid solution. Finally, it is noted that, FMG solution reaches the converged result at the same time with V-cycle MG without any oscillation.

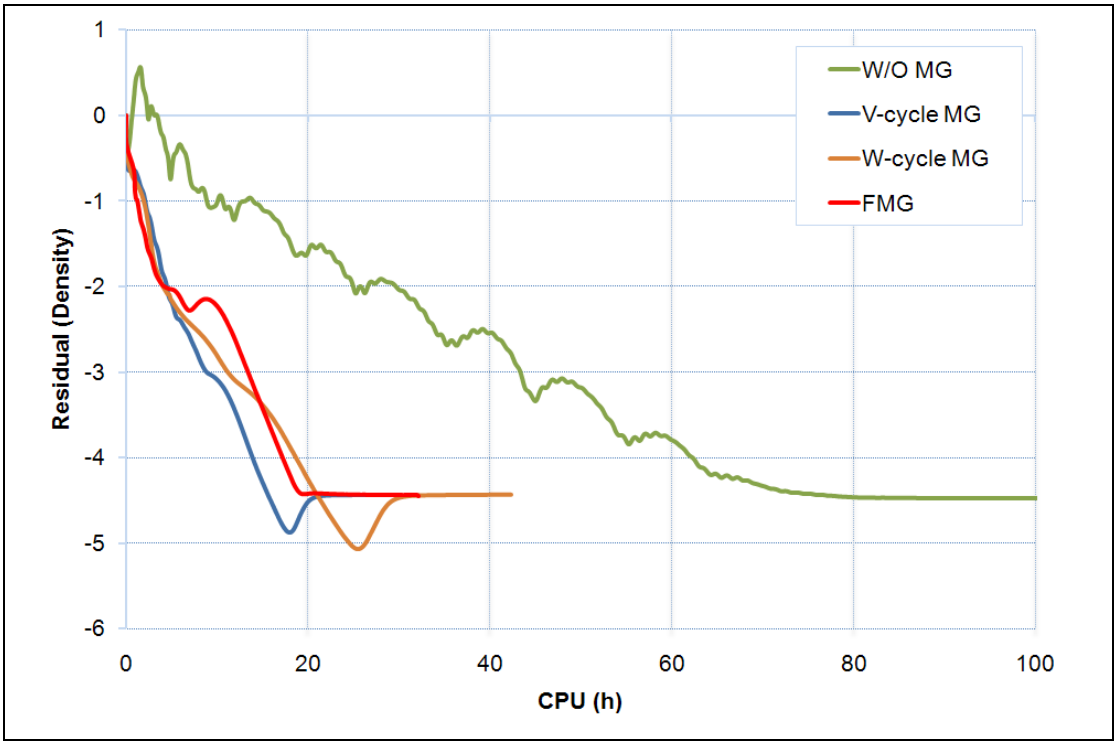


Figure 4.42 Residual histories of single and MG solutions for case 4

Then, the convergence histories of aerodynamic loads given in Figure 4.43 are investigated with and without multigrid solutions using V-cycle, W-cycle and FMG cycling strategies. Like residual convergence characteristics, the convergence histories of aerodynamic loads for MG solutions are similar and reach to at the same values after a few numbers of iterations.

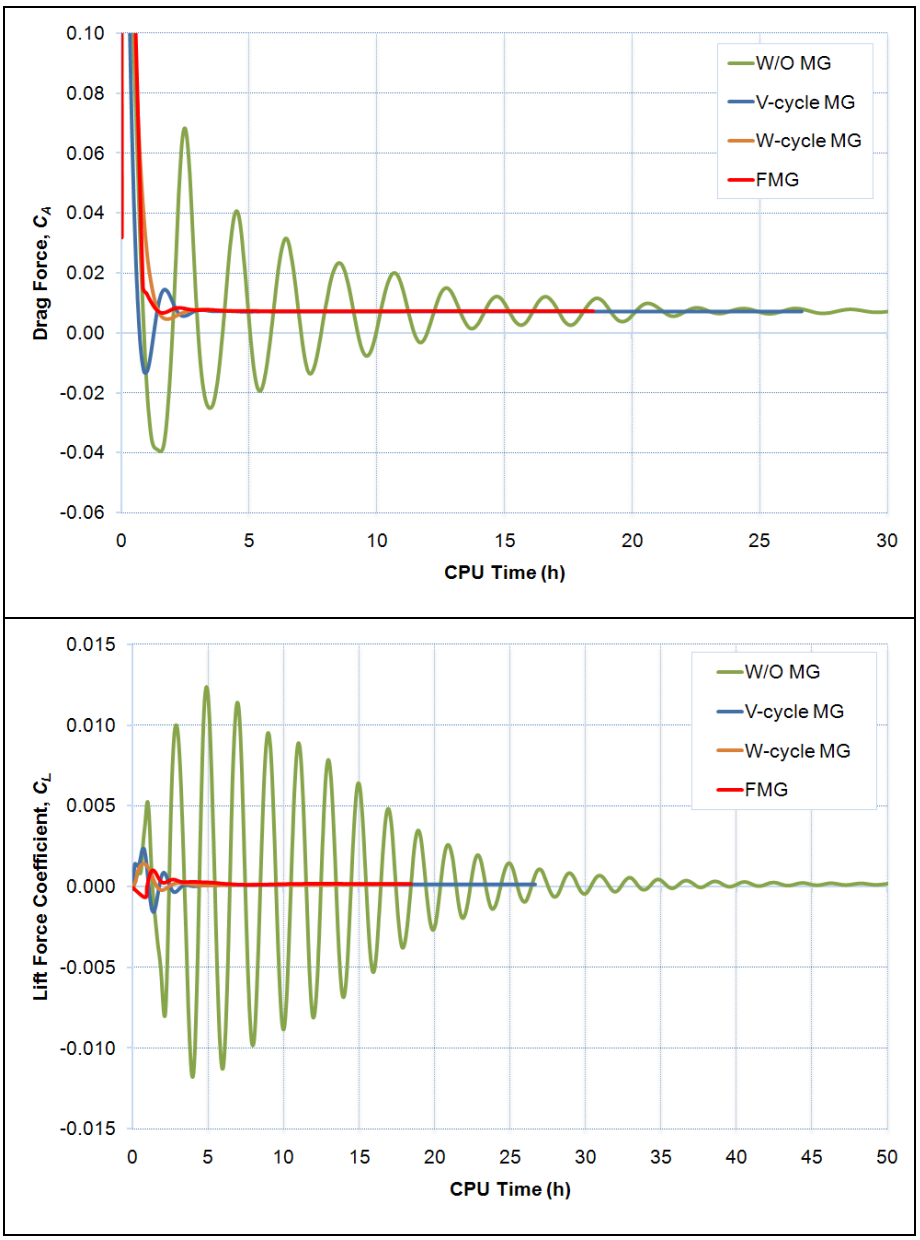


Figure 4.43 Convergence histories of aerodynamic coefficients for case 4

The CPU times required to reach 20 %, 10 %, 5 %, 1 % and 0.1 % error bands of aerodynamic force coefficients with all MG cycling types and the fine grid solution are also analyzed and tabulated in Table 4.3 and Table 4.4

. FMG solution is taken as a reference value and it is seen that the single grid solution did not reach 0.1 % error band in 75,000 time steps. The speed up ratios relative to the time required for the single grid solution are presented for drag and lift force coefficients in Figure 4.45. It is observed that, the other MG cycling strategies excluding the W-cycle reach to 1 % error band in the prediction of force coefficients very fast, about 5 and 8 times faster than the single grid solution for the convergence of drag and lift forces.

Table 4.3 The runtime required to reach 20 %, 10 %, 5 %, 1 % and 0.1 % error bands of drag force coefficient

%	CPU time (seconds)			
	FINE	V-cycle MG	W-cycle MG	FMG
20	78,479	9,149	8,043	4,787
10	99,372	10,062	8,643	9,271
5	107,150	12,334	13,480	13,251
1	166,350	18,596	19,420	15,093
0.1	210,880	30,869	26,703	41,206

Table 4. 4 The runtime required to reach 20 %, 10 %, 5 %, 1 % and 0.1 % error bands of lift force coefficients

%	CPU time (seconds)			
	FINE	V-cycle MG	W-cycle MG	FMG
20	185100	30869	57616	33736
10	203390	38966	66664	35926
5	218310	45780	76606	46080
1	254630	66625	104850	55583
0.1	-	101530	142560	88424

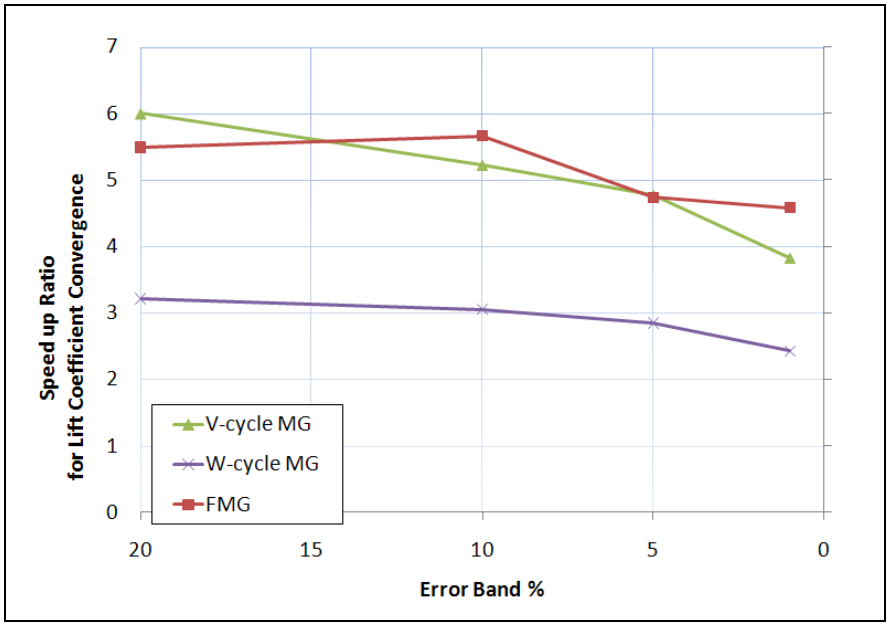


Figure 4.44 Speed up ratios for the convergence of drag and lift coefficient to reach 20 %, 10 %, 5 %, 1 % and 0.1 % error band for case 4

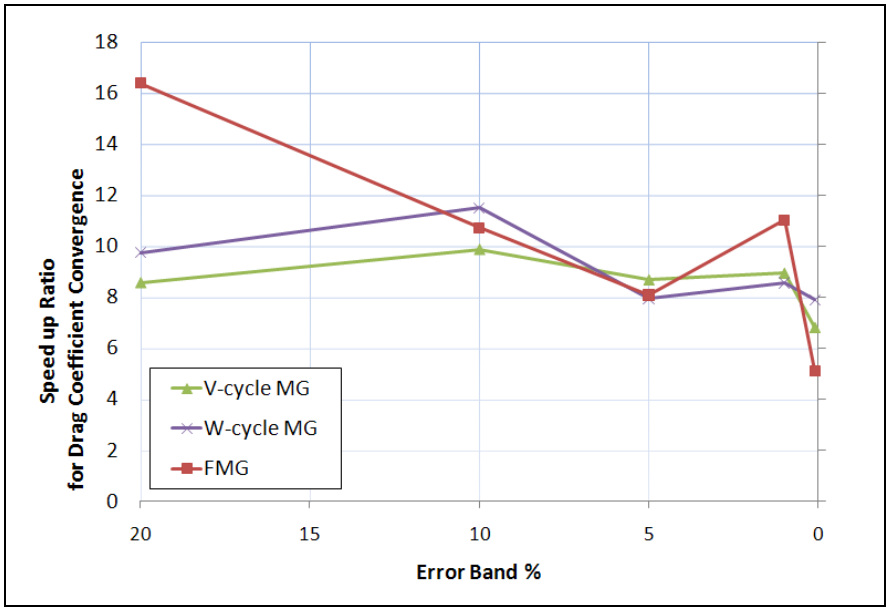


Figure 4.45 Speed up ratios for the convergence of drag and lift coefficient to reach 20 %, 10 %, 5 %, 1 % and 0.1 % error band for case 4

4.4.3 Effect of Iteration Time Step at Coarse Grid Level on MG Efficiencies

The efficiency of iteration count at coarse grid levels is investigated for three dimensional inviscid solutions. The iteration step sizes are taken equivalent for fine and coarse grid levels and MG solutions are obtained by 10, 20, 30 and 40 iteration steps using V-cycle strategy. The convergence histories of the solutions in terms of the variation of the normalized density residual are presented in Figure 4.46. It is seen that, except for 10 iteration steps, all V-cycle MG solutions converge approximately at the same time.

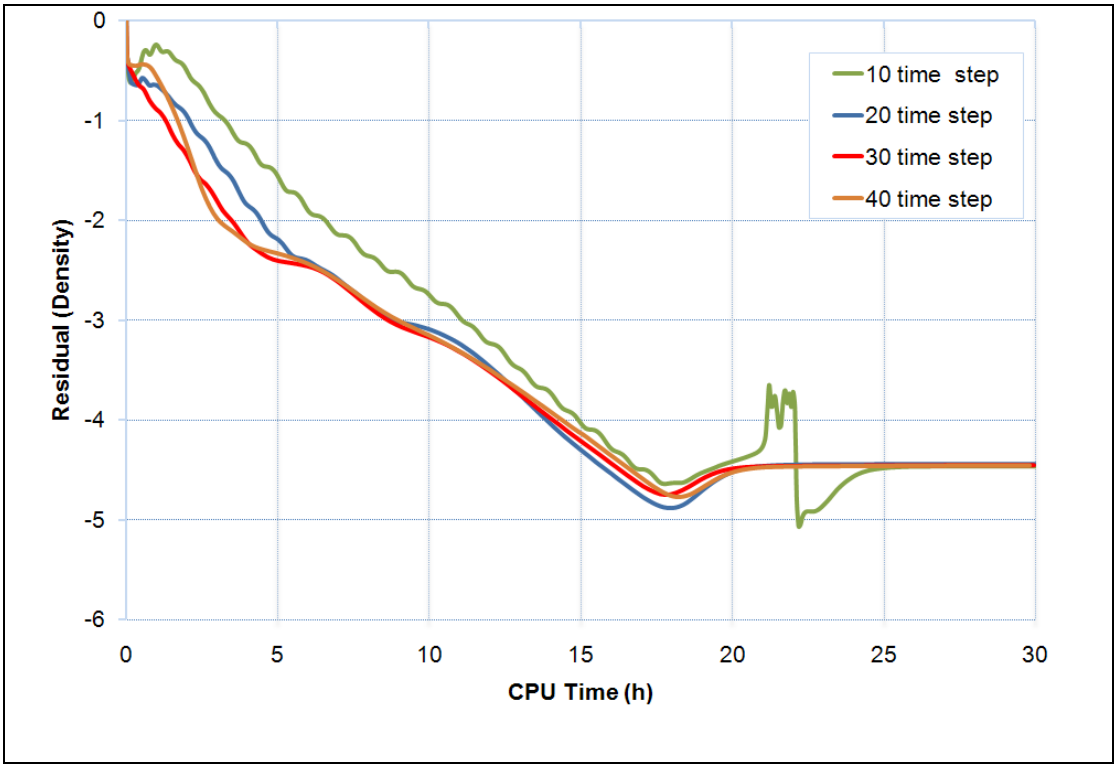


Figure 4.46 Residual histories of MG solutions with a variety of iteration count at coarse grid levels for case 4

Then, the load convergence histories for all alternatives are investigated and the convergence histories of drag and lift force are shown in Figure 4.47 by focusing the initial behavior of the solution. It is seen as the number of iteration steps are increased, the number of oscillations and overshoots before convergence decreases.

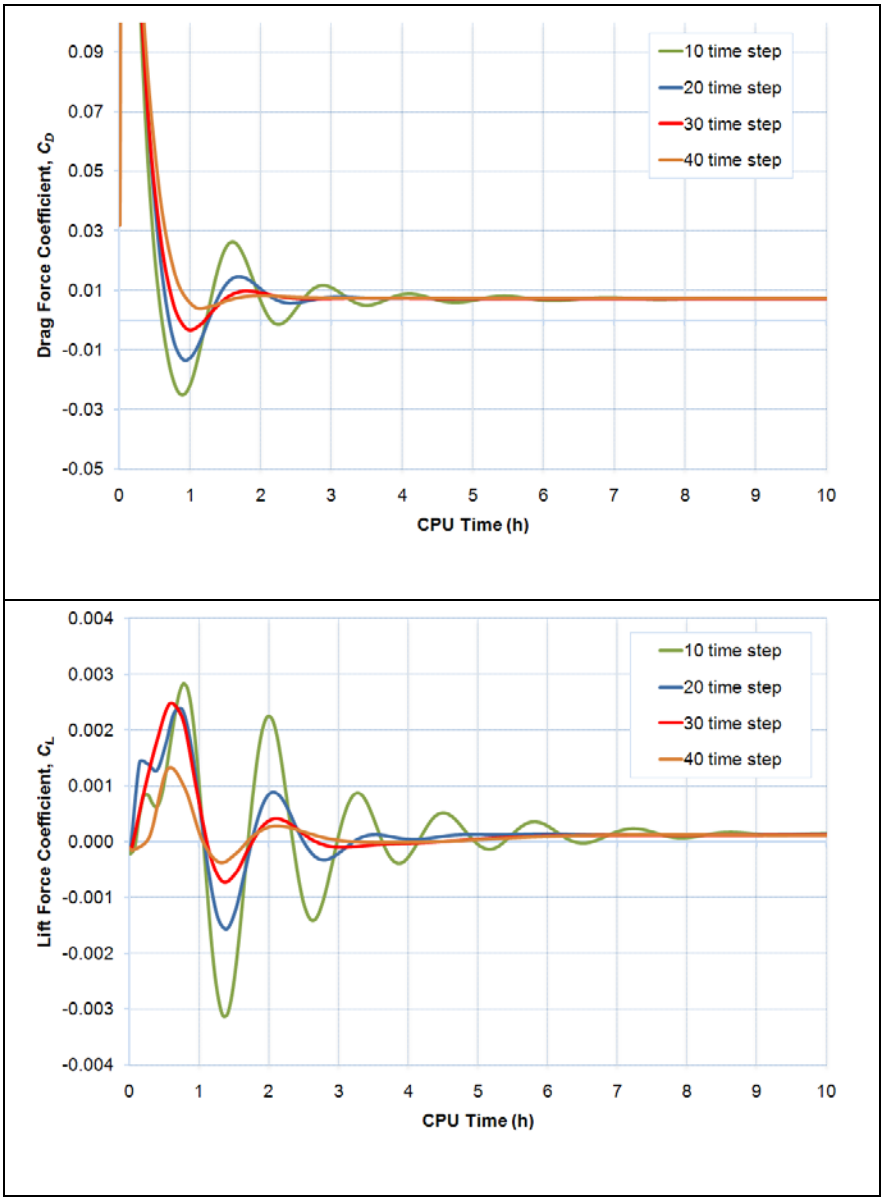


Figure 4.47 The convergence histories of drag and lift force coefficients for a variety of iteration counts at coarse grid levels for case 4

4.5 Case 5: Laminar Flow Solution over a Wing with NACA0012 Airfoil

4.5.1 Grid Coarsening

An automated octree based data structure is next implemented on 3D fully structured grid over a 0.25 chord wing with NACA0012 airfoil section. The fine grid contains 679560 nodes and 640000 cells, as shown in Figure 4.48. A sequence of four coarse hybrid grids is generated with again a maximum coarsening ratio of about 40%. The coarse level grids for MG application have 152678, 60103, 16080 and 4506 cells. Coarse grid levels at the symmetry axis are presented in Figure 4.49, while the surface meshes of fine and the highest level coarse grid are presented in Figure 4.50.

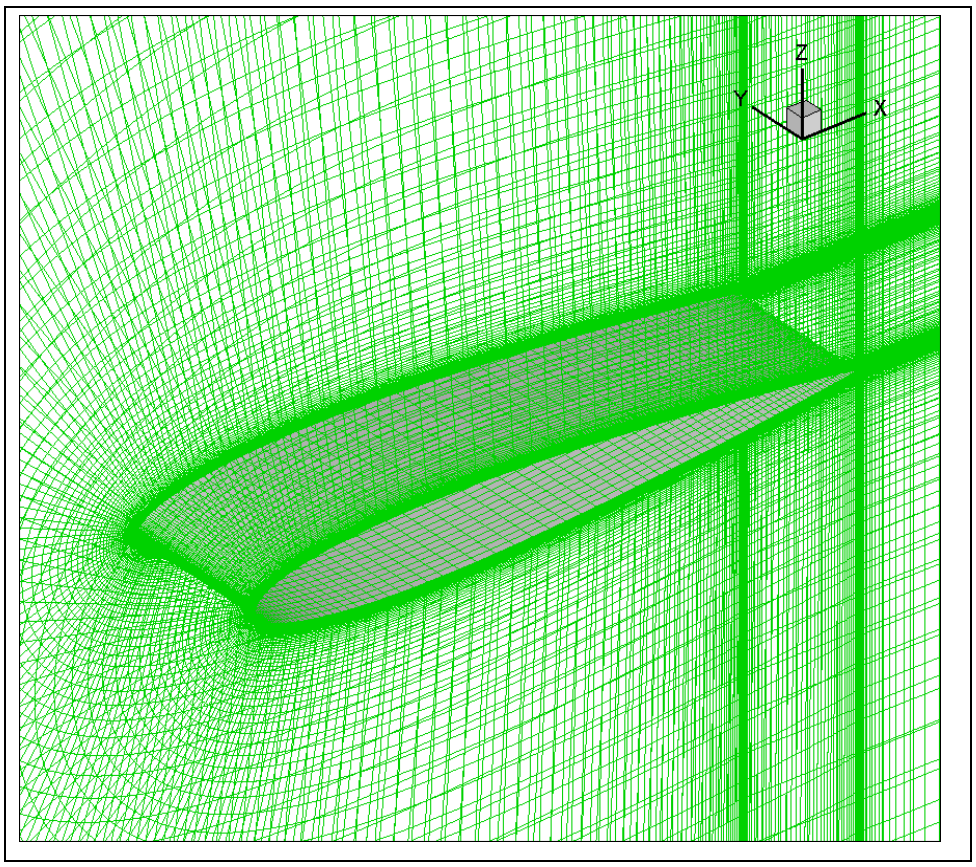


Figure 4.48 The fine grid for case 5

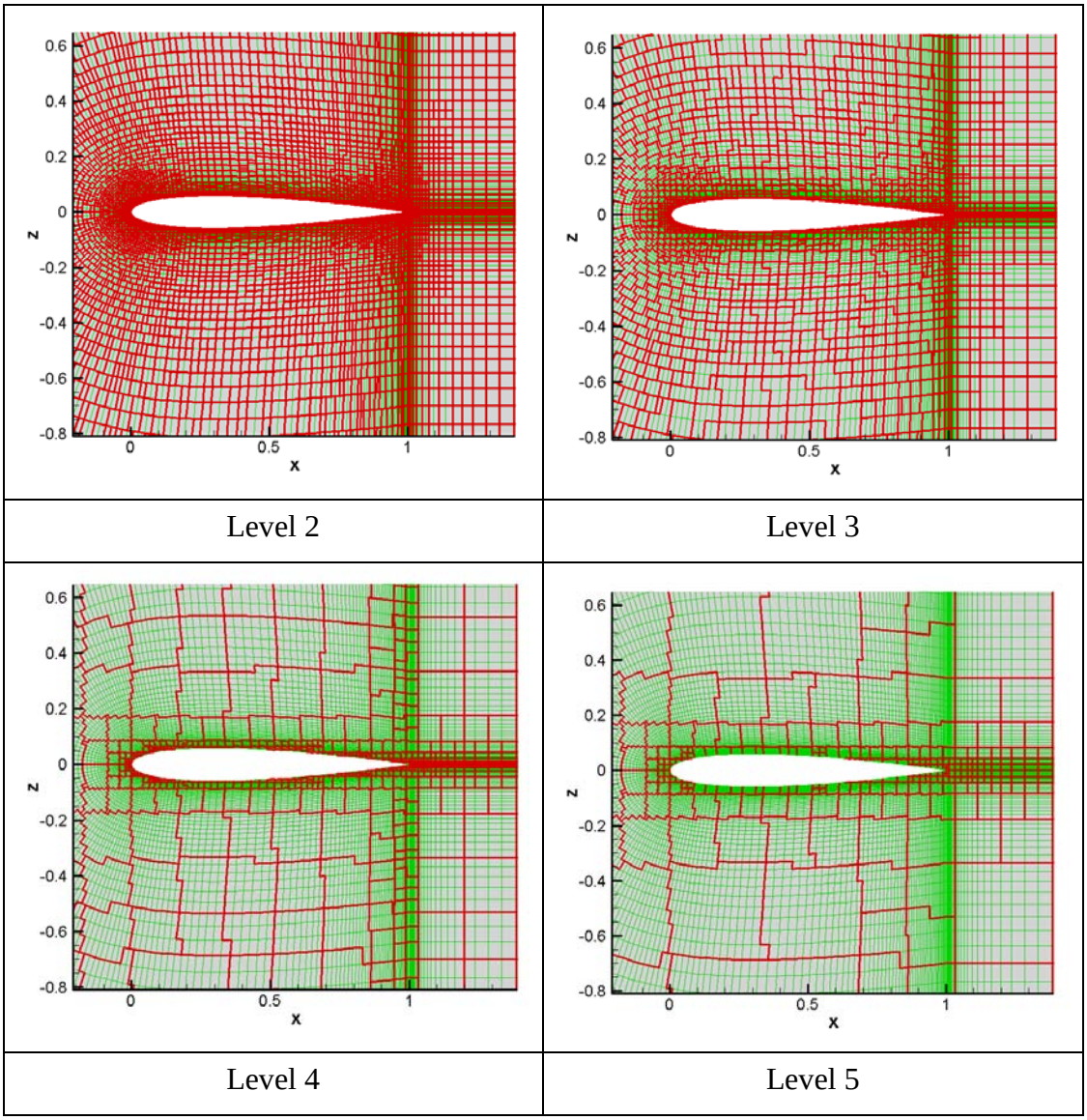


Figure 4.49 Coarse grid levels at the symmetry axis for case 5

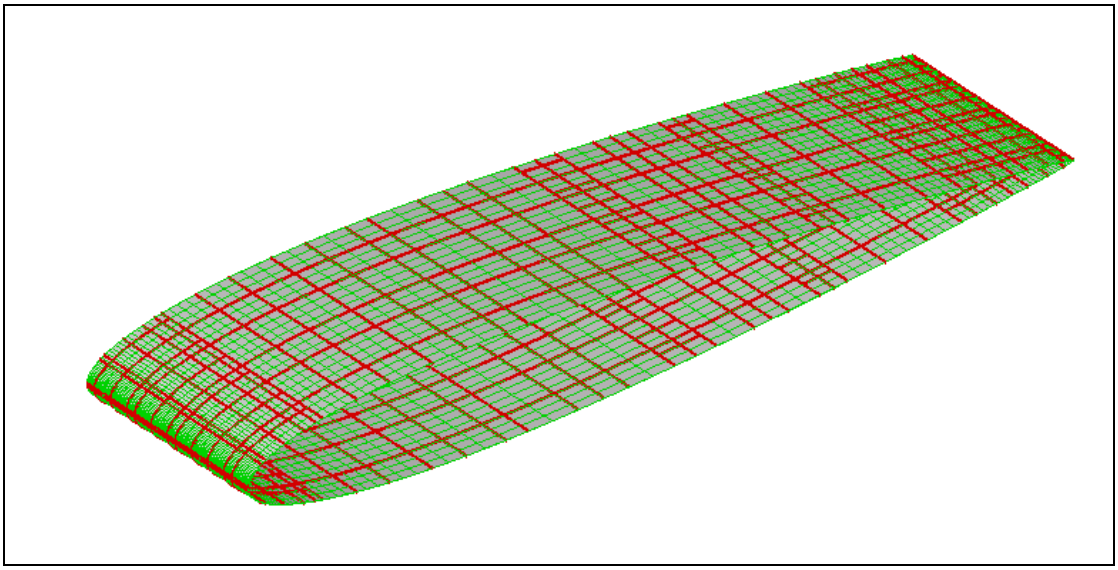


Figure 4.50 The surface meshes of fine and the highest level coarse grid for case 5

4.5.2 Effect of the MG Efficiencies on Laminar Flow

The single and MG solutions over a wing with NACA0012 airfoil are obtained at a low Reynolds number of 10000, a Mach number of 0.1 and an angle of attack of 3° with CFL number of 0.1. V-cycle, W-cycle and FMG solutions are obtained with 20 time steps at all grid levels including the fine grid. For coarse grid iterations, the inviscid solver is used but the calculated laminar face fluxes at fine grid level are added as a constant term at active faces during flux calculations at the coarse grid level.

The Mach number distribution and boundary layer velocity profile, which are computed by the single grid and MG solutions, are similar to each other and they are presented in Figure 4.51.

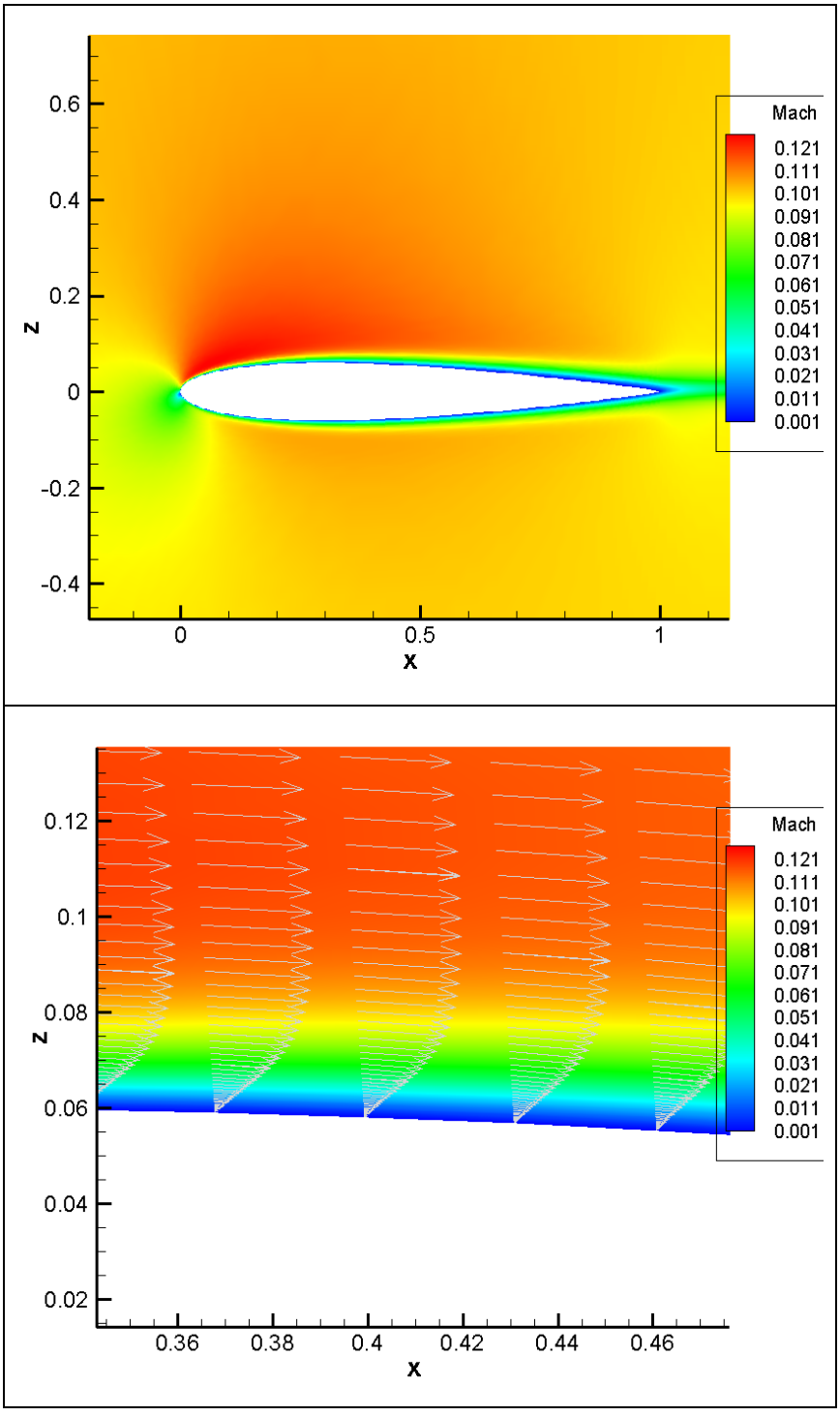


Figure 4.51 The Mach number distribution and boundary layer velocity profile for case 5

The convergence histories of the solutions in terms of the variation of the normalized density residual are shown in Figure 4.52 and compared with the convergence history of the single grid solution. It is realized that, the residual convergence behavior in two dimensional and three dimensional laminar solutions are similar to each other. All cycling strategies give approximately same convergence on laminar solutions except the initial stage of W-cycle.

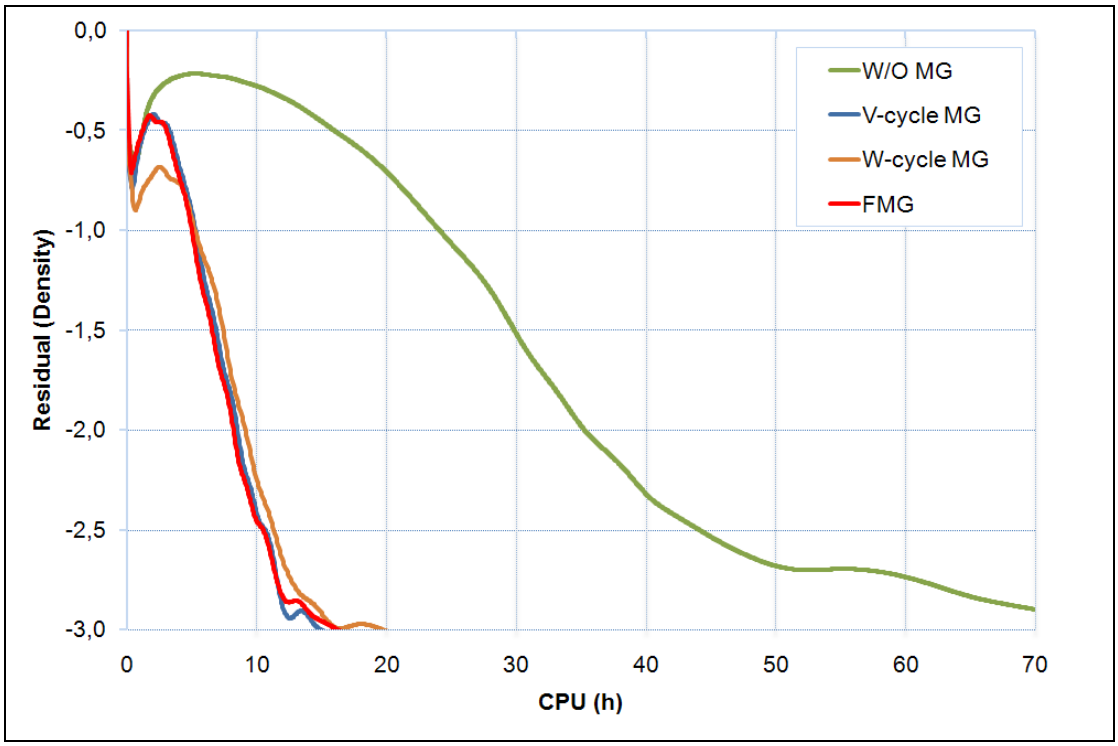


Figure 4.52 Residual histories of single and MG solutions for case 5

Then, the cycling strategies are investigated according to aerodynamic load convergence in Figure 4.53 for the solutions with and without MG. Like residual convergence characteristics, the histories of aerodynamic loads convergence are similar and reach the same values after a few number of iterations.

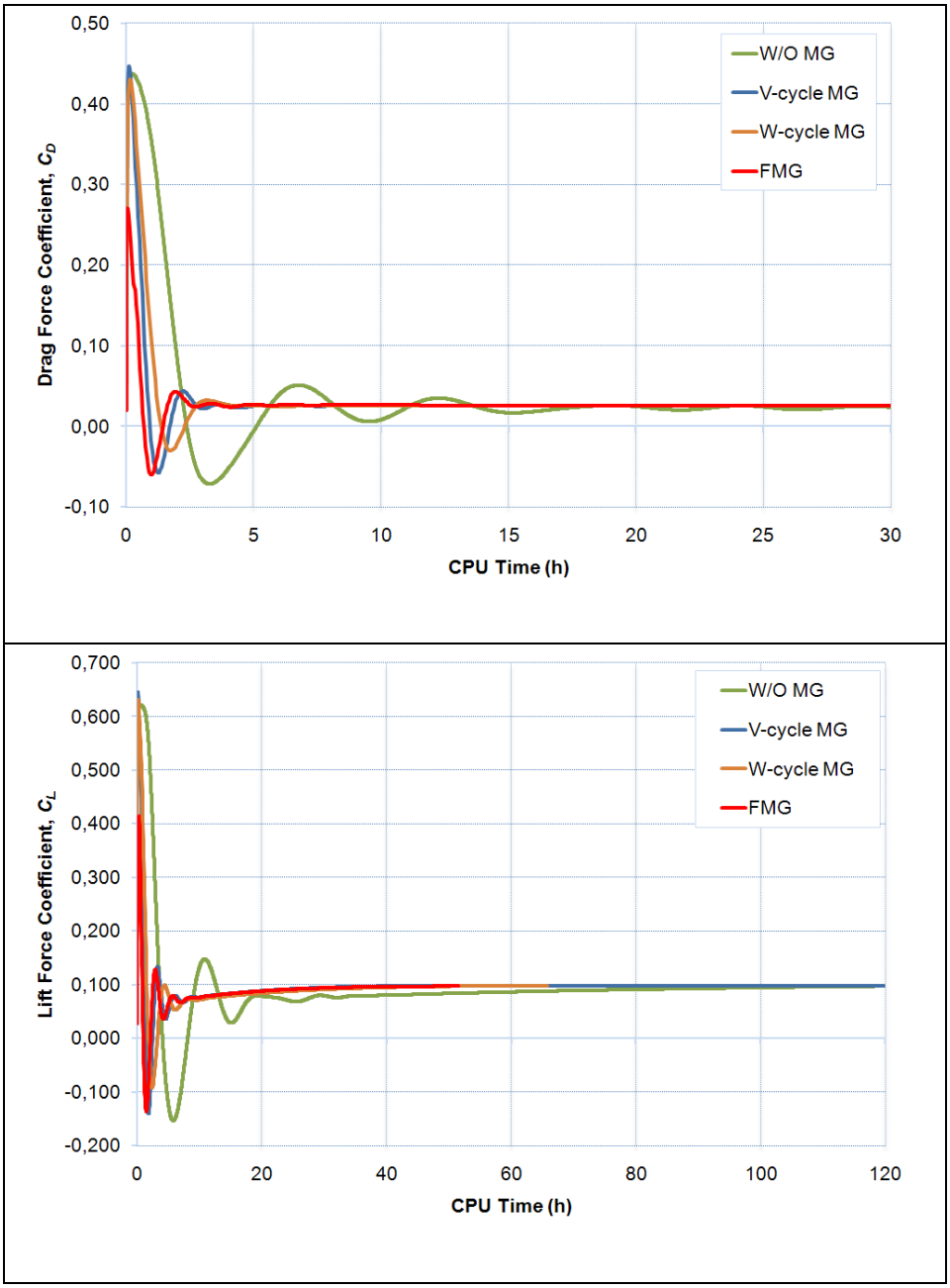


Figure 4.53 Convergence histories of drag and lift force coefficients for case 5

The CPU time needed to reach 20 %, 10 %, 5 %, 1 % and 0.1 % error bands of aerodynamic force coefficients with FMG cycling type and the fine grid solution is analyzed. FMG solution is taken as a reference value and it is seen that, the single

grid solution did not reach 1 % and 0.1 % error band within 150,000 time steps. The speed up ratios relative to time required for the single grid solution are presented for drag and lift force coefficients in Figure 4.54. It is observed that, FMG solution reach 5 % error band about 13 and 4 times faster than the single grid solution for lift and drag forces, respevtively.

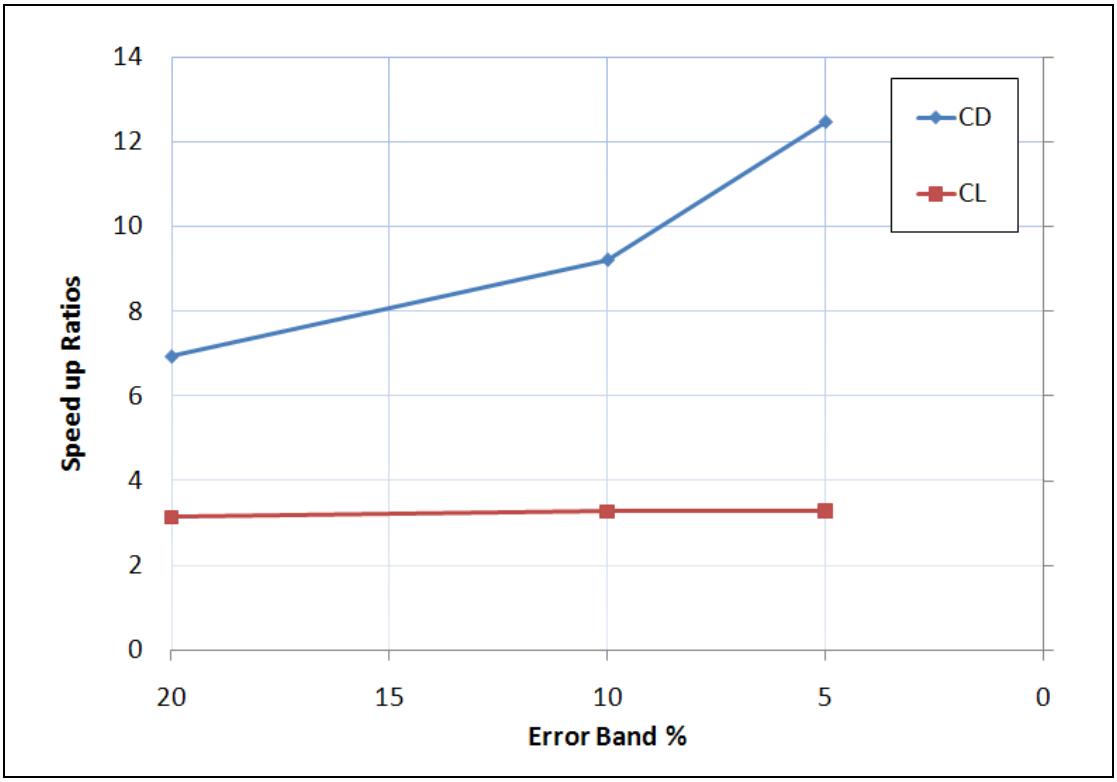


Figure 4.54 Plot of speed up ratios of drag and lift coefficient convergences to reach 20%, 10%, 5% error band for case 4

4.5.3 Effect of the Solver Type at Coarse Level Solution on MG Efficiency

The turbulent MG solutions are obtained by solving the coarse levels by inviscid solver. In this section, the effect of solver at coarse grid levels on MG efficiency is

investigated. V-cycle MG solutions are obtained by solving laminar equations on coarse levels at the same flow conditions which are given in previous section.

The convergence histories of both V-cycle MG solutions which differ at coarse level solutions are presented in Figure 4.55. It is seen that, the residual convergence behaviors for these two MG solutions are different and the residual of V-cycle MG solution which has laminar solution at coarse levels did not reduce like the V-cycle MG solution which has inviscid solution at coarse levels. It is concluded that the errors coming from the coarse grid levels are fixed and they could not be reduced by fine grid iterations.

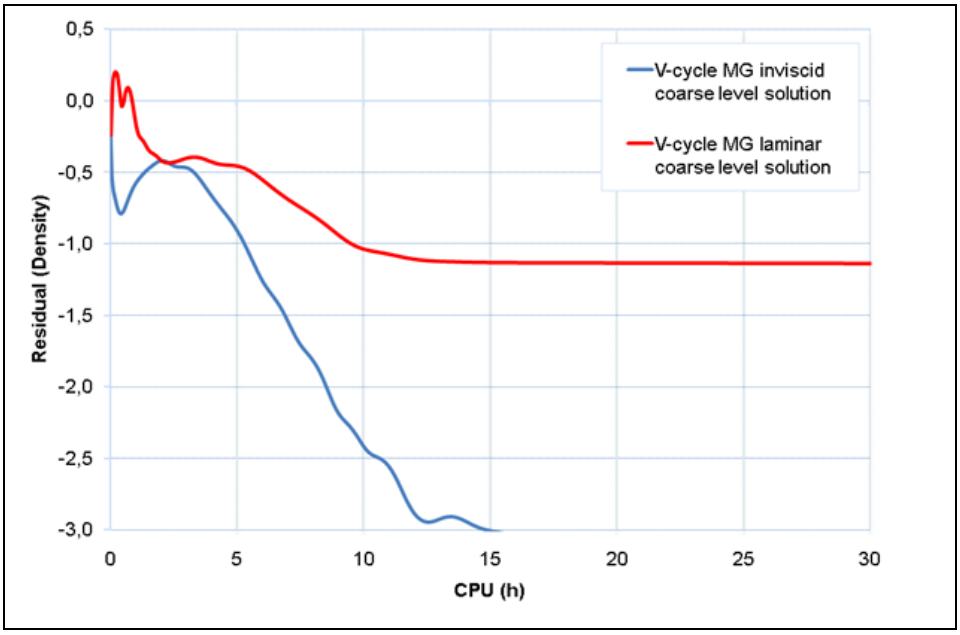


Figure 4.55 V-cycle MG solutions which differs at coarse level solution for case 4

Then, the solutions are investigated according to aerodynamic load convergence in Figure 4.56. It is seen that, although both results converge to the same aerodynamic parameters after some iterations, V-cycle MG solution which has inviscid solution at coarse level give more efficient results.

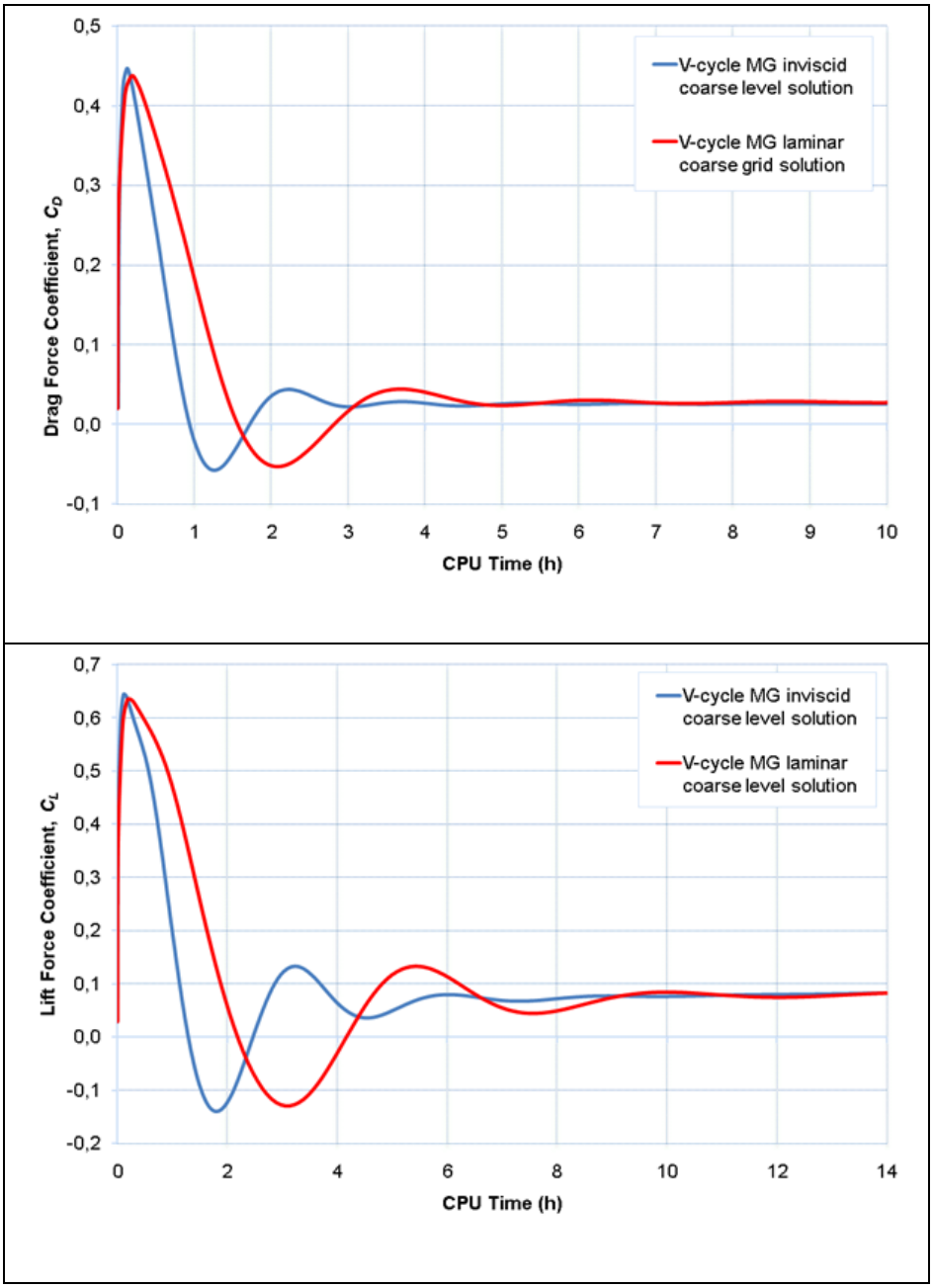


Figure 4.56 Convergence histories of drag and lift force coefficients obtained from two types of V-cycle MG solutions for case 4

4.5.4 Effect of Level Number on MG Efficiency

The efficiency of coarse grid level number on MG applications is investigated for three dimensional problems in this section. The laminar flow solutions at a low Reynolds number of 10000, a Mach number of 0.1 and an angle of attack of 3° are obtained by using 2, 3 and 4 grid levels and compared to the solution with 5 grid levels. The convergence histories of the solutions in terms of the variation of the normalized density residual are shown in Figure 4.57. The same investigation is presented in Section 4.2.4 for two dimensional laminar problems. The comparison showed that, 5-level MG solution is more efficient for both 2D and 3D, but, the efficiency of is more realizable in 3D laminar solutions.

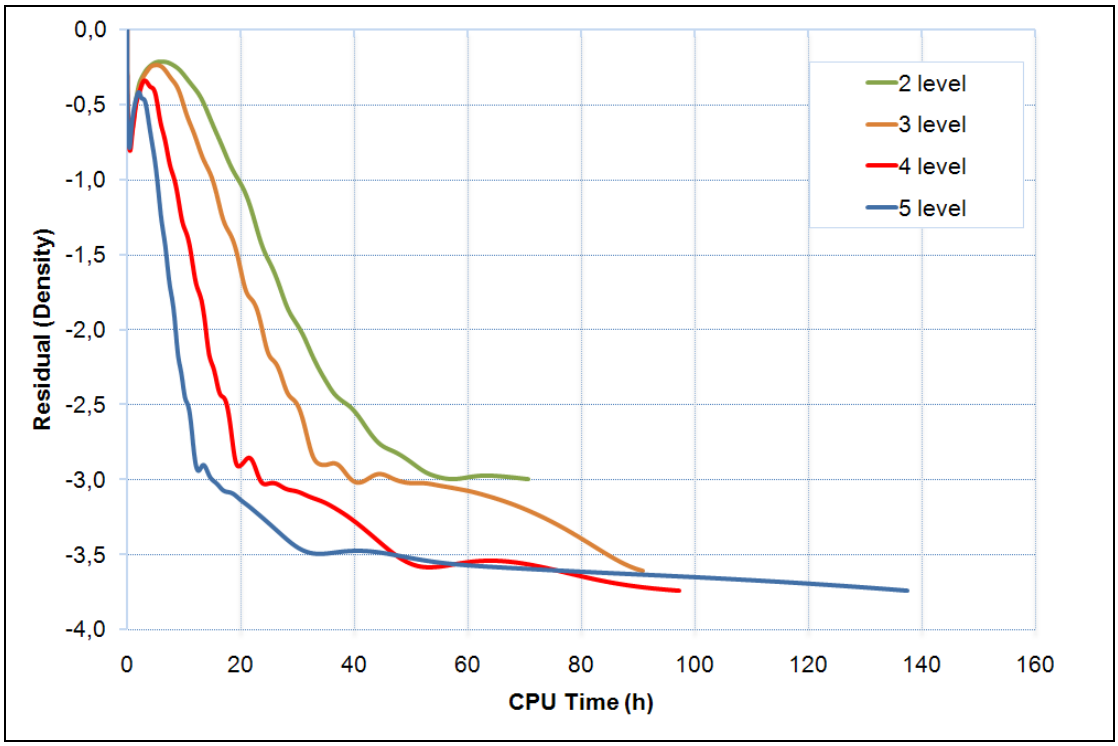


Figure 4.57 Residual histories of V-cycle MG solutions with different number of coarse grid levels for case 5

The load convergence histories of V-cycle MG solutions with different coarse grid levels are shown in Figure 4.58. It is seen that 5-level MG solution is the most efficient solution when the residual convergence history is considered. Finally, using high number of coarser grid levels damps the oscillatory characteristics coming from fine grid solution more successfully.

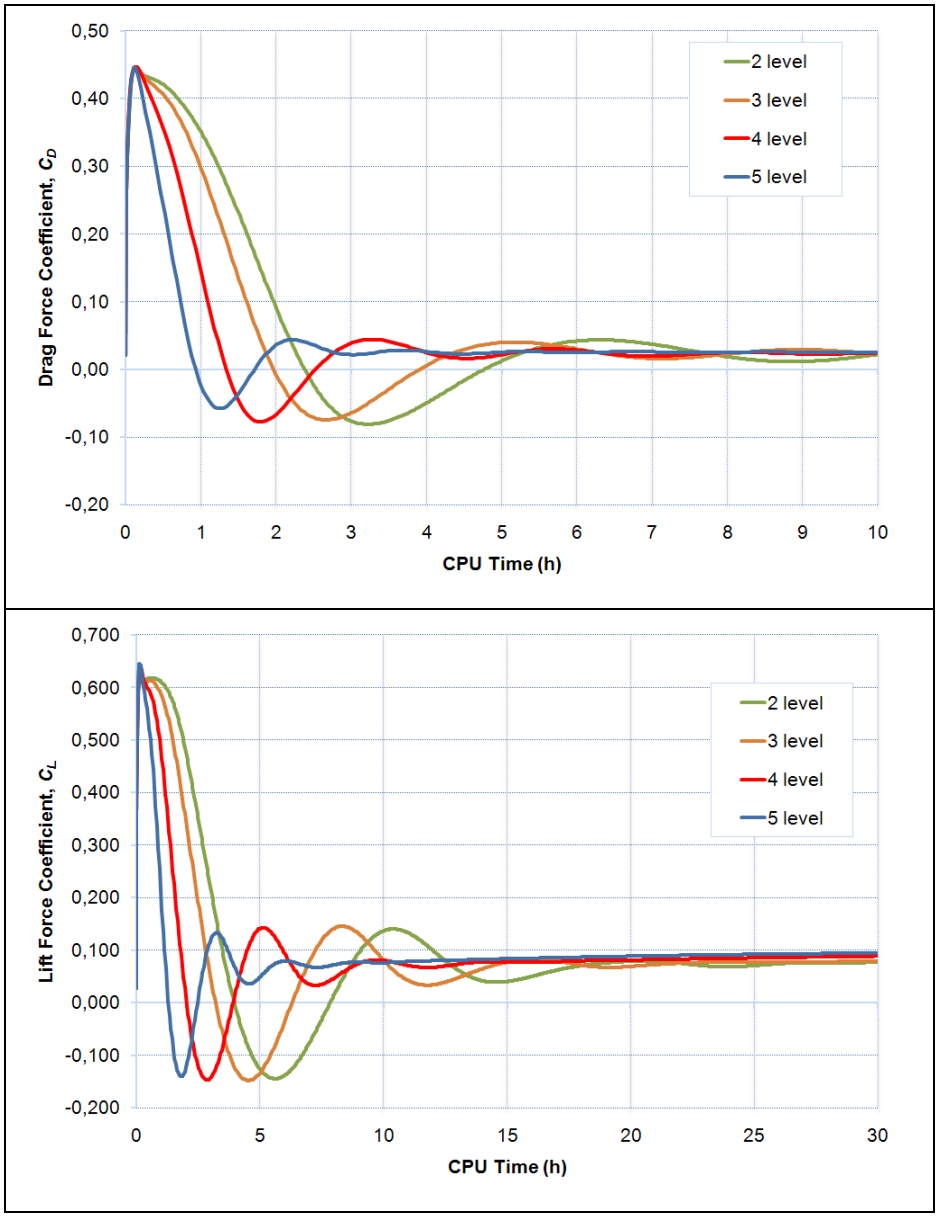


Figure 4.58 The convergence histories of drag and lift force coefficients with different number of coarse grid level for case 5

4.5.5 Effect of the Iteration Time Step at Coarse Grid Levels on MG Efficiency

The efficiency of iteration count at coarse grid levels are investigated for 2D laminar solution in Section 4.2.5 and it found that the solution with 20 iteration steps provides better convergence. In 3D inviscid flow solutions in Section 4.4.3, it is found that the solution with 40 iteration steps provides best convergence. In this section, the number of iterations for best convergence is again investigated for laminar flows. The MG solutions with 10, 20, 30 and 40 coarse level iteration steps using V-cycle strategy are studied. The convergence histories of the solutions in terms of the variation of the normalized density residual are presented in Figure 4.59.

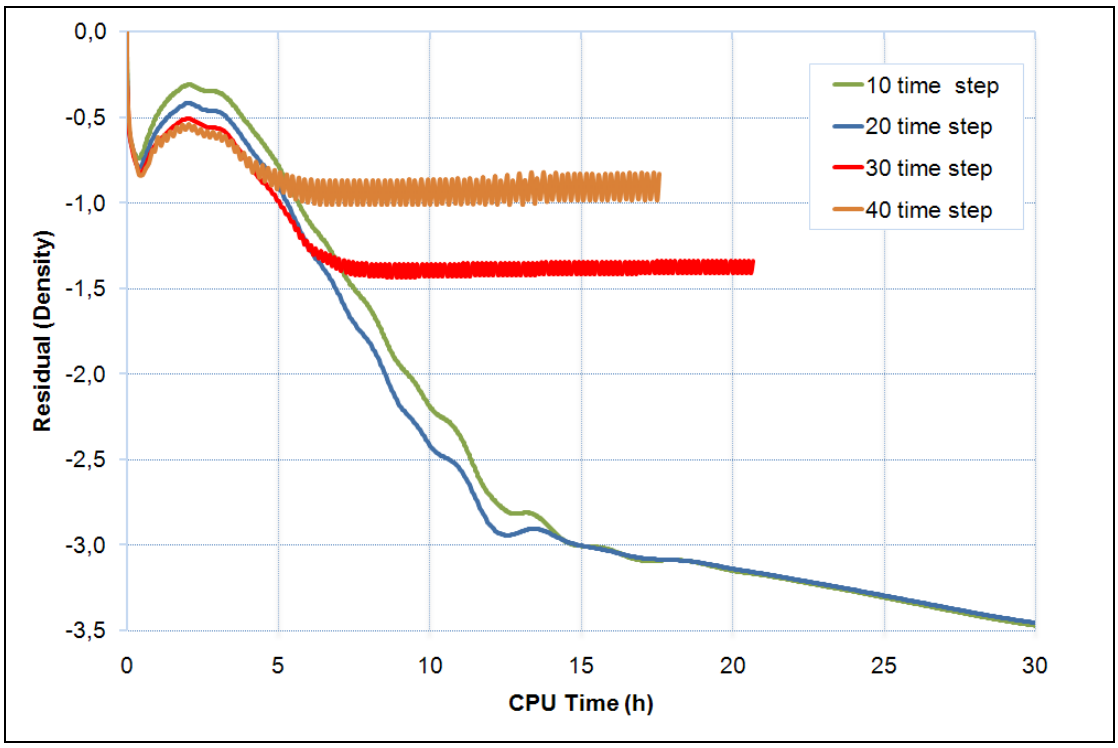


Figure 4.59 Residual histories of MG solutions with a variety of iteration count at coarse grid levels for case 5

It is seen that, contrary to the 3D inviscid case, the solution with 20 iteration steps is more successful in convergence histories when the variation of the normalized density residual is considered. Therefore, 20 time steps at coarse grid levels should be used for turbulent solutions and more iteration steps on inviscid solutions whether it is two or three dimensional. This is due to the fact that the inviscid solver is used at coarse grid levels for both inviscid and turbulent solutions. Therefore the turbulent solution in fine grid should be updated more frequently.

4.6 Case 6: A Transonic Laminar Flow Solution over a Wing with RAE2822 Airfoil

4.6.1 Grid Coarsening

An automated octree based data structure is finally implemented on 3D hybrid/unstructured grid over a 0.25 chord wing with RAE2822 airfoil section. The fine grid contains 645718 nodes and 1570568 cells. The meshes on the wing and the symmetry axis are demonstrated in Figure 4.60. A sequence of four coarse hybrid grids is generated in this time with a maximum coarsening ratio of about 50%. The coarse level grids for MG application have 655788, 154885, 33223 and 12067 cells. Coarse grid levels at the symmetry axis are presented in Figure 4.61.

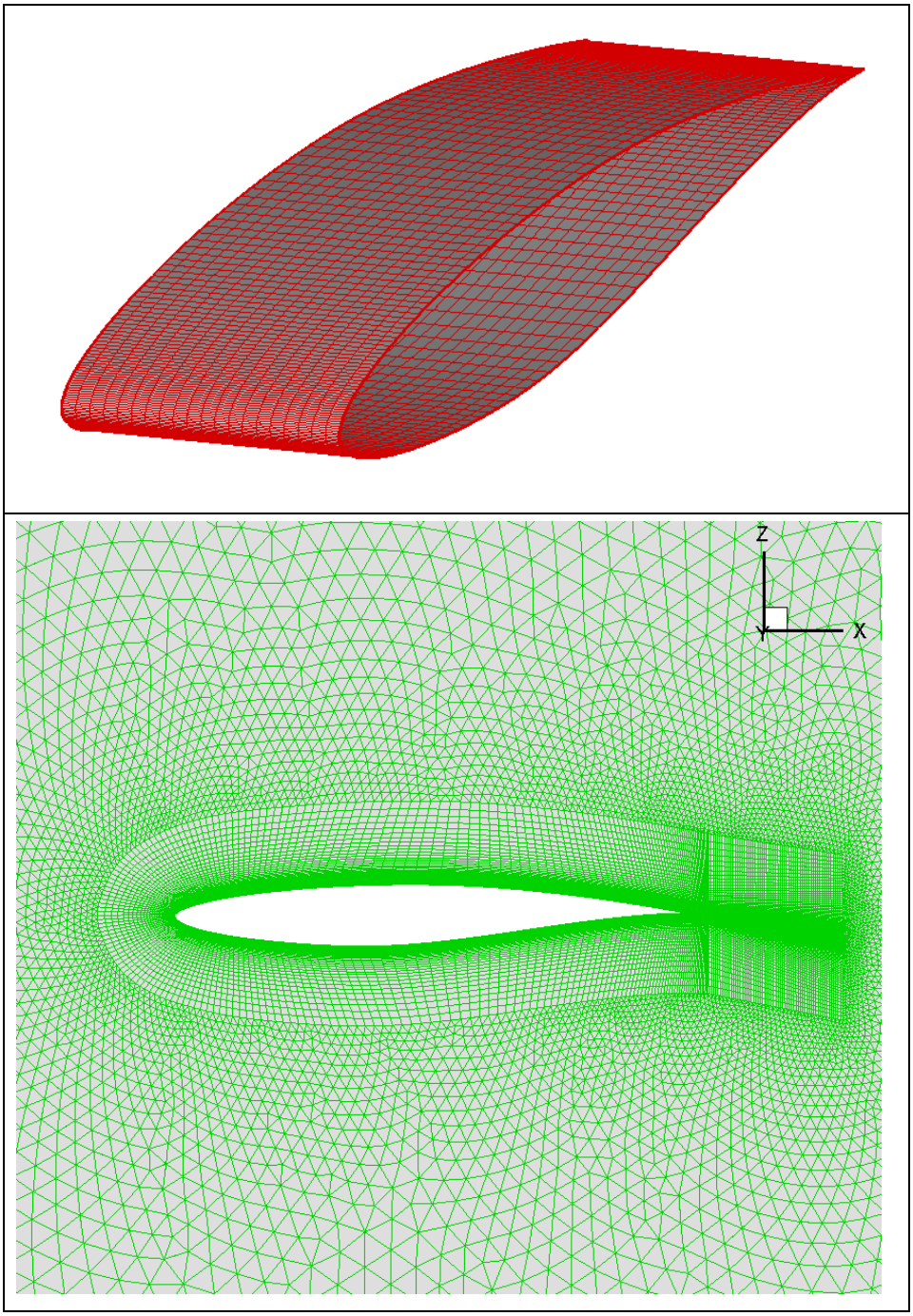
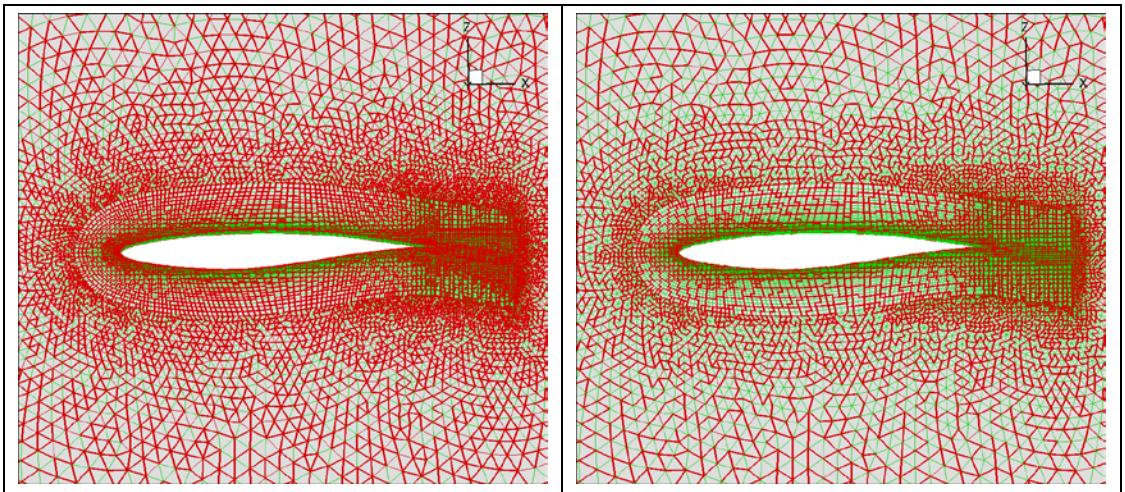
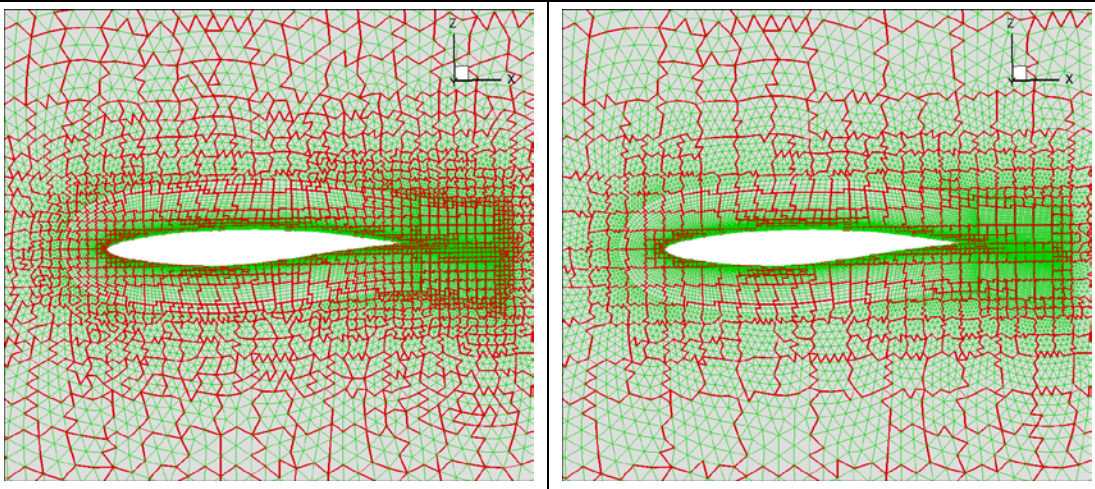


Figure 4.60 The meshes on the wing and the symmetry axis for case 6



Level 2	Level 3



Level 4	Level 5

Figure 4.61 Coarse grid levels at the symmetry axis for case 6

4.6.2 Effect of the MG Efficiencies on Laminar Flow

The single and FMG solution over a wing with RAE2822 airfoil are obtained at a transonic Mach number of 0.729, Reynolds number of 6,500,000 and an angle of attack of 2.31° with CFL number of 0.1. FMG solution is obtained again with 20 time steps at all grid levels including the fine grid. For coarse grid iterations, the

inviscid solver is used but the calculated laminar face fluxes at fine grid level are added as a constant term at active faces during flux calculations at the coarse grid level. The pressure coefficient distributions over the airfoil, which are computed by FMG solutions and the experimental values which are already given in Section 4.2 are presented in Figure 4.62. It is seen that, the first order laminar solution catched the shock location but the strength of the shock is calculated less according to experimental values.

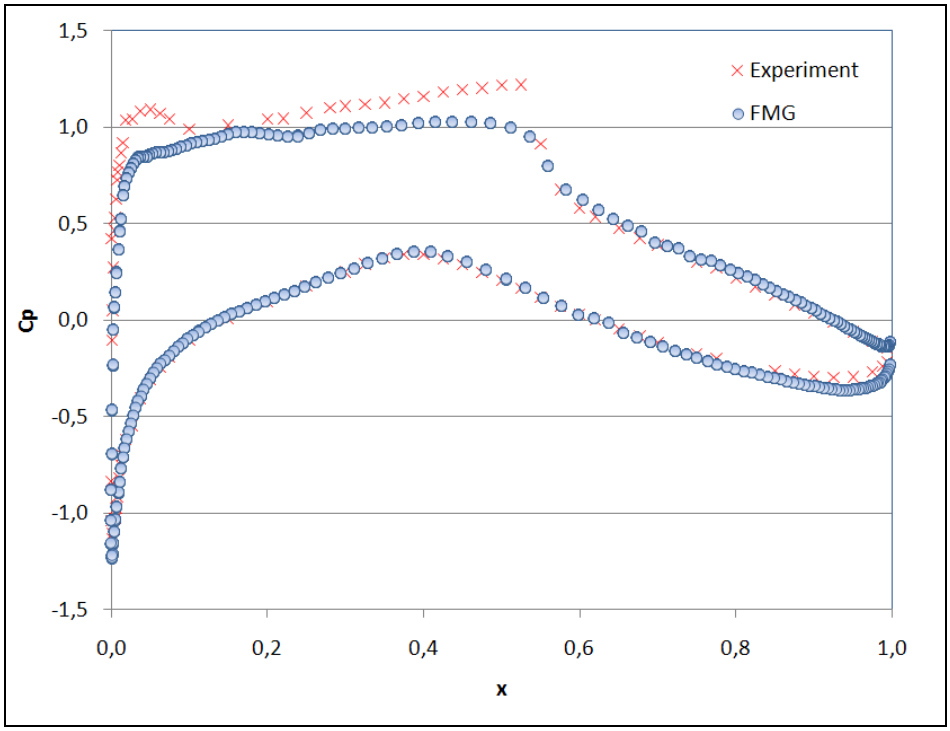


Figure 4.62 The pressure coefficient distributions over the airfoil for case 6

The convergence history of FMG solution in terms of the variation of the normalized density residual is shown in Figure 4.63 and compared with the convergence history of the single grid solution. It is realized that, the residual convergence behavior in two dimensional and three dimensional laminar solutions are similar to each other.

Figure 4.63 Residual histories of single and FMG solution for case 6

Then, the solutions are investigated according to aerodynamic load convergence in Figure 4.64 for the solutions with and without MG. FMG solution converges rapidly like the other test cases, but it is seen that, in transonic flow regime, MG solution is less efficient compared with the other laminar test cases.

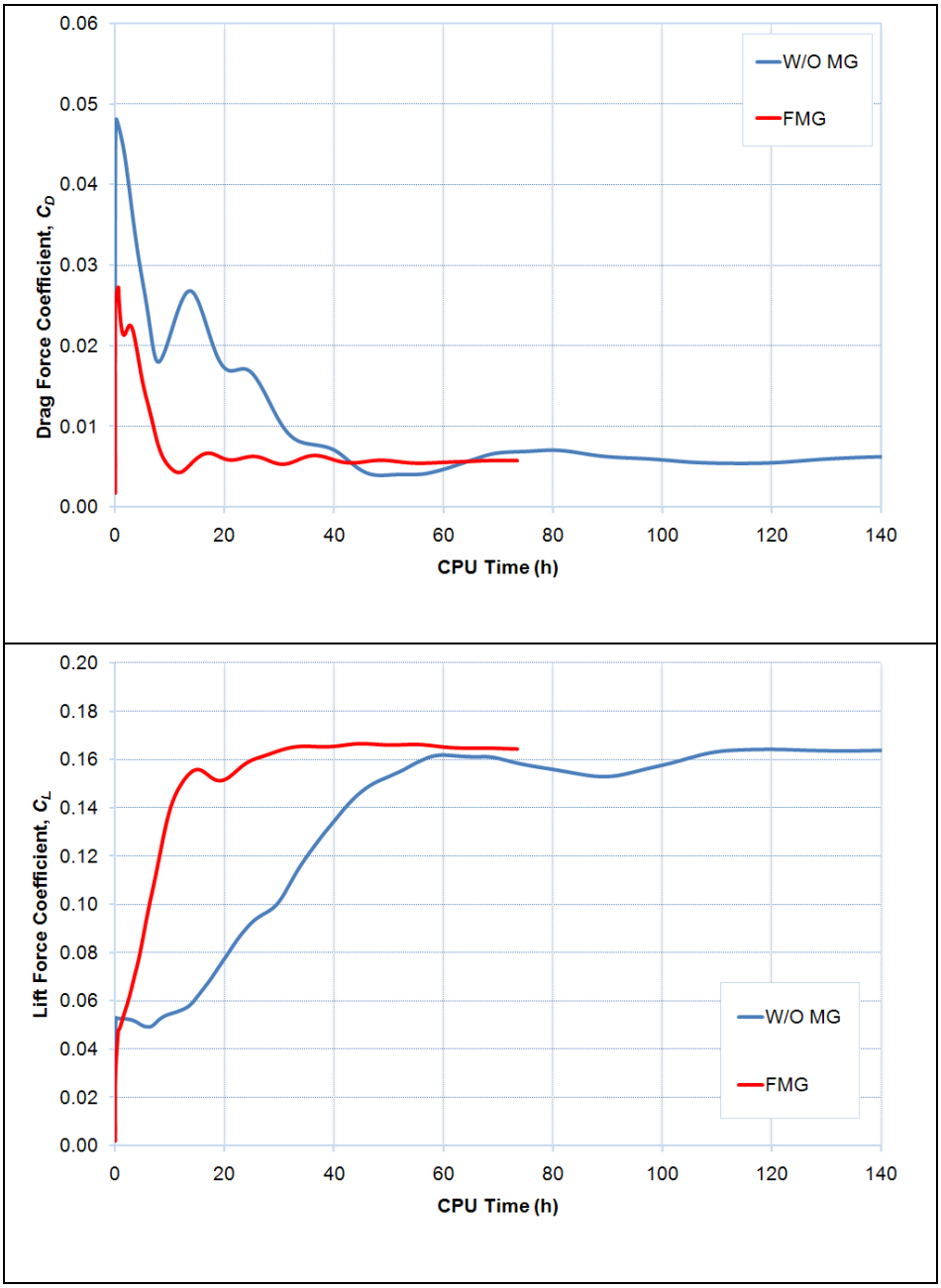


Figure 4.64 Convergence histories of drag and lift force coefficients for case 6

CHAPTER 5

CONCLUSION

In this thesis work, it is aimed to accelerate SAGE Euler / Navier Strokes Equation Solver developed by TUBITAK-SAGE. While preparing the CFD tool, unstructured-grid methodology was chosen for rapid aerodynamics analysis and design of complex configurations. According to literature survey about unstructured grids, there are many studies about accelerating the CFD studies, generally by convergence acceleration techniques and parallelization techniques, rather than solution technique directly. These acceleration techniques are invested and it is seen that multigrid is likely to be the most effective technique to achieve this goal.

Since, the basic idea of a multigrid strategy is to accelerate the solution of a set of fine grid equations by computing corrections on a coarser grid; it has been successfully demonstrated for specially structured and Cartesian grid due to the easy generation of the coarse grid levels. Besides, in the past decade, various multigrid strategies have also been demonstrated for the unstructured grid flow solvers. It is clearly seen that, the most difficult subject on implementation multigrid strategy to code with unstructured grid is generating the coarse grid levels. It is obviously seen that, the agglomeration coarsening approach is the most powerful technique and is a widely used method due to being fully nested, easily automated, no geometry loss and high solution accuracy. In an agglomeration method, grid cells are fused together to form a smaller set of larger polygonal (or polyhedral in three dimensions) control volumes. Since, it satisfies the requirement of this study, agglomeration coarsening approach is chosen for generating coarse grid levels. The main difficulty of agglomeration approach with unstructured MG methods is the selection of the cells to be agglomerated.

In this research, a novel grid coarsening method for hybrid/unstructured grids is developed to provide an aspect ratio of the cells about one at all coarse grid levels. The new grid coarsening technique relies on the agglomeration of hybrid/unstructured cells based on cell center localization on a quadtree and octree data structure for 2D and 3D applications respectively. This agglomeration strategy or point removal algorithm can be define as globally coarsening method by merging cells according to parent quadrant/octant or sub-groups. The coarsening algorithm can be summarized in two steps. In first one, the quadrant or octant cells are created as imaginary cells over the cell domain such a way that each quadrant covers maximum four; each octant covers maximum eight cell center points. Then, the finest mesh cells are grouped according to quadtree / octree data structure, in other words the parent/child structure. The coarsening algorithm is automated and implemented to baseline flow solver to make user-friendly.

Besides generating the coarse grid levels, the baseline flow solver is modified during this study. The viscous flow solver, SAGE Euler / Navier Stokes Equation Solver (SENSE) is a hybrid / unstructured finite volume method (FVM) solver, flow variables are stored at cell centers and second order Roe's upwind flux computations are employed. The time dependent equations are solved explicitly using the third order Runge-Kutta method with variable time-stepping. For easily adaptation of multigrid application, it is decided to change SENSE solver flux calculation algorithm by changing the cell based loops to edge/face based loops without changing the cell centered scheme. By doing so, due to the nested coarse grid levels, which are obtained by using the agglomeration coarsening method, the information can be easily transferred between grid levels. The second advantage is that, following the edges/faces instead of cells needs less memory and it is time consuming by reducing calculation once for each edges/faces. After that, the modifications are validated in two and three dimension by obtaining the same L2 norm residuals (which shows the sum of all variables flux calculation) at both versions. Final modification on baseline flow solver about multigrid adaptation is

adding the checking algorithm that shows the edge/face is active or deactive. If the edge/face is deactive, the process is skipped and continues for the active edge/face. Doing so, some arrays are updated to store the information with the responding coarsening level.

Finally, the multigrid algorithms developed are implemented to the baseline code. The full approximation storage scheme is chosen to directly handle non-linear problems. In this scheme, the defect-correction term appears on the right-hand side and it ensures that the fine grid problem is represented by the coarse grid discretization. Therefore, fine and coarse grid levels are solved using exactly same free flight conditions, numerical discretization methods, CFL number and boundary conditions. Both coarse and fine grid equations converge to the same solution with different accuracies. Due to the simplicity and robustness, the fixed cycling strategy where has a fixed pattern of coarse and fine grid is preferred in this thesis. All common cycling patterns, V-cycle, W-cycle and FMG, are adapted to the solver. The restriction of the solution from a finer to coarser grid and the prolongation of the correction from a coarser grid to a finer grid are implemented by using simple injection. For 2D application, the area weighting rule and for 3D application the volume weighting rule is used to restrict the flow variables. The restriction for residual is just summing the residuals of finer meshes that form the coarse mesh. The prolongation of errors from coarse to fine meshes are made by simple injection.

The performance of the developed grid coarsening algorithm is investigated with six test cases with unstructured, structured and hybrid grids for both 2D and 3D. The quadtree based grid coarsening algorithm is used for two dimensional, octree based grid coarsening algorithm is used for three dimensional test cases. Firstly, fully unstructured fine grids are used over NACA0012 airfoil for 2D and ONERA M6 wing for 3D. Then hybrid grid over NACA0012 airfoil for 2D and structured grid over a wing with NACA0012 airfoil for 3D which are suitable for low Reynolds number laminar flow are preferred. Finally, a hybrid grid over RAE2822

airfoil for 2D with very small initial boundary layer cells and a hybrid grid over a wing with RAE2822 airfoil for 3D, which is suitable for laminar mesh for transonic flow solution, are prepared. A sequence of four coarse hybrid grids for each grid is generated with a maximum coarsening ratio of about 40% and 50 %. When the coarse grid levels are investigated, the coarse meshes have high quality cells with aspect ratios of about one like Cartesian type grid. The coarsening ratio effect is seen around the airfoil very clearly. In addition, the coarse level grids keep the original anisotropic grid distribution at all coarse levels.

Then the multigrid efficiencies of SENSE2D and SENSE3D are investigated using three different solvers: inviscid, laminar and turbulent.

The performance of MG solution on inviscid flow solutions for 2D and 3D is investigated in the first and fourth validation cases, using V-cycle, W-cycle and FMG cycling strategies at subsonic flow conditions. For both dimension, all multigrid solutions converge very fast in terms of the variation of the normalized density residual and they are approximately 10 times for 2D and 5 times for 3D faster than the single grid solution without the MG. The MG cycling strategies shows vey similar behavior and especially at initial iterations, FMG gives the fastest convergence. It is found that to reach 5% error band, the drag coefficients in MG solutions have approximately 20 fold in 2D and 12 fold in 3D, the lift coefficients in MG solutions have approximately 5 fold for both 2D and 3D faster convergence ratio than the baseline solution.

The performance of MG solution on laminar flow solutions for 2D and 3D is investigated in the second and fifth validation cases at low Reynolds number. It is seen that for viscous case, the single grid solution converges very hardly and therefore MG solution acceleration rate is very efficient. To reach 5% error band, the drag coefficient in FMG solutions has approximately 25 fold for 2D and 13 fold for 3D, the lift coefficient in FMG solutions has approximately 4 fold for both 2D and 3D faster convergence ratio than the single grid solution. In addition the

efficiency on laminar flow at transonic regime is investigated in the sixth test case for 3D applications. The performance of MG strategy for this case is still very efficient but it has less speed up ratio with respect to other laminar solutions.

The performance of MG solution on turbulent flow solutions for 2D is investigated in the third validation cases at transonic Mach number. It is seen that, the explicit turbulent flow solution restricted the CFL condition considerably. Furthermore, the number of iterations for convergence is about one order of magnitude greater than the laminar flow solutions. Finally, flow solution does not predict the shock location accurately, which may be attributed to the fully turbulent first order solution and the performance of the one-equation turbulence model for transonic flows. In addition to that, to reach 0.1% error band, the drag coefficient in FMG solutions has approximately 7 fold, the lift coefficient in FMG solutions has approximately 6 fold faster convergence ratio than the single grid solution.

In the validation cases, the dependency of the multigrid solutions on the flow parameters and MG variables is also investigated. The solutions at Mach number of 0.15, 0.3, 0.45, 0.6 and 0.75 are obtained at the same flow conditions and except the low subsonic case, all MG solutions converge at the same time. Therefore while decreasing Mach number; MG solutions are more efficient about residual convergence. With respect to the angle of attack, 0°, 3°, 6°, 9° and 12° are studied at the same flow conditions, the multigrid convergence rates are approximately same and do not depend on the angle of attack parameter. The efficiency of coarsening ratio between coarse grid levels on MG applications is investigated. It is seen that all coarse grid levels with different coarsening ratio give approximately same convergence histories, but, coarsening ratio of 40% gives the most rapid initial load convergence without any oscillation. The efficiency of coarse grid level number on MG applications is investigated using the coarse grid levels with the coarsening ratio of %40. Since, using high number of coarser grid levels prevents the oscillation characteristics coming from fine grid solution, 5-level MG solutions,

especially in 3D applications, is more efficient on convergence. The efficiency of iteration count at coarse grid levels on MG convergence is investigated. Since, the inviscid solver is used at coarse grid levels for both inviscid and turbulent solutions, the step number should be chosen according to solver. Although, 20 time step at coarse grid levels gives most powerful results for turbulent solution, increasing the step number in inviscid cases prevents the oscillations and overshoots before convergence.

As an overall conclusion, the automated grid coarsening algorithm and multigrid flow solvers are developed for unstructured/hybrid grids and cell-centered scheme. It is shown that the computational cost is reduced significantly. As a future work it can be stated that, the solvers should be updated by adding different type turbulence models besides one equation Spallart Almaras model. In addition, especially for three dimension applications, the parallel solving capability to multigrid flow solver is absolutely necessary. Finally, on multigrid application, the variety of the transfer mechanisms and turbulent solver techniques at coarse grid level can be increased.

REFERENCES

1. Jameson, A., *Solution of the Euler Equations for Two-Dimensional Transonic Flow by a Multigrid Method,* Applied, Mathematics and Computation, 13:327–356, 1983.

2. Jameson A., *Computational Algorithms for Aerodynamic Analysis and Design*, Applied Numerical Mathematics, 13:383–422, 1993.

3. Jameson A., Baker T. J., and Weatherhill N. P., *Calculation of Inviscid Transonic Flow Over A Complete Aircraft,* AIAA 86-0103, 1986.

4. Jameson A., Schmidt W., and Turkel E., *Numerical Simulation of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time Stepping Schemes,* AIAA-81-1259, 1981.

5. Van Leer B., *Flux-Vector Splitting for the Euler Equations,* Technical Report 81-11, ICASE, 1981.

6. Van Leer B., *Upwind-Difference Methods for Aerodynamic Problems Governed By the Euler Equations,* Lectures in Applied Mathematics, 22, 1985.

7. Van Leer B., Thomas J., Roe P., and Newsome R., *A Comparison of Numerical Flux Formulas for the Euler and Navier-Stokes Equations,* AIAA Paper 87-1104, 1987.

8. Roe P. L., *Approximate Riemann Solvers, Parametric Vectors, and Difference Schemes*, Journal of Computational Physics, 43:357–372, 1981.

9. Roe P. L., *Characteristic-Based Schemes for The Euler Equations,* Ann. Rev. Fluid Mech., 18:337–65, 1986.

10. Frink, N.T., *Three-Dimensional Upwind Scheme for Solving the Euler Equations on Unstructured Tetrahedral Grids*, PhD Thesis, Virginia Polytechnic Institute and State University, September 1991.

11. Anderson W. K., Rausch R. D., and Bonhaus D. L., *Implicit/Multigrid Algorithms for Incompressible Turbulent Flows on Unstructured Grids*. J. Comput. Phys., 128:391–408, 1996.

12. Bassi F., and Rebay S., *High-Order Accurate Discontinuous Finite Element Solution of the 2-D Euler Equations*, Journal of Computational Physics, 138:251–285, 1997.

13. Gönç L. O., *Computation of External Flow Around Rotating Bodies*, PhD Thesis, METU, March 2005.

14. Mavriplis D.J., *Multigrid Techniques for Unstructured Meshes,* ICASE Report N0: 95-27, NASA Contractor Report 195070, April 1995.

15. Mavriplis D.J., *On Convergence Acceleration Techniques for Unstructured Meshes,* AIAA 1998-2966.

16. Grand Chanllenges: *High-Performance Computing and Communications,* A Report By The Committee On Physical, Mathematical, And Engineering Sciences, Federal Coordinating Council For Science, Engineering And Technology, Office Of Science And Technology Policy, 1992.

17. Bergen B.K., *A Framework for Efficient Multigrid on High Performance Architectures,* PhD Thesis, Friedrich-Alexander-Universitat Erlangen-Nurnberg

18. Pandya M.J., Frink N.T., *Agglomeration Multigrid for an Unstructured Grids Flow Solver*, AIAA-2004-759.

19. Okamoto N., Nakahashi K., and Obayashi S., *A Coarse Grid Generation Algorithm for Agglomeration Multigrid Method on Unstructured Grids*, AIAA-98-0615.

20. Parthasarathy V., Kallinderis Y., *A New Multigrid Approach for 3D Unstructured, Adaptive Grids*. AIAA 94-0078, January 1994.

21. Mavriplis D.J., *A Three Dimensional Multigrid Reynolds Averaged Navier-Stokes Solver for Unstructured Meshes*. AIAA-94-1878-CP.

22. Mavriplis D.J., Venkatakrishnan V., *A 3D Multigrid Solver for the Reynolds Averaged Navier-Stokes Equations on Unstructured Meshes*, AIAA-95-0345.

23. Waltz J., Löhner R., *A Grid Coarsening Algorithm for Unstructured Multigrid Applications*, AIAA 2000-0925, January 2000.

24. Bonhaus D.L., *An Upwind Multigrid Method for Solving Viscous Flows on Unstructured Triangular Meshes*. NASA-TM-109343, August 1993.

25. Ollivier-Gooch C.F., *On Mesh-to-Mesh Transfer Operators for Unstructured Multigrid Solvers*, AIAA-97-2026.

26. Lassaline J. V., *A Navier Stokes Equation Solver Using Agglomerated Multigrid Featuring Directional Coarsening and Line Implicit Smoothing*, PhD Thesis, University of Toronto, 2003

27. Federenkro P., *The Speed of Convergence of One Iterative Process*, U.S.S.R. Computational Mathematics and Mathematical Physics, Vol. 4, 1964, pp. 227-235.

28. Brandt A., *Multi-Level Adaptive Solutions to Boundary-Value Problems*, Mathematics of Computation, Vol. 31, No. 138, April 1977, pp. 333-390.

29. Nicolaides A., *On the l^2 Convergence of an Algorithm for Solving Finite Element Equations*, Mathematics of Computation, Vol. 31, No. 140, October 1977, pp. 892-906.

30. Hackbusch W., *On the Multi-grid Method Applied to Difference Equations*, Computing, Vol. 20, 1978, pp. 291-306.

31. Ni R. H., *A Multiple-Grid Scheme for Solving Euler Equations*, AIAA Journal, Vol. 20, No. 11, November 1982, pp. 1565-1571.

32. Jameson A., *Multigrid Algorithms for Compressible Flow Calculations*, Lecture Notes in Mathematics, edited by W. Hackbusch and U. Trottenberg, Vol. 1228, Springer-Verlag, 1986, pp. 166-201.

33. Jameson A., Mavriplis D. J., *Finite Volume Solution of the Two-Dimensional Euler Equations on a Regular Triangular Mesh*, AIAA Paper 85-0435, 1985.

34. Mavriplis D. J., *Solution of the Two-Dimensional Euler Equations on Unstructured Meshes*, PhD Thesis, Princeton University, June 1987.

35. Peraire J., Peiro J., and Morgan K., *3D Finite-Element Multigrid Solver for the Euler Equations*", AIAA Paper 92-0449, AIAA, 1992.

36. Morano E., Dervieux A., *Looking for O(N) Navier-Stokes Solutions on Non-Structured Meshes*, in Sixth Copper Mountain Conference on Multigrid Methods, N. D. Melson, T. A. Manteuffel, and S. F. McCormick, eds., vol. CP 3224, Hampton, VA, 1993, NASA, pp. 449-463.

37. Riemslagh K., Dick E., *A Multigrid Method For Steady Euler Equations On Unstructured Adaptive Grids,* in Sixth Copper Mountain Conference on Multigrid Methods, N. D. Melson, T. A.

38. Ollivier-Gooch C. F., *Multigrid Acceleration of an Upwind Euler Solver on Unstructured Meshes,* AIAA J., 33 (1995), pp. 1822-1827.

39. Zuliani G. L., *Aerodynamic Flow Calculations Using Finite-Differences and Multigrid*, PhD Thesis, University of Toronto, 2004.

40. Fidkowski K.L., *A High-Order Discontinuous Galerkin Multigrid Solver for Aerodynamic Applications*, Ms Thesis, MIT, 2004.

41. South, Jr., Jerry C, and Brandt, Achi. *Application of a Multilevel Grid Method To Transonic Flow Condition*, In Adamson, T. C. and Platzer, M. C., editors, Transonic Flow Conditions in Turbomachinary. Hemisphere Publications, 1977.

42. Perez E., *A 3D Finite-Element Multigrid Solver For The Euler Equations*, INRIA Report No. 442, September 1985.

43. Parthasarathy V., Kallinderis Y., *New Multigrid Approach for Three-Dimensional Unstructured, Adaptive Grids*. AIAA J., 32(5):956-963, 1994.

44. Conneu S. D., Holmes D.G., *A 3D Unstructured Adaptive Multigrid Scheme for the Euler Equations,* AIAA J., 32(8):1626-1632; 1994.

45. Leclercq M. P., *Resolution des Equations d'Euler par des Methods Multigrilles Conditions aux Limites en Regime Hypersonique*. PhD thesis, Univerite de Saint-Etienne, Applied Math, April 1990.

46. Guillard H., *Node Nested Multigrid with Delaunay Coarsening*, INRIA Report No. 1898, 1993.

47. Chan T. F., Smith B., *Domain Decomposition and Multigrid Algorithms for Elliptic Problems on Unstructured Meshes*. UCLA CAM Report 93-42, Dept. of Mathematics, University of California, Los Angeles, December 1993.

48. Lallemand M., Steve H., and Dervieux A., *Unstructured Multigridding by Volume Agglomeration: Current status*, Computers and Fluids, 21(3):397-433, 1992.

49. Smith W. A., *Multigrid Solution of Transonic Flow on Unstructured Grids*, In Recent Advances and Applications in Computational Fluid Dynamics, November 1990. Proceedings of the ASME Winter Annual Meeting, Ed. O. Baysal.

50. Venkatakrishnan V., Mavriplis D. J., *Agglomeration Multigrid for The Three-Dimensional Euler Equations*, AIAA Paper 94-0069, June 1994; to appear in AIAA J.

51. Brandt A., *Multigrid Techniques with Applications to Fluid Dynamic,*: 1984 guide. In VKI Lecture Series, pages 1-176, March 1984.

52. McCormick S., Thomas J., *The Fast Adaptive Composite Grid (Fac) Method For Elliptic equations*, Mathematics of Computations, 46:439-456, 1986.

53. Parthasarathy V., Kallinderis Y., *A New Multigrid Approach for 3D Unstructured, Adaptive Grids*, AIAA 94-0078, January 1994.

54. Ruge J. W., Stiiben K., *Algebraic Multigrid*, In S. F. McCormick, editor, Multigrid Methods, SIAM Frontiers in Applied Mathematics, pages 73-131, Philadelphia, 1987. SIAM.

55. Francescatto, J., Dervieux, A., *A Semi-coarsening Strategy for Unstructured Multigrid Based on Agglomeration*, International Journal for Numerical Methods in Fluids 26: 927–957, 1998.

56. Ollivier-Gooch., C.F., *Robust Coarsening of Unstructured Meshes for Multigrid Methods*, AIAA-99-3250, 1999.

57. Ahlawat, S., Johnson, R., Vanka, S.P., *Object Based Library for 3D Unstructured Grid Representation and Volume Based Agglomeration*, AIAA 2006-888, 2006.

58. Chan, T.F., and Zikatanov, L., *An Agglomeration Multigrid Method for Unstructured Grids*, Contemporary Mathematics, Volume 218, BO-8218-0988-1-03002-8, 1998.

59. Okamoto N., Nakahashi K., and Obayashi S., *A Coarse Grid Generation Algorithm for Agglomeration Multigrid Method on Unstructured Grids*, AIAA-98-0615.

60. Samet H., *The Quadtree and Related Hierarchical Data Structures*, Computing Surveys, Vol. 16, No.2, June 1984

61. Lambropoulos N.K., Koubogiannis D.G., Giiannakogluo K.C., *Agglomeration Multigrid and Parallel Processing For The Numerical Solution of The Euler*

Equations on 2D and 3D Unstructured Grids, 4th GRACM Congress on Computational Mechanics, GRACM 27-29 June, 2002

APPENDIX A

SAMPLE INPUT FILE FORMAT

```
-------------------------------------------------------------------------------
                        SENSE 3D INPUT FILE
-------------------------------------------------------------------------------
                              SOLVER
-------------------------------------------------------------------------------
1       --> Solver selection 1: Explicit, 2: Implicit
1       --> Solver order 1:1st order, 2:2nd order
1       --> Flux Algorithm 1: ROE, 2:HARTEN
0       --> Viscosity 0: Inviscid; 1: Laminar; 2: Spalart-Allmaras
0       --> Implicit Residual Smoothing 0: Closed; 1:Open
0       --> Time Step Selection 0: local, 1: global
0.5     --> Courant-Friedrich-Levy Number
-------------------------------------------------------------------------------
                        FLOW & GRID PROPERTIES
-------------------------------------------------------------------------------
0.1     --> Inflow Mach Number
2.33e+6         --> Reynolds #
1.      --> Inflow Pressure [Non-dim]
1.      --> Inflow Temperature [Non-dim]
1.      --> Outflow Pressure [Non-dim](For Cascade Flows)
0.      --> Normalized base Pressure [Pbase/Pin]
0.      --> Alpha [°]
0.      --> Beta [°]
1.      --> Reference Length [m]
0., 0., 0 --> Center Of Mass [m]
0.      --> Min. time step (=0 dtmin calculated, else dtmin=amindt)
-------------------------------------------------------------------------------
        MULTIGRID PROPERTIES
-------------------------------------------------------------------------------
2       --> Multigrid Type
        0: without MG,1: Sawtooth MG,
        2: Classical V MG,3: Full MG
5       --> Multigrid Level
10      --> # of iteration at each level during cycle
40      --> The maximum coarsening ratio (%) between grid levels
```

```
1       --> Printing coarse grids (1) or not (0)
1       --> Preparing the techplot for coarse levels (1) or not (0)
0       --> Reading pre-prepared coarse grids
-------------------------------------------------------------------------------------
        PARAMETERS & FILES FOR INITIALIZING
-------------------------------------------------------------------------------------
1.0     --> Initial value for turbulent working variable - Spalart-Allmaras
0.0     --> Input data file is used (1) or not (0)
q.end   --> Input file name
3       --> Tecplot output 0:no output; 1:Surface; 2:Volume; 3:Both
-------------------------------------------------------------------------------------
        ITERATION PROPERTIES
-------------------------------------------------------------------------------------
1000    --> Total # of cycle (Required if MG active)
1       --> Total # of iteration (Required if MG off)
1       --> Frequency of the residual to be printed on screen
1.e10   --> Maximum physical time for run [sec](for unsteady problems)
10      --> Frequency for the solution to be saved (If MG is active it is cycle freq.)
5.      --> Order of magnitude for the residual to be reduced
```

APPENDIX B

SAMPLE INFORMATION OUTPUT FILE FORMAT

20100409

SOLVER PROPERTIES

Solver	: Explicit
Solver Order	: First Order
Flux Model	: ROE
Viscosity	: Inviscid
Imp. Residual Smoothing	: Closed
Time Stepping	: Local
CFL Number	: 0.900000000000000

FLOW & GRID PROPERTIES

Inflow Mach Number	: 0.150000000000000
Reynolds #	: 3000000.00000000
Angle of Attack	: 3.00000000000000
Reference Length [m]	: 1.00000000000000
Center Of Mass [m]	: 0.000000E+000 0.000000E+000
Mesh Size	: 2D 18390

MULTIGRID PROPERTIES

Multigrid Level	: 5
Multigrid Cycle	: V cycle
Grid level cells	: 9012 3204 1118 441
# of ite. at each level	: 10

RESULTS

Computational Time	: 38.4092874547207
# of Iteration	: 40000
# of cycle	: 2000
CD	: 8.985221953226218E-003

CL	: 0.319051662827522
Cm	: 4.002696485797786E-003
Logarithmic residual l2	: -11.6768130465795
Logarithmic residual rho	: -11.7030125704998
Logarithmic residual u	: -11.3314484052987
Logarithmic residual v	: -11.6153427179718
Logarithmic residual e	: -11.7155838143928
First residual l2	: 3.623711568108460E-005
First residual rho	: 1.248872619352622E-005
First residual u	: 5.869368324626266E-006
First residual v	: 1.166253722652242E-005
First residual e	: 3.141171774763823E-005
CPU time (seconds)	: 3553.82173600000

CURRICULUM VITAE

PERSONAL INFORMATION :

Surname : MAHMUTYAZICIOĞLU

Name : Emel

Date of birth : 24 March 1975

Place of birth : Ankara - TURKEY

Nationality : Turkish

Permanent address : Tara Park Sitesi, 3151.Sok, 6A Blok, No 34,
Yaşamkent, Ankara ,TURKEY
Tel: +90 (312) 217 05 18
Mobile Tel: +90 (532) 731 07 99

Work address : TÜBİTAK-SAGE
Turkish Scientific & Technical Research Council
Defense Industries Research & Development Institute
P.K. 16, 06261 Mamak – ANKARA / TURKEY
Tel: +90 (312) 590 91 70
Fax: +90 (312) 590 91 48-49
E-mail: emel.mahmut@sage.tubitak.gov.tr
emel.mahmut@hotmail.com

Marital status : Married

Profession : Aeronautical Engineer

EDUCATION :

School	*Years*	*Diploma, Grade and Subject*
Primary School		
Şehitlik Primary School Ankara, TÜRKİYE	1981 – 1986	Primary School Diploma Honors
Secondary School		
29 Ekim Secondary School Ankara,TÜRKİYE	1986 - 1989	Secondary Diploma / Honors
Secondary & High School		
Başkent High School Ankara,TÜRKİYE	1989 - 1992	High School Diploma / in math. & science division / Honors *Second degree
University		
METU Department of Aeronautical Engineering İnönü Blv., Ankara, TÜRKİYE	1992 - 1997	B.Sc. “System Engineering; 8 lectures from Electric-Electronics Department”
METU Department of Aeronautical Engineering İnönü Blv., Ankara, TÜRKİYE	1997 - 2000	M.Sc. “Calibration and Instrumentation of Ankara Wind Tunnel”
METU Department of Mechanical Engineering İnönü Blv., Ankara, TÜRKİYE	2001 - 2010	Ph.D. “Development of an Octree Based Grid Coarsening and Multigrid Flow Solutions”

LANGUAGE SKILLS :

Native Language : Turkish
Other Languages : English

PUBLICATIONS:

1. Atik H., Başoğlu O., Ilgaz M., Mahmutyazıcıoğlu E., and Yalçin L., Geniş Kanatlara Sahip Bir Füze Geometrisinin Aerodinamik İncelemesi, HASEM 2004, Kayseri 2004.

2. Mahmutyazıcıoğlu E., Tuncer İ.H., and Aksel M.H., Dörtdal Yöntemi İle 2 Boyutlu Düzensiz ve Hybrid Çözüm Ağlarının Seyrekleştirilmesi, UHUK 2008, İstanbul, 2008.

3. Mahmutyazıcıoğlu E., Tuncer İ.H., and Aksel M.H., Development of A Quadtree Based Agglomeration Method for Unstructured Hybrid Grids, 5th AIAC Conference, August 2009.

4. Mahmutyazıcıoğlu E., Tuncer İ.H., and Aksel M.H., Development of A Quadtree Based Agglomeration Method for Viscous Solver on Unstructured Grids, AIAA Conference, January 2010.

5. Mahmutyazıcıoğlu E., Tuncer İ.H., and Aksel M.H., Octree Based Unstructured Grid Coarsening Method For 3D Multigrid Applications, 5th ECCOMAS CFD 2010, June 2010.

6. Mahmutyazıcıoğlu E., Tuncer İ.H., and Aksel M.H., Sekizdal Yöntemi İle 3 Boyutlu Düzensiz Çözüm Ağlarının Seyrekleştirilmesi ve Çok Katmanlı Akış Çözümlerinin Elde Edilmesi, UHUK 2010, Eskişehir, September 2010. –Accepted-

7. Mahmutyazıcıoğlu E., Tuncer İ.H., and Aksel M.H., Development of Quadtree Based Grid Coarsening Method for an Unstructured Multigrid Turbulent Flow Solver, AIAA Journal, 2010. –Full Paper is Submitted-

8. Mahmutyazıcıoğlu E., Tuncer İ.H., and Aksel M.H., *Octree Based Unstructured Grid Coarsening And Multigrid Turbulence Flow Solution,* AIAA Conference, January 2011. –Accepted-

TRAINING :

Summer Training

1. 1. HİBM Komutanlığı, Eskişehir, TÜRKİYE July - August 1995

2. ROKETSAN, Ankara, TÜRKİYE July - August 1996

Technical Training

Strategic Planning ve Managing Seminars, TÜBİTAK-TÜSSİDE

Advance FLUENT Training, ANOVA

Project Management Seminars; PYO (Project Management Organization)

Management Skills, Masers Training International

Occupational Health And Safety for Technical Members, İDEAL

MISSILE DATCOM, USAF DATCOM, AEROPREDICTION, NLRAero, VORLAX, SET3D Trainings, TÜBİTAK-SAGE

Process Management, TÜSİDE

System Engineering, TÜBİTAK-SAGE

Basic Quality Concepts & Institution Culture, TÜSİDE

ISO 9000 Quality Assurance Management Model, TÜBİTAK-SAGE

Configuration Management, TÜBİTAK-SAGE

Occupational Health And Safety, TÜBİTAK-SAGE

Marketing Seminars; TÜBİTAK-SAGE

Time Management, TÜBİTAK-SAGE

Quality Assessment Seminar; TÜBİTAK-SAGE

Metrology and Calibration Seminar; TAI

WORK EXPERIENCE :

Organization	***Years***	***Job***
TÜBİTAK-SAGE Flight Dynamics Group / Aerodynamic Division Ankara, TÜRKİYE	September 1997 -	Research Engineer Senior Research Engineer Chief Research Engineer