CONVERGENCE ACCELERATION PROCEDURES FOR THE COMPUTATION OF 2-D TRANSONIC FLOWS

by

Uğur TÜRK

B.S., Mechanical Engineering, Deniz Harp Okulu, 1989

M.S., Mechanical Engineering, Naval Postgraduate School, 1995

Submitted to the Institute for Graduate Studies in
Mechanical Engineering in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

Graduate Program in
Boğaziçi University
2014

ACKNOWLEDGEMENTS

The present study was carried out at Boğaziçi University, Department of Mechanical Engineering under the guidance of Assoc. Prof. Ali Ecder. First I would like to express my gratitude to my advisor, Prof. Ecder, for his support and his guidance throughout the research.

It is a pleasure to be a part of Prof. Ecder's CFD-FMS Research Group and I would like to thank for their contributions to the present work. In particular to my colleagues, Mehmet Orhan, Erhan Turan and Yalın Kaptan, for their substantial inputs and advices during the many interesting discussions and for the excellent personal relations. Additionally, I thank all members of the Department of Mechanical Engineering for their support and the enjoyable work environment.

I would like to thank and express my gratitude to my wife Buket, my son Berk and my daughter Beril for their loving support and patience. I also would like to thank my mother Gülser, my sister Gülçin and my aunt Müzeyyen for their endless support, which was a great motivation needed during this research.

I dedicate my thesis to my father Mustafa TÜRK (1984).

ABSTRACT

CONVERGENCE ACCELERATION PROCEDURES FOR THE COMPUTATION OF 2-D TRANSONIC FLOWS

This study addresses a novel adaptive time stepping procedure, which leads to selection of larger time steps allowed by the physics of the problem. Information about the gradients of the flow variables can be regarded as an indicator for determining proper amount of time step, in which the system evolved. The signals from the pressure sensors, which act according to the pressure gradients, are chosen as a measure to determine the magnitude of the local CFL number. Thus, the aimed methodology for the selection of the local time step with the use of Pressure Sensor introduces optimal time steps to the implicit solution method by accounting for the pressure gradient in the solution domain, such that sharp pressure gradients encourages small time steps and vice versa. To illustrate the effect of proposed procedure, Newton Krylov (NK), with implicit pseudo time stepping method, has been employed to solve the compressible Euler equations for steady transonic case by turning on the pressure switch. Numerical experiments show that the introduced adaptive time stepping procedure decreases the computation time and the number of iterations, effectively. Additionally, a comparison study on the performances of Newton Krylov (NK) and nonlinear multigrid (FMG-FAS) methods are presented. The longer computation time required by NK can be a result of the requirement of Newtons method for a better initial guess. When the free stream values are used as initial guess, a more sophisticated method for time step selection is needed for a better NK performance especially at the start up phase.

ÖZET

2 BOYUTLU TRANSONİK AKIŞLARIN HESAPLANMASI İÇİN YAKINSAMA İVEMELENDİRME PROSEDÜRLERİ

Bu çalışmada, problemin fiziğinin izin verdiği ölçüde büyük zaman adımını seçen yeni bir uyarlamalı zaman adım prosedürü ele alınmıştır. Uygun zaman adımlarının belirlenmesi için, basınç sensörlerinden alınan basınç gradyanlarına bağlı sinyaller, yerel CFL sayısının büyüklüğünü belirler. Basınç sensörü kullanımı ile yerel zaman adımı seçimini amaçlayan metodoloji, basınç gradyanlarını kullanarak implisit çözüm metodları için optimum zaman adımlarını oluşturur, öyle ki yüksek basınç gradyanları küçük zaman adımlarını veya düşük gradyanlar büyük zaman adımlarını oluşturur. Önerilen prosedürün etkinliğini göstermek için, implisit sözde zaman adımı yöntemi ile Newton Krylov (NK) metodu istikrarlı transonik sıkıştırılabilir Euler denklemlerini çözmek için basınç sensörü ile kullanılmıştır. Sayısal deneyler, önerilen adaptif zaman adım prosedürünün hesaplama süresini ve hesap adımlarının sayısını azalttığını göstermiştir. İlave olarak, Newton Krylov (NK) ve nonlineer multigrid (FMG-FAS) yöntemlerinin karşılaştırmasında, NK çözümleri için gereken daha uzun hesaplama zamanının, Newton methodunun daha iyi bir başlangıç tahminine ihtiyaç duymasından kaynaklandığı değerlendirilmiştir. Serbest akış değerlerinin ilk tahmin olarak kullanıldığı durumlarda, NK performansının arttırılması için zaman adım seçiminde daha gelişmiş bir yönteme, özellikle de, akışın geliştiği ilk safhada ihtiyaç olduğu değerlendirilmiştir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

a	speed of sound
D	Artificial dissipation operator
d_h or d_H	defect or residual in multigrid
e	total energy per unit volume
f_h or f_H	right hand side vector
Fc	convective fluxes
Fv	viscous fluxes
$F(u)$	nonlinear vector function
f_c	convective fluxes (x component)
f_v	viscous fluxes (x component)
g_c	convective fluxes (y component)
g_v	viscous fluxes (y component)
H	Hessenberg matrix
I	Identity matrix
$J_{i,j}$	Jacobian matrix
l_m	length of the vector
M	Mach number
M^{-1}	left preconditioning matrix
m	Krylov subspace dimension
N	the size of the Jacobian
n_x,n_y	unit normal vectors in x and y directions
p	pressure
P_H^h (or I_H^h)	Prolongation operator ("coarse-to-fine operator") in multigrid
Pr	Prandtl number
Pr_t	turbulent Prandtl number
Q	vector of conservative variables
q	local flow speed
R	Residual
R_h^H (or I_h^H)	Restriction operator ("fine-to-coarse operator") in multigrid

r	continuation parameter
S	area
$\vec{S}_m$	face vector
T	temperature
t	time
u	solution vector or state vector
u	x component of velocity
v	y component of velocity
v	guess to the solution
$\vec{v}$	velocity vector
$\vec{n}$	unit normal vector (outward pointing) of cell face
V	velocity
α	angle of attack
δu	Newton update
$\varepsilon^{(2)}$	parameter for artificial dissipation
$\varepsilon^{(4)}$	parameter for artificial dissipation
ε_k	stopping criteria for nonlinear problem
ε_{GMRES}	GMRES stopping criteria
$\varepsilon_{DISTURB}$	disturbance value for numerical Jacobian
γ	ratio of specific heats
μ	laminar viscosity
μ_t	turbulent viscosity
ν	pressure sensor for artificial dissipation
Ω	(edges of) control volume
Φ or ϕ	velocity potential
ρ	density
∂	partial derivative
k	nonlinear iteration index (counter)
n	time level

c	convective
h	fine grid in multigrid
H	coarse grid in multigrid
m	approximate values at the cell face
up	upstream cell/value
v	viscous
∞	free stream (farfield) properties
a	free stream value for BC calculation
b	value belong to the cell face at the boundary
d	value belong to the center of boundary cell
0	value at the interior cell center next to the boundary

LIST OF ACRONYMS / ABBREVIATIONS

ADI	Alternating Direction Implicit
AF	Approximate Factorization
CFD	Computational Fluid Dynamics
FAS	Full Approximation Storage
FMG	Full MultiGrid
GMRES	Generalized Minimum Residual
ILU	Incomplete LU preconditioner
MG	MultiGrid
MUSCL	Monotone Upstream Centered Schemes for Conservation Laws
NK	Newton Krylov

1. INTRODUCTION

Computational Fluid Dynamics (CFD) enables research of various engineering problems by using different numerical methods with the use of computers. In the recent decades, there have been numerous developments in CFD methods as well as in the computers.

The improvements in CFD made CFD an indispensable tool for researchers, scientists and engineers. CFD provided an advanced insight to a large range of phenomena of interest, from engineering to health, from education to high technology. Nowadays, CFD is an important step and proven to be a fruitful tool to be conducted in analysis of different kinds of phenomena before experiments or prototype productions start. The problems, which CFD has to tackle, become more complex and difficult as a result of increasing interest from different fields.

With the successes from the problems solved by CFD, the encouragement to solve harder and more complex problems prevails. Even though the advances in computer technology helps making life easier in CFD world, the challenge arises and the tackled problems gets tougher to handle everyday.

With the recent status of CFD given above, the required time to solve complicated problems is the main constraint nowadays. More improvements in algorithms by reducing computation time and memory usage is still needed. In CFD, the required time to reach convergence needs to be reduced by improving faster algorithms (i.e. Convergence Acceleration Procedures) in addition to advances in computer technology. In this research area, so far extensive work has been performed on the compressible (transonic) flow solution by using finite volume methods on structured and unstructured grids. Aerodynamics has been one of the key areas, where CFD has started and been evolved.

Two outstanding methods to solve the nonlinear equations arising from the finite-

volume discretization of the compressible (transonic) equations are "Newton Krylov Methods (NK) with preconditioning" and "nonlinear multigrid methods (FMG-FAS)". These methods are known to give satisfactory results and converge to steady state with reasonable time and computer sources.

In the first part of this thesis, the research is focused on;

(i) the comparison of NK and nonlinear multigrid (FAS) methods and investigate the effects of different numerical methods on the performance from different perspectives and report on the justification/selection criteria (advantages and disadvantages) as to the reduction of CPU time, possible savings in the memory requirements for the transonic flow problems specifically,
(ii) the comparison of the use of different grids in implicit solution of transonic flow calculations.

In the discussion and comparison of algorithms (NK and FMG-FAS methods); transonic flow is used as the model problem. The numerical investigation of Euler equations, which model transonic flow regime, is an extensively studied model problem to be used in development / comparison / improvement of numerical methods due to nonlinear character of the equations and the coupling of equations and unknowns.

In this study, the emphasis is on the numerical methods rather than physical model. The aim is to investigate the numerical solution of the transonic flow using different methods such as implicit solution methods, nonlinear multigrid and preconditioning methodologies.

The second part of this thesis is focused on the following:

In Steady Transonic Flow solution, we have used a novel method for the Selection of Local CFL Parameter by using "Pressure Sensor" in Implicit Methods. For a flow around an airfoil, when the free-stream Mach number (M_∞) and the angle of attack (α) are given relatively high values, some discontinuities like strong shocks are observed

in the flow. The shocks, depending on their severity, create steep gradients in the flow variables. Consequently, the strength of the shock determines the difficulty of the problem. In other words, the higher M_∞ and α values are, the more difficult the problem is.

Newton's method requires an initial guess which is as close to the solution as possible. Especially in case of the transonic flow problems with strong shocks, the use of the free-stream values as initial guess, is not a good representation of transonic flow because it does not contain any shocks. So, using free-stream values as initial guess does not satisfy this requirement for Newton's method and slows down the solution process or causes divergence.

An implicit method, Newton Krylov (NK), with pseudo time stepping, has been used to solve the compressible Euler equations for steady transonic flow. In such problems usually the free-stream flow values are used as initial guess.

In Newton's method, if free-stream values are used as the initial guess, to eliminate divergence in the beginning of iterations, the time step is kept as low as possible to take smaller time steps until the flow develops around the object (an airfoil in our test cases) which creates disturbance. This is called start-up or defect correction phase.

On the other hand, the use of relatively lower time step leads to relatively longer convergence time. For that reason to accelerate convergence, the time step is gradually increased as the flow field develops. The increase of time step is encouraged for the acceleration of solution process but discouraged by the possibility of divergence of Newton's method. This dilemma requires controlled increase of time step.

Normally, for time dependent problems, the use of global time step is necessary. Where as for the steady state problems the use of local time step is possible. The possibility of using different time steps is an advantage and it can be manipulated to achieve faster convergence by selecting local time step.

In theory, implicit time integration methods are assumed to achieve convergence with very high time steps (in theory $\Delta t \rightarrow \infty$). In practice, on the other hand, very high time steps may lead to divergence for higher α and M_∞. As a remedy, it is possible to start iterations by taking relatively smaller time steps until the flow field develops.

In steady state flow solutions, it is possible to use local time steps (i.e. different Δt for each cell) such that;

(i) in the regions where the flow variables change rapidly (the existence of steep gradients), relatively lower local time step is used,
(ii) in the smoother regions, relatively higher local time step is selected.

In the selection of local time step (CFL_{local}) "Pressure Sensor" is used, which can make the distinction between the regions with the sharp gradients and smoother gradients.

With the use of "Pressure Sensor", which detects the existence of sharp gradients, CFL_{local} (Δt_{local}) for each cell is selected appropriately as the iterations progress.

Implicit schemes are widely used in engineering problems with iterative methods. Implicit schemes and iterative methods are preferred also because of their robustness and convergence properties. On the other hand, depending on the existence and the strength of shock(s) (i.e. the difficulty level of the problem determined by α and M_∞), implicit schemes may require more computation time and iterations for transonic flow solutions.

So far extensive work has been performed on the compressible inviscid (transonic) flow solution on structured and unstructured grids by using finite volume and Newton Krylov (NK) Methods. NK as an implicit method is known to give satisfactory results and converge to steady state together with pseudo time stepping with the use of reasonable time and computer sources.

When the free-stream values are used as initial guess, which do not inherit the shocks in transonic flow, it is possible;

(i) to use a different method other than Newton's method at start-up until the flow develops;
(ii) to keep timesteps (i.e. Δt or CFL number) as low as possible and take smaller time steps until the flow develops. Use of relatively low timesteps leads to longer convergence (run) time. On the other hand, if one uses relatively higher time steps to accelerate the convergence, this may cause divergence of Newton's method at start-up.

The main purpose of the second part of this thesis is (without the use of a different method);

(i) To introduce a novel CFL_{local} (i.e. Δt_{local}) selection methodology by the use of "Pressure Sensor" in transonic flow calculation by using NK, which makes it possible to take larger time steps, Δt, in smooth regions and lower Δt in the areas with sharp gradients,
(ii) To investigate the effects of different parameters on the solution methodology, such as the subspace size of GMRES (m), fill-in level for ILU preconditioner, ILU (f_{in}) and CFL_{Rate} are conducted.

The transonic flow around an airfoil is used as the physical model to conduct numerical experiments on the use of "Pressure Sensor" in the determination of CFL_{local} (i.e. Δt_{local}). To achieve the objective of this work; the convergence properties of NK are tested. The effects of "Pressure Sensor" on the selection of CFL_{local} (i.e. Δt_{local}) and on the convergence performance are investigated.

1.1. Background

Computational Fluid Dynamics has become one of the major tools for the aerodynamic design in aircraft industry and other areas of engineering which deals with com-

pressible or incompressible flows. Higher costs of experimental design especially wind tunnel testing, can be counted as the main reason for CFD's increasing importance. The need for CFD as a design tool in aircraft industry led to various methodologies for compressible flow simulation.

Jameson [21] reported that "Aerodynamic design is aided by increasing performance of computers. In addition to increasing capacity of computers, further development in numerical methods has a positive influence on the development of CFD. With more efficient numerical methods, it is possible to create more efficient algorithms for compressible flow equations. Efficient algorithms increase the reliability of flow simulations and with these algorithms it is possible to simulate more complex flows".

The research is still going on for developing algorithms, which can solve Euler equations efficiently. The research for more efficient algorithms is focused on both physical models and numerical methods (implicit iterative methods, nonlinear multigrid, preconditioning, etc).

Generally, the motivation of this work and the others in this specific field is to develop computer simulation of compressible flow to be used in the aircraft design and similar compressible flow applications of engineering.

The objective of this research is to develop / compare / improve / investigate / discuss algorithms by comparing different numerical methods for transonic flow. Existing and newly developed algorithms are tested to achieve reduction of computation time and memory usage, which contribute to faster convergence.

The physical model is transonic flow around an airfoil. For this model problem inviscid flow equations are solved.

Their convergence properties are tested and the results are compared and the effects of different numerical methods on the performance are investigated. As summary, the methods, mentioned above, have been tested and discussed for Euler Equation.

1.2. Literature Review

In this research area, the primary goal of the research in computational fluid dynamics is to develop algorithms, which solve the governing equations for compressible flow efficiently and accurately.

1.2.1. Explicit Schemes

1.2.1.1. Explicit Runge-Kutta with Artificial Dissipation. Jameson *et al.* [15] solved Euler Equations around an airfoil by using the Runge-Kutta Time-Stepping Schemes and Artificial Dissipation (Appendix C). Runge-Kutta explicit time stepping scheme with spatial central finite volume discretization is one of the widely used explicit time stepping schemes.

The solution of discretized Euler equations yields oscillations, when there are strong jumps like shocks in the transonic flow. They have used central differentiation with artificial dissipation (or "JST dissipation" named after the authors Jameson, Schmidt and Turkel) for stabilization. Acceptable results are achieved especially with the use of JST dissipation scheme for various problems in compressible flows including transonic regime. Runge-Kutta Time-Stepping Scheme with Artificial Dissipation has become a classical method and widely used by many researchers and in different applications.

Jameson *et al.* [15] used a blend of second-and fourth order differences in order to control oscillations. This scheme is called the Jameson-Schmidt-Turkel (JST) scheme or artificial dissipation. In the work of Jameson *et al.* [15], Runge-Kutta time stepping scheme is used as the explicit solution technique for Euler equations. In the presence of discontinuities (like shock waves), some oscillations are introduced by the third order diffusive flux. In order to eliminate the oscillations a switch is used to sense the shock wave and the artificial diffusion is switch to first order locally. The switch decreases the level of diffusion in the smooth regions of the flow, but increases the level of diffusion near the oscillations near discontinuities [22].

1.2.1.2. Explicit Runge-Kutta with Multigrid Method. Nonlinear multigrid methods have proven to perform well in solution of nonlinear problems. In the nonlinear multigrid solution, the linearization of the problem is not required. In Jameson [16], solution of finite volume discretized Transonic Euler equations are sought by using nonlinear multigrid method. In the application of multigrid methods, the selection of the following is highly important;

(i) The number of levels and grid sizes,
(ii) Smoothing method and
(iii) Prolongation (interpolation) and restriction operators.

With the right selection of above components, it is possible to achieve better convergence properties, such as reduction in memory requirement, computer time and residuals as reported by Washio and Oosterlee [8], Fidkowski and Darmofal [10] and Sidilkover [11].

1.2.2. Implicit Schemes

Implicit methods are robust and have preferable convergence speed but require more computational effort per iteration. With implicit schemes, Newton's method can be used as a nonlinear system solver and it is possible to use larger time steps. At each time step of an implicit scheme, the time step (CFL number) has to be carefully chosen.

Some properties of the implicit schemes are as follows:

(i) It is possible to use larger time steps in implicit schemes.
(ii) With implicit schemes, Newton's method can be used (because of some practical application issues, the theoretical quadratic convergence property may be downgraded).
(iii) The implicit schemes are robust and have higher convergence speed but require more computational effort per iteration.

(iv) In some applications it is hard to vectorize and/or parallelize implicit schemes.

Since the computer resources, required by direct methods, are extremely high, it is better to use the implicit schemes with iterative methods. If Newton's method is used, a linear system must be solved at each time step. The approximation of Jacobian is the limiting factor on the quadratic convergence speed of Newton's method. For example, when first order Jacobian approximation is used, the time step (CFL number) has to be finite.

1.2.2.1. Implicit ADI and multigrid. Alternating Direction Implicit - Approximate Factorization (ADI/AF) scheme is one of the first iterative implicit schemes used for the solution of compressible flows. Beam and Warming approximate factorization type, implicit schemes are in use of compressible flow solution since 1970s. Beam and Warming [37] have developed an implicit approximate factorization scheme for the compressible flow equations. ADI/AF can be used only with structured grids. One ADI/AF iteration is performed at each implicit time step.

Anderson and Bonhaus [20] used an implicit Navier Stokes solution algorithm for the computation of turbulent flow. They have used upwinding in the calculation of inviscid fluxes. They have performed numerical experiments on NACA 0012 and RAE 2822 airfoils. In general their results were in agreement with the experimental data available.

1.2.2.2. LU-SGS and Multigrid. The implicit Lower- Upper Symmetric Gauss-Seidel (LU-SGS) scheme has similar benefits to an explicit RK scheme, with its lower memory requirements compared to the other implicit schemes. The implicit LU-SGS scheme can be used both with structured and unstructured grids.

Caughey and Jameson [17] has reported on multigrid solution of the compressible Euler and Navier Stokes equations by using a new version of implicit method, which implements a nonlinear Symmetric Gauss Seidel (SGS) algorithm with Lower Upper

(LU) preconditioner.

For transonic flows, Shen *et al.* [23] compared the Implicit Gauss-Seidel Line Iteration Method with the LU-SGS when the Roe scheme is used. The researchers studied the sweep direction effect on the convergence rate and CPU time of these methods for compressible flows. According to the results of their numerical experiments, they concluded that the best convergence rate is obtained with sweeping in streamwise direction for an inviscid transonic internal flow. As another outcome, they determined that one sweep (a forward sweep plus a backward sweep) per time step is sufficient within each time step.

1.2.2.3. Newton-Krylov. Jiang and Forsyth [12–14] have reported their work on transonic flow solutions using Full Potential equation, Euler equations and Laminar Navier Stokes equations respectively. In their studies for the solution of Full Potential equation, they have used numerical methods like upstream weighted discretization, ILU preconditioning, flux-biasing finite volume method. They have solved the nonlinear systems by using a continuation method with full Newton iteration. For transonic Euler equations, they used an incomplete LU factorization with CGSTAB, which is a Krylov subspace method. They have tested their algorithms at various airfoil shapes at transonic and supersonic conditions. They have concluded that full Newton approach and high order Jacobian technique is more robust and efficient for transonic flow solution. In reference [12], high speed laminar compressible Navier Stokes equations are solved by using full Newton nonlinear iteration with block ILU preconditioner and CGSTAB acceleration.

Choquet and Erhel [24] have applied Newton-Krylov algorithms to compressible flow but not transonic case. They used implicit time discretization for the solution of nonlinear equations of compressible flow. In the finite volume formulation, they used upwinding for convective terms. The Riemann problem at the edges of finite volume cells is solved by Osher scheme, where the flux is approximated. Jacobian is approximated by a finite difference scheme. At the linear system solution, restarted

GMRES [2] is used. They have concluded that, (i) GMRES directs the solution process in the right path if the convergence tolerance of the linear solver is small enough, (ii) If the Jacobian approximation is good enough, a linear convergence of Newton is possible. Their final conclusion is that, the use of a second order finite difference scheme to approximate the initial residual at each start of GMRES, improves the convergence of Newton method.

Blanco and Zinng [25] have developed a Newton-GMRES (unstructured) solver for Euler equations.

Pueyo and Zinng [26, 27] reported their work on aerodynamic calculations by using Newton-Krylov methods. Chisholm and Zinng [28] have extended their Newton-Krylov solver to a fully coupled one, which is capable of solving turbulent aerodynamic flows.

1.2.3. More on Compressible Flow Solution by using NK and FMG-FAS

Regarding the literature review, more recent studies are listed below, which aim to investigate the performance of different numerical methods like nonlinear multigrid, implicit iterative methods and preconditioning. The mentioned numerical studies use transonic flow as a test case, which means the research for more efficient numerical methods for transonic flow is still going on. In connection, the aim of this thesis work and the organization of the thesis are given in the next section.

Manzano *et al.* [29] have developed a Newton-Krylov Algorithm for the solution of the Euler Equations using unstructured grids.

Caughey and Jameson [17] developed a compressible flow solver, which implements preconditioned multigrid for the Euler and Navier Stokes Equations.

Rossow [30] reported on enhanced Runge-Kutta/Implicit Methods for Solving the Navier-Stokes Equations.

Michalak and Ollivier-Gooch [31] reported on numerical solution of compressible flow using implicit methods which implements matrix-explicit GMRES and higher order (higher than first-order which is the typical case) Jacobian for preconditioning on unstructured grids. In their study, the researchers focused on forming the higher-order Jacobian at a reasonable cost and results in improvements in (i) preconditioning and (ii) reduced efforts for inner GMRES (linear iterations). (matrix-explicit vs. matrix-free). They have concluded that the matrix-explicit methods requires less time than the matrix-free methods when overall computation time is considered.

Michalak and Ollivier-Gooch [31] have used matrix-free approach, which is typical in Newton GMRES applications, is used. Instead of multiplying full Jacobian by a vector, the matrix-vector products required by GMRES are computed by using Fréchet derivatives. GMRES requires preconditioning for efficient convergence of linear system. When incomplete LU factorization is used as the preconditioner, the Jacobian matrix is required for preconditioning. Since ILU preconditioning is used only to precondition the linear system, the first-order approximation of Jacobian matrix is used for typical applications.

Maciel [32] used the Steger and Warming and the Van Leer Schemes to suppress oscillations in the existence of strong shocks in a diffuser and a ramp as the model problems. Maciel has implemented a MUSCL approach to obtain second order accuracy and the Minmod limiter. Both the Steger and Warming and the Van Leer Schemes are flux vector splitting schemes and can employ approximate factorizations to be solved by implicit ADI solver. In his implicit solutions, CFL number ranged from 1.3 to 3.6 and the convergence was achieved in less than 100 iterations.

1.2.4. CFL Selection Strategies

Different strategies exist for the determination of Δt_{local} (CFL_{local}). These strategies are mainly based on the residual and act on the CFL value globally by considering the increase or decrease of residual value as outlined below.

Bücker *et al.* [43] introduced a novel CFL evolution strategy and compared this strategy with two existing strategies in implicit methods for the solution of linearized Euler equations. After CFL number reaches to a predetermined limit, CFL number is ramped up and this increase is controlled by one of the following strategies: (i) Exponential Progression (EXP); (ii) Switch Evolution Relaxation (SER); (iii) Residual Difference Method (RDM). RDM is proposed by Bücker *et al.* [43]. They tried to achieve a CFL evolution strategy, to find the balance between large CFL numbers to achieve fast convergence and small time steps to guarantee the convergence of Newton's method and resolve all flow features. They have concluded that that all three CFL evolution strategies have their advantages and disadvantages. CFL control still requires application-specific knowledge, intuition, and trial and error.

Pollul and Reusken [44] have investigated some strategies for choosing the time step for iterative methods in implicit time integration for compressible flow simulations. In Pollul and Reusken [44], the size of the time step is determined by a CFL number, which is not limited by the Courant-Friedrichs-Levy (CFL) condition. Initialized by CFL_{MIN}, the CFL number is increased by an evolution method in every time step until a previously determined upper bound CFL_{MAX} is reached. Their main goal was to improve efficiency and robustness of the iterative method used in the flow solver by careful selection of the time step size. They have presented results of numerical experiments using the different CFL evolution strategies with different CFL Control Parameters. They have investigated two known (EXP and SER) and a novel time step selection strategy (RDM) and an alternative one. They have concluded that the investigated CFL evolution strategies, the choice of the control parameters has a great impact on the number of time steps needed for convergence and the total execution time. The results have shown that some increase of the residual can be accepted to achieve rapid overall convergence. RDM, proposed by the authors, has been compared with the existing strategies EXP and SER. For the residual-based strategies SER and RDM, RDM has turned out to be faster than SER. They have concluded that application-specific knowledge, intuition, and trial and error are still needed in order to determine appropriate values for the CFL control parameters.

1.3. Organization of Thesis

In this thesis, by keeping the focus on the implicit solution algorithms, nonlinear multigrid, Newton-Krylov methods with preconditioning and convergence acceleration procedures, the compressible flow equations are solved for transonic case. Navier Stokes equations and Euler equations, which are discretized by finite volume method, are investigated.

In Chapter 2 (Mathematical Modeling of Transonic Flow), the governing equations of compressible flow are given. The derivation, discretization and other related details are given for the model problem, which is transonic flow around an airfoil. The details related to governing equations, boundary conditions, finite volume discretization, upwind scheme (Roe's flux difference splitting), and limiters are covered.

In Chapter 3 (Computational Modeling and Numerical Methods), the details of numerical methods like (i) implicit Newton-Krylov methods and preconditioning, (ii) nonlinear multigrid are given.

In Chapter 4, (An Adaptive Time-Stepping Strategy for the Implicit Solution of Steady Transonic Flows), Newton Krylov (NK), with implicit pseudo time stepping method, has been used to solve the compressible inviscid flow (Euler) equations for steady transonic case. In this part, a novel local time step (Δt_{local} or CFL_{local}) selection method based on the gradients of flow variables is introduced with the use of "Pressure Sensor". Numerical experiments showed that the introduced local time step selection method can decrease the computation time and the number of iterations.

In Chapter 5, the details of numerical experiments and obtained results are presented and comparisons are made. In Chapter 6, the conclusions, which are drawn from the thesis studies, are given.

2. MATHEMATICAL MODELING OF TRANSONIC FLOW

For the numerical experimentation, the viscous compressible flow around an airfoil is described by Navier Stokes equations and the inviscid compressible flow around an airfoil is described by Euler equations.

In the following sections, the derivation and related details for compressible Navier Stokes Equations and compressible Euler Equations are given, which can be used in the analysis of transonic flow.

2.1. Navier-Stokes Equations for Compressible Flow

Navier Stokes equations for compressible flow, where (i) only laminar flow considered and (ii) turbulent viscosity are introduced below.

The conservation equations for mass, momentum and energy can be written in the form of a volume and a surface integral. In the below formulation the source terms and the external forces are not shown. The below formulation forms Navier Stokes equations for compressible flow with the viscous terms.

$$\frac{\partial}{\partial t}\int_{\Omega} Q d\Omega + \oint_{\partial\Omega}(F_c - F_v)dS = 0 \tag{2.1}$$

where

$$F_c = f_c + g_c \tag{2.2}$$

$$F_v = f_v + g_v. \tag{2.3}$$

2.1.1. Laminar Navier-Stokes Equations for Compressible Flow

Navier Stokes equations in conservative form for compressible flow around an airfoil is given below. The conservation equations for mass, momentum an energy can be written in the below form without the source terms as in Forsyth and Jiang [14].

$$\frac{\partial Q}{\partial t} + \frac{\partial f_c}{\partial x} + \frac{\partial g_c}{\partial y} = \frac{\partial f_v}{\partial x} + \frac{\partial g_v}{\partial y} \tag{2.4}$$

Q is the vector of conservative variables. f_c, g_c are the convective fluxes.

$$Q = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ e \end{bmatrix} \quad ; \quad f_c = \begin{bmatrix} \rho u \\ \rho u^2 + P \\ \rho uv \\ u(\rho e + P) \end{bmatrix} \quad ; \quad g_c = \begin{bmatrix} \rho v \\ \rho uv + P \\ \rho v^2 + P \\ v(\rho e + P) \end{bmatrix}. \tag{2.5}$$

f_v and g_v are viscous fluxes;

$$f_v = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xy} v + q_x \end{bmatrix} \quad ; \quad g_v = \begin{bmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ \tau_{xy} u + q_y \end{bmatrix} \tag{2.6}$$

where the shear stress and heat conduction terms are given as;

$$\tau_{xx} = \frac{2M_\infty \mu}{3Re_\infty}(2\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y}) \tag{2.7}$$

$$\tau_{yy} = \frac{2M_\infty \mu}{3Re_\infty}(2\frac{\partial v}{\partial y} - \frac{\partial u}{\partial x}) \tag{2.8}$$

$$\tau_{xy} = \frac{M_\infty \mu}{Re_\infty}(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}) \tag{2.9}$$

$$(q_x, q_y) = \frac{M_\infty \mu}{(\gamma - 1)PrRe_\infty}\nabla T \tag{2.10}$$

where; ρ is density, u is x-direction velocity, v is y-direction velocity, e is total energy per unit volume, p is pressure, μ is viscosity, M_∞ is freestream Mach number, Re_∞ is freestream Reynolds number, Pr is Prandtl number, γ is ratio of specific heats, T is temperature.

Sutherland's law is used to compute the laminar viscosity.

$$\mu = \frac{(1 + C^*)T^{3/2}}{T + C^*} \tag{2.11}$$

where $C^* = 198.6/460.0$ assuming the freestream temperature is 460 R. The equations are closed with the equation of state for a perfect gas given by Equations 2.20 and 2.21.

2.1.2. Turbulent Navier-Stokes Equations for Compressible Flow

In case the turbulent effects are added to the solution, shear stress and heat conduction terms given as follows. (Anderson [20]).

$$\tau_{xx} = (\mu + \mu_t)\frac{2M_\infty}{3Re_\infty}(2\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y}) \tag{2.12}$$

$$\tau_{yy} = (\mu + \mu_t)\frac{2M_\infty}{3Re_\infty}(2\frac{\partial v}{\partial y} - \frac{\partial u}{\partial x}) \tag{2.13}$$

$$\tau_{xy} = (\mu + \mu_t)\frac{M_\infty}{Re_\infty}(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}) \tag{2.14}$$

$$q_x = \frac{-M_\infty}{(\gamma - 1)Re_\infty}(\frac{\mu}{Pr} + \frac{\mu_t}{Pr_t})\frac{\partial a^2}{\partial x} \tag{2.15}$$

$$q_y = \frac{-M_\infty}{(\gamma-1)Re_\infty}(\frac{\mu}{Pr} + \frac{\mu_t}{Pr_t})\frac{\partial a^2}{\partial y} \tag{2.16}$$

where μ_t and Pr_t are turbulent viscosity and turbulent Prandtl number respectively. a is speed of sound.

In order to calculate turbulent viscosity a turbulence model has to be selected from one of the algebraic models, one- or two equation turbulence models.

2.2. Euler Equations for Compressible Flow

If the viscous terms are neglected from Equation 2.1, Euler Equations are obtained. Euler equations for compressible flow for a control volume Ω with boundary $\partial\Omega$ are given below without the source terms and the external forces.

$$\frac{\partial}{\partial t}\int_\Omega Q d\Omega + \oint_{\partial\Omega} F_c dS = 0 \tag{2.17}$$

In this formulation, Q is the solution vector of conservative variables and F_c is the vector of the convective (inviscid) fluxes.

$$Q = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho e \end{bmatrix} \quad ; \quad F_c = \begin{bmatrix} \rho V \\ \rho u V + n_x P \\ \rho v V + n_y P \\ (\rho e + P)V \end{bmatrix} \tag{2.18}$$

Other variables are as follows: ρ is the density, u is the x-direction velocity, v is the y-direction velocity, e is the total energy per unit volume, P is the pressure and V is the contravariant velocity;

$$V \equiv \vec{v} \cdot \vec{n} = n_x u + n_y v. \tag{2.19}$$

$\vec{n}_x$ and $\vec{n}_y$ are the components of the unit normal vector pointing outward from the flux face. Total energy per unit volume and pressure are related as follows:

$$e = \frac{P}{(\gamma - 1)\rho} + \frac{1}{2}(u^2 + v^2).$$

The equations are closed with the equation of state for a perfect gas

$$P = (\gamma - 1)[e - \frac{\rho(u^2 + v^2)}{2}] \tag{2.20}$$

$$T = \frac{\gamma P}{\rho} \tag{2.21}$$

γ is the ratio fo the specific heats and T is the temperature.

2.2.1. Finite Volume Discretization of Euler Equations

Euler equations are discretized with finite volume method by using quadrilateral grid. As it can be seen in Figure 2.1 and cell centered finite volume approach is used.

$$\frac{\partial}{\partial t}\int\int_{\Omega} Q dx dy + \oint_{\partial\Omega}(f_c dy - g_c dx) = 0 \tag{2.22}$$

with

$$Q = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho e \end{bmatrix} \quad ; \quad f_c = \begin{bmatrix} \rho u \\ \rho u^2 + P \\ \rho uv \\ (\rho e + P)u \end{bmatrix} \quad ; \quad g_c = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + P \\ (\rho e + P)v \end{bmatrix} \tag{2.23}$$

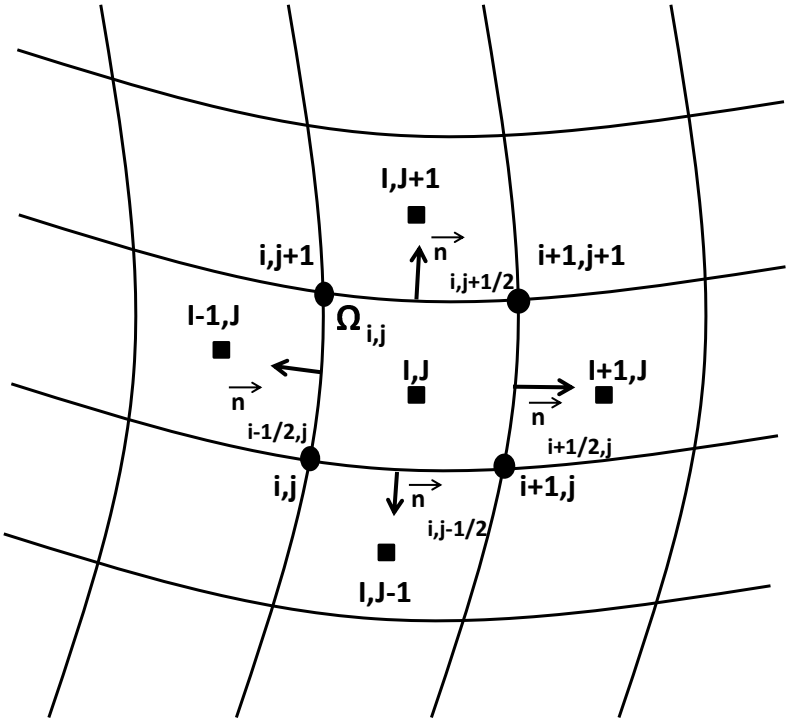

Figure 2.1. Cell centered finite volume discretization indexing.

2.2.2. Spatial Discretization of Euler Equations

Euler equations 2.17 are discretized with finite volume method by using quadrilateral grid. The cell centered finite volume approach is used. With the finite volume discretization of Euler equations, residual R is obtained.

$$\frac{dQ_{I,J}}{dt} = -\frac{1}{\Omega_{I,J}} \sum_{m=1}^{4} (F_c)_m \Delta S_m = -\frac{1}{\Omega_{I,J}} R_{I,J}, \tag{2.24}$$

where the face vector is

$$\vec{S}_m = \vec{n}_x \Delta S_{x,m} + \vec{n}_y \Delta S_{y,m}.$$

When cell centered finite volume discretization is applied to continuity, x momentum, y momentum and energy equations the following discretized equations are obtained.

$$\frac{\partial}{\partial t}(\Omega \rho_m) + \sum_{m=1}^{4} (\triangle y_m \rho_m u_m - \triangle x_m \rho_m v_m) = 0 \tag{2.25}$$

$$\frac{\partial}{\partial t}(\Omega (\rho u)_m) + \sum_{m=1}^{4} (\triangle y_m (\rho u)_m u_m + \triangle y_m P_m - \triangle x_m (\rho u)_m v_m) = 0 \tag{2.26}$$

$$\frac{\partial}{\partial t}(\Omega(\rho v)_m) + \sum_{m=1}^{4}(\triangle y_m(\rho v)_m u_m - \triangle x_m(\rho v)_m v_m - \triangle x_m P_m) = 0 \tag{2.27}$$

$$\begin{aligned}\frac{\partial}{\partial t}(\Omega(\rho e)_m) + \sum_{m=1}^{4}(&\triangle y_m(\rho e)_m u_m + \triangle y_m P_m u_m \\ &- \triangle x_m(\rho e)_m v_m - \triangle x_m P_m v_m) = 0\end{aligned} \tag{2.28}$$

As an example where $m = 1$ at South cell face between (I, J) and $(I, J-1)$; $(\rho u)_m$ is approximated as follows:

$$(\rho u)_{m=1} = \frac{1}{2}(\rho u)_{I,J} + \frac{1}{2}(\rho u)_{I,J-1}. \tag{2.29}$$

At the other faces and for the other variables like ρ_m, $(\rho u)_m$, $(\rho v)_m$ and $(\rho e)_m$ the same methodology shall apply.

2.2.3. Implicit Time Integration Scheme

The discretization of the governing equations (Equation 2.17) by using finite volume and the computation of the inviscid fluxes at the cell faces by using Roe's flux splitting scheme yield a system of nonlinear equations. Starting from Equation 2.24, below formulation constitutes implicit scheme where $R(Q)$ is the residual.

$$\frac{\Delta Q}{\Delta t} + \frac{1}{\Omega} R^{n+1}(Q) = 0, \quad where \quad \Delta Q = Q^{n+1} - Q^n \tag{2.30}$$

Where ΔQ represents the update to the vector of conserved variables for a time step Δt. In equation (2.30), n is the current time level, $n+1$ is the sought time level.

The residual can be linearized at the current time level by using below Equation 2.31.

$$R^{n+1} = R^n + \left(\frac{\partial R}{\partial Q}\right)^n \Delta Q \tag{2.31}$$

$\frac{\partial R}{\partial Q}$ is the Jacobian matrix or flux Jacobian. Combining Equations 2.30 and 2.31 yields Equation 2.32. The resulting system from Equation 2.32 is solved by two different methods, NK and FMG-FAS, for the steady state solution.

$$\left[\frac{\Omega}{\Delta t} I + \left(\frac{\partial R}{\partial Q}\right)^n\right] \Delta Q^n = -R^n(Q) \tag{2.32}$$

I is the identity matrix, which can be used for steady state solutions.

2.2.4. Upwind Schemes

Roe's approximate Riemann solver with Van Leer's MUSCL (Monotone Upstream Centered Schemes for Conservation Laws) approach is implemented. Roe's scheme is one of the a few remarkable upwind schemes that have been successfully applied to the

Euler equations and used to control oscillations and provide positivity.

Roe's Flux Difference Splitting [38] evaluates the convective fluxes at a face of a control volume. In this scheme, the values on either side of the face of the control volume are called "left" and "right" states. The "left" and "right" states are used to calculate the flux at the interface between two cells (or more generally control volumes). With Roe's scheme, it is possible to capture shocks at a single point for stationary normal shocks [17].

Roe linearized the equations locally by using mean values of Jacobian [38]. The use of Roe's approximate Riemann solver enables the exploration of the wave motions and implementation of Godunov-type schemes efficiently. On the other hand Roe's approximate Riemann solver is an efficient flux difference splitting scheme, where it is possible to capture shocks at a single point for stationary normal shocks as reported by Caughey and Jameson [17].

In order to control oscillation and sustain positivity, Van Leer [39], [40] has devised MUSCL (Monotone Upstream-Centered Schemes for Conservation Laws). In Van Leer's MUSCL approach, the required monotonicity is provided by the use of limiters and the solution accuracy is extended from first order to higher orders. Venkatakrishnan's Limiter [41] is used to suppress oscillations near strong discontinuities like shocks in transonic flow. The use of limiters increases the CPU usage and the time required for the convergence.

2.2.4.1. Limiters. Limiters are used to suppress oscillations near strong discontinuities like shocks in transonic flow. The use of limiters increases the CPU usage and the time required for the convergence. Venkatakrishnan's Limiter [41] is tested in this study.

Michalak and Ollivier-Gooch [31] have used Venkatakrishnan's limiter [41] in the Jacobian matrix. They have presented and compared to their adaptation of the limiter in order to eliminate over shoots.

Maciel [32] has implemented a MUSCL approach and the Minmod limiter.

2.2.5. Boundary Conditions for Euler Equations

2.2.5.1. Solid Boundary Conditions (Inviscid). On the wing surface, slip boundary conditions for inviscid flow are used for Euler equations, where the surface normal of the flow velocity is zero.

$$\vec{v} \cdot \vec{n} = 0 \text{ at the wing surface.} \tag{2.33}$$

The pressure at the wing surface is specified at the cell faces which forms the airfoil. p_{wall} is calculated by following extrapolation;

$$p_{wall} = \frac{1}{8}(15p_I - 10p_{I+1} + 3p_{I+2}) \text{ at the wing surface.} \tag{2.34}$$

p_I, p_{I+1} and p_{I+2} are the pressure values at the cell centers of 3 cells next to the surface in the normal direction, p_I being the closest to the wing.

2.2.5.2. Farfield Boundary Conditions. Farfield boundary conditions for subsonic and supersonic case is summarized below.

Subsonic farfield boundary conditions are specified by the following formulae, where subscript b denotes the values belong to the cell face at the boundary, subscript a denotes the free stream values and subscript d denotes the values belong to the center of boundary cell. ρ_0 and c_0 are taken as the values at the interior cell center next to

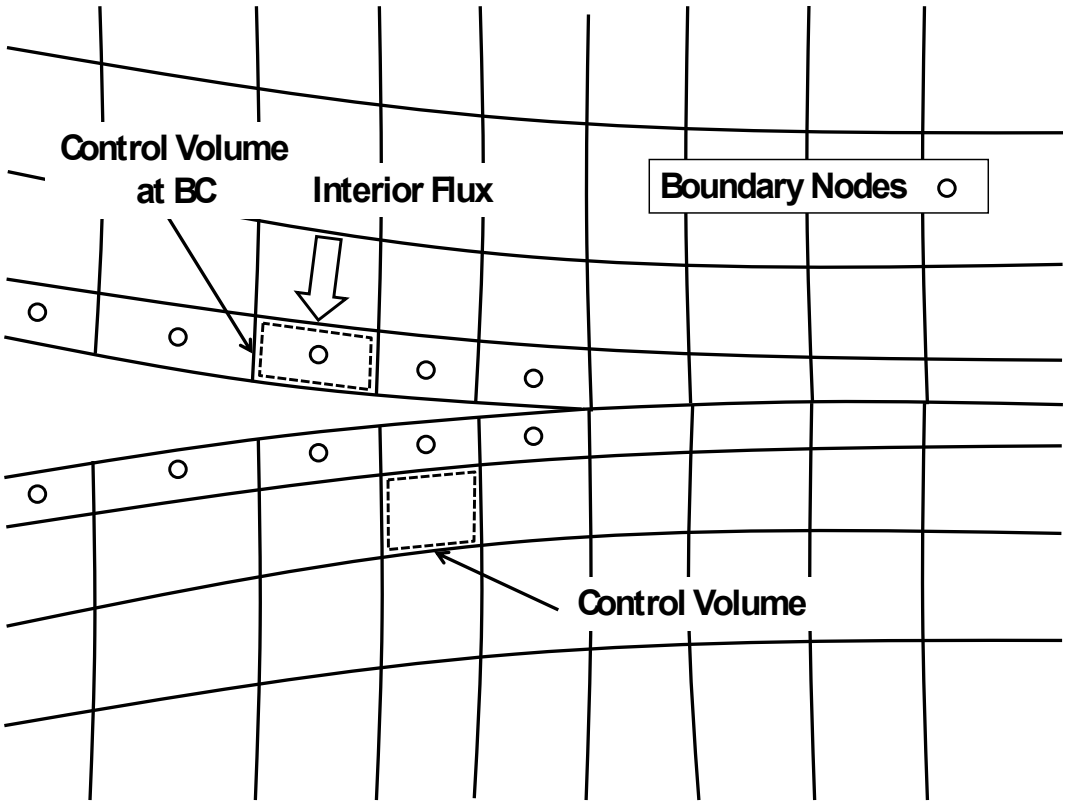

Figure 2.2. Circles denote the cell centers, where p_I's are defined.

the boundary.

At the farfield boundary, the flow is considered either as subsonic inflow or subsonic outflow depending on the location of the cell and orientation of the cell face and flow direction. The subsonic inflow case is given as follows:

$$p_b = \frac{1}{2}\{p_a + p_d - \rho_0 c_0[n_x(u_a - u_d) + n_y(v_a - v_d)]\} \tag{2.35}$$

$$\rho_b = \rho_a + \frac{(p_b - p_a)}{{c_0}^2} \tag{2.36}$$

$$u_b = u_a - n_x \frac{(p_a - p_b)}{\rho_0 c_0} \tag{2.37}$$

$$v_b = v_a - n_y \frac{(p_a - p_b)}{\rho_0 c_0} \tag{2.38}$$

The subsonic outflow case is as follows:

$$p_b = p_a \tag{2.39}$$

$$\rho_b = \rho_d + \frac{(p_b - p_d)}{{c_0}^2} \tag{2.40}$$

$$u_b = u_d + n_x \frac{(p_d - p_b)}{\rho_0 c_0} \tag{2.41}$$

$$v_b = v_d - n_y \frac{(p_d - p_b)}{\rho_0 c_0} \tag{2.42}$$

At the cuts, the neighboring cells in the physical domain are introduced to the flux calculations with careful implementation of cell and cell face indexes.

3. COMPUTATIONAL MODELING AND NUMERICAL METHODS

Computational Fluid Dynamics (CFD), with its increasing importance, is an area, where the solution methods for engineering problems are developed with the use of computers. In the recent decades, there have been numerous developments in CFD methods as well as in the computers. The main purpose of this study is to utilize the modern methods in CFD, to solve the nonlinear governing equations arising from the transonic flow problem.

3.1. Newton-Krylov Methods

In a nonlinear problem solution, Newton-Krylov methods use Newton's method for nonlinear system solution and one of the Krylov subspace methods for linear system solution at each Newton step. The use of these two methods together for nonlinear problems leads to matrix-free implementation, where the solution can be achieved without forming or inverting Jacobian as explained in the following parts. The use of Krylov subspace methods in conjunction with the Newton's method, are simply called Newton-Krylov methods.

The use of robust algorithms with Newton's method to solve various problems of interest is growing in popularity mainly due to the rapid progress in computer speed and available memory and advances in iterative solution methods. Newton's method is a robust technique which converges rapidly for nonlinear problems if the initial guess is close to the solution. In transonic flow problem, the discretization of flow field results in nonlinear systems of equations, which can be solved by Newton's method.

3.1.1. Newton's Method; Nonlinear Problem Solution

The nonlinear system of equations obtained from the discretization of governing equations can be solved by Newton's method. The large sparse nonlinear system of

equations is

$$F(u) = 0, \tag{3.1}$$

where F is the nonlinear vector function of the discrete governing equations and boundary conditions. u is the solution vector with all unknowns. Newton's method has quadratic convergence rate, if the initial guess, $u^{(0)}$, is close enough to the solution.

$$J^{(k)}\delta u^{(k+1)} = -F(u^{(k)}) \quad \text{where} \quad J_{i,j} = \frac{\partial F_i(u)}{\partial u_j}, \tag{3.2}$$

where $J_{i,j}$ is the Jacobian matrix and where k nonlinear iteration counter.

At kth Newton step with the initial guess, $u^{(k)}$, a linear problem is solved to determine the Newton update, $\delta u^{(k+1)}$. Then the new approximate solution is calculated by,

$$u^{(k+1)} = u^{(k)} + \delta u^{(k+1)}$$

The stopping criteria for the solution of nonlinear problem can be determined by selecting ε_k and applying the one of the following two equations.

$$\|F(u^{(k)}) - F(u^{(k+1)})\| \leq \varepsilon_k \|F(u^{(k)})\|$$

or

$$\|F(u^{(k)})\| \leq \varepsilon_k \|F(u^{(0)})\|$$

Each Newton step requires the solution of a large, sparse, non-symmetric linear system, in order to find update vector, $\delta u^{(k+1)}$. A Krylov subspace method, Generalized Minimum Residual (GMRES) [2], is used for the solution of the linear system of equations. Krylov subspace methods with emphasis on GMRES are outlined below.

3.1.2. Krylov Subspace Methods; Iterative Linear System Solver

Krylov subspace methods are fast iterative methods and used to solve the linear systems at each Newton step. The updates, which are required by the nonlinear problem solution, are found by approximately solving,

$$J^{(n+1,k)}\delta u^{(k+1)} = -F(u^{(n+1,k)}) \tag{3.3}$$

$$\text{where} \qquad u^{(n+1,k+1)} = u^{(n+1,k)} + \delta u^{(k+1)} \qquad \text{and} \qquad J_{i,j} = \frac{\partial F_i(u)}{\partial u_j}$$

Here; J is Jacobian matrix, $F(u)$ is nonlinear system of equations, u is the state vector, k is the nonlinear iteration index.

In the numerical solution of the linear system in Equation 3.3 the iteration is said to converge if

$$\|\delta u^{(k+1)}\|_2 \leq \varepsilon_{GMRES}\|F(u^{(k)}) - J^{(k)}\delta u^{(k)})\|_2.$$

where

A striking advantage of iterative methods like Krylov subspace methods is the re-

1: Compute $r_0 = b - Ax_0$, $b = \|r_0\|_2$ and $v_1 = r_0/b$
2: Define $H = [h_{i,j}]_{1 \le i \le m+1, 1 \le j \le m}$ ▷ Set $H_{m+1,m} = 0$
3: **for do**$j = 1, 2, ..., m$
4: Compute $w_j = Av_j$
5: **for do**$i = 1, ..., j$
6: $h_{i,j} = (w_j, v_j)$
7: $w_j = w_j - h_{i,j}v_j$
8: **end for**
9: $h_{j+1,j} = \|w_j\|_2$ ▷ If $h_{j+1,j} = 0$ set $m = j$ and go to 12
10: $v_{j+1,j} = w_j/h_{j+1,j}$
11: **end for**
12: Compute y_m the minimizer of $\|\beta e_1 - y\|_2$ and $x_m = x_0 - V_m y_m$

Figure 3.1. GMRES algorithm.

duced memory requirements (compared to the factorization based methods, like Gaussian elimination). On the other hand, it is possible to start iterations with a relaxed tolerance and decrease the magnitude of the tolerance with increasing iteration number, while the convergence takes place as reported by McHugh and Knoll [18].

3.1.3. Generalized Minimum Residual-GMRES

GMRES is an iterative solver for large non-symmetric linear systems developed by Saad *et al.* [2]. It is a projection based method and uses Krylov subspaces.

GMRES does not require the Jacobian to be symmetric and/or positive definite. GMRES minimizes the norm of the computed residual vector at every step over a Krylov subspace, which contains a certain number of orthogonal search directions. The algorithm is derived from the Arnoldi process for constructing an I_2-orthogonal basis of Krylov subspaces.

Using GMRES, the norm of the residual is non-increasing, and the norm can be monitored without constructing intermediate iterates.

The cost of allowing the Jacobian to be nonsymmetric in Krylov methods is that the current iterate depends on all previous iterates that causes operation count and storage requirements to grow quadraticly and linearly, respectively, in the iteration index. To overcome this problem, one can restart the algorithm periodically or restrict the orthogonalization process to a moving window of the most recent iterates. Both of these variences with bounded recurrence relation suffer the loss of finite termination property. Instead of restarting the algorithm, convergence can be improved by preconditioning techniques.

In GMRES, the eigenvalues are approximately given at no extra cost, for spectral analysis, that enables convergence studies.

GMRES requires only matrix-vector products. When GMRES is used to solve the linear system at a Newton step, the matrix of the linear system is Jacobian, which can be approximated by directional differencing. With this consideration, it is possible to develop a matrix free implementation of GMRES, where it is not required to evaluate or store Jacobian.

GMRES is parallelizable depending on the preconditioner used.

3.1.4. Linear Preconditioning (Left Preconditioning)

For Krylov subspace like linear iterative solvers the preconditioning is a requirement for improving their robustness. When the Krylov subspaces used as iterative methods, their advantage is that they can converge to the solution in m iterations. ($m << N$, where m is the Krylov subspace dimension and N is the size of the Jacobian). To keep the number of iterations low, the linear system can be transformed to one which has a more favorable eigenvalue spectrum. Numerical experience and existing theory indicates that the convergence depends on the distribution and clustering of the eigenvalues.

In order to accelerate the convergence of the linear solver, an effective precon-

ditioner must be employed. Incomplete LU preconditioner is used in the numerical solution of the linear problems. For a linear system, $J\delta u = -F(u)$, the left preconditioning can be shown as,

$$M^{-1}J\delta u = M^{-1}(-F(u)) \quad \text{where} \quad J_{i,j} = \frac{\partial F_i(u)}{\partial u_j}$$

The multiplication of the system by the preconditioning matrix does not change the solution, but improves the condition number of the system matrix. Preconditioning is inevitable for the successful convergence of Krylov subspace like iterative methods.

If M^{-1} is close to J, $M^{-1}F(u)$ yields the sought solution vector δu.

In Krylov algorithms, the Jacobian is required only in the form of matrix-vector products and the 'matrix-free' implementation is possible. However, the Jacobian is needed periodically during the outer Newton iteration to generate an effective preconditioner for the inner Krylov iteration.

Luo *et al.* [1] reports in their work dated 1998 that the condition number of the system matrix plays a major role in the convergence rate of the iterative solver. The iterative solvers perform better with the linear systems, for which the eigenvalues of the system matrix are clustered. The method, called the preconditioning, attempts to change the linear system to another one, which has the same solution, but with eigenvalues clustered at a single value. Preconditioning, (i) accelerates the convergence of the linear system, which means, the decrease in computation time and effort; (ii) decreases the number of time steps to reach a steady state solution.

Solving the system of $Ax = b$; Left preconditioning involves premultiplying the linear system with a matrix as $M^{-1}Ax = M^{-1}b$ where M^{-1} is preconditioning matrix. The best preconditioning matrix for A would cluster as many eigenvalues as possible at unity. Obviously, the optimal choice of M^{-1} is A, in which case the underlying matrix problem for GMRES is trivially solved with one Krylov subspace vector. Precondition-

ing will be cost effective only if the additional computational work incurred for each subiteration is compansated for by a reduction in the total number of iterations to converge. In this way, the total cost of solving the overall nonlinear system is reduced. [1]

ILU (Incomplete LU) preconditioner is employed for better spectral properties of the system matrix.

ILU creates non zero elements during factorization, which causes fill in in the sparse matrix. In order to eliminate the fill in but compansate for the discarded elements, Modified ILU (MILU) type precoditioner must be used, which is based on adding up the elements, which are dropped out, and substracting them from the main diagonal in U. It is not guaranteed that MILU performs better than ILU. In some problems MILU is better, in others ILU may surpass MILU in convergence performance.

In order to reduce the memory requirements for a large Jacobian storage, GMRES can be preconditioned by a smaller Jacobian, which comes from the same governing equations but a smaller size domain. Some details about Coarse Grid Preconditioning of GMRES are given in Appendix A.

3.1.5. Numerical Jacobian Evaluation

In Newton Krylov methods the Jacobian matrix only is only needed for the preconditioner. The elements of Jacobian matrix can be approximated by numerically evaluated (Fréchet) derivatives. If ε is the disturbance to calculate the derivatives. The Jacobian is formed by

$$J_{i,j} = \frac{\partial F_i}{\partial u_j} = \frac{F_i(u_1, ..., u_j + \varepsilon_{DISTURB}, ..., u_N) - F_i(u_1, ..., u_j, ..., u_N)}{\varepsilon_{DISTURB}} \tag{3.4}$$

where $i = 1, ..., N$ and $j \in \kappa_i$. The set κ_i consists of the all nodes used in the finite difference approximation of the derivatives in function $F(u)$.

(Note: The numerically approximating the elements of Jacobian by numerically evaluated (Fréchet) derivatives is different than the matrix-free formation of Jacobian times a vector. In the previous one, Jacobian matrix is formed numerically. In the latter one, for the multiplication of the Jv, the matrix is not formed at all.)

If κ_i had contained all $j = 1, ..., N$, then the numerical calculation of Jacobian would have required $N \times N + N$ function evaluations. For a 2-D PDE, which is discretized by using 5 point stencil finite differences, each $J_{i,j}$ calculation in Equation 3.4 requires u_j itself and its North, South, East, and West neighboring nodes.

It is possible to choose such u_j's which are not connected and these can be disturbed accordingly, by $u_j + \varepsilon$. So with just one function evaluation, it is possible to calculate many elements of Jacobian matrix.

It is possible to create Jacobian matrix in 1+5 function evaluations where the derivatives are approximated by 5 point stencil finite differences for a 2-D problem. In 5 point stencil the calculation of derivatives at a certain node is only related to nodes. In the first function evaluation no variables are disturbed and $F_i(u_j)$ is calculated.

In the following 5 steps, the variables are disturbed one by one in the center and at N, S, W, and E of the cross shapes at each function evaluation. At the end of 1+5 steps, all the derivatives are calculated at all nodal points and Jacobian is constructed.

3.1.6. Directional Differentiation

In the solution of a nonlinear system with Newton's method, a linear system must be solved at each Newton step. The linear system matrix is the Jacobian matrix as given in Equation 3.3. Jacobian matrix does not have to be stored for matrix-vector product. Jacobian in Newton's method can be approximated by directional differencing

at each time Jv product is necessary, where v is the guess to the solution at any time. Such a matrix-free solver saves incredible computation time and memory.

This property can be used in Krylov subspace methods, where the Jacobian is only used in matrix-vector product form. This can be seen in GMRES algorithm Figure 3.1, where the linear system matrix is only required at Step 01 ($r_0 = b - Ax_0$) and at Step 04 ($w_j = Av_j$).

With above considerations, it is possible to manipulate the properties of Newton's method and Krylov solvers and solve the linear system without storing or evaluating the Jacobian matrix but approximating the matrix-vector product by finite differences as follows,

$$Jv \approx \frac{F(u + \varepsilon v) - F(u)}{\varepsilon}. \tag{3.5}$$

This is called the 'matrix-free' implementation.

3.2. Nonlinear Multigrid

In this study, nonlinear multigrid solution methods (FAS) are investigated for the solution of transonic flows and the results are compared with the results obtained from NK method. Nonlinear multigrid techniques help to improve convergence and enhance the efficiency of the algorithm, because they are extremely fast to eliminate high frequency errors.

One of the papers, which may be attributed as the origin of the multigrid, is the work of Brandt [5]. Two other excellent references to multigrid are the work of Hackbusch [6] and Wesseling [7].

While multigrid can be used to solve linear problems, different versions of multigrid can be a solution tool for nonlinear problems such as Full Approximation Storage (FAS). In addition to FAS, Full Multigrid scheme (FMG) can be used to provide a better initial guess by starting multigrid solutions from coarser meshes. Some detailed information and algorithms of FAS and FMG are given below.

It is possible to use multigrid in conjunction with Newton's method for nonlinear problems such as Krylov accelerated multigrid method (KMG) by Oosterlee and Washio [9]. When Newton's method is applied to a nonlinear problem at each Newton step the solution of a linear system is required in order to calculate the updates. Multigrid can be used for the arising linear system. On the other hand multigrid can be applied to nonlinear problems by using Full Approximation Storage (FAS) algorithm.

3.2.1. Full Approximation Storage (FAS)

The Full Approximation Storage (FAS) is chosen to solve the nonlinear systems of equations emerged from finite volume discretization of transonic flow equations. The data obtained from numerical experimentation is given in Section 5 Results. Multigrid (FAS) algorithm, used in this study, is introduced below.

A nonlinear system can be given as $L(u) = 0$ and it can be said that $L(.)$ is the nonlinear operator. Even though a zero right hand side is sought, during the solution process, a nonzero right hand side is generally the case, where $L(u) = f$, f being the right hand side vector. The nonlinear residual equation can be expressed as follows;

$$L(u) - L(v) = f - L(v), \tag{3.6}$$

$$L(u) - L(v) = r, \tag{3.7}$$

where residual $r = f - L(v)$. When e is used to denote the error vector, the nonlinear residual equation can be written as follows:

$$L(v + e) - L(v) = r. \tag{3.8}$$

The residual equation in coarse grid is as follows:

$$L_H(v_H + e_H) - L_H(v_H) = r_H. \tag{3.9}$$

In multigrid methods, the coarse grids provide faster convergence for the smooth errors on fine grid results. The correction terms provided by fine grid solutions makes it possible to achieve fine grid accuracy on coarse grid.

In the explanations for FAS, following subscripts are used: Subscript h indicates finer grid and subscript H indicates coarser grid, ie $H = 2h$.

Restriction Operator R (or I_h^H), also called "fine-to-coarse operator" can be shown as,

$$d_H = Rd_h. \tag{3.10}$$

Prolongation or Interpolation Operator P (or I_H^h), also called "coarse-to-fine operator" can be shown as,

$$v_h = Pv_H. \tag{3.11}$$

FAS algorithm applied to finite volume discretized Euler equations is given in Figure 3.2. $\tilde{u}_h$ is the approximate solution at any time of the solution process. Step by step explanation of the algorithm is as follows and below explanation of FAS algorithm can be followed from Figure 3.2.

FAS starts the downward cycle with pre-smoothing. In implicit ADI solver (pre-smoothing step), approximate solutions for f_h and $\tilde{u}_h$ are obtained. f_h and $\tilde{u}_h$ values obtained from this MG Level are stored to be used in the next level of coarse grid solution.

After presmoothing step, $\tilde{u}_h$ is obtained. Nonlinear operator is applied to $\tilde{u}_h$ and the result is restricted, such that $RL_h(\tilde{u}_h)$ is obtained.

Then $\tilde{u}_h$ is restricted to obtain $R\tilde{u}_h$. Next thing is first to restrict $\tilde{u}_h$ and to apply nonlinear operator to restricted $\tilde{u}_h$ to get $L_H(R\tilde{u}_h)$.

At this stage, τ_H can be formed as follows:

$$\tau_H = L_H(R\tilde{u}_h) - RL_h(\tilde{u}_h). \tag{3.12}$$

The right hand side vector, f_h, from the presmoothing step, is restricted to form Rf_h. The coarse right hand side (RHS) to be used in next coarse grid solution is calculated.

$$f_H = Rf_h + \tau_H. \tag{3.13}$$

At the coarsest level of MG cycles, implicit ADI solver iteration provides new approximations to f_H and $\tilde{u}_H$.

After the relaxation on the coarsest level, it is time for prolongation, which means to go up to next finer grid level. With the start of upward stroke of V cycle, the correction is calculated at the coarser grid level.

First $\tilde{u}_h$ is restricted to get $R\tilde{u}_h$. (Note: $\tilde{u}_h$ here does not come from the previous relaxation but from the corresponding downward cycle.) But $\tilde{u}_H$ comes from the previous relaxation level. Now it is time to calculate "tau correction (τ_h)" by subtraction and prolongation of $R\tilde{u}_h$ and $\tilde{u}_H$ as follows;

$$\tau_h = P(\tilde{u}_H - R\tilde{u}_h). \tag{3.14}$$

The new approximation for the solution, $\tilde{u}_h^{new}$, can be calculated by the following

formula.

$$\tilde{u}_h^{new} = \tilde{u}_h + \tau_h, \tag{3.15}$$

or similarly.

$$\tilde{u}_h^{new} = \tilde{u}_h + P(\tilde{u}_H - R\tilde{u}_h). \tag{3.16}$$

Next stage is to relax on this level, by postsmoothing application. In post-smoothing step, u_h is obtained by using $\tilde{u}_h^{new}$ as initial guess implementing implicit ADI solver.

3.2.2. Full Multigrid (FMG)

In nonlinear problem solution, it is an excellent advantage to start with a good initial guess for an iterative solution procedures. Multigrid methods may provide a cheap and good initial guess by using the solutions at coarser level, which is already a part of the solution process.

In FMG approach, the idea is to provide a better initial guess to the finest level computations by solving the problem successively starting from the coarsest grid. The initial calculations are performed on the coarsest grid and the solution is fed to the next grid level as an initial guess until the finest grid level is reached.

Full Multigrid (FMG) Algorithm for solving $A_L(u_L) = f_L$ is in Figure 3.3 and FMG Cycle can be seen in Figure 3.4.

1: **for** **do** $= 1, 2, ...$ ▷ Nested iteration loop
2: **for** **do** $= 1, 2, ..., maxcyc$ ▷ V-cycle loop
3: **for** **do** $= Vmax, Vmax - 1, ..., 2$ ▷ Downward stoke of V cycle
4: **for** **do** $= 1, 2, ..., npre$ ▷ Pre-smoothing
5: Solve $L_h(u_h) = f_h$ for $\tilde{u}_h$
6: **end for**
7: Determine RHS: $L_h(\tilde{u}_h)$
8: Restriction: $RL_h(\tilde{u}_h)$
9: Restriction: $R\tilde{u}_h$
10: Determine RHS: $L_H(R\tilde{u}_h)$
11: Matrix Subtraction: $\tau_H = L_H(R\tilde{u}_h) - RL_h(\tilde{u}_h)$
12: Restriction: Rf_h
13: Matrix Addition: $f_H = Rf_h + \tau_H$
14: **end for**
15: Solve $L_H(u_H) = f_H$ for $\tilde{u}_H$ ▷ Solve the coarsest grid
16: **for** **do** $= 2, 3, ..., Vmax$ ▷ Upward stoke of V cycle
17: Restriction: $R\tilde{u}_h$
18: Matrix Subtraction: $\tilde{u}_H - R\tilde{u}_h$
19: Interpolation: $\tau_h = P(\tilde{u}_H - R\tilde{u}_h)$
20: Matrix Addition: $\tilde{u}_h^{new} = \tilde{u}_h + \tau_h$
21: **for** **do** $= 1, 2, ..., npost$ ▷ Post-smoothing
22: Solve $L_h(\tilde{u}_h^{new}) = f_h$ for u_h
23: **end for**
24: **end for**
25: **end for**
26: **end for**

Figure 3.2. Full Approximation Storage (FAS) Algorithm; nested iteration, downward and upward V cycles, pre-smoothing, post-smoothing and the coarsest level solutions given with restriction and interpolation functions.

3.3. Description of the Codes

In this study, for FMG-FAS, Morrison's ISAAC code [34] and for NK Praveen's *flo2d* code [33] is used.

Morrison's ISAAC [34] is a CFD code, which solves inviscid/viscous flows, uses structured grids, ADI/AF implicit solution methods and nonlinear multigrid. In this study, ISAAC is used for comparison of the solution of transonic flow around airfoils.

Morrison [34] reports on ISAAC which is developed as a compressible flow solver. In ISAAC, the convective terms are discretized with an upwind scheme, which is Roe's approximate Riemann solver coupled with a MUSCL scheme. In order to solve the discretized governing equations, (cell-centered finite-volume approach is applied), Morrison [34] uses a spatially split, approximate factorization (AF) scheme in order to achieve steady state solution with time integrated conservation and transport equations and nonlinear multigrid algorithm to speed up convergence. The following are some papers/reports on the studies which are performed partly/fully by using ISAAC.

Abid *et al.* [35] used ISAAC for the analysis of turbulent compressible flows. They compared the experimental data with the data achieved from numerical experimentation for turbulence models.

1: Set (compute) u_l on the working level l.
2: **if** $l < L$ **then** then FMG interpolate to the next finer working level.
3: $u_{l+1} = I_l^{l+1} u_l$
4: Apply FAS (or MG) scheme to $A_{l+1}(u_{l+1}) = f_{l+1}$ starting with u_{l+1}
5: **if** $l + 1 < L$ **then**
6: set $l := l + 1$
7: go to 2
8: **end if**
9: **end if**

Figure 3.3. Full Multigrid (FMG) Algorithm for solving $A_L(u_L) = f_L$.

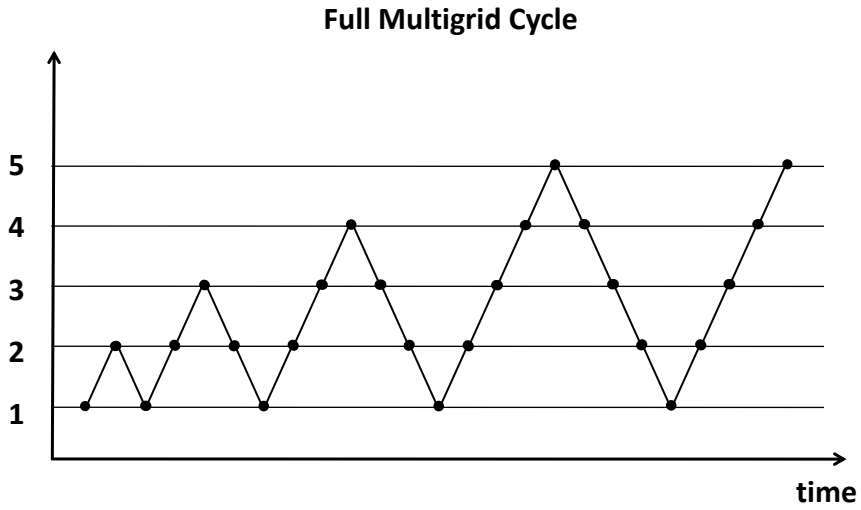

Figure 3.4. Full Multigrid (FMG) Cycle.

Campobasso *et al.* [36] report on the research on CFD modeling of wind turbine computational aerodynamics. The aerodynamics of the airfoil of wind turbine is investigated by comparing the data from experiments and numerical analysis. The flow around the airfoil is studied by using ISAAC flow solver.

Praveen's *flo2d* [33] is a 2 dimensional flow solver for inviscid and viscous fluids. *flo2d* solves the compressible flow equations on unstructured triangular grids using a vertex-centroid finite volume scheme. *flo2d* uses the following methods; finite volume (cell-centered scheme), triangular grids, second order scheme (MUSCL-type), Roe's flux difference splitting, explicit Runge Kutta time stepping, implicit LU-SGS, implicit NK (GMRES). The Jacobian is formed by using the automatic differentiation tool TAPENADE. In this work, left (ILU(2)) preconditioned GMRES is used for the solution of linear systems.

The required modifications has been done on the aforementioned codes to come up with a reasonable basis for the comparisons to be conducted. As such, *flo2d* is modified to solve quadrilateral structured grids.

4. AN ADAPTIVE TIME STEPPING STRATEGY FOR THE IMPLICIT SOLUTION OF STEADY TRANSONIC FLOWS

In this part of the thesis, Newton Krylov (NK), with implicit pseudo time stepping method, has been used to solve the compressible inviscid flow (Euler) equations for steady transonic case. As usually done, the free-stream flow values are used as the initial guess. The free-stream values are far from the solution of transonic flow, because the flow field with free stream values does not contain any shocks. Since Newton's method requires an initial guess to be as close to the solution as possible, the free-stream values as the initial guess may lead to divergence. In this study, a novel local time step (Δt_{local} or CFL_{local}) selection method based on the gradients of flow variables is introduced with the use of "Pressure Sensor". The introduced local time step (Δt_{local} or CFL_{local}) selection methodology with the use of "Pressure Sensor" makes it possible to take larger time steps, Δt, in smooth regions and lower Δt in the areas where there are sharp gradients.

4.1. Adaptive Time Stepping

Since I in Equation 2.32 operates only on the diagonal, a smaller Δt creates a diagonally dominant linear system. A smaller Δt leads to a better condition number, which makes it a lot easier to solve the linear system given in Equation 3.2.

The downside of a smaller Δt is that the number of time steps required to reach steady state increases. This means, a larger time step is needed for less iterations and faster convergence to steady state. On the other hand, larger time steps may cause divergence, if the free stream values are used as initial guess at the start-up of Newton's method.

4.1.1. Time Step (CFL) Selection Strategies

In the first part of this section, CFL_{local} (Δt_{local}) selection methodologies for the time dependent and steady state problems are mentioned.

In the second part, the use of "Pressure Sensor" in the selection of local CFL number (local time step) for steady transonic flow calculations in implicit methods like NK by the use of pseudo time stepping is covered.

4.1.1.1. Time dependent problem solution. In the time dependent problem solution, Δt_{global} (CFL_{global}) is used. Δt is determined by the use of CFL_{global} for each cell.

The relationship between $\Delta t_{i,j}$ and CFL_{global} can be seen below.

$$\Delta t_{i,j} = CFL_{global} \frac{\Delta x_{i,j}}{u_{i,j}} \tag{4.1}$$

After the calculation of time step Δt for all cells, the minimum value of time step (Δt_{min}) is assigned as Δt_{global} by using Equation 4.2 and used for all cells.

$$\Delta t_{global} = min(\Delta t_{i,j}). \tag{4.2}$$

4.1.1.2. Steady state flow solution with pseudo time-stepping. For steady state flow solution (Equation 2.32), it is not necessary to use the same time step in the whole domain, which means it is possible to use local time step. This is called pseudo time stepping (or integration).

The use of pseudo time integration scheme in the steady state problem solution and the role of Δt selection for improved system properties by increasing Δt and providing diagonally dominant system is mentioned in Section (4.1).

The relationship between Δt_{local} and CFL_{local} for Steady State Problems is given below (where $\Delta t_{i,j} = \Delta t_{local}$ and $CFL_{i,j} = CFL_{local}$):

$$\Delta t_{local} = CFL_{local} \frac{\Delta x_{i,j}}{u_{i,j}} \tag{4.3}$$

After the calculation of local time step Δt_{local} for all cells, each Δt_{local} is used for the cell it belongs.

In this study we propose a novel Δt_{local} (CFL_{local}) selection method based on the flow properties, i.e. by considering the gradients of flow variables, which can be attributed as adaptive time stepping.

4.1.2. Determination of CFL Number

Implicit time integration methods are assumed to achieve convergence with very high time steps (in theory $\Delta t \to \infty$) (Equation 2.32). In practice, on the other hand, very high values for Δt may lead to divergence.

4.1.2.1. CFL Number (Δt) Strategy Used without "Pressure Sensor". CFL increase is controlled by Equation 4.4. In the first part up to a predetermined CFL value, CFL_{predet}, a constant factor (CFL_{coef1}) and the residual rate, $\frac{Fresidual_{previous}}{Fresidual_{current}}$, are used to increase the CFL number.

After CFL_{predet} is reached, another constant factor, CFL_{coef2} is used to increase

the CFL number.

$$CFL_{new} = \begin{cases} CFL_{coef1} CFL_{previous} \frac{Fresidual_{previous}}{Fresidual_{current}} & if \quad CFL < CFL_{predet} \\ CFL_{coef2} CFL_{previous} & if \quad CFL \geq CFL_{predet} \end{cases} \tag{4.4}$$

Different values of CFL_{predet}, CFL_{coef1} and CFL_{coef2} are tested.

After each CFL increase (i.e. Δt increase) a certain number of nonlinear solution steps (implicit time steps) are conducted. Each time CFL number has been changed, a number of iterations has been conducted before the next update of CFL number. During these iterations, it is possible to encounter some increase in the residual.

4.1.2.2. Determination of Δt_{local} (CFL_{local}) by using "Pressure Sensor". The use of the pressure value to construct a sensor for determining higher gradients is possible as in Jameson [15]. "Pressure Sensors" (in i- and j-directions) are given as follows respectively:

$$dpi_{i,j} = \frac{|p_{i+1,j} - 2p_{i,j} + p_{i-1,j}|}{|p_{i+1,j}| + 2|p_{i,j}| + |p_{i-1,j}|} \quad dpj_{i,j} = \frac{|p_{i,j+1} - 2p_{i,j} + p_{i,j-1}|}{|p_{i,j+1}| + 2|p_{i,j}| + |p_{i,j-1}|} \tag{4.5}$$

"Density Sensor" ($drho_i$ or $drho_j$) can be calculated with a similar method, which uses density value instead of pressure.

Interpolation equations for CFL_{local} (Δt_{local}) by using "Pressure Sensor" is below.

$$CFL_{i,j} = CFL_{min} + \left[1 - \frac{(dpi_{i,j} - dpi_{min})}{(dpi_{max} - dpi_{min})}\right](CFL_{max} - CFL_{min}) \tag{4.6}$$

where dpi is the "Pressure Sensor" in i-direction.

$$CFL_{i,j} = CFL_{min} + \left[1 - \frac{(dpj_{i,j} - dpj_{min})}{(dpj_{max} - dpj_{min})}\right](CFL_{max} - CFL_{min}) \tag{4.7}$$

where dpj is the "Pressure Sensor" in j-direction.

The idea is to use relatively lower CFL number (Choose CFL_{local} such that it is close to the predetermined CFL_{min}) in the areas with steep gradients like shocks) but use larger CFL number (Choose CFL_{local} such that it is close to the predetermined CFL_{max}) in the smooth regions.

In the determination of CFL_{min} and CFL_{max}, first it is required to determine a CFL_{min}, which ensures the convergence for the problem depending on its difficulty level which depends on M_∞ and α. At start-up; CFL_{ini} is set to 10.0 and CFL_{min} is set to CFL_{ini}. CFL_{Rate} is defined with the following formula:

$$CFL_{Rate} = \frac{CFL_{max}}{CFL_{min}} \tag{4.8}$$

For a given CFL_{Rate} and CFL_{min}, CFL_{max} is calculated by using Equation 4.8.

Each time step has been repeated for a fixed number of times (6 times here).

At the end of each time step, the next CFL_{min} is calculated by using Equation 4.4 and CFL_{max} is calculated by using CFL_{min} and current value of CFL_{Rate} by using Equation 4.8.

The aim is to come up with a CFL selection methodology to sustain acceleration of convergence for implicit NK methods without leading to divergence at start-up.

5. RESULTS

5.1. Results for the Comparison of NK and FMG-FAS Methods

In this work, transonic inviscid flow calculations are performed for a flow around NACA0012 (symmetric) and RAE2822 (unsymmetric) airfoils. Test cases demonstrate that both methods tested are capable of effectively solving transonic flow regime. Various grids are used in calculations to show the behavior of the results with changing grid properties.

NACA 0012 and RAE2822 are used extensively for experiments in wind tunnels as well as the numerical experiments. The studies on transonic flow around these airfoils are well documented and the documented results can be used for the validation and test of numerical methods.

The shocks in the transonic flow constitute a challenge for the tested numerical methods. Our results show that the numerical methods have good features to simulate shocks accurately.

NK and FMG-FAS methods for the solution of transonic flow are tested by using Grid01 (in Figure 5.1), Grid02 (in Figure 5.2) and Grid03 (in Figure 5.3). The details of these structured C-grids around airfoils are given in Table 5.1.

For all tests; T_∞ is 273.15 K, p_∞ is 101325 N/m^2, $\gamma = 1.4$, specific gas constant is 287.87 J/(kg.K). In test problems with NACA0012, Mach number at the free stream, M_∞, is 0.80 and angle of attack , α, is 1.25^o. For RAE2822, $M_\infty = 0.729$ and $\alpha = 2.31^o$. The flow field is set to free stream conditions in the beginning of computations for all numerical experiments. The stopping criteria is set to less than $1x10^{-12}$.

The convergence histories are presented in two different types of plots; first of which shows L_2 norm vs iteration count and the other one shows L_2 norm vs compu-

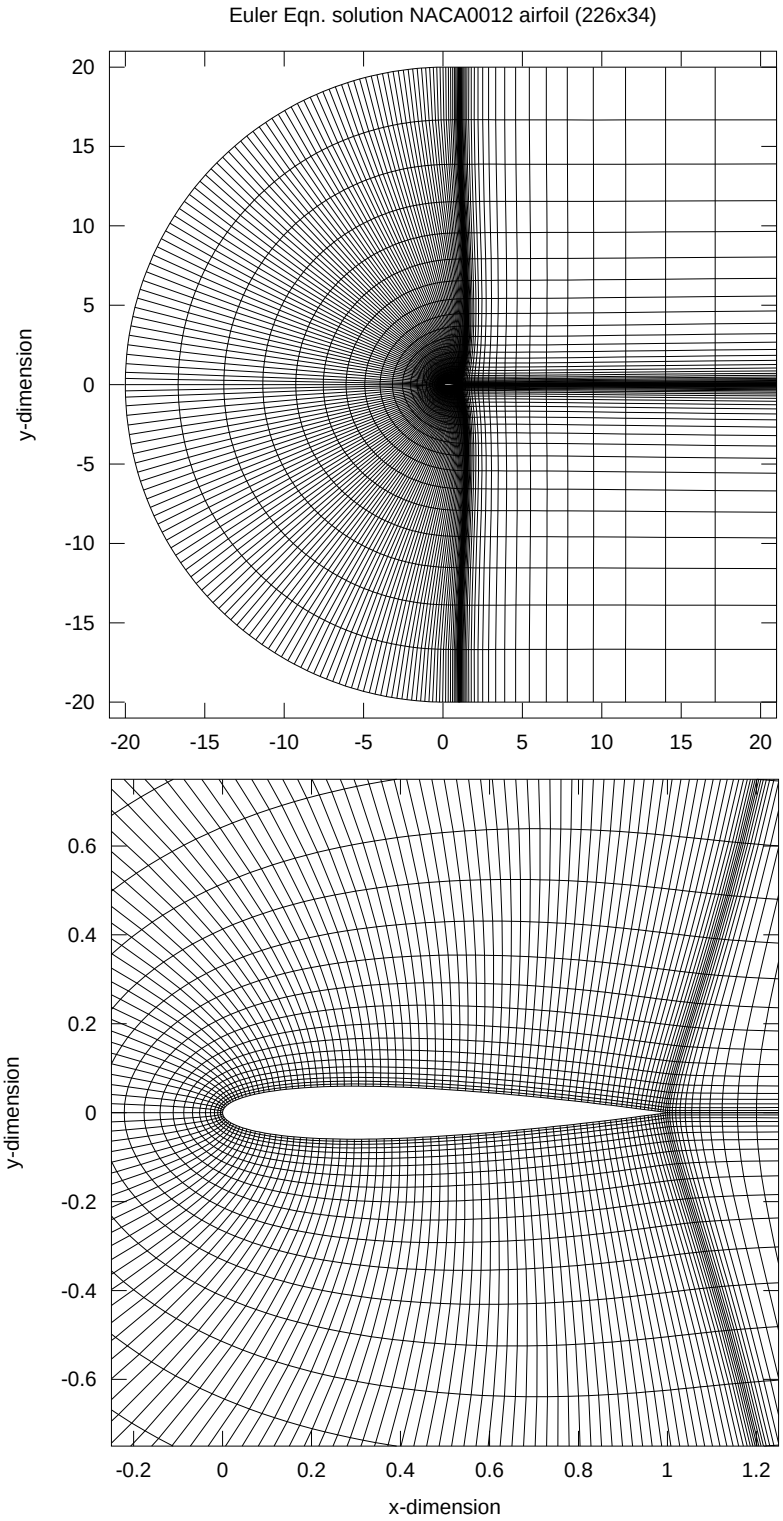

Figure 5.1. The Grid01 (225x33, C-grid) used in NK and FMG-FAS analysis of transonic flow around NACA0012 airfoil. For FMG-FAS analysis, the multigrid levels are given in Table 5.1.

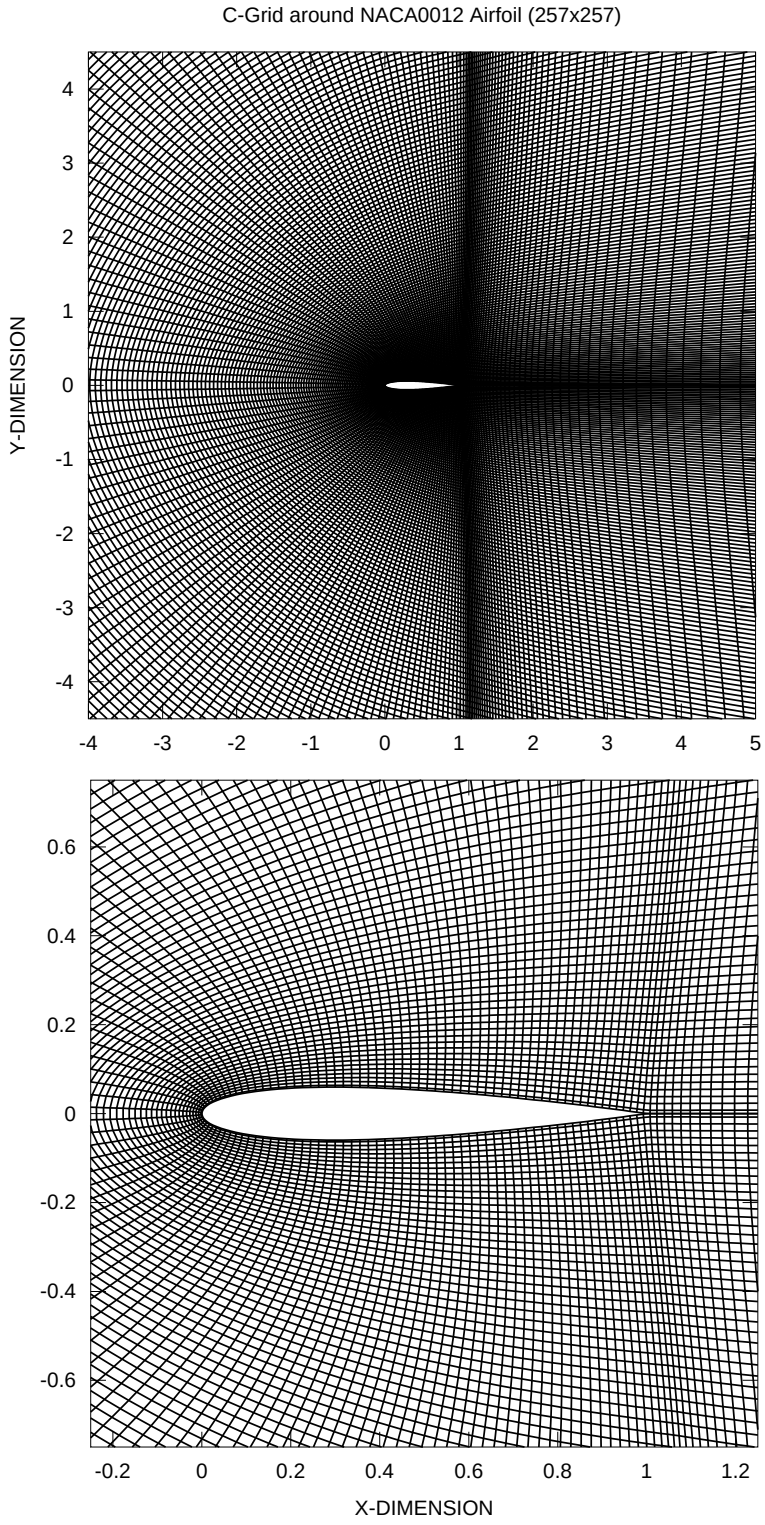

Figure 5.2. The Grid02 (257x257, C-grid) used in NK and FMG-FAS analysis of transonic flow around NACA0012 airfoil. For FMG-FAS analysis, the multigrid levels are given in Table 5.1.

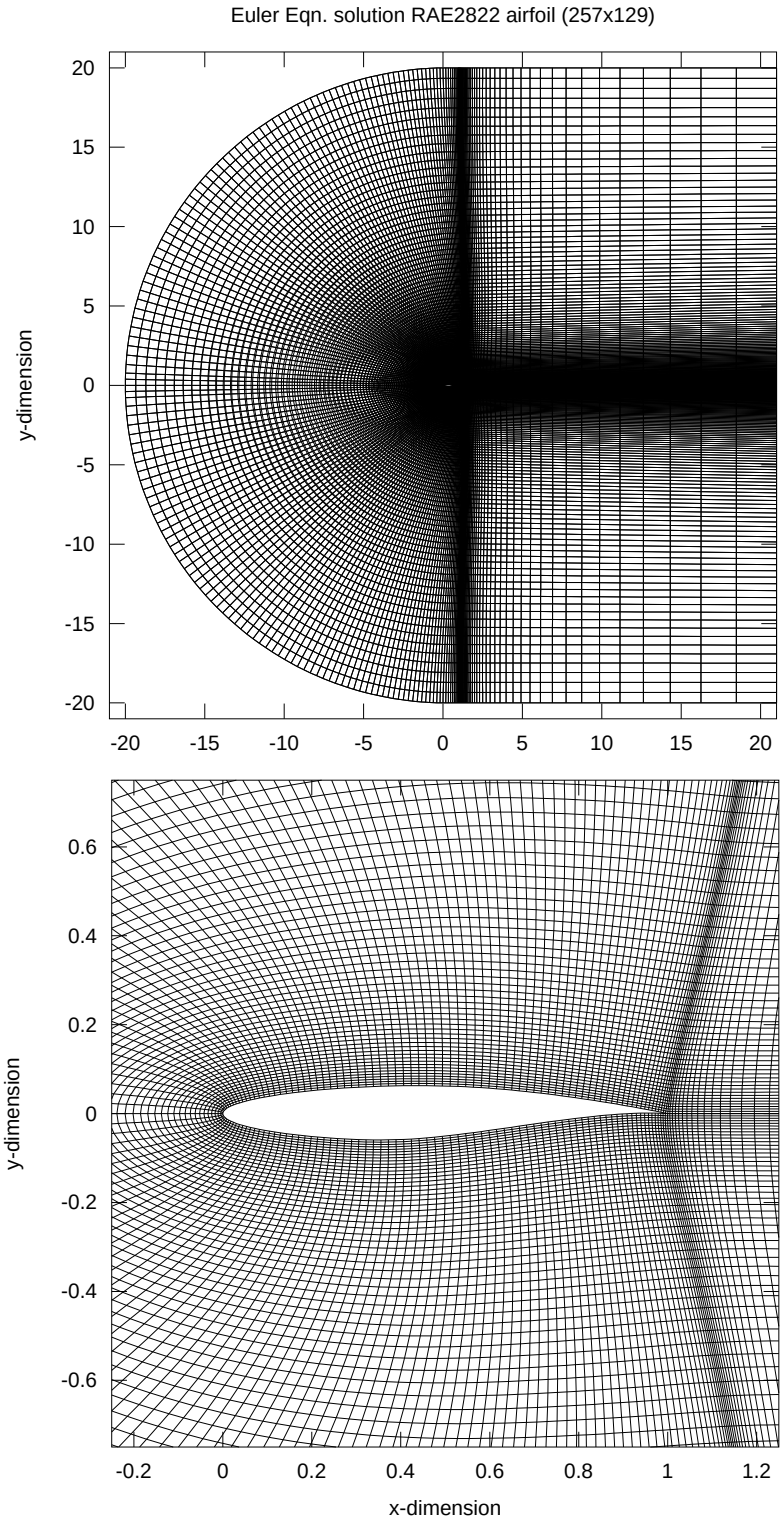

Figure 5.3. The Grid03 (257x129, C-grid) used in NK and FMG-FAS analysis of transonic flow around RAE2822 airfoil. For FMG-FAS analysis, the multigrid levels are given in Table 5.1.

Table 5.1. The structural details of Grid01 (225x33), Grid02 (257x257) and Grid03 (257x129) used in NK (only the finest grid) and FMG-FAS analysis of transonic flow. Total number of grid nodes are given in parentheses.

	Grid01	Grid02	Grid03
Airfoil	NACA 0012	NACA 0012	RAE 2822
The finest grid MG Level 1	224x32 (7168)	256x256 (65536)	256x128 (32768)
MG Level 2	112x16 (1792)	128x128 (16384)	128x64 (8192)
MG Level 3	56x8 (224)	64x64 (4096)	64x32 (2048)
MG Level 4	28x4 (112)	32x32 (1024)	32x16 (512)

tation time. Numerical tests are conducted in single processor mode not parallel.

The computer and the operating system used in these work are as follows: Intel(R) Core(TM)2 Quad CPU Q8300 @ 2.50GHz with 3.9 GiB memory.

The compiler used in this study is "gfortran", version: GNU Fortran (Ubuntu / Linaro 4.5.2-8ubuntu4) 4.5.2.

5.1.1. Results of Numerical Experiments for NK

In the numerical experiments on Newton Krylov method with preconditioning, three different grid structures are used. The summary of the numerical experiments with NK and their results can be seen in Table 5.2 and Figure 5.4, 5.5 and 5.6.

The flow field is set to free stream conditions in all numerical experiments and the starting CFL number is set to 10.0. After the startup of NK, with the development of flow, CFL number is ramped up. The CFL increase w.r.t. the number of iterations

and computation time can be seen in Figure 5.4, 5.5 and 5.6.

Figure 5.4 shows 4 different comparison plots for NK run with Grid01 (225x33). In these test cases, where the initial guess for the solution is set to free stream values, it is obvious that the startup takes some number of iterations and some time, before the residual starts decreasing with a considerable rate. Sometimes after the CFL number was increased, an increase in residual has been observed followed by a limited number of iterations. Later in the computation process, the required decrease in residual towards convergence is sustained. The process of CFL number increase, which is followed by some number of iterations, before next CFL number increase has to be kept under control. Otherwise, uncontrolled increase in CFL number leads to larger time steps and divergence may occur. On the other hand, if the rate of increase in CFL number is restrained, then the computation time becomes unnecessarily longer than it should be.

Figure 5.5 presents the results of NK solution by using Grid02 (257x257). The comparison of iteration numbers in Figure 5.4 and Figure 5.5 shows that the iteration numbers is in the same order of magnitude, 65 and 90 respectively. On the other hand, the comparison of computation time show that increasing the problem size (from 225x33 to 257x257) increases the computation time drastically (from 42.63 seconds for Grid01 (225x33) to 560.74 seconds for Grid02 (257x257). Other than the computation time, solution process for Grid02 (257x257) behaves in a similar manner as Grid01 (225x33).

In NK runs, for Grid01 (225x33) it took 65 iterations to converge, on the other hand, for Grid02 (257x257) the convergence is reached after 90 iterations. The sizes of the grids are the major contributors to the computation time. It took 42.63 seconds for the code to reach convergence for Grid01 (225x33) and 560.74 seconds for Grid02 (257x257). For NK run with Grid03 (RAE2822 257x129), the convergence is achieved in 74 iterations and 228.43 sec.

The comparison of iteration numbers in Figure 5.4 and 5.5 shows that the iteration numbers is in the same order of magnitude, 65 and 90 respectively. On the other hand,

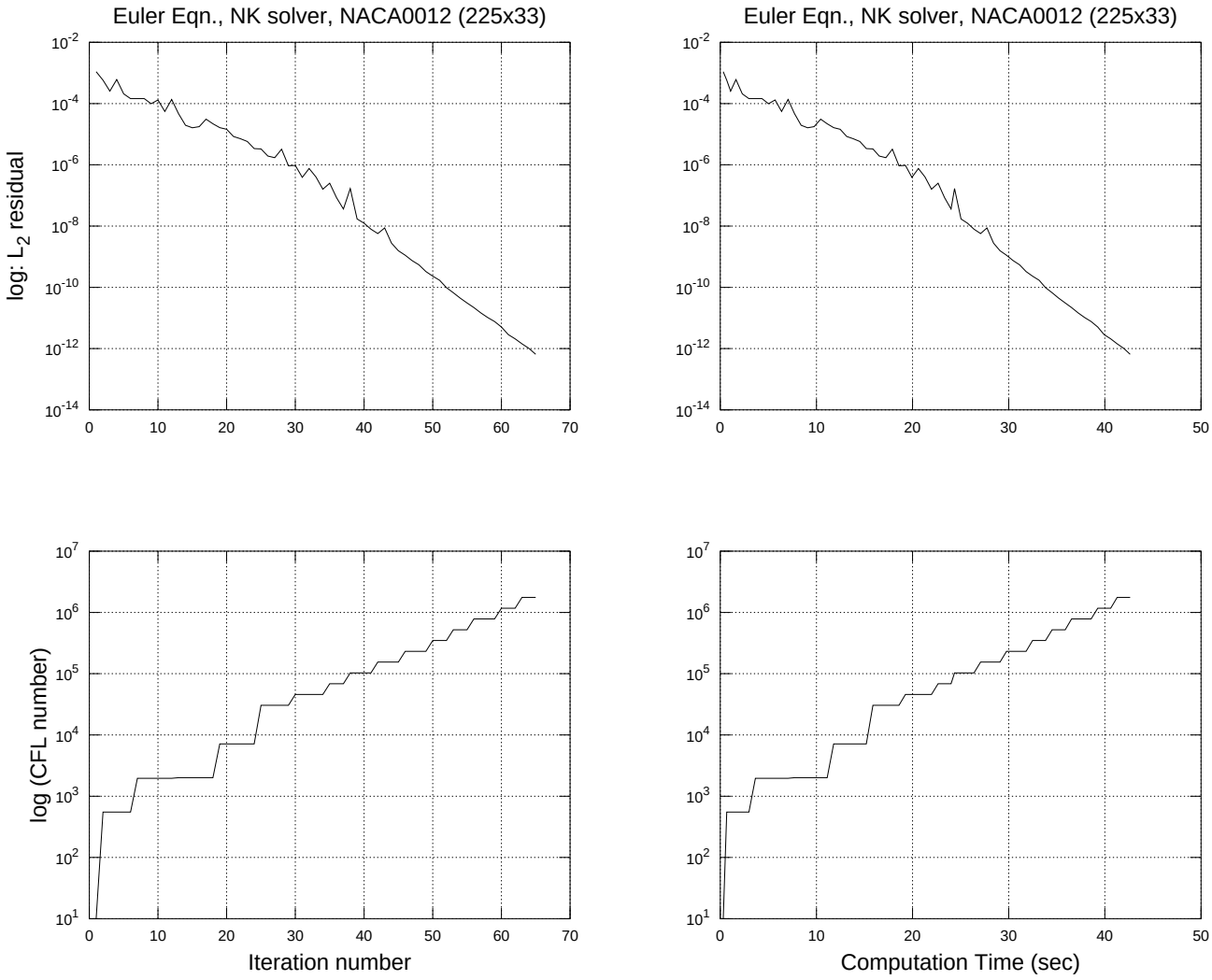

Figure 5.4. The convergence histories of (L_2 norm vs iteration count and L_2 norm vs computation time) Newton-Krylov analysis of C-grid around NACA 0012 airfoil. Grid01 (225x33). The grid used is in Figure 5.1. T_∞=273.15 K, p_∞=101325 N/m^2, $\gamma = 1.4$, M_∞=0.80, $\alpha = 1.25^o$.

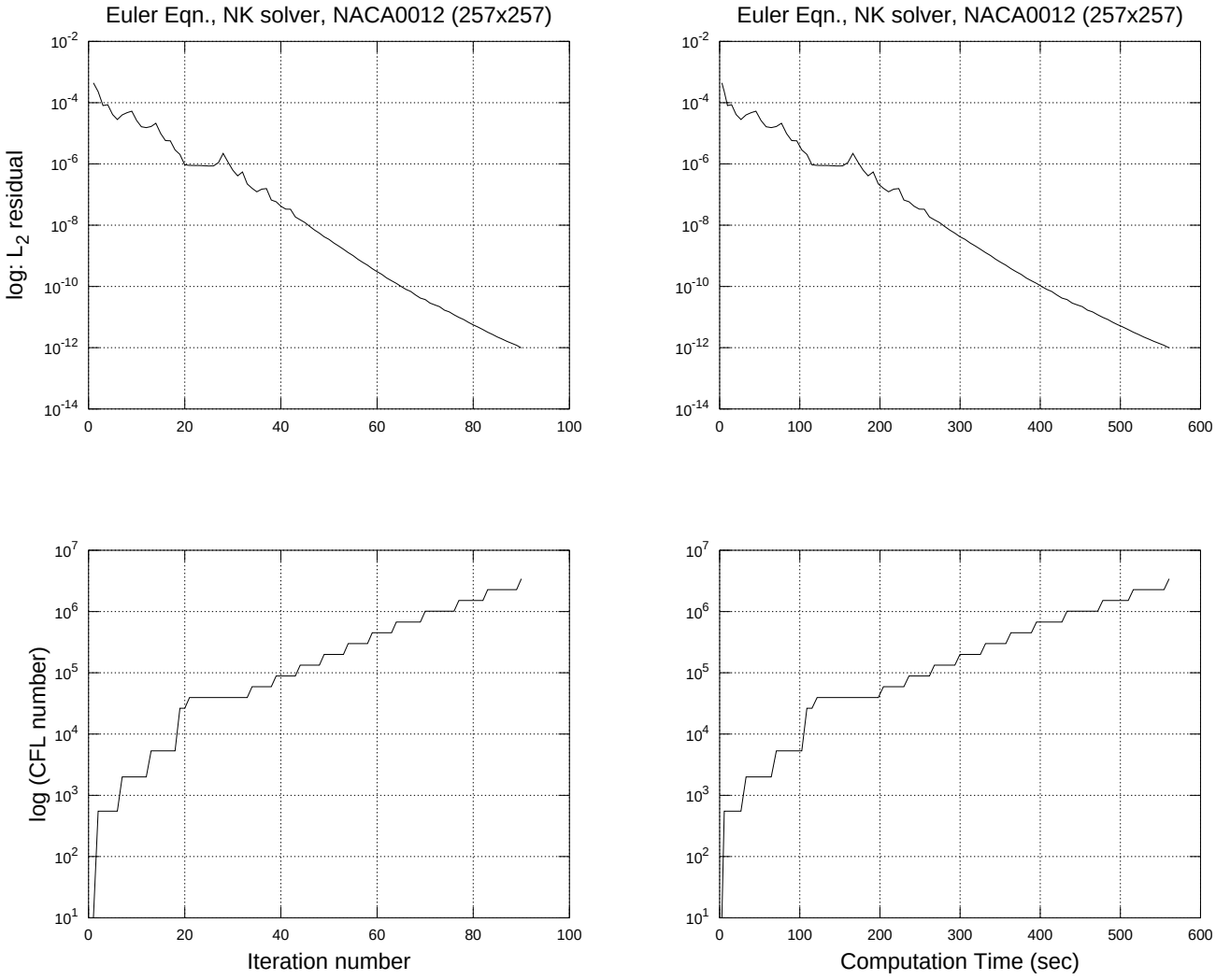

Figure 5.5. The convergence histories of (L_2 norm vs iteration count and L_2 norm vs computation time) and CFL number increase for Newton-Krylov analysis of C-grid around NACA 0012 airfoil; Grid02 (257x257) in Figure 5.2. T_∞=273.15 K, p_∞=101325 N/m^2, $\gamma = 1.4$, M_∞=0.80, $\alpha = 1.25^o$.

the comparison of computation time shows that increasing the problem size (from 225x33 to 257x257) increases the computation time drastically (from 42.63 seconds for Grid01 (225x33) to 560.74 seconds for Grid02 (257x257)).

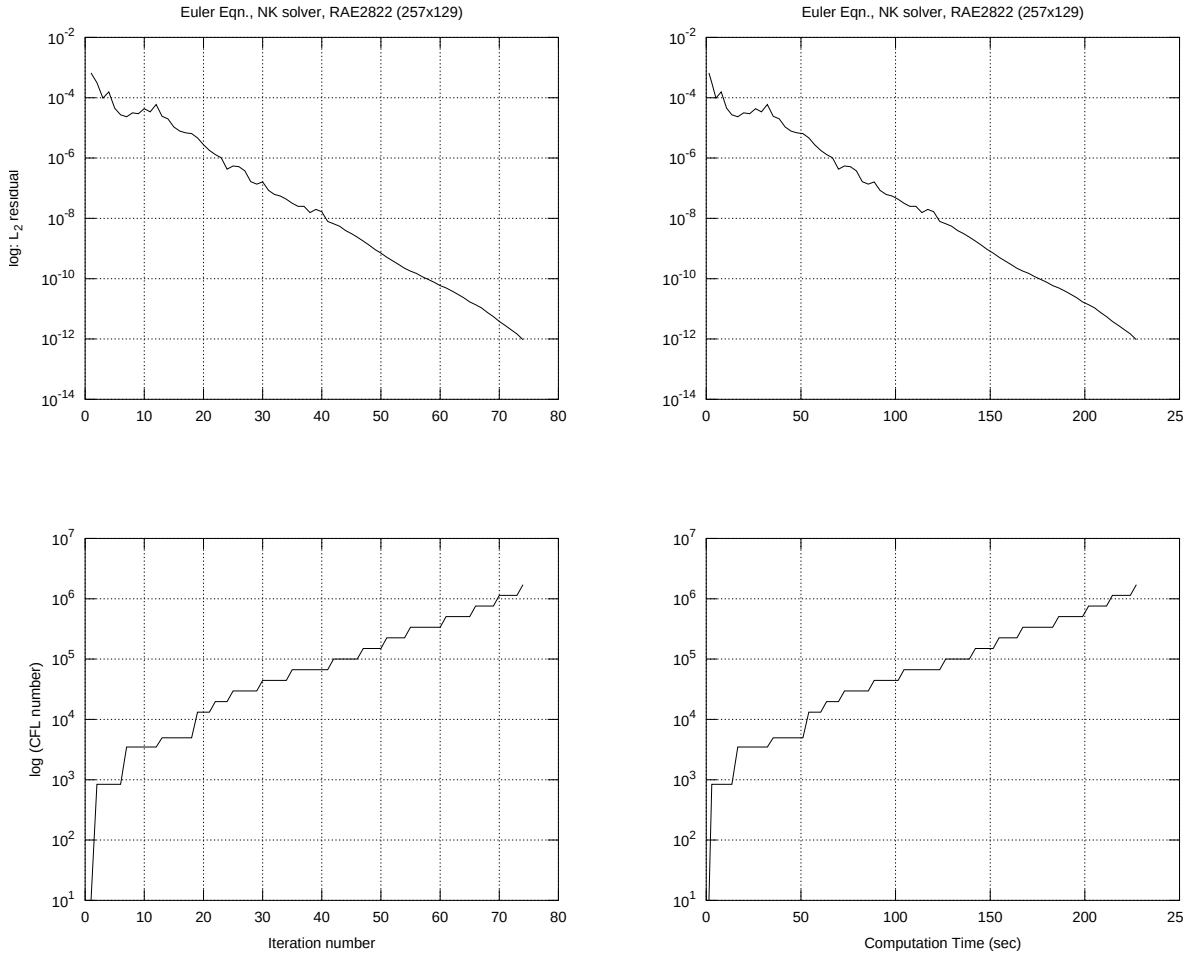

Figure 5.6. The convergence history of and CFL number increase for Newton-Krylov analysis of transonic flow around RAE2822 airfoil (L_2 norm vs iteration count and L_2 norm vs computation time). Grid03 (257x129) is given in Figure 5.3. T_∞=273.15 K, p_∞=101325 N/m^2, $\gamma = 1.4$, M_∞=0.729, $\alpha = 2.31^o$.

5.1.2. Results of Numerical Experiments for FMG-FAS

The convergence histories of are presented in L_2 norm vs iteration count and L_2 norm vs computation time plots are given in Figures 5.7, 5.8 and 5.9 for Grid01 (225x33) and in Figures 5.10, 5.11 and 5.12 for Grid02 (257x257) and in Figure 5.13 for Grid03 (257x129).

In the numerical experiments 4, 3 and 2 multigrid levels are tested. The finest grid is kept as the highest level but the intermediate and coarse MG levels are changed. The runs with four multigrid levels have the shortest computation time among other multigrid runs. Elimination of coarser multigrid levels increases the computation time.

Regarding the numerical experiments with 4 multigrid levels and 225x33 cells, the convergence history is given in Figure 5.7. Another run with 3 multigrid levels is also conducted. In order to solve this problem by using FMG-FAS with 3 multigrid levels MG Level 4 in Table 5.1 is removed. In the last attempt for this grid structure only the finest level and one coarse level is used. The convergence histories are given in the figures and the computation time is given in Table 5.3.

Following FMG-FAS runs are conducted with Grid02 (257x257). The details of Grid02 (257x257) are given in the Table 5.1. The results and comparison of FMG-FAS solutions to transonic flow using Grid02 (257x257) is given in Table 5.4. 4-level multigrid solution provides the best performance compared to the other runs with 2 and 3 levels of multigrid.

The results of transonic flow solutions by using FMG-FAS showed that C_l and C_d values obtained from the conducted analysis comply with Vassberg and Jameson [42], which validates the algorithms used in this work.

5.2. Comparison of Results from NK and FMG-FAS

The results achieved from the numerical experiments of NK and FMG-FAS are summarized in Table 5.5. The results obtained from the numerical experiments with NK and FMG-FAS around RAE2822 airfoil (Grid03-257x129) show that convergence is reached in 228.43 sec for NK and 64,65 sec for FMG-FAS. Similar behavior is observed in the runs with Grid01 and Grid02. These results reflect that FMG-FAS is more efficient than NK when the free stream values applied as the initial guess for the flow field.

Table 5.2. The results of NK solutions of transonic flow. For all tests; T_∞=273.15 K, p_∞=101325 N/m^2, $\gamma = 1.4$.

	Grid01	Grid02	Grid03
Airfoil	NACA0012 225x33	NACA0012 257x257	RAE2822 257x129
M_∞	0.80	0.80	0.729
α	1.25	1.25	2.31
Residual	.65E-12	.98E-12	.89E-12
Iter. No	65	90	74
t (sec)	42.63	560.74	228.43
CFL_{start}	10.00	10.00	10.00
CFL_{max}	1.76E06	3.41E06	1.71E06
Figure	Figure 5.4	Figure 5.5	Figure 5.6

Table 5.3. The results and comparison of FMG-FAS solutions of transonic flow using 225x33 grid. (* C_l and C_d values are from Vassberg and Jameson's work conducted in 2010 [42].) T_∞=273.15 K, p_∞=101325 N/m^2, $\gamma = 1.4$, M_∞=0.80, $\alpha = 1.25^o$.

Problem parameters	2 MG levels	3 MG levels	4 MG levels
NACA0012 M_∞=0.80 $\alpha = 1.25^o$	225x33 113x17	225x33 113x17 57x9	225x33 113x17 57x9 29x5
L2 residual	0.99E-12	0.99E-12	0.99E-12
Computation time (seconds)	26.75	18.01	17.30
Number of iterations	100 +913	100+100 +505	100+100+100 +461
Total number of iter.s	1013	705	761
C_l (0.3517)*	0.32356E+00	0.32356E+00	0.32356E+00
C_d (0.02245)*	0.22480E-01	0.22480E-01	0.22480E-01
Figure	5.9	5.8	5.7

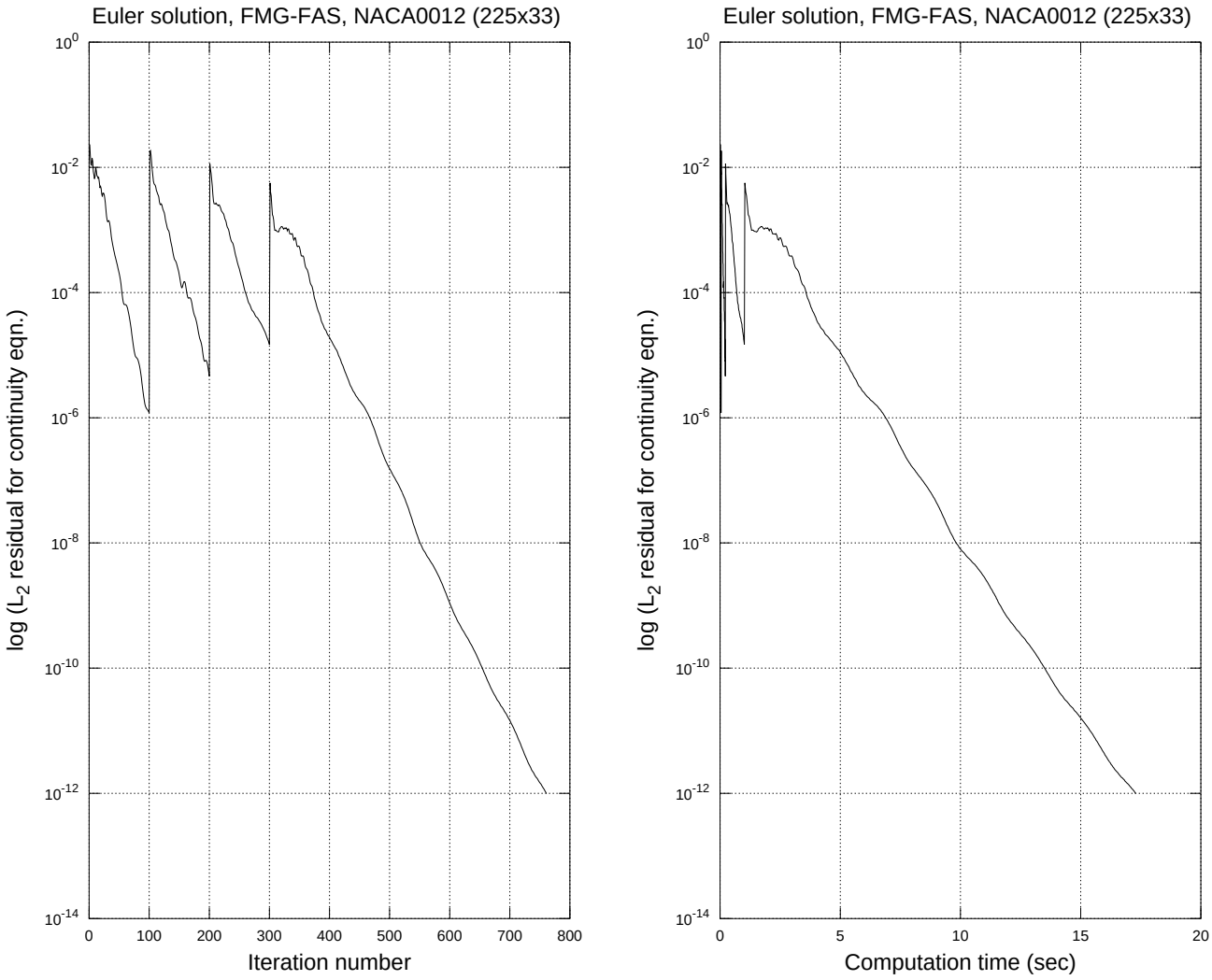

Figure 5.7. The convergence histories of FMG-FAS analysis of C-grid around NACA 0012 airfoil. Fine grid: 225x33; intermediate grid-1: 113x17; intermediate grid-2: 57x9; coarse grid: 29x5. The structural details of the grid is given in the Table 5.1. T_∞=273.15 K, p_∞=101325 N/m^2, $\gamma = 1.4$, M_∞=0.80, $\alpha = 1.25^o$.

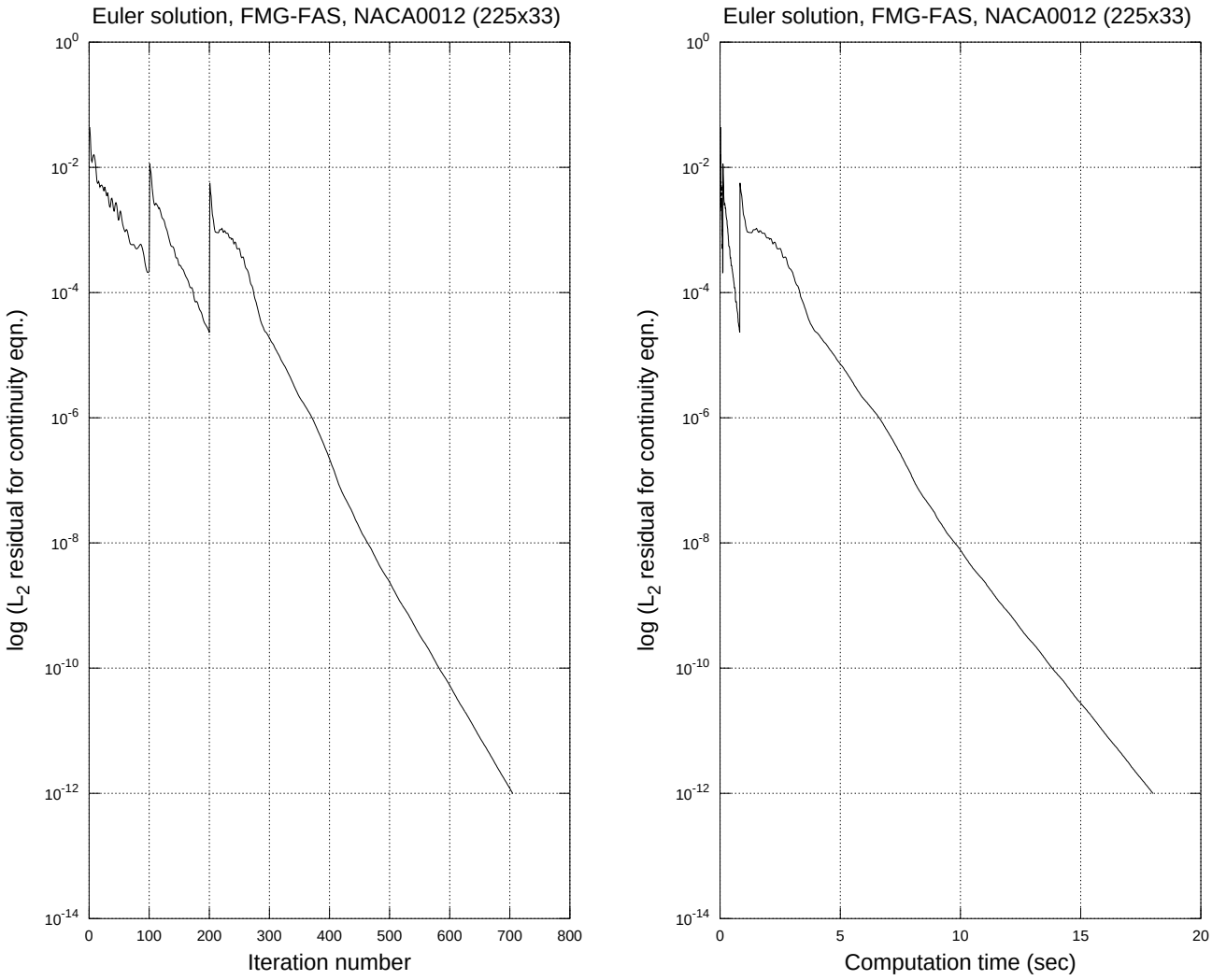

Figure 5.8. The convergence histories of FMG-FAS analysis of C-grid around NACA 0012 airfoil. Fine grid: 225x33; intermediate grid: 113x17; coarse grid: 57x9. The structural details of the grid is given in the Table 5.1 without MG Level 4. T_∞=273.15 K, p_∞=101325 N/m^2, $\gamma = 1.4$, M_∞=0.80, $\alpha = 1.25^o$.

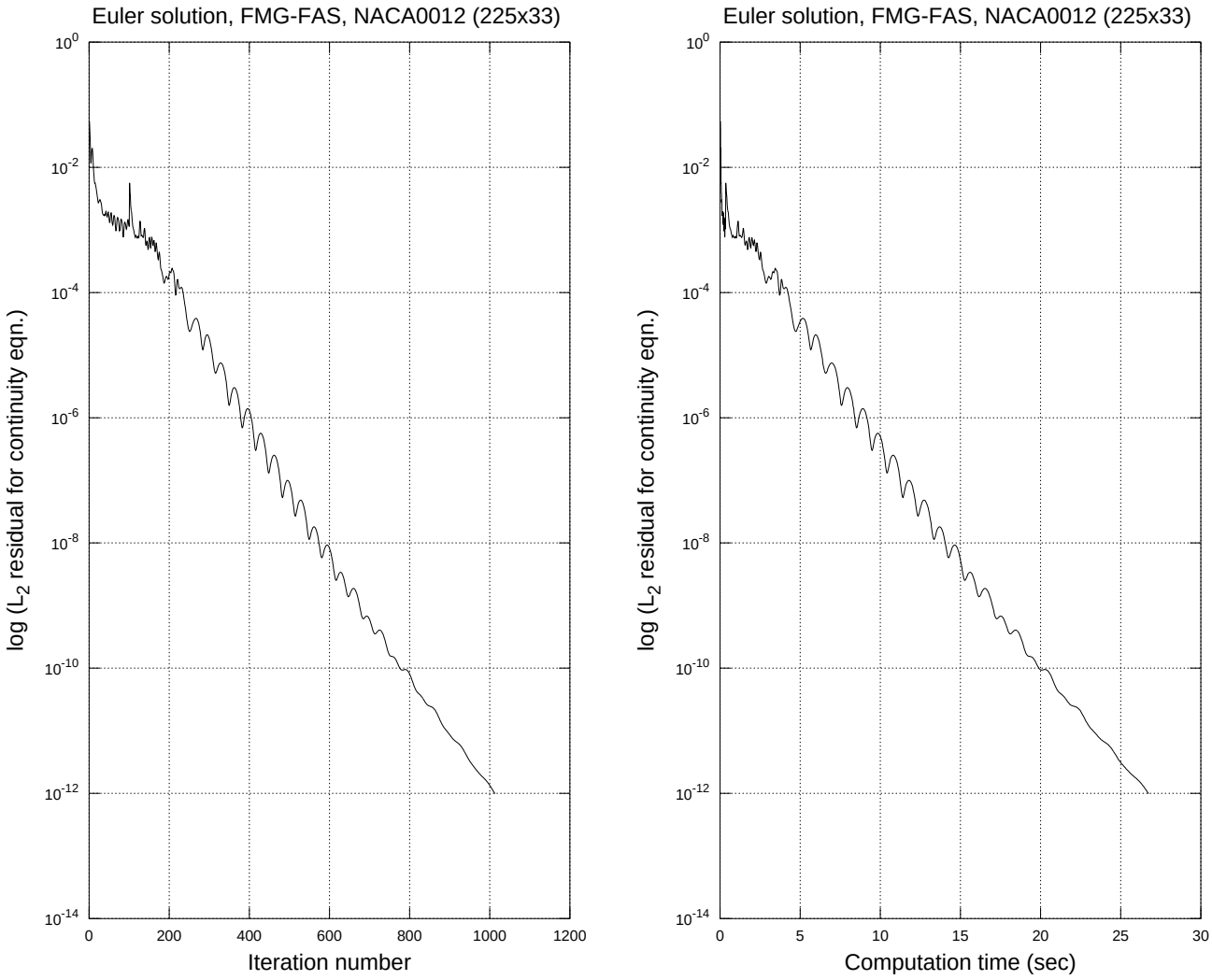

Figure 5.9. The convergence histories of FMG-FAS analysis of C-grid around NACA 0012 airfoil. Fine grid: 225x33; coarse grid: 113x17. The structural details of the grid is given in the Table 5.1 without MG Levels 3 and 4. T_∞=273.15 K, p_∞=101325 N/m^2, $\gamma = 1.4$, M_∞=0.80, $\alpha = 1.25^o$.

Table 5.4. The results and comparison of FMG-FAS solutions of transonic flow using 257x257 grid. (* C_l and C_d values are from Vassberg and Jameson's work conducted in 2010 [42]). T_∞=273.15 K, p_∞=101325 N/m^2, $\gamma = 1.4$, M_∞=0.80, $\alpha = 1.25^o$.

Problem parameters	2 MG levels	3 MG levels	4 MG levels
NACA0012 M_∞=0.80 $\alpha = 1.25^o$	256x256 128x128	256x256 128x128 64x64	256x256 128x128 64x64 32x32
L2 residual (total)	0.96E-12	0.99E-12	0.99E-12
Computation time (seconds)	372.74	198.10	181.71
Number of iterations	100 +1268	100+100 +600	100+100+100 +534
Total number of iter.s	1368	800	834
C_l (0.3517)*	0.331961E+00	0.331961E+00	0.331961E+00
C_d (0.02245)*	0.226488E-01	0.226488E-01	0.226488E-01
Figure	5.12	5.11	5.10

Table 5.5. The comparison of NK and FMG-FAS solutions of transonic flow. For all runs; T_∞=273.15 K, p_∞=101325 N/m^2, $\gamma = 1.4$.

	NK	NK	NK	FMG-FAS	FMG-FAS	FMG-FAS
	Grid01	Grid02	Grid03	Grid01	Grid02	Grid03
Airfoil	NACA 0012 225x33	NACA 0012 257x257	RAE 2822 257x129	NACA 0012 225x33	NACA 0012 257x257	RAE 2822 257x129
M_∞ α	0.80 1.25	0.80 1.25	0.729 2.31	0.80 1.25	0.80 1.25	0.729 2.31
Res.	.65E-12	.98E-12	.89E-12	.99E-12	.99E-12	.89E-12
IterNo	65	90	74	761	834	686
t (sec)	42.63	560.74	228.43	17.30	181.71	64.65
Figure	5.4	5.5	5.6	5.7	5.10	5.13

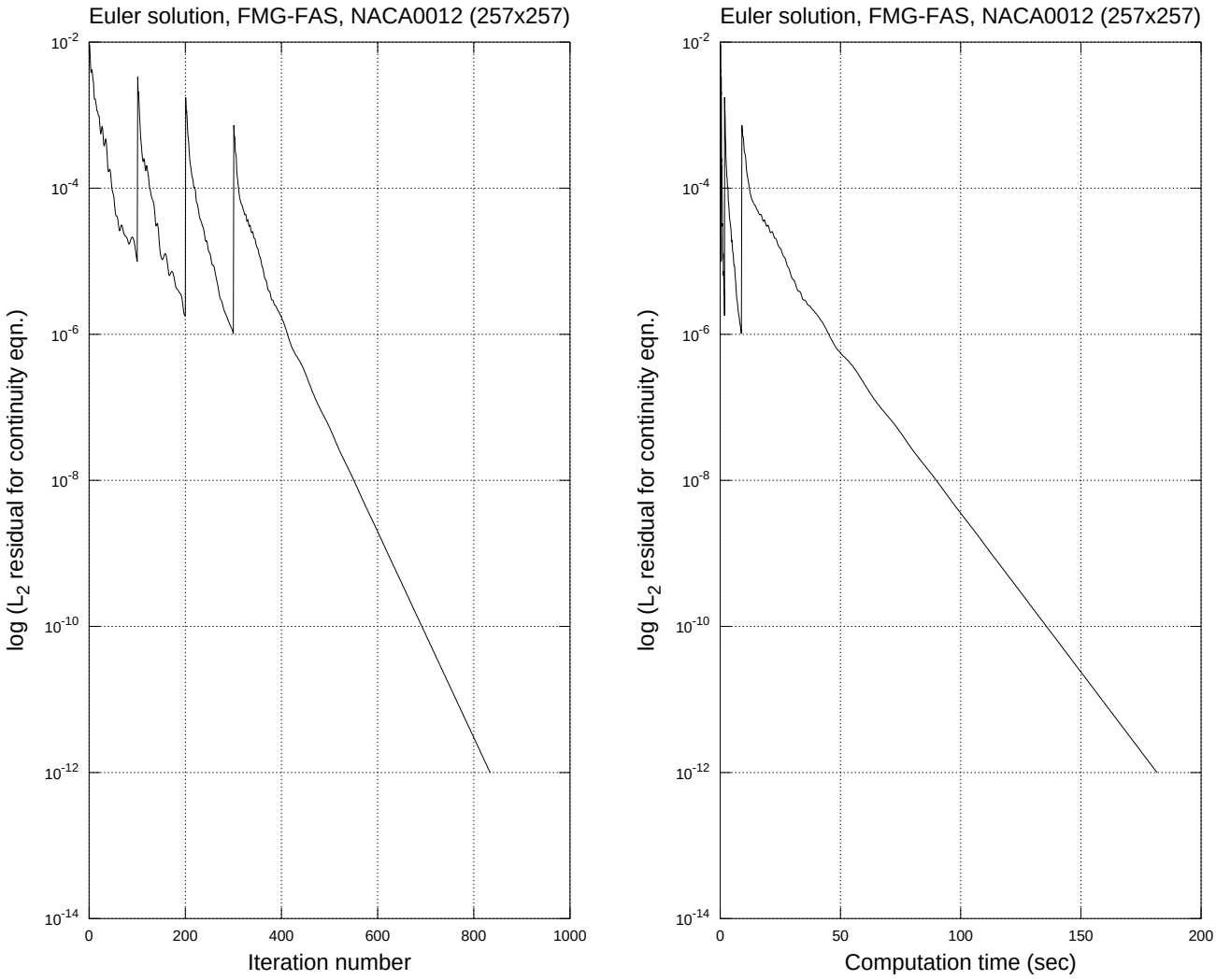

Figure 5.10. The convergence histories of FMG-FAS analysis of C-grid around NACA 0012 airfoil. Fine grid: 257x257; intermediate grid-1: 129x129; intermediate grid-2: 65x65; coarse grid: 33x33. The structural details of the grid is given in the Table 5.1. T_∞=273.15 K, p_∞=101325 N/m^2, $\gamma = 1.4$, M_∞=0.80, $\alpha = 1.25^o$.

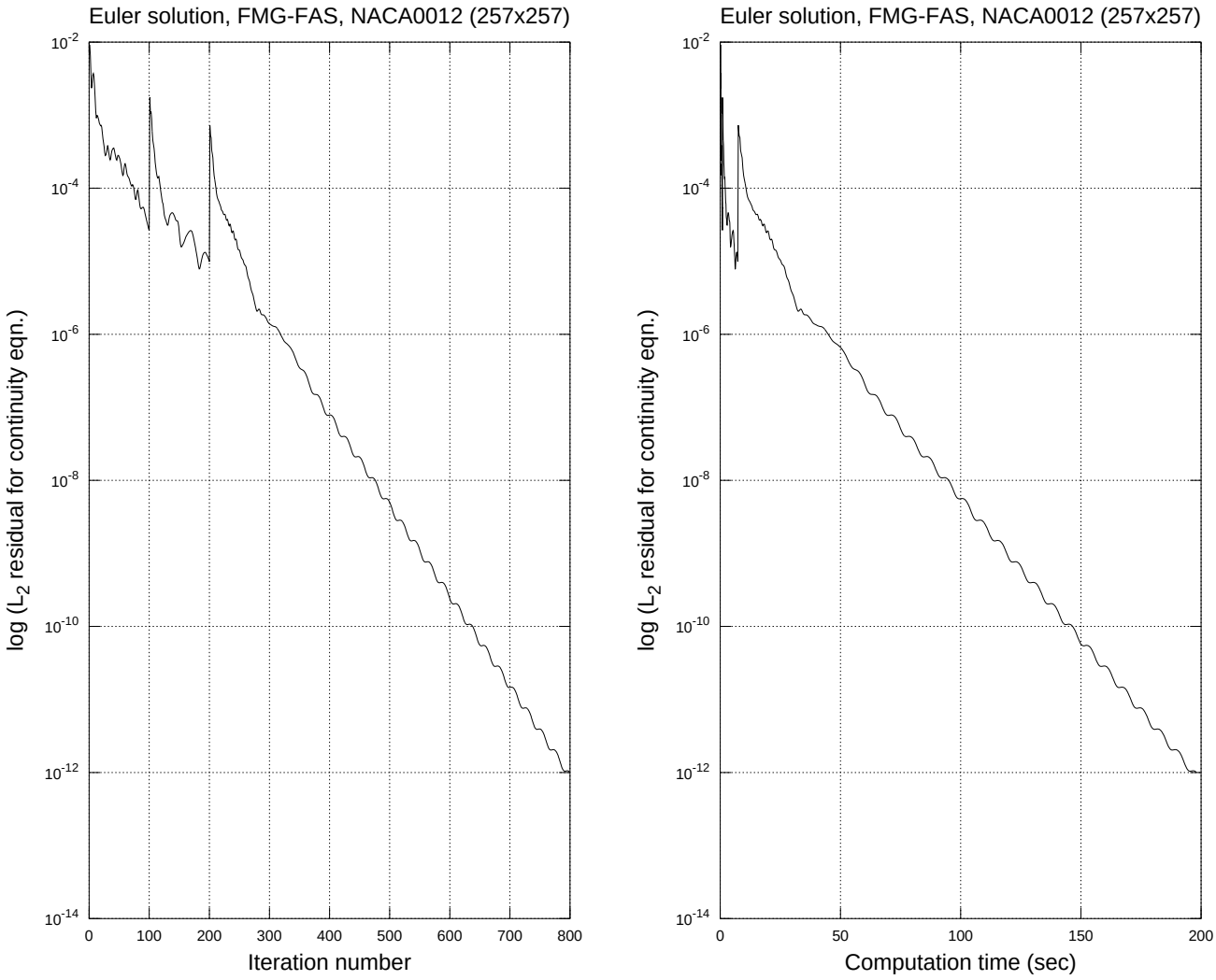

Figure 5.11. The convergence histories of FMG-FAS analysis of C-grid around NACA 0012 airfoil. Fine grid: 257x257; intermediate grid: 129x129; coarse grid: 65x65. The structural details of the grid is given in the Table 5.1 without MG Level 4. T_∞=273.15 K, p_∞=101325 N/m^2, $\gamma = 1.4$, M_∞=0.80, $\alpha = 1.25^o$.

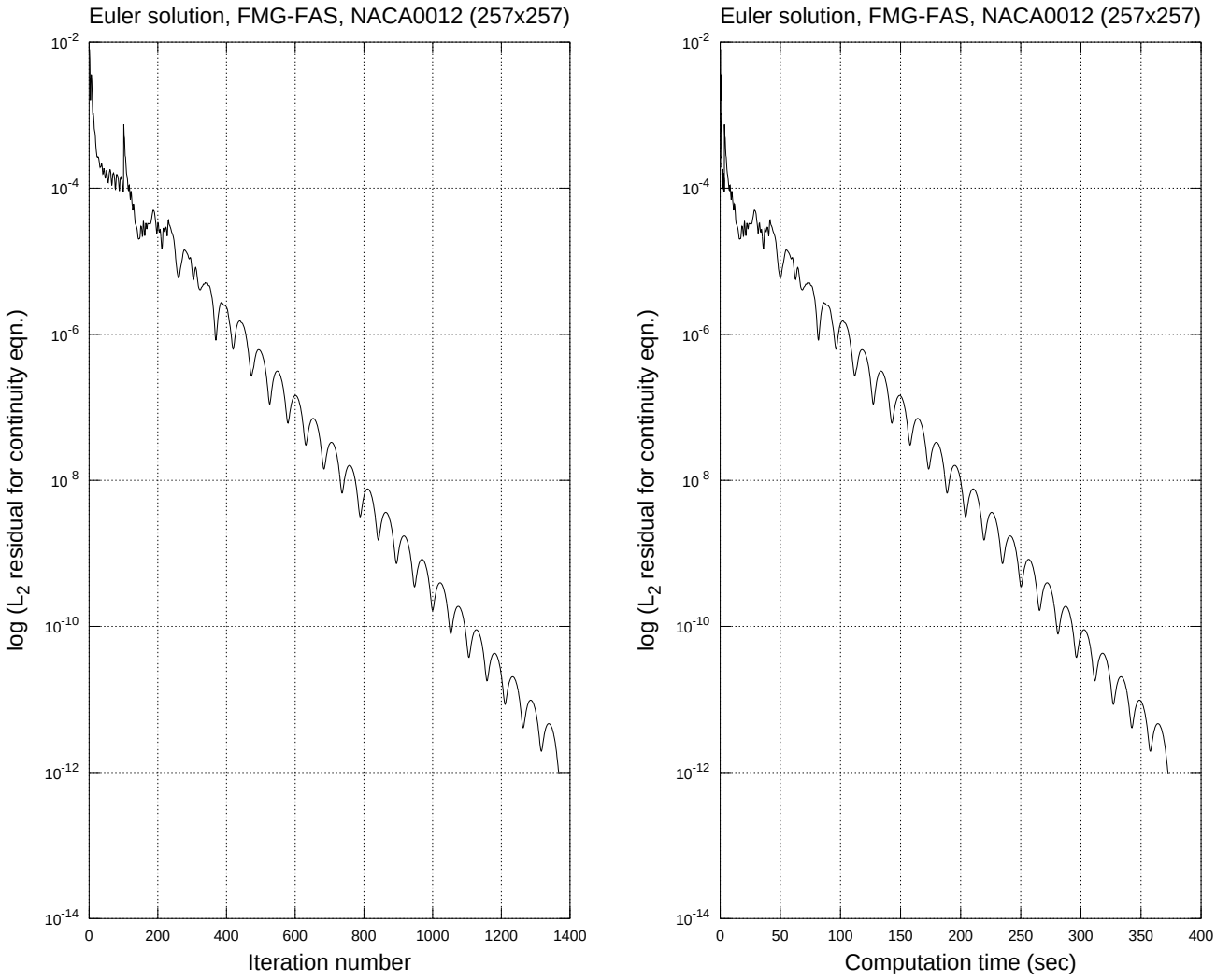

Figure 5.12. The convergence histories of FMG-FAS analysis of C-grid around NACA 0012 airfoil. Fine grid: 257x257; coarse grid: 129x129. The structural details of the grid is given in Table 5.1 without MG Levels 3 and 4. T_∞=273.15 K, p_∞=101325 N/m^2, $\gamma = 1.4$, M_∞=0.80, $\alpha = 1.25^o$.

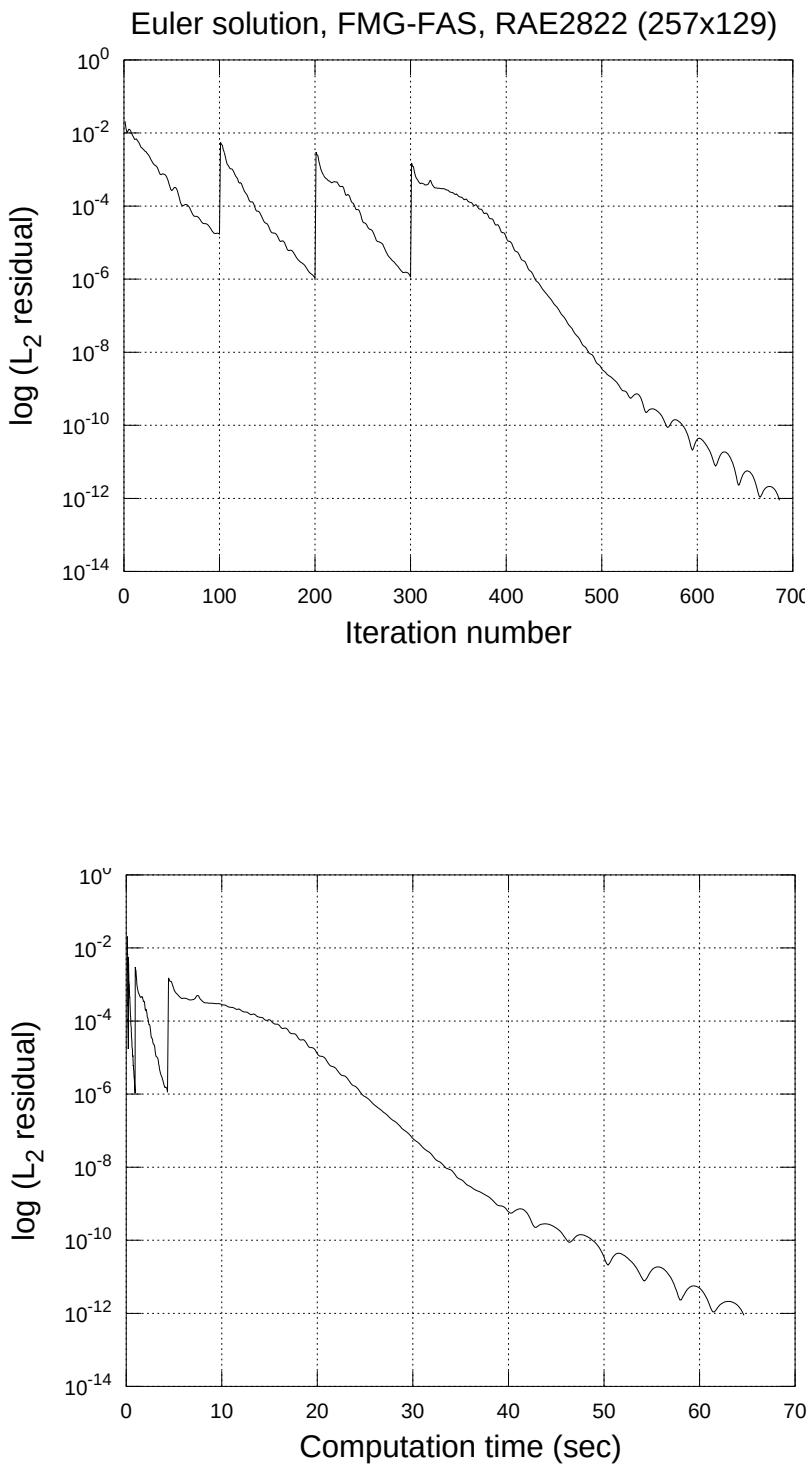

Figure 5.13. The convergence history of FMG-FAS analysis of transonic flow around RAE2822 airfoil. The structural details of Grid003 (257x129) are given in Table 5.1. T_∞=273.15 K, p_∞=101325 N/m^2, $\gamma = 1.4$, M_∞=0.729, $\alpha = 2.31^o$.

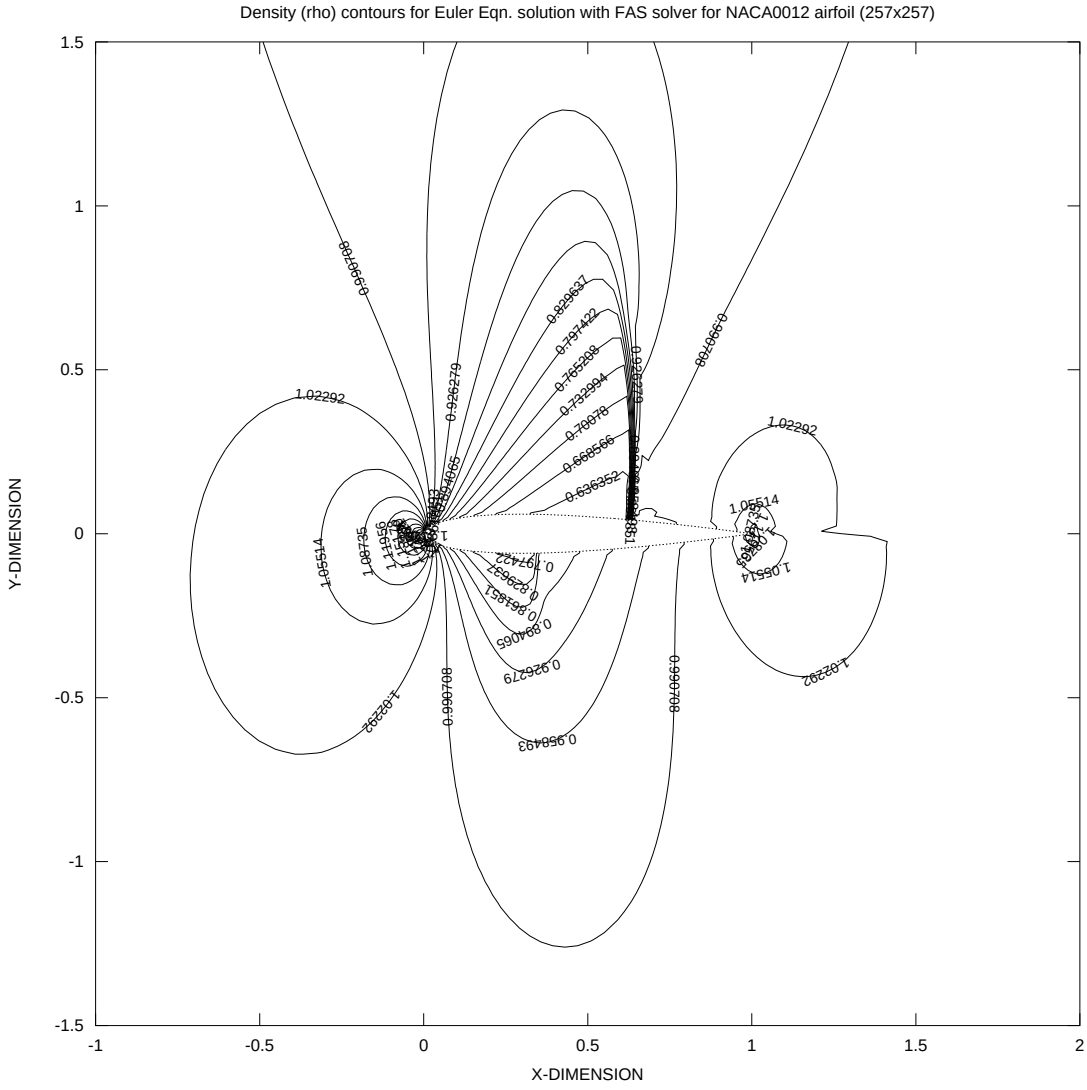

Figure 5.14. Density contour plot as a result of FMG-FAS analysis of C-grid around NACA 0012 airfoil. The structural details of the grid used is given in Table 5.1. T_∞=273.15 K, p_∞=101325 N/m^2, $\gamma = 1.4$, M_∞=0.80, $\alpha = 1.25^o$.

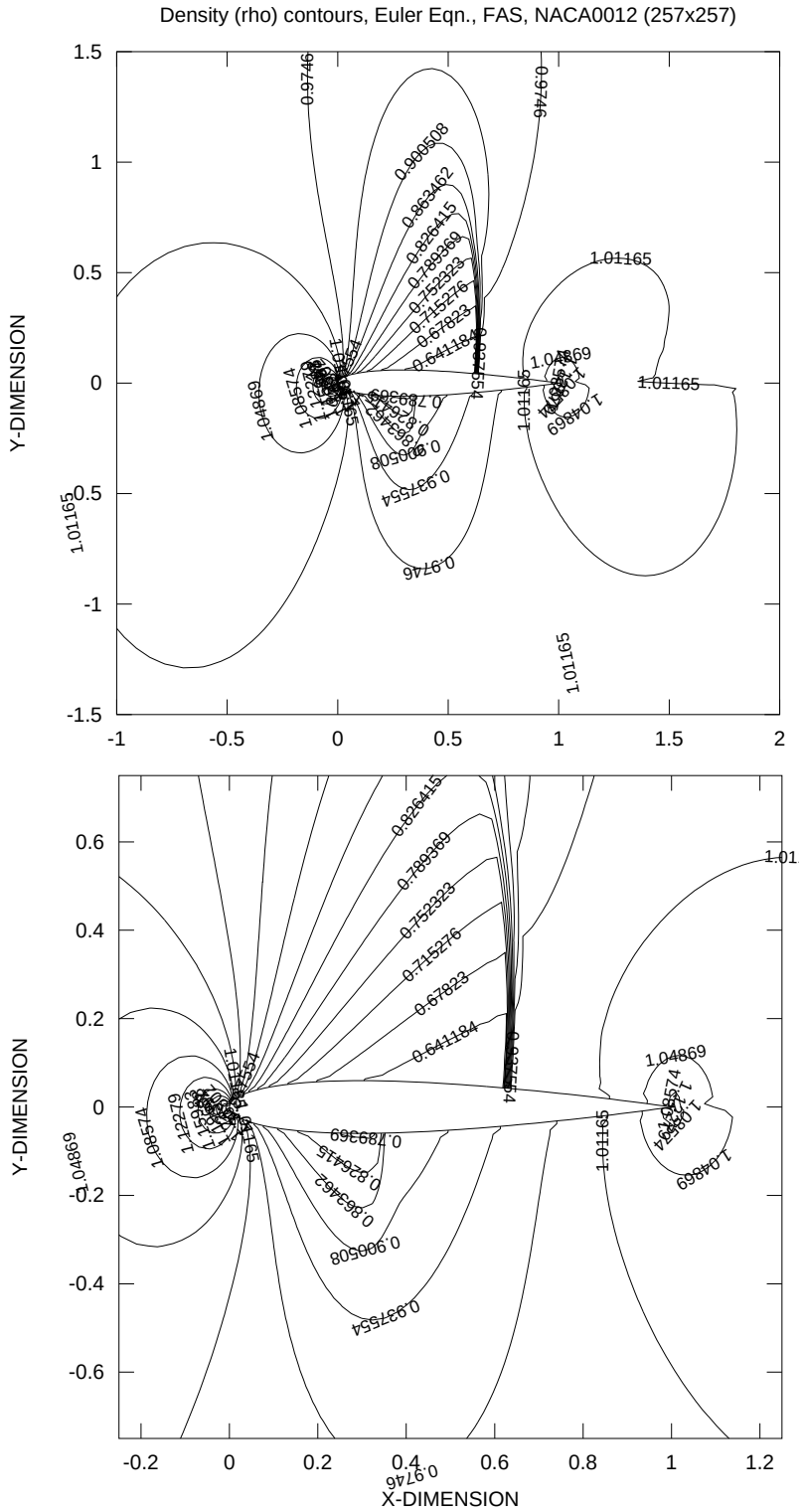

Figure 5.15. Density contour plot as a result of FMG-FAS analysis of transonic flow around NACA0012 airfoil. NACA0012 Grid002 (257x257) is in Figure 5.2. T_∞=273.15 K, p_∞=101325 N/m^2, $\gamma = 1.4$, M_∞=0.80, $\alpha = 1.25^o$.

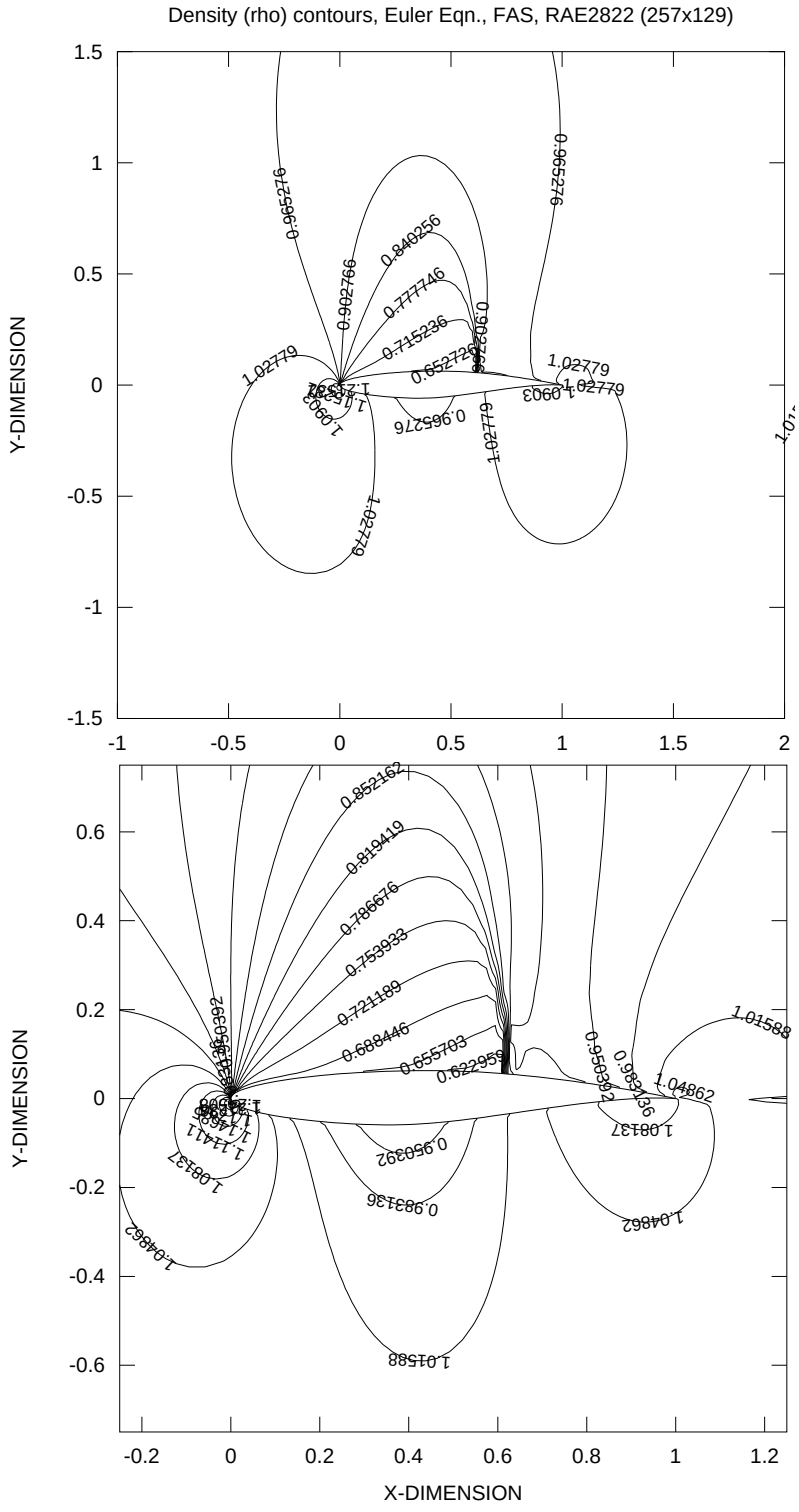

Figure 5.16. Density contour plot as a result of FMG-FAS analysis of transonic flow around RAE2822 airfoil. RAE2822 Grid003 (257x127) is in Figure 5.3. T_∞=273.15 K, p_∞=101325 N/m^2, $\gamma = 1.4$, M_∞=0.729, $\alpha = 2.31^o$.

5.3. Results for "Adaptive Time Stepping Strategy for the Implicit Solution of Steady Transonic Flows"

The results of the numerical experiments, conducted on the newly introduced method for the selection of local CFL number (local time step) in transonic flow problems, are given below.

In the numerical experiments, two different C-type grids around airfoils are used. The grids are around NACA0012 226x34 (symmetric) airfoil and RAE2822 257x129 (non-sysmmetric) airfoils. Test Cases A1, A2, B3, and B4 are conducted on NACA0012 and Test Case C is tested on RAE2822. NACA0012 (226x34) and RAE2822 (257x129) grids are given in Figures 5.1 and 5.3 respectively.

Convergence tolerance is set to 10^{-12} for all problems. If the convergence tolerance is not achieved without the code breakdown, the next CFL_{Rate} is tested. In this study, the further CFL_{Rate} values are not tested after the breakdown of the code.

In all numerical experiments, $ILU(2)$ is used for the preconditioning of GMRES.

In all numerical experiments, CFL_{Rate}=1 corresponds to deactivation of Pressure Sensor. Tests of CFL_{Rate} are set to start from 1 with the increments of 1, up to break down of the code (i.e. 79, 69) but limited to maximum of 100.

The computer used in these work has Intel(R) Core(TM)2 Quad CPU Q8300 @ 2.50GHz with 3.9 GiB memory.

Praveen's *flo2d* [33] code is used with necessary modifications required to conduct numerical experiments on the introduced method.

Outline of Test Cases for the numerical experiments are given in Tables 5.6 and 5.7.

Table 5.6. The outline of numerical experiments for NK solutions with "Pressure Sensor". (CFL_{repeat}=6; CFL_{predet}=10000, ILU(2)). TC: Test Case. For all runs, m=20 in GMRES(m).

Test Case	Airfoil	Pressure (Density) Sensor	GMRES (m)	CFL_{Rate}	CFL_{coef1}	CFL_{coef2}
A1	NACA0012	dpi	20	1:1:79	2.5	1.5
A2	NACA0012	dpj	20	1:1:69	2.5	1.5
B3	NACA0012	drhoi	20	1:1:79	2.5	1.5
B4	NACA0012	dpi	20	1:1:79	2.5	2.5
C	RAE2822	dpi	20	1:1:100	2.5	1.5

Table 5.7. The numerical experiments for NK solutions with "Pressure Sensor". For all tests; T_∞ is 273.15 K, p_∞ is 101325 N/m^2, $\gamma = 1.4$.

Test Case	A1	A2	B3	B4	C
M_∞	0.80	0.80	0.80	0.80	0.729
α	1.25	1.25	1.25	1.25	2.31

5.3.1. Results for Test Cases A1 and A2

5.3.1.1. Test Case A1: With "Pressure Sensor" *dpi*. The results of Test Case A1 is given in Table 5.8 and Figure 5.17. For "Pressure Sensor" in i-direction, GMRES(20), $M_\infty = 0.80$ and $\alpha = 1.25^o$, it is possible to achieve 18.3% reduction in computation time (Percent time reduction= $(52.48 - 42.89)/52.48 = 18.3\%$) and 30.3% reduction in iteration numbers (Percent iteration reduction= $(89 - 62)/89 = 30.3\%$) with the use of proposed adaptive time stepping method.

Figure 5.17 shows the reductions in both computation time and number of iterations, which are more pronounced in CFL_{Rate} range from 40 to 60. After $CFL_{Rate} =$ 65 some peaks are observed both in computation time and number of iterations. When CFL_{Rate} increased to 80, the code has diverged due to the extreme increase in Δt.

5.3.1.2. Test Case A2: With "Pressure Sensor" *dpj*. The results of Test Case A2 is given in Table 5.9 and Figure 5.18. For "Pressure Sensor" in j-direction, GMRES(20), $M_\infty = 0.80$ and $\alpha = 1.25^o$, it is possible to achieve 19.1% reduction in computation time (Percent time reduction= $(52.69 - 42.63)/52.69 = 19.1\%$) and 30.3% reduction in iteration numbers (Percent iteration reduction=$(89 - 62)/89 = 30.3\%$) with the use of proposed adaptive time stepping method.

As it can be seen in Figure 5.18, the reductions in both computation time and number of iterations achieved are similar to the results of Test Case A1. Figure 5.18 shows the reductions in both computation time and number of iterations, which are more pronounced in CFL_{Rate} range from 35 to 65. For Test Case A2 no peaks are observed, but at CFL_{Rate}=70, the code has diverged due to the extreme increase in Δt.

5.3.2. Results for Test Cases B3 and B4

Table 5.8. Test Case A1: The results and comparison of CFL selection in NK by the use of "Pressure Sensor" in i-direction. CFL_{Rate}=1 corresponds to no CFL selection. For all runs; m=20 in GMRES(m), f_{in}=2 in ILU (f_{in}). Test Case A1: T_∞=273.15 K, p_∞=101325 N/m^2, $\gamma = 1.4$.

Pressure Sensor (*dpi*) in i-direction	Without CFL selection	With CFL selection	With CFL selection	With CFL selection
GMRES (m)	m=20	m=20	m=20	m=20
ILU (f_{in})	f_{in}=2	f_{in}=2	f_{in}=2	f_{in}=2
NACA0012 M_∞ α	226x34 0.80 1.25	226x34 0.80 1.25	226x34 0.80 1.25	226x34 0.80 1.25
CFL_{Rate}	1	46	55	70
Total number of iterations	89	63	**62**	**62**
Computation time (sec)	52.48	42.97	**42.89**	43.00
L2 residual (overall)	0.98E-12	0.88E-12	0.99E-12	0.99E-12
CFL_{max}	1.50E06	1.9E06	1.7E06	1.4E06

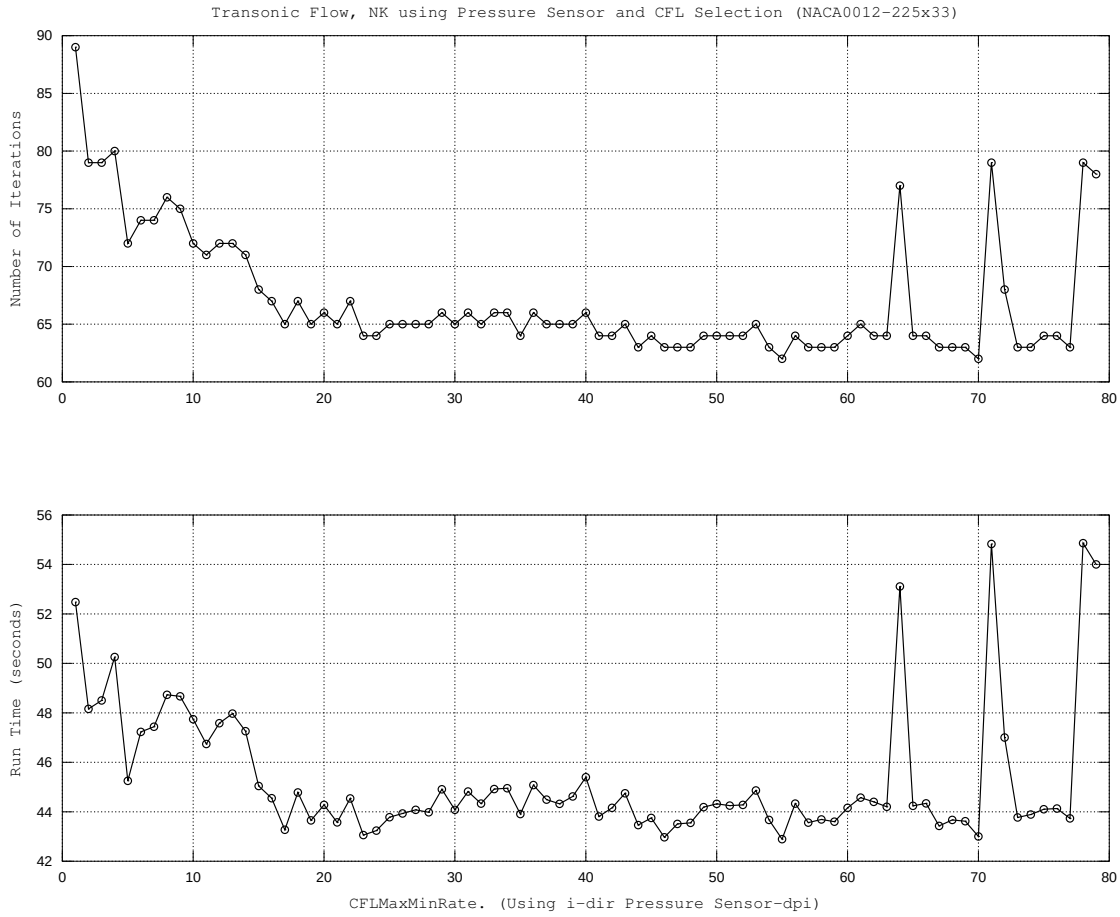

Figure 5.17. Test Case A1: CFL_{Rate} runs from 1 to 79 with GMRES (20) and "Pressure Sensor" in i-direction. T_∞=273.15 K, p_∞=101325 N/m^2, $\gamma = 1.4$, M_∞=0.80, $\alpha = 1.25^o$.

Table 5.9. Test Case A2: The results and comparison of CFL selection in NK with the use of "Pressure Sensor" in j-direction. CFL_{Rate}=1 corresponds to no CFL selection. For all runs; m=20 in GMRES(m), f_{in}=2 in ILU (f_{in}), NACA0012 (grid 226x34). Test Case A2: T_∞=273.15 K, p_∞=101325 N/m^2, $\gamma = 1.4$.

Pressure Sensor (dpj) in j-direction	Without CFL selection	With CFL selection	With CFL selection	With CFL selection
GMRES (m)	m=20	m=20	m=20	m=20
ILU (f_{in})	f_{in}=2	f_{in}=2	f_{in}=2	f_{in}=2
NACA0012	226x34	226x34	226x34	226x34
M_∞	0.80	0.80	0.80	0.80
α	1.25	1.25	1.25	1.25
CFL_{Rate}	1	25	39	59
Total number of iterations	89	63	**62**	**62**
Computation time (sec)	52.69	42.65	**42.63**	42.98
L2 residual (overall)	0.98E-12	0.95E-12	0.75E-12	0.70E-12
CFL_{max}	1.5E06	2.3E06	1.8E06	1.7E06

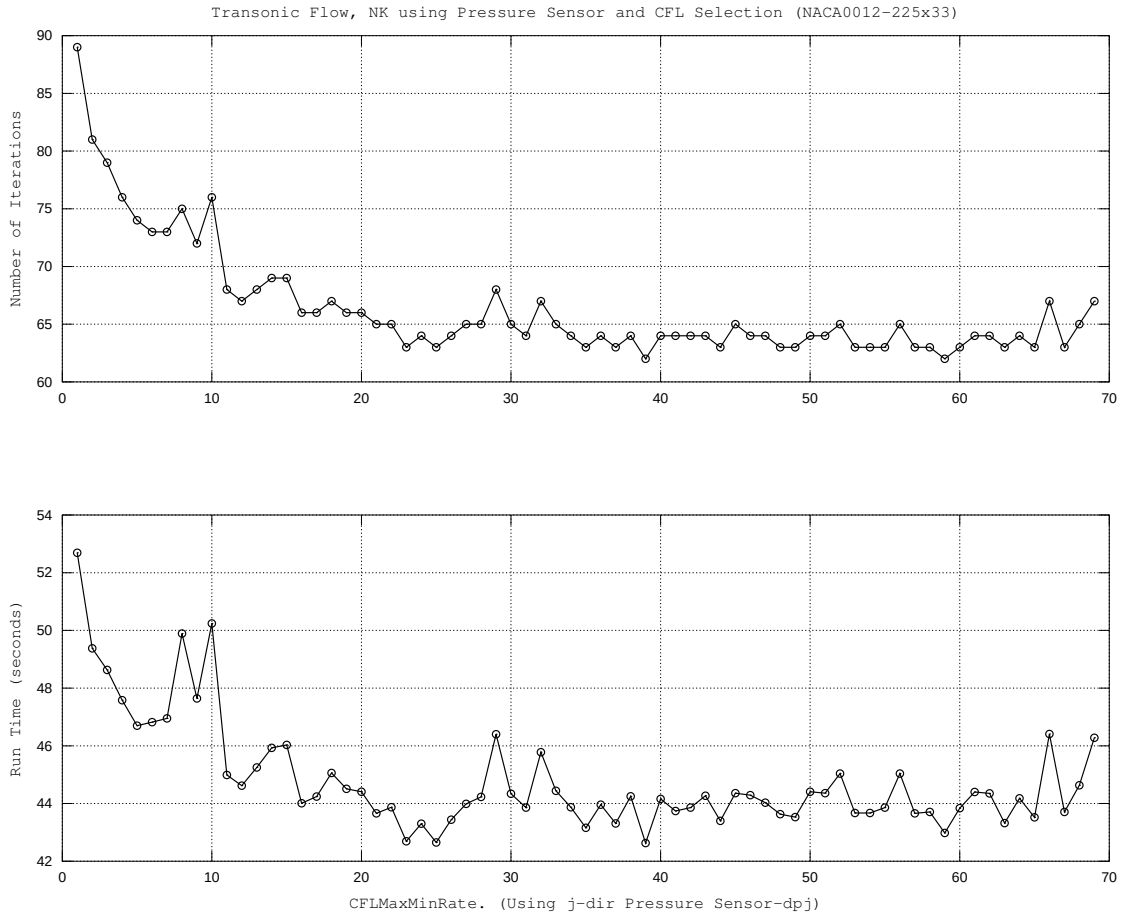

Figure 5.18. Test Case A2: CFL_{Rate} runs from 1 to 69 with GMRES (20) and "Pressure Sensor" in j-direction. T_∞=273.15 K, p_∞=101325 N/m^2, $\gamma = 1.4$, M_∞=0.80, $\alpha = 1.25^o$.

Table 5.10. Test Case B3: The results and comparison of CFL selection in NK with the use of "Density Sensor" in j-direction. CFL_{Rate}=1 corresponds to no CFL selection. For all runs; m=20 in GMRES(m), f_{in}=2 in ILU (f_{in}), NACA0012 (grid 226x34). Test Case B3: T_∞=273.15 K, p_∞=101325 N/m^2, $\gamma = 1.4$.

Density Sensor ($drhoj$) in j-direction	Without CFL selection	With CFL selection	With CFL selection	With CFL selection
GMRES (m)	m=20	m=20	m=20	m=20
ILU (f_{in})	f_{in}=2	f_{in}=2	f_{in}=2	f_{in}=2
NACA0012	226x34	226x34	226x34	226x34
M_∞	0.80	0.80	0.80	0.80
α	1.25	1.25	1.25	1.25
CFL_{Rate}	1	54	68	73
Total number of iterations	89	**62**	**62**	**62**
Computation time (sec)	52.47	**42.86**	43.02	43.06
L2 residual (overall)	0.98E-12	0.80E-12	0.94E-12	0.81E-12
CFL_{max}	1.5E06	3.5E06	1.1E06	1.4E06

5.3.2.1. Test Case B3. In Test Case B3, NACA0012 airfoil with Density Sensor is used. The parameters for this test case are $\alpha = 1.25^o$ and $M_\infty = 0.80$. The results of this test case (B3) are given in Table 5.10. In this test case instead of "Pressure Sensor", "Density Sensor" ($drho_i$) is used and 18.3% reduction in computation time (Percent time reduction= $(52.47 - 42.86)/52.47 = 18.3\%$) and 30.3% reduction in iteration numbers (Percent iteration reduction= $(89 - 62)/89 = 30.3\%$) with the use of proposed adaptive time stepping method are achieved. Since Test Case B3 behaves in a similar way to Test Cases A1 and A2, no figure for this test case is given.

5.3.2.2. Test Case B4. In Test Case B4, NACA0012 airfoil is used. The parameters for this test case are $\alpha = 1.25^o$ and $M_\infty = 0.80$. The results of this test case (B4)

Table 5.11. Test Case B4: The results and comparison of CFL selection in NK with the use of "Pressure Sensor" in i-direction. CFL_{coef2} is set to 2.5 instead of 1.5. CFL_{Rate}=1 corresponds to no CFL selection. For all runs; m=20 in GMRES(m), f_{in}=2 in ILU (f_{in}), NACA0012 (grid 226x34). T_∞=273.15 K, p_∞=101325 N/m^2, $\gamma = 1.4$.

Pressure Sensor (dpi) in i-direction	Without CFL selection	With CFL selection	With CFL selection	With CFL selection
GMRES (m)	m=20	m=20	m=20	m=20
ILU (f_{in})	f_{in}=2	f_{in}=2	f_{in}=2	f_{in}=2
NACA0012 M_∞ α	226x34 0.80 1.25	226x34 0.80 1.25	226x34 0.80 1.25	226x34 0.80 1.25
CFL_{Rate}	1	48	67	70
Total number of iterations	90	**62**	**62**	**62**
Computation time (sec)	53.38	**42.77**	42.88	42.83
L2 residual (overall)	0.76E-12	0.99E-12	0.95E-12	0.86E-12
CFL_{max}	413.E06	1041E06	742E06	93E06

are given in Table 5.11. In this test case, CFL_{coef2} is set to 2.5 instead of 1.5. Test Case B4: The results and comparison of CFL selection in NK with the use of "Pressure Sensor" in i-direction. CFL_{coef2} is set to 2.5 instead of 1.5. CFL_{Rate}=1 corresponds to no CFL selection. For all runs; m=20 in GMRES(m), f_{in}=2 in ILU (f_{in}), NACA0012 (grid 226x34), $M_\infty = 0.80$, $\alpha = 1.25^o$. Percent time reduction=$\frac{(53.38-42.77)}{53.38} = 19.9\%$. Percent iteration reduction=$\frac{(89-62)}{89} = 30.3\%$.

5.3.3. Results for Test Case C: RAE2822, $\alpha = 2.31^o$ and $M_\infty = 0.729$

In Test Case C, $\alpha = 2.31^o$, $M_\infty = 0.729$ and "Pressure Sensor" (dp_i) are used and the results are given in Table 5.12 and Figure 5.19. In this test case, it is possible to achieve 13.6% reduction in computation time (Percent time reduction = $(271.25 - 234.28)/271.25 = 13.6\%$) and 28.2% reduction in iteration numbers (Percent iteration reduction = $(103 - 74)/103 = 28.2\%$) with the use of proposed adaptive time stepping method.

Figure 5.19 shows the reductions in both computation time and number of iterations, which are more pronounced in CFL_{Rate} range from 40 to 100. In Test Case C, no peaks are observed in computation time and number of iterations. The numerical experiments are stopped, when CFL_{Rate} is equal to 100. In this test case no divergence is observed.

Table 5.12. Test Case C: The results and comparison of CFL selection in NK by the use of "Pressure Sensor" in i-direction for RAE2822 grid (257x129). CFL_{Rate}=1 corresponds to no CFL selection. For all runs; m=20 in GMRES(m), f_{in}=2 in ILU (f_{in}). T_∞=273.15 K, p_∞=101325 N/m^2, $\gamma = 1.4$.

Pressure Sensor in i-dir.	Without CFL selection	With CFL selection	With CFL selection	With CFL selection	With CFL selection
GMRES (m)	m=20	m=20	m=20	m=20	m=20
ILU (f_{in})	f_{in}=2	f_{in}=2	f_{in}=2	f_{in}=2	f_{in}=2
RAE2822 M_∞ α	257x129 0.729 2.31	257x129 0.729 2.31	257x129 0.729 2.31	257x129 0.729 2.31	257x129 0.729 2.31
CFL_{Rate}	1	84	90	94	99
Number of iter.	103	**74**	**74**	**74**	**74**
Comput. time (sec)	271.25	**234.28**	234.48	234.71	235.11
L2 resid. (overall)	0.94E-12	0.89E-12	0.90E-12	0.95E-12	0.95E-12
CFL_{max}	795,000	2.2E06	2.0E06	1.8E06	2.6E06

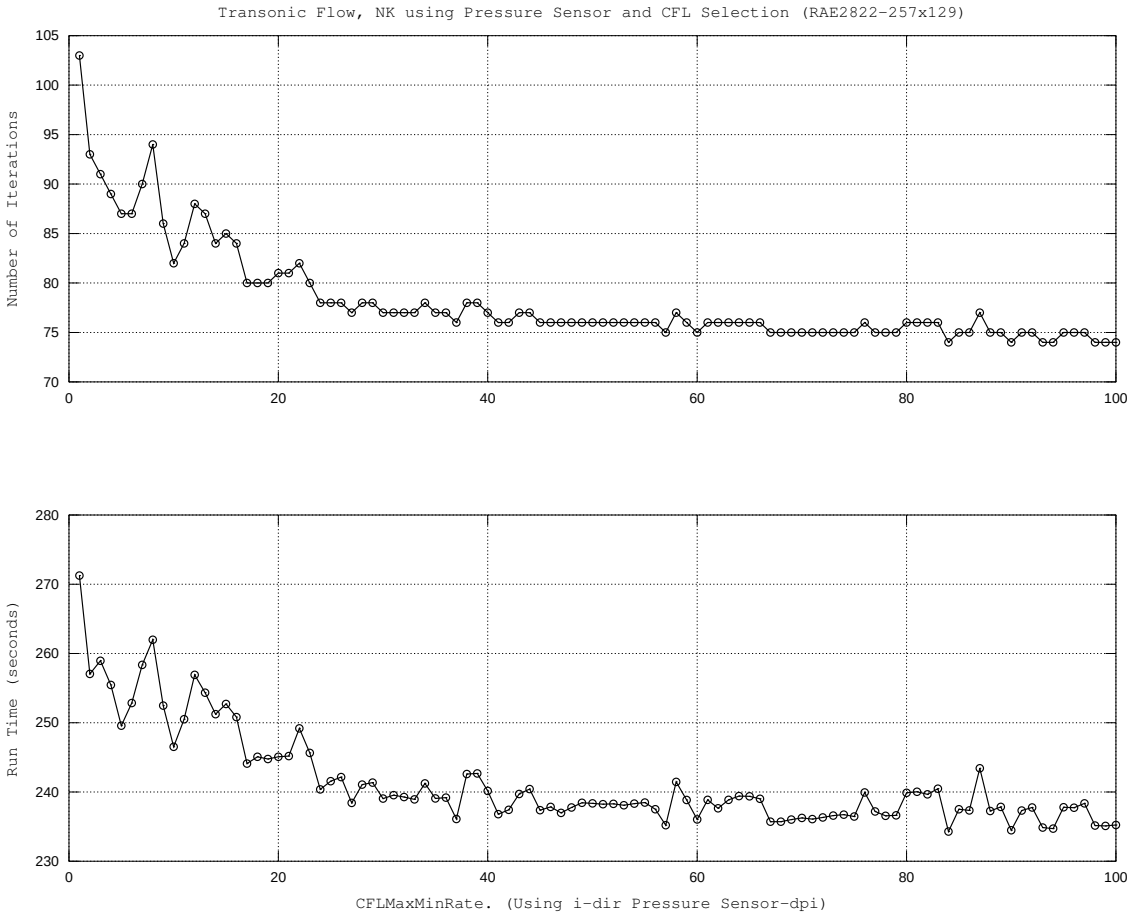

Figure 5.19. Test Case C: CFL_{Rate} runs from 1 to 100 with GMRES (20) and "Pressure Sensor" in i-direction. T_∞=273.15 K, p_∞=101325 N/m^2, $\gamma = 1.4$, M_∞=0.729, $\alpha = 2.31^o$.

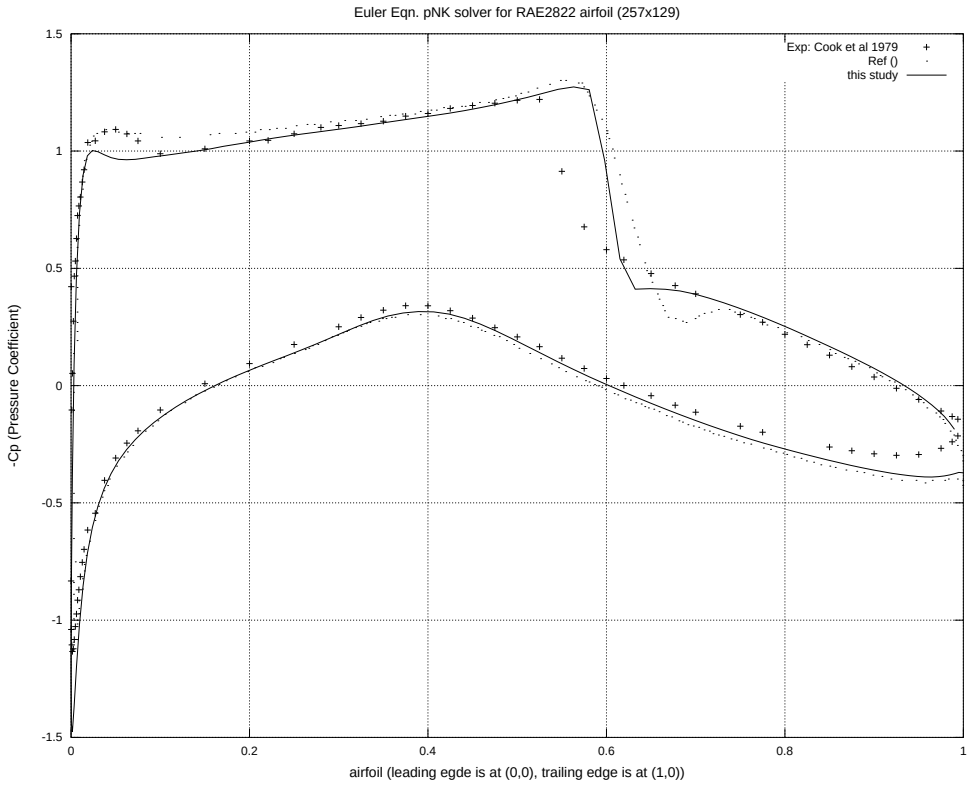

Figure 5.20. C-grid around RAE2822 grid 257x129 Cp plot comparison. Experimental values are from Cook *et al.* [45]. T_∞=273.15 K, p_∞=101325 N/m^2, $\gamma = 1.4$, M_∞=0.729, $\alpha = 2.31^o$.

6. CONCLUSIONS

6.1. Conclusions for the Comparison of NK and FMG-FAS Methods

A number of numerical investigations has been conducted, based on Euler equation solutions of transonic flow, to study the influence of different parameters and grids that can influence computation time and convergence behavior of two different implicit and nonlinear solution methods namely NK and FMG-FAS.

In the numerical experiments, the initial guess for the solution is set to free stream values. In NK, the startup takes some time, before the residual starts decreasing with a considerable rate. Sometimes after the CFL number was increased, an increase in the residual has been observed followed by a limited number of iterations. Later in the computation process, the required decrease in residual towards convergence is sustained. The process of CFL number increase has to be kept under control. Otherwise, uncontrolled increase in CFL number leads to larger time steps and divergence may occur. On the other hand, if the rate of increase in CFL number is unnecessarily restrained, then the computation time becomes longer than it should be. For NK implicit time integration scheme, a CFL selection method has been implemented which provides start up procedure for the flow solution to develop from freestream values and after the development of flowfield, CFL selection method decides the incrementation of CFL number till convergence is achieved.

The nonlinear multigrid method, used in this study, is based on the full approximation storage (FAS) and full multigrid (FMG). Pseudo time stepping is used to achieve steady state solution. At each time step, a number of coarse to fine and fine to coarse grid iterations are conducted.

The utmost care has been paid to bring $flo2d$ and ISAAC to the same basis for comparison reasons. $flo2d$ is converted to be a structured solver and the same FV discretization methodology is used in both iterative methods for the comparisons. Test

cases demonstrate that both methods are capable of effectively solving transonic flow. As it has been outlined in the results section, in case the free stream values are used as initial guess, the convergence times required for convergence show that less time is required for FMG-FAS to reach convergence compared to NK.

Numerical tests showed that CFL selection methods play a major role in the performance of NK by determining larger time steps, where it is applicable but using smaller time steps to prevent divergence.

With better CFL selection methods and preconditioning techniques, NK may reach a convergence speed similar to FMG-FAS.

6.2. Conclusions for "Adaptive Time Stepping Strategy for the Implicit Solution of Steady Transonic Flows"

In transonic flow solutions, Newtons method suffers from the use of smooth flow field (farfield values) as initial guess. In Newton-Krylov (NK) methods, "Pressure Sensor" is effectively used to detect steep gradients like shocks in the flow and CFL_{local} (i.e. Δt_{local}) selection has been performed as follows: (i) In the vicinity of steep gradients, small time steps are used. (ii) In the relatively smoother regions, larger time steps are used to achieve faster convergence. This method, which is used to determine local time step and based on the local CFL value, can be attributed as an adaptive time stepping method.

In order to investigate the performance of the introduced CFL_{local} (Δt_{local}) selection methodology, numerical experiments on various sensors (pressure and density) and different airfoil geometries like NACA0012 (symmetric) and RAE2822 (unsymmetric) with different grid densities have been conducted and the obtained computation time and iteration numbers are evaluated to see the influence on the convergence behavior.

The results show that the introduced CFL selection methodology (with "Pressure Sensor"), which decides the incrementation of CFL number (i.e. Δt) locally, can be

used as a start-up procedure for the flow to develop from free-stream values within NK implicit time integration scheme. In this study, we have focused on structured grids and tested the code with and without "Pressure Sensor".

The investigation of the adaptive time stepping strategy addressed in this study shows that the introduced method significantly improves the convergence rate for inviscid compressible flow, which is solved for steady transonic flow with the initial condition of free stream values. In other words, transonic flow calculations conducted by using NK with CFL number (time step) selection methodology showed that it is possible to reduce the computation time.

In addition to reported 2-D transonic flow problem which shows the capability of reducing the computation time and number of iterations, this method needs to be tested in a 3-D transonic flows as well.

In this study, for the sake of simplicity, CFL_{local} has been calculated by using either dp_i, dp_j or $drho_i$. As a future study, it is possible to combine dp_i and dp_j so that the effects of both pressure sensors (gradients in both directions) are taken into account.

APPENDIX A: COARSE GRID PRECONDITIONED GMRES

This section can be named as "Operator-Preconditioner Mismatch for Jacobian-Free Implementation of GMRES".

It has been explained that the preconditioner applications increase the efficiency of iterative linear solvers. Since ILU preconditioner needs Jacobian matrix, it is not possible to talk about a real matrix-free Newton-ILU preconditioned GMRES application. The storage requirement of Jacobian matrix increases the load of the solver. In order to decrease the load of storing Jacobian matrix, it is possible to use coarse grid preconditioner instead of using the real size matrix.

Table A presents the results of a 2-D Elliptic Grid Generation around an airfoil problem solution with grid size (193x225). Instead of using (193x225) grid to set up Jacobian matrix, two other grids, (49x57) and (97x113) are used. The results show that, the preconditioner utilizes Jacobian matrix from (193x225) grid converges with less GMRES iterations. On the other hand, other two preconditioners with Jacobian matrices from (49x57) and (97x113) grids converges well in spite of the increased iteration numbers.

Table A.1. Convergence of the Elliptic Grid Generation around an airfoil with grid size (193x225). The size of the preconditioner is the same as the problem itself.

Preconditioner size (193x225)

Newton iteration number	Newton residual	GMRES residual	GMRES iteration number/(m)
0	0.214E-01	0.000E+00	(00/20)
1	0.147E-02	0.819E-02	(05/20)
2	0.334E-04	0.234E-02	(09/20)
3	0.400E-07	0.484E-03	(41/20)

Table A.2. Convergence of the Elliptic Grid Generation around an airfoil with grid size (193x225). Preconditioner size (97x113)

Newton iteration number	Newton residual	GMRES residual	GMRES iteration number/(m)
0	0.214E-01	0.000E+00	(00/20)
1	0.148E-02	0.819E-02	(05/20)
2	0.325E-04	0.234E-02	(11/20)
3	0.642E-07	0.493E-03	(51/20)

Table A.3. Convergence of the Elliptic Grid Generation around an airfoil with grid size (193x225). Preconditioner size (49x57)

Newton iteration number	Newton residual	GMRES residual	GMRES iteration number/(m)
0	0.214E-01	0.000E+00	(00/20)
1	0.153E-02	0.821E-02	(08/20)
2	0.333E-04	0.216E-02	(15/20)
3	0.136E-06	0.707E-03	(42/20)

APPENDIX B: FULL POTENTIAL EQUATION (FPE)

The solution of Finite Volume (FV) discretized Full Potential Equation (FPE) is attempted by using a combination of Newton-Krylov nonlinear iterative solution technique and multigrid FAS methods. The upwinding and flux biasing with FPE are performed. Multigrid FAS approach is combined with Newton-Krylov methods in order to accelerate convergence and achieve smaller residual with the same computer performance, i.e. less computer time and memory usage.

As a model problem for the numerical experimentation, Transonic Full Potential Equations (FPE) around a NACA 0012 airfoil is used. Nonlinear governing equations are discretized by using Finite Volume method.

Transonic Full Potential Equations can be solved by upwinding, where the values at the upstream nodes are biased as Jiang and Forsyth reported in [12].

B.1. Transonic Full Potential Equations

In this section derivation of Transonic Full Potential equations from Euler equations is outlined.

In Euler equations, which consist of the continuity, x momentum and y momentum equations, the mass conversation in steady state is,

$$\frac{\partial}{\partial x}(\rho u) + \frac{\partial}{\partial y}(\rho v) = 0, \tag{B.1}$$

and the conservation of momentum in x and y directions,

$$\frac{\partial}{\partial x}\left(\rho u^2 + p\right) + \frac{\partial}{\partial y}\left(\rho uv\right) = 0 \quad \text{and} \quad \frac{\partial}{\partial x}\left(\rho uv\right) + \frac{\partial}{\partial y}\left(\rho v^2 + p\right) = 0, \tag{B.2}$$

where ρ,p,u and v are the density, the pressure and the velocity components in x and y directions respectively. For an isentropic ideal gas the entropy is constant,

$$\frac{p}{\rho^\gamma} = constant. \tag{B.3}$$

Full Potential Equation (FPE) formulation for 2-D steady state can be derived as follows. The full velocity potential equation is an exact equation for irrotational flow over arbitrary bodies, at any angle of attack. FPE holds for an irrotational, inviscid, and compressible flow. The continuity, momentum and energy equations can be expressed in one equation with one independent variable, which is the velocity potential in two dimensions, $\Phi(x, y)$. The velocity potential is defined as $V = \nabla\Phi$ with

$$u = \Phi_x \quad \text{and} \quad v = \Phi_y.$$

$$\frac{\partial}{\partial x}\left(\rho\Phi_x\right) + \frac{\partial}{\partial y}\left(\rho\Phi_y\right) = 0, \tag{B.4}$$

Bernoulli's equation can be derived from x- and y-momentum equations with some assumptions,

$$\frac{q^2}{2} + \frac{a}{\gamma - 1} = constant, \tag{B.5}$$

where q is the local flow speed with $q^2 = u^2 + v^2$. The sound speed a is defined by $a^2 = dp/d\rho$ and Mach number is defined by $M = q/a$. ρ_∞, M_∞, a_∞ are the freestream properties.

$$\frac{q^2}{2} + \frac{a}{\gamma - 1} = \frac{{q_\infty}^2}{2} + \frac{a_\infty}{\gamma - 1}. \tag{B.6}$$

$$\frac{a^2}{{a_\infty}^2} = 1 + \frac{\gamma - 1}{2} {M_\infty}^2 \left(1 - \frac{q^2}{{q_\infty}^2}\right) \tag{B.7}$$

The above equation can be changed with $(\frac{a}{a_\infty})^2 = \left(\frac{\rho}{\rho_\infty}\right)^{(\gamma-1)}$, which is derived from the definition of speed of sound with isentropic flow consideration.

$$\frac{\rho}{\rho_\infty} = \left(1 + \frac{\gamma - 1}{2} {M_\infty}^2 \left(1 - \frac{q^2}{{q_\infty}^2}\right)\right)^{(\frac{1}{\gamma-1})}, \tag{B.8}$$

$$\left(1 - \frac{\Phi_x^2}{a^2}\right) \Phi_{xx} + \left(1 - \frac{\Phi_y^2}{a^2}\right) \Phi_{yy} - \left(\frac{2\Phi_x \Phi_y}{a^2}\right) \Phi_{xy} = 0. \tag{B.9}$$

$$a^2 = a_\infty^2 + \left(\frac{\gamma - 1}{2}\right)\left(V_\infty^2 - V^2\right). \tag{B.10}$$

$$V^2 = \Phi_x^2 + \Phi_y^2. \tag{B.11}$$

B.2. Boundary Conditions for FPE

The flow must be tangent to the wing surface, which has a shape function of $y = f(x)$.

$$\frac{df}{dx} = \frac{\Phi_y}{\Phi_x} = \frac{v}{u}.$$

The far field boundary conditions, $u = u_\infty$, $v = v_\infty$ and $\rho = \rho_\infty$ are set away from the wing.

B.3. Finite Volume Discretization of FPE

FPE is discretized by using finite volume method with quadrilateral grid and cell centered approach is used. After the integration of Equation B.4, the term at the left hand side is obtained and approximating the flux values as shown below is the first step to finite volume formulation of FPE.

$$\int\int_{\forall}\left(\frac{\partial}{\partial x}\left(\rho\Phi_x\right)+\frac{\partial}{\partial y}\left(\rho\Phi_y\right)\right)d\forall=\sum_{m=1}^{4}\int_{S_m}\rho\nabla\phi\cdot\vec{n}dS=0, \tag{B.12}$$

On face S_m the values can be approximated and the integral can be converted to sum of fluxes on the cell faces.

$$\sum_{m=1}^{4}\int_{S_m}\rho\nabla\phi\cdot\vec{n}dS=\sum_{m=1}^{4}\rho_m(\nabla\phi)_m\cdot n_m S_m, \tag{B.13}$$

where m denotes the approximate values at cell faces for 2-D problems. The neighbors of the cell (I,J), which can be named as North(N), South(S), West(W) and East(E). The cell centers of N, S, W and E cells are donated by $(I,J)_m$.

$$\begin{aligned}
&(S) \quad m=1 \quad (I,J) \text{ and } (I,J-1)_{m=1}\\
&(E) \quad m=2 \quad (I,J) \text{ and } (I+1,J)_{m=2}\\
&(N) \quad m=3 \quad (I,J) \text{ and } (I,J+1)_{m=3}\\
&(W) \quad m=4 \quad (I,J) \text{ and } (I-1,J)_{m=4}
\end{aligned} \tag{B.14}$$

$$\phi_{(I,J)_m}-\phi_{I,J}\approx(\nabla\phi)_m\cdot n_m l_m.$$

where l_m is the length of the vector between the center node (I, J) and the point $(I, J)_m$. So Equation B.13 can be approximated as follows:

$$\sum_{m=1}^{4} \rho_m(\phi_{(I,J)_m} - \phi_{I,J})\frac{S_m}{l_m} = 0, \tag{B.15}$$

B.4. Upwinding for Potential in FPE

In order to bias the values at the upstream cells known as upwinding, it is required to set up the connection between the potential, ϕ, and velocity , V.

$$\vec{V} = \vec{\nabla}\phi \cdot \vec{n} = n_x u + n_y v = n_x \Phi_x + n_y \Phi_y. \tag{B.16}$$

The calculation of $\nabla\phi$ requires ϕ values at (I, J) and m. In order to determine the value to be used in the NS direction and EW direction the following method is proposed by Jiang and Forsyth [12].

$$m_{EW} = \begin{cases} m_E & \text{if } \phi_{m_W} - \phi_{m_E} \geq 0 \\ m_W & \text{otherwise} \end{cases} \qquad m_{NS} = \begin{cases} m_S & \text{if } \phi_{m_N} - \phi_{m_S} \geq 0 \\ m_N & \text{otherwise} \end{cases} \tag{B.17}$$

$\nabla\phi$ can be calculated by using the below equation considering the selections made accordingly for m_{EW} and m_{NS}.

$$\begin{bmatrix} (x_{m_{EW}} - x_{I,J}) & (y_{m_{EW}} - y_{I,J}) \\ (x_{m_{NS}} - x_{I,J}) & (y_{m_{NS}} - y_{I,J}) \end{bmatrix} \begin{bmatrix} \phi_x \\ \phi_y \end{bmatrix} = \begin{bmatrix} \phi_{m_{EW}} - \phi_{I,J} \\ \phi_{m_{NS}} - \phi_{I,J} \end{bmatrix} \tag{B.18}$$

B.5. Density as Continuation Parameter

As Jiang and Forsyth [12] reports, the density value, ρ_m, in Equation B.15 is replaced by $\overline{\rho}_m$ as follows;

$$\sum_{m=1}^{4} \overline{\rho}_m (\phi_{(I,J)_m} - \phi_{I,J}) \frac{S_m}{l_m} = 0, \tag{B.19}$$

where $\overline{\rho}_m = (1 - r)\rho_m + r\rho_{up}$.

The continuation parameter, r, starts from one and goes to zero after each converged continuation step. ρ_{up} provides convergence by using upwinding as below at the intermediate steps.

$$\rho_{up} = \begin{cases} \rho_{(I,J)} & \text{if } \phi_m - \phi_{(I,J)} \geq 0 \\ \rho_m & \text{otherwise} \end{cases} \tag{B.20}$$

The first Newton step starts with $r = 1$, after the residual decreases to a certain value, r shall be decreased a certain amount and the solution for the previous continuation parameter shall be feed to the new iteration as an initial guess. The intermediate step does not require to converge fully, but the last step with $r = 0$ shall converge to the desired accuracy.

APPENDIX C: ARTIFICIAL DISSIPATION

As outlined in Section (1.2.1.1), to prevent oscillations in the transonic flow solutions, in central difference schemes artificial dissipation is introduced. Artificial Dissipation [15] as defined below is added to Equations 2.25- 2.28.

$$\frac{\partial}{\partial t}(\Omega\rho_m) + \sum_{m=1}^{4}(\triangle y_m \rho_m u_m - \triangle x_m \rho_m v_m) - D\rho = 0 \tag{C.1}$$

Artificial Dissipation operator D for continuity equation is as follows and the others can be derived accordingly.

$$\begin{aligned} D\rho &= D_x\rho + D_y\rho \qquad \text{where;} \\ D_x\rho &= d_{I+\frac{1}{2},J} - d_{I-\frac{1}{2},J} \\ D_y\rho &= d_{I,J+\frac{1}{2}} - d_{I,J-\frac{1}{2}} \end{aligned} \tag{C.2}$$

$$\begin{aligned} d_{I+\frac{1}{2},J} = \frac{h_{I+\frac{1}{2},J}}{\triangle t}[\varepsilon^{(2)}_{I+\frac{1}{2},J}(\rho_{I+1,J} - \rho_{I,J}) \\ - \varepsilon^{(4)}_{I+\frac{1}{2},J}(\rho_{I+2,J} - 3\rho_{I+1,J} + 3\rho_{I,J} - \rho_{I-1,J})] \end{aligned} \tag{C.3}$$

The pressure sensor ν is defined to determine $\varepsilon^{(2)}$ and $\varepsilon^{(4)}$

$$\nu_{I,J} = \frac{|P_{I+1,J} - 2P_{I,J} + P_{I-1,J}|}{|P_{I+1,J}| + |2P_{I,J}| + |P_{I-1,J}|} \tag{C.4}$$

where $\varepsilon^{(2)}$ and $\varepsilon^{(4)}$

$$\varepsilon^{(2)}_{I+\frac{1}{2},J} = \kappa^{(2)} max(\nu_{I+1,J}, \nu_{I,J}) \tag{C.5}$$

$$\varepsilon^{(4)}_{I+\frac{1}{2},J} = max\left[0, (\kappa^{(4)} - \varepsilon^{(2)}_{I+\frac{1}{2},J})\right] \tag{C.6}$$

Different values for $\kappa^{(2)}$ and $\kappa^{(4)}$ have been applied in similar calculations, the values given below are taken from Jameson's original paper [15].

$$\kappa^{(2)} = 1/4 \qquad \kappa^{(4)} = 1/256 \tag{C.7}$$

REFERENCES

1. Luo, H., J.D. Baum, and R. Lohner, "Fast Matrix-free Implicit Method for Compressible Flows on Unstructured Grids", *Journal of Computational Physics,* Vol. 146, No. 2, pp. 664 – 690, 1998.

2. Saad Y. and M.H. Schultz, "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems", *SIAM Journal on Scientific and Statistical Computing,* Vol. 7, No. 3, pp. 856 – 869, 1986.

3. Lotz R.D., B.E. Thompson, and C.A. Konings, "Numerical Uncertainities in Transonic Flow Calculations for Airfoils with Blunt Trailing Edges", *International Journal for Numerical Methods in Fluids,* Vol. 24, No. 4, pp. 355 – 373, 1997.

4. Emwin P. and L. Davidson, "A Local Mesh Refinement Algorithm Applied to Turbulent Flow, *International Journal for Numerical Methods in Fluids",* Vol. 24, No. 5, pp. 519 – 530, 1997.

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

6. Hackbusch, W., *Multi-Grid Methods and Applications,* Springer-Verlag, Tokyo, 1985.

7. Wesseling, P., *An Introduction to Multigrid Methods,* Wiley, New York, 1991.

8. Washio, T., and C. W. Oosterlee, "Krylov Subspace Acceleration Method for Nonlinear Multigrid Schemes", *Electronic Transactions on Numerical Analysis,* Vol. 6, pp. 271 – 290, 1997.

9. Oosterlee, C. W. and T. Washio, "Krylov Subspace Acceleration of Nonlinear Multigrid with Application to Recirculating Flows", *SIAM Journal on Scientific Com-*

puting, Vol. 21, No. 5, pp. 1670 – 1690, 2000.

10. Fidkowski, K.J., and D.L. Darmofal, "Development of A Higher-Order Solver for Aerodynamic Applications", *42nd AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV.,* Paper Number 2004-0436, 2004.

11. Sidilkover, D., "Some Approaches Towards Constructing Optimally Efficient Multigrid Solvers for The Inviscid Flow Equations", *Institute for Computer Applications in Science and Engineering (ICASE),* Technical Report: TR-97-39, 1997.

12. Jiang, H., and P.A. Forsyth, "Robust Numerical Methods for Transonic Flows", *International Journal for Numerical Methods in Fluids,* Vol. 24, No. 5, pp. 457 – 476, 1997.

13. Jiang, H., and P.A. Forsyth, "Robust Linear and Nonlinear Strategies for Solution of The Transonic Euler Equations", *Computers and Fluids,* Vol. 24, No. 7, pp. 753 – 770, 1995.

14. Forsyth, P.A. and H. Jiang, "Nonlinear Iteration Methods for High Speed Laminar Compressible Navier-Stokes Equations", *Computers and Fluids,* Vol. 26, No. 3, pp. 249 – 268, 1997.

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

16. Jameson A., "Solution of the Euler Equations for Two-Dimensional, Transonic Flow by a Multigrid Method", *Applied Mathematics and Computation,* Vol. 13, No. 3 – 4 pp. 327 – 355, 1983.

17. Caughey, D.A. and A. Jameson, "Fast Preconditioned Multigrid Solution of the Euler and Navier Stokes Equations for Steady, Compressible Flows", *International Journal for Numerical Methods in Fluids,* Vol. 43, No. 5, pp. 537 – 553, 2003.

18. McHugh, P.R. and D.A. Knoll, "Comparison of Standard and Matrix-Free Implementations of Several Newton-Krylov Solvers", *AIAA Journal,* Vol. 32, No. 12, pp. 2394 – 2400, 1994.

19. Ecder, A., and R. Ünsal, "Multiple-Domain Analysis of Combustion of a Spherical Fuel Droplet", *International Journal for Computational Methods in Engineering Science and Mechanics,* Vol. 8, No. 2, pp. 103 – 106, 2007.

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

21. Jameson, A., "Computational Aerodynamics for Aircraft Design", *Science,* Vol. 245, No. 4916, pp. 361 – 371, 1989.

22. Jameson, A., "A Perspective on Computational Algorithms for Aerodynamic Analysis and Design", *Progress in Aerospace Sciences,* Vol. 37, No. 2 pp. 197 – 243, 2001.

23. Shen, Y., B. Wang, and G. Zha, "Comparison Study of Implicit Gauss-Seidel Line Iteration Method for Transonic Flows", *AIAA Paper 2007-4332,* 2007.

24. Choquet, R., and J. Erhel, "Newton-GMRES Algorithm Applied to Compressible Flows", *International Journal for Numerical Methods in Fluids,* Vol. 23, No. 2, pp. 177 – 190, 1996.

25. Blanco, M., and D.W. Zingg, "A Fast Solver for the Euler Equations on Unstructured Grids Using a Newton-GMRES Method", *AIAA Paper 97-0331,* 1997

26. Pueyo, A., and D.W. Zingg, "Progress in Newton-Krylov Methods for Aerodynamic Calculations", *AIAA Paper 97-0877,* 1997.

27. Pueyo, A., and D.W. Zingg, "Improvements to a Newton-Krylov Solver for Aerodynamic Flows", *AIAA Paper 98-0619,* 1998.

28. Chisholm, T., and D.W. Zingg, "A Fully-Coupled Newton-Krylov Solver for Turbulent Aerodynamic Flows", *23rd Congress of International Council of the Aeronautical Sciences, Toronto, Canada,* Paper ICAS 2002-1.9.3, 2002.

29. Manzano, L.M., J.V. Lassaline, P. Wong and D.W. Zingg, "A Newton-Krylov Algorithm for the Euler Equations Using Unstructured Grids", *AIAA Paper 2003-0274,* 2003.

30. Rossow, C.-C., "Enhanced Runge-Kutta/Implicit Methods for Solving the Navier-Stokes Equations", *New Results in Numerical and Experimental Fluid Mechanics VI, Notes on Numerical Fluid Mechanics and Multidisciplinary Design (NNFM),* Vol. 96, pp. 202 – 209, 2008.

31. Michalak, K., and C.F. Ollivier-Gooch, "Matrix-Explicit GMRES for A Higher-Order Accurate Inviscid Compressible Flow Solver", *AIAA Paper 2007-3943,* 2007.

32. Maciel, E. S. G., "Solutions of the Euler Equations Using Implicit TVD High Resolution Algorithms in Three-Dimensions", *Mecanica Computacional Journal, Argentina*, Vol. XXVIII, No. 18, pp. 1517 – 1541, 2009.

33. Praveen, C., "Adjoint code development and optimization using automatic differentiation", *National Aerospace Laboratories, NAL PD CF-0604,* 2006.

34. Morrison, J. H., "A compressible Navier-Stokes solver with two-equation and Reynolds stress turbulence closure models", *NASA CR4440,* 1992.

35. Abid, R., J.H. Morrison, T.B. Gatski, and C.G. Speziale, "Prediction of Complex Aerodynamic Flows with Explicit Algebraic Stress Models", *AIAA Paper 96-0565,* 1996.

36. Campobasso, M. S., A. Zanon, M. Foerster, F. Fraysse, and A. Bonfiglioli, "CFD Modelling of Wind Turbine Airfoil Aerodynamics", *63 Congresso Nazionale ATI Energia per lo Sviluppo Sostenibile, Palermo,* 2008.

37. Beam, R.M., and R.F. Warming, "An Implicit Factored Scheme for the Compressible Navier-Stokes Equations", *AIAA Journal,* Vol. 16, No. 4, pp. 393 – 402, 1978.

38. Roe, P. L., "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes", *Journal of Computational Physics,* Vol. 43, No. 2, pp. 357 – 372, 1981.

39. Van Leer, B., "Towards the Ultimate Conservative Difference Scheme. II: Monotonicity and Conservation Combined in a Second-Order Scheme", *Journal of Computational Physics,* Vol. 14, No. 4, pp. 361 – 370, 1974.

40. Van Leer, B., "Towards the Ultimate Conservative Difference Scheme V. A second Order Sequel to Godunov's Method", *Journal of Computational Physics,* Vol. 32, No. 1, pp. 101 – 136, 1979.

41. Venkatakrishnan, V., "On the Accuracy of Limiters and Convergence to Steady-State Solutions", *AIAA Paper 93-0880,* 1993.

42. Vassberg, J. C. and A. Jameson, "In Pursuit of Grid Convergence for Two-Dimensional Euler Solutions", *Journal of Aircraft,* Vol. 47, No. 4, pp. 1152 – 1166, 2010.

43. Bücker, H. M., B. Pollul, and A. Rasch, "On CFL evolution strategies for implicit upwind methods in linearized Euler equations", *International Journal for Numerical Methods in Fluids*, Vol. 59, No. 1,, pp. 1 – 18, 2009.

44. Pollul, B. and A. Reusken, "Iterative Solvers for Discretized Stationary Euler Equations", *Summary of Flow Modulation and Fluid-Structure Interaction Findings (W. Schröder, ed.), Notes on Numerical Fluid Mechanics and Multidisciplinary Design,* Vol. 109, pp. 295 – 223, 2010.

45. Cook, P.H., M.A. McDonald, and M.C.P. Firmin, "Aerofoil RAE 2822 - Pressure Distributions, and Boundary Layer and Wake Measurements", *Experimental Data Base for Computer Program Assessment,* AGARD Report AR 138, 1979.