

IMPLEMENTATION OF ROTATION INTO A 2-D EULER SOLVER

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

BY

ENVER DORUK ÖZDEMİR

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

SEPTEMBER 2005

Approval of the Graduate School of Natural and Applied Sciences

Prof. Dr. Canan ÖZGEN
Director

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

Prof. Dr. S. Kemal İDER
Head of Department

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

Prof. Dr. M. Haluk AKSEL
Supervisor

Examining Committee Members

Prof. Dr. Kahraman ALBAYRAK	(METU, ME)	_____
Prof. Dr. M. Haluk AKSEL	(METU, ME)	_____
Instructor Dr. Tahsin ÇETİNKAYA	(METU, ME)	_____
Asst. Prof. Dr. Cüneyt SERT	(METU, ME)	_____
Dr. L. Oktay GÖNÇ	(TÜBİTAK-SAGE)	_____

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

Name, Last name : Enver Doruk ÖZDEMİR

Signature :

ABSTRACT

IMPLEMENTATION OF ROTATION INTO A 2-D EULER SOLVER

ÖZDEMİR, Enver Doruk

M.Sc., Department of Mechanical Engineering

Supervisor : Prof. Dr. M. Haluk AKSEL

September 2005, 91 pages

The aim of this study is to simulate the unsteady flow around rotating or oscillating airfoils. This will help to understand the rotor aerodynamics, which is essential in turbines and propellers.

In this study, a pre-existing Euler solver with finite volume method that is developed in the Mechanical Engineering Department of Middle East Technical University (METU) is improved. This structured pre-existing code was developed for 2-D internal flows with Lax-Wendroff scheme.

The improvement consist of firstly, the generalization of the code to external flow; secondly, implementation of first order Roe's flux splitting scheme and lastly, the implementation of rotation with the help of Arbitrary Lagrangian Eulerian (ALE) method.

For the verification of steady and unsteady results of the code, the experimental and computational results from literature are utilized. For steady conditions, subsonic and transonic cases are investigated with different angle of attacks. For the verification of unsteady results of the code, oscillating airfoil case is used.

The flow is assumed as inviscid, unsteady, adiabatic and two dimensional. The gravity is neglected and the air is taken as ideal gas.

The developed code is run on computers housed in METU Mechanical Engineering Department Computational Fluid Dynamics High Performance Computing (CFD-HPC) Laboratory.

Keywords: CFD, Finite Volume Method, Arbitrary Lagrangian Eulerian, Unsteady Aerodynamics, Rotation, Oscillating Airfoil

ÖZ

2 BOYUTLU BİR EULER ÇÖZÜCÜSÜNE DÖNÜŞ EKLENMESİ

ÖZDEMİR, Enver Doruk

Yüksek Lisans, Makina Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. M. Haluk AKSEL

Eylül 2005, 91 sayfa

Bu çalışmanın amacı dönen ve salınım yapan kanatçıklar üzerindeki zamana bağımlı aerodinamik akışı çözmektir. Bu sayede, türbinlerin ve pervanelerin aerodinamiğini anlamak kolaylaşacaktır.

Bu çalışmada, daha önceden Orta Doğu Teknik Üniversitesi (ODTÜ) Makina Mühendisliği Bölümü'nde iç akışlar için yazılmış olan düzenli çözüm ağı, 2 boyutlu ve sonlu hacim metodu kullanılmış Euler çözücüsü geliştirilmiştir.

Bahsedilen geliştirme bu çözücüye dış akışları çözer hale getirmeyi; bu çözücüye Roe'nun uzayda birinci derece akı ayırım yöntemini uygulamayı, son olarak da dönüş etkisini keyfi Lagrangian Eulerian yöntemiyle eklemeyi içerir.

Ortaya çıkarılan kodun zamana bağımlı ve zamana bağımlı olmayan aerodinamik koşullarda doğrulanması için daha önceki yapıtlardaki deneysel ve hesaplamalı sonuçlar kullanılmıştır. Zamana bağımlı olmayan aerodinamik koşullardaki doğrulama için, sınaama örnekleri ses-altı ve ses seviyesi hızlarında ve değişik hücum açılarında incelenmiştir. Kodun zaman bağımlı aerodinamik kısmının doğrulanması için salınım yapan kanatçık örneği kullanılmıştır.

Akış, ağdasız, zaman bağımlı, ısı aktarımsız ve 2 boyutlu kabul edilmiştir. Yer çekimi yok sayılmış ve hava mükemmel gaz kabul edilmiştir.

Geliştirilen kod, ODTÜ Makina Mühendisliği Bölümündeki Hesaplamalı Akışkanlar Dinamiği – Yüksek Başarımlı Hesaplama (HAD-YBH) Laboratuvarında koşturulmuştur.

Anahtar Kelimeler: HAD, Sonlu Hacim Metodu, keyfi Lagrangian Eulerian, zaman bağımlı aerodinamik, dönüş, salınım yapan kanatçık

“To know the road ahead, ask those coming back.”
-Chinese Proverb

I dedicate this work to all, who have bothered to answer even my tiniest question.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my supervisor Prof. Dr. M. Haluk AKSEL and Dr. Oktay GÖNÇ for their continuous guidance and support.

I also want to thank Asst. Prof. Dr. Cüneyt SERT and Prof. Dr. Zafer DURSUNKAYA for their suggestions and comments about my thesis.

I express my sincere appreciation to Asst. Tahsin Çağrı ŞİŞMAN for being an excellent teammate. Without his support, this study will be unbearable.

I offer my sincere thanks to Serkan KASAPOĞLU, Etkin ÖZEN, Mehmet Ali PİŞKİN and Nihan TANIŞER for their precious support and sharing.

Finally, I gratefully thank my family for their ever lasting support.

TABLE OF CONTENTS

PLAGIARISM.....	iii
ABSTRACT.....	iv
ÖZ.....	vi
DEDICATION.....	viii
ACKNOWLEDGMENTS.....	ix
TABLE OF CONTENTS.....	x
LIST OF TABLES.....	xii
LIST OF FIGURES.....	xiii
LIST OF SYMBOLS.....	xvii
CHAPTERS	
1) INTRODUCTION.....	1
1.1 GENERAL.....	1
1.2 REVIEW OF LITERATURE.....	2
1.3 PRESENT STUDY.....	11
2) GOVERNING EQUATIONS.....	13
2.1 EULER EQUATIONS.....	13
2.2 EULER EQUATIONS FOR MOVING MESH.....	15
3) NUMERICAL METHODS.....	17
3.1 PHYSICAL AND COMPUTATIONAL DOMAIN.....	17
3.2 DATA STRUCTURE.....	20
3.3 CELL GEOMETRY.....	22
3.4 SPATIAL DISCRETIZATION.....	23
3.5 TEMPORAL DISCRETIZATION.....	28

3.6 INITIAL CONDITIONS.....	30
3.7 BOUNDARY CONDITIONS.....	31
3.8 THE GRID MOVEMENT.....	35
3.9 CONVERGENCE CRITERIA FOR STEADY FLOWS.....	38
4) VERIFICATION OF THE CODE, RESULTS AND DISCUSSION.....	40
4.1 STEADY SOLUTIONS.....	40
4.2 UNSTEADY SOLUTIONS.....	60
5) CONCLUSION.....	78
REFERENCES.....	81
APPENDICES	
A) SAMPLE INPUT FILES.....	85
B) AIRFOIL PROFILES.....	88
C) CALCULATION OF PRESSURE, LIFT AND DRAG COEFFICIENT.....	91

LIST OF TABLES

Table 4.1 Summary of Steady Test Cases.....	41
Table 4.2 Grid types for subsonic flow over symmetric airfoil (case 4.1.1).....	41
Table 4.3 Grid types for transonic flow over symmetric airfoil (case 4.1.2).....	51
Table 4.4 Grid types for Test Case 4.1.3.....	55
Table 4.5 Properties of Unsteady Test Case.....	60
Table 4.6 Summary of oscillating airfoil Test Cases.....	61
Table 4.7 Grid types for the oscillating airfoil.....	62
Table B.1 The Geometry of NACA 0012 profile.....	88
Table B.2 The Geometry of RAE 2822 profile.....	89

LIST OF FIGURES

Figure 1.1	The method of “rotating frame of reference”	7
Figure 1.2	1-D representation of Riemann approach.....	10
Figure 3.1	Physical domain for O-grid solver.....	18
Figure 3.2	Computational domain for O-grid solver.....	18
Figure 3.3	Physical domain for C-grid solver.....	19
Figure 3.4	Computational domain for C-grid solver.....	20
Figure 3.5	The structured mesh with cell centered scheme.....	21
Figure 3.6	The geometry of cell (I,J)	22
Figure 3.7	Solid wall boundary for a stationary grid (O-grid solver).....	32
Figure 3.8	Geometry for grid velocities.....	36
Figure 4.1	Overall view for NACA 0012 profile O-grid with (121x121) nodes.....	42
Figure 4.2	Overall view for NACA 0012 profile C-grid with (211x121) nodes.....	42
Figure 4.3	Close-up view of NACA 0012 profile O-grid with (121x121) nodes.....	43
Figure 4.4	Close-up view of NACA 0012 profile O-grid with (243x257) nodes.....	43
Figure 4.5	Close-up view of NACA 0012 profile C-grid with (211x121) nodes.....	44
Figure 4.6	Residual history for NACA 0012 (121 x 121) nodes, $M_\infty=0.60$ $\alpha=0.0$	45
Figure 4.7	Residual history for NACA 0012 (243 x 257) nodes, $M_\infty=0.60$ $\alpha=0.0$	45

Figure 4.8	Residual history for NACA 0012 C-grid (211 x 121) nodes, $M_\infty=0.60$ $\alpha=0.0$	46
Figure 4.9	Figure 4.0.9 c_p distribution over the NACA 0012 airfoil, $M_\infty=0.60$ $\alpha=0.0$	47
Figure 4.10	Mach Contour, NACA0012 airfoil (243x257) nodes, $M_\infty=0.60$ $\alpha=0.0$	48
Figure 4.11	c_p contour, NACA0012 airfoil (243x257) nodes, $M_\infty=0.60$ $\alpha=0.0$	48
Figure 4.12	Mach Contour and Velocity vectors, NACA0012 airfoil Leading edge close-up view (243x257) nodes, $M_\infty=0.60$ $\alpha=0.0$	49
Figure 4.13	Close-up view of NACA0012 profile O-grid with (101x201) nodes.....	50
Figure 4.14	Residual history for NACA 0012 (121 x 121) nodes, $M_\infty=0.80$ $\alpha=0.0$	51
Figure 4.15	Residual history for NACA 0012 (101 x 201) nodes, $M_\infty=0.80$ $\alpha=0.0$	52
Figure 4.16	c_p distribution over the NACA 0012 airfoil, $M_\infty=0.80$ $\alpha=0.0$...	53
Figure 4.17	Mach Contour, NACA0012 airfoil, $M_\infty=0.80$ $\alpha=0.0$, Solution of Wu and Li.....	54
Figure 4.18	Mach Contour, NACA0012 airfoil (101x212) nodes, $M_\infty=0.80$ $\alpha=0.0$	54
Figure 4.19	Close-up view of RAE 2822 profile O-grid with (801x301) nodes.....	56
Figure 4.20	Close-up view of RAE 2822 profile C-grid with (601x121) nodes.....	56

Figure 4.21	Residual history for RAE 2822 profile O-grid with (801x301), $M_\infty=0.7290$ $\alpha=2.310$	57
Figure 4.22	Residual history for RAE 2822 profile C-grid with (601x121), $M_\infty=0.7290$ $\alpha=2.310$	57
Figure 4.23	c_p distribution over RAE 2822 profile, $M_\infty=0.7290$ $\alpha=2.310$	58
Figure 4.24	Mach Contour, RAE 2822 profile, $M_\infty=0.7290$ $\alpha=2.310$, Solution of Kudinov [32].....	59
Figure 4.25	Mach Contour, RAE 2822 profile, O-grid 801x301 nodes, $M_\infty=0.7290$ $\alpha=2.310$, present computation.....	59
Figure 4.26	Oscillating Airfoil Test cases in the sinusoidal oscillation cycle.....	62
Figure 4.27	Numerical mesh of NACA0012 for oscillating airfoil problem...	63
Figure 4.28	Instantaneous c_p distribution, $\alpha = 1.090^\circ$, case 4.2.a.....	64
Figure 4.29	Instantaneous c_p distribution, $\alpha = 2.340^\circ$, case 4.2.b.....	65
Figure 4.30	Instantaneous c_p distribution, $\alpha = 2.010^\circ$, case 4.2.c.....	66
Figure 4.31	Instantaneous c_p distribution, $\alpha = 0.520^\circ$, case 4.2.d.....	67
Figure 4.32	Instantaneous c_p distribution, $\alpha = -1.250^\circ$, case 4.2.e.....	68
Figure 4.33	Instantaneous c_p distribution, $\alpha = -2.410^\circ$, case 4.2.f.....	69
Figure 4.34	Instantaneous c_p distribution, $\alpha = -2.000^\circ$, case 4.2.g.....	70
Figure 4.35	Instantaneous c_p distribution, $\alpha = -0.540^\circ$, case 4.2.h.....	71
Figure 4.36	Instantaneous Mach contours, $\alpha = 1.090^\circ$, case 4.2.a.....	72
Figure 4.37	Instantaneous Mach contours, $\alpha = 2.340^\circ$, case 4.2.b.....	72
Figure 4.38	Instantaneous Mach contours, $\alpha = 2.010^\circ$, case 4.2.c.....	73
Figure 4.39	Instantaneous Mach contours, $\alpha = 0.520^\circ$, case 4.2.d.....	73
Figure 4.40	Instantaneous Mach contours, $\alpha = -1.250^\circ$, case 4.2.e.....	74
Figure 4.41	Instantaneous Mach contours, $\alpha = -2.410^\circ$, case 4.2.f.....	74

Figure 4.42	Instantaneous Mach contours, $\alpha = -2.000^\circ$, case 4.2.g.....	75
Figure 4.43	Instantaneous Mach contours, $\alpha = -0.540^\circ$, case 4.2.h.....	75
Figure 4.44	Comparison of C_L versus Angle of Attack values.....	76

LIST OF SYMBOLS

a	integer
$\hat{\mathbf{a}}$	eigenvalues vector
$ \hat{\mathbf{a}}_k ^*$	modified eigenvalues vector
\bar{c}_p	specific heat at constant pressure
\bar{c}_v	specific heat at constant volume
c	speed of sound
c_p	pressure coefficient
c_D	drag coefficient
c_L	lift coefficient
CFL	Courant Friedrichs Lewy coefficient
e	total energy per unit mass
f_j	the interpolation function of FEM
$\vec{\mathbf{F}}$	flux vector
F_1	magnitude of flux vector in x direction
F_2	magnitude of flux vector in y direction
F_{face}	magnitude of flux at the defined face
H	total enthalpy per unit mass.

\vec{i}	unit vector in x direction
\vec{i}_n	unit vector in normal direction
\vec{i}_t	unit vector in tangential direction
\vec{j}	unit vector in y direction
L	chord length
M	Mach number
NX	maximum number of nodes in x direction
NY	maximum number of nodes in y direction
p	static pressure
P_o	stagnation pressure
Q_k	any conserved property
\vec{Q}	vector of conservative variables
\vec{r}	position vector relative to the rotating frame of reference
R	universal gas constant
$\hat{\mathbf{R}}_k$	matrix for flux calculation
$\mathbf{R}_{(I,J)}$	residual term for the cell (I,J)
$R1^n$	calculated residual for mass conservation at the n th time step

$R2^n$	calculated residual for x momentum conservation at the n th time step
$R3^n$	calculated residual for y momentum conservation at the n th time step
$R4^n$	calculated residual for energy conservation at the n th time step
$RS1$	normalized residual for mass conservation
$RS2$	normalized residual for x momentum conservation
$RS3$	normalized residual for y momentum conservation
$RS4$	normalized residual for energy conservation
\bar{S}	normal area vector
SS	node number of trailing edge in C-grid
SP	node number of leading edge in C-grid
$S_{(i,j)}^{kx}$	magnitude of normal area vector at k th face in x direction in (i,j) th cell
$S_{(i,j)}^{ky}$	magnitude of normal area vector at k th face in y direction in (i,j) th cell
Δs	the length of a face
t	time
Δt	time increment
Δt^{Global}	global time increment

T	temperature
T_o	stagnation temperature
u	magnitude of fluid velocity in x direction
u^n	magnitude of fluid velocity vector in normal direction
u^t	magnitude of fluid velocity vector in tangential direction
u_{CV}	magnitude of grid velocity in x direction
u_{CV}^n	magnitude of grid velocity in normal direction
u_{CV}^t	magnitude of grid velocity in tangential direction
\vec{V}_{rel}	velocity relative to the rotating coordinate
\vec{V}	absolute velocity of fluid
\forall	volume of a cell
v	magnitude of fluid velocity in y direction
v_{CV}	magnitude of grid velocity in y direction
$\Delta \mathbf{V}$	vector for flux calculation
\vec{W}	grid velocity vector
x, y, z	coordinates of absolute reference frame
$x_{(i,j)}$	x coordinate of the (i,j) th node
$x_{cellcenter(I,J)}$	x coordinate of the (i,j) th cell center

$y_{(i,j)}$	y coordinate of the (i,j) th node
$y_{cellcenter(I,J)}$	y coordinate of the (i,j) th cell center

Greek Symbols

α	angle of attack
$\dot{\alpha}$	predefined arbitrary angular velocity for grid
β	base angle
ϕ	any property
$\hat{\phi}$	weighted average at the cell face of any property
ρ	density
$\overline{\omega}$	angular velocity

Subscripts

x	derivative of a variable with respect to x
n	iteration number
$Cell(I,J)$	values at the (I,J) th cell
$cellcenter(I,J)$	values at the (I,J) th cell center

(i, j)	values at the (I,J) th node
L	values at the left cell
R	values at the right cell
ALE	values for ALE method
$ghost$	values at the ghost cell
∞	freestream values

Superscript

k	determines the face of area vector
$farfield$	values at the far field boundary
$solid$	values at the solid boundary

CHAPTER 1

INTRODUCTION

*“There is no such thing as a long piece of work,
except one that you dare not start.”*

Charles Baudelaire

*“‘We must do something’ is the unanimous refrain.
‘You begin’ is the deadening refrain.”*

Walter Dwight

1.1 General

There are three different sources of information for making predictions about fluid flows. Analytic methods give the exact solution, however one can deal with difficult partial differential equations, most of which are even impossible to solve. Experimental studies give the *real* solution, nonetheless they are very costly. Furthermore, experimentation can be practically impossible for large scale applications like aircrafts and wind turbines. One of the solutions to this problem may be performing experiments with scaled down model of the prototype. But this will bring additional difficulty in similarity studies. The last method is the computational method. By this method, basic laws of the nature are solved numerically. Computational Fluid Dynamics (CFD) is the simulation of physical properties involving fluid flow, heat and mass transfer, and

chemical reactions through computer modeling. If there are no chemical reactions in the physical flow, then only mass conservation, momentum conservation and energy conservation laws are used. The drawback of computational methods is that they need verification with either experimental data or analytical solutions. [1]

CFD is used in this study. To validate the results, experimental results or computational results of previous researches are utilized.

1.2. Review of Literature

Any CFD code needs three main structural elements; namely pre-processor, solver and post-processor.

1.2.1. Pre-processor

First of all, the geometry of interest should be defined and this domain should be divided into smaller sub-domains, which could be thought as control volumes where the basic laws of nature will be applied separately on each of them. The grid, or mesh, is composed of these small control volumes (or cells). [2]

Generally speaking, the computation time and accuracy increase as the grid becomes finer. But after a limit, the enhancements in the accuracy will not compensate the increase in the computation time. The aim in CFD codes is to reach that limit where the accuracy and the time are optimized.

There are two major types of grids: Structured and unstructured grids. Hybrid grids are (as the name implies) the co-existence of both. Structured-body fitted grids are used in this study. In this type, each node has four neighbors in 2-D and six neighbors in 3-D. [3]. General known advantage of structured grid other than its simplicity is that neighbor

cell indices are inherently known for every cell. This feature reduces the memory requirement.

The major drawback of the structured grids is that the grid generation process is difficult for complex geometries. Another disadvantage is that, increasing the intensity of the mesh in a particular part of the domain, also increases the intensity in other places which are not that critical, in order to maintain the structure. As a result, the number of cells and thus, computation time is increased inefficiently [4].

Pre-processing also includes the definition of fluid properties, specification of appropriate boundary conditions and selection of the physical and chemical phenomena [2].

1.2.2. Solver

After the pre-processing part, the solver should be analyzed. There are four different approaches to solvers. These are Finite Difference Method (FDM), Finite Element Method (FEM), Spectral Method (SM), and lastly Finite Volume Method (FVM).

1.2.2.1 Solver Approaches

Finite difference methods use truncated Taylor series expansions and the simple definition of the derivative. For any property ϕ , the derivative is defined as below:

$$\phi_x = \left(\frac{\partial\phi}{\partial x}\right) = \lim_{\Delta x \rightarrow 0} \frac{\phi(x + \Delta x) - \phi(x)}{\Delta x} \quad (1.1)$$

For finite but small $\Delta\phi$, the above equation turns to an approximation. Actually the error is the truncation error of the Taylor series expansion. [5] All the derivatives in the

governing equations are expressed in terms of finite differences. [2] The application of this method is easy, but it needs highly regular mesh. [4]

Finite element method is originally developed for structural stress analysis, but it can also be used for fluid mechanics. This method is more mathematically based and uses the principle that the field variables can be approximated by linear combinations of simple piecewise functions locally [2, 5].

In 1-D domain with N nodes, $\bar{\phi}$, which is the approximation of any property ϕ , is shown below:

$$\phi \approx \bar{\phi} = \sum_{i=1}^N \phi_i f_i \quad (1.2)$$

Standard FEM uses locally defined polynomials for the interpolation functions f_i . Each interpolation function has a zero value outside its element. The aim is to find the local coefficients ϕ_i so that the physical property ϕ can be approximated. [5]

The difference of SM and FEM is that the approximation functions are truncated Fourier series or series of Chebyshev polynomials, and defined over the whole domain in SM. However, the use of SM is limited in aerodynamics. [2, 5]

FVM is developed as a special case of FDM. However, it gained much popularity due to the fact that FVM is more physically based. That is to say, each term in the calculation represent a physical phenomenon. Most of the commercial CFD package programs use FVM.

In any FVM code, first, the governing equations are integrated over the control volumes (cells). Secondly, the discretisation is applied by representing all the physical terms of the flow (like convection, diffusion and sources) as finite difference approximations. Lastly, the unknown properties in each cell are solved simultaneously and iteratively.[2]

The basic idea behind the FVM is that all the governing equations represent the conservation of some property, which can be stated as below.

$$\left(\begin{array}{c} \text{Rate of change of } \phi \\ \text{in the Control Volume} \\ \text{with respect to time} \end{array} \right) = \left(\begin{array}{c} \text{Net flux of } \phi \text{ due to} \\ \text{convection into the} \\ \text{Control Volume} \end{array} \right) + \left(\begin{array}{c} \text{Net flux of } \phi \text{ due to} \\ \text{diffusion into the} \\ \text{Control Volume} \end{array} \right) + \left(\begin{array}{c} \text{Net rate of} \\ \text{creation of} \\ \phi \text{ inside the} \\ \text{Control Volume} \end{array} \right)$$

(1.3)

1.2.2.2 Governing Equations

Fluid motion is described by Navier-Stokes equations for those flows that satisfy continuum postulate. However, as the viscosity is neglected in this work, Navier-Stokes equations reduce to Euler equations. Euler equations describe an unsteady, inviscid and compressible flow. Mathematically speaking, these equations are hyperbolic equations. Here, the characteristics are real and distinct. However, if the Euler equations are simplified to steady and subsonic flow, then the nature turns to elliptic, where the characteristics are imaginary or complex. [3]

Although there are steady flow cases in this study, the flow is considered unsteady and steadiness is considered to be achieved when the terms including time are getting

sufficiently small. Thus, in this study the mathematical nature of governing equations are always hyperbolic.

1.2.2.3 Boundary Conditions

Thompson [6] states that *“The purpose of boundary conditions is to supply whatever information is needed at the boundaries of the computational volume in order to complete the definition of the behavior of the system.”*

The needed information could be a physical or a numerical boundary condition. A physical boundary condition (BC) specifies a previously known physical variable at the boundary. If these physical variables are not enough to define the behavior of the system then some numerical BC's should be used. The simplest way to handle numerical BC's is to extrapolate the needed physical variables. However, one can also use the conservation equations on the boundaries to obtain the unknown quantities. [7]

The boundary conditions can be calculated or taken from the physics of the problem, as stated above, at the boundary. However, another method is to define “ghost” cells at the boundaries. Dadone et al. [8] call this method as symmetry technique. The advantage of this method is that one should only define the values of the conserved variables (or primitive variables) at the “ghost” cells, and the flux calculation is the same as an interior cell.

To define the conservative variables at the “ghost” cells, one could implement characteristic Riemann invariants as in the references [4, 7, 6 and 9]. However, this superior method is not used in this study. Instead, a much easier method is utilized. The detailed explanation of all BC's for steady and unsteady solvers are present in Chapter 3.

1.2.2.4 Implementation of Rotation in Solver

There are two methods to implement rotation. Either rotational frame of reference should be used or the mesh should be rotated.

The method of “rotating frame of reference” is used for a steadily rotating system with angular velocity of $\vec{\omega}$, \vec{V}_{rel} is the velocity relative to the rotating coordinate and \vec{V} is the absolute velocity of fluid. [5, 10]

$$\vec{V} = \vec{V}_{rel} + (\vec{\omega} \times \vec{r}) \quad (1.4)$$

It is obvious that \vec{r} is the position vector relative to the rotating frame of reference. The Figure 1.1 should make the concept much clearer.

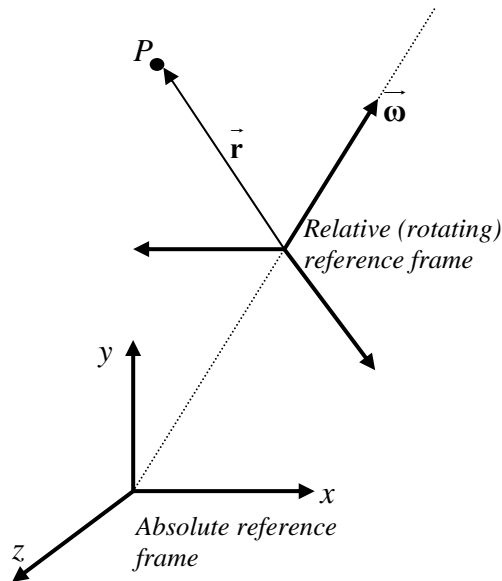


Figure 1.1 The method of “rotating frame of reference”

The mass conservation equation does not change since the entrainment velocity does not contribute to the mass balance. However, the *Coriolis* and *centrifugal* forces should be included into the momentum equation. For the energy equation, only modification comes from work done by *centrifugal* forces, since *Coriolis* forces do not contribute to energy equation. [5]

The second approach is to rotate the mesh, rather than having a non-inertial rotating reference frame. This goal could be achieved by utilizing “Arbitrary Lagrangian Eulerian” (ALE) method. This method is first developed by Hirt et al. [11] for a FDM solver. Many codes with ALE followed this publication, due to its simplicity.

It is well known from fluid mechanics text books that there are two classical description of fluid flow: Eulerian and Lagrangian. [12, 13]

In Lagrangian approach, the attention is on the fluid particle. So identified fluid particles are traced. The difficulty here is that every fluid particle should be traced in order to have a full understanding of the flow field. Therefore, this method is much more suitable to solid mechanics, rather than fluid mechanics. Following fluid particles would mean that the control volume is moving at the fluid particle velocity. [12, 13]

On the other hand, Eulerian approach focuses on a fixed point in the space. The changes of the flow variables are investigated on that particular point. This would mean having a stationary control volume. [12, 13]

Eulerian approach is frequently used in fluid mechanics. If an engineer is going to investigate, for example, the flow around an aircraft, the method is to “*stop*” the aircraft, and let the air “*blow*” onto it. This approach can be used for analytical, experimental and also for computational studies. This is due to relative simplicity of this approach.

In ALE, the focus is neither a fixed point in space, nor a fluid particle. The “*attention*” or the control volume is moving arbitrarily; not stationary (Eulerian) or not at the particle velocity (Lagrangian). The governing equations are presented in Chapter 2. [11, 14]

The advantage of ALE method is that, with ALE, it is convenient to simulate moving boundary problems or deforming structure problems. [4, 15]

Although the history of ALE is dating back to 1960’s and it is presented in [16], some examples from literature will be mentioned in this review. Batina [17] uses this method to solve the flow around deforming aircraft. Michler et al. [18] utilizes ALE to simulate flows with rotating (non-deforming) meshes and also deforming meshes. Uzun et al. [15, 19] concentrate on 3-D oscillating aircraft wings. Darlington [16] uses Large Eddy Simulation together with ALE. Finally, Gönç [4] presented a compressible 3-D flow around rotating missile.

Most of the above researchers [18, 20 and 14] are concerned about the Geometric Conservation Law (GCL) which is explained in Chapter 3 when implementing ALE. The general consensus among the researchers is that GCL should be satisfied.

1.2.2.5 Spatial Discretization Methods for Solvers

There are many spatial discretisation methods. In this study, Flux Difference Splitting (Godunov) type discretisation is used. This type is also referred to as Riemann approach, since the Riemann problem (or the shock tube problem) is solved on each cell face. However, as the exact solution of the Riemann problem is computationally expensive, some approximate Riemann solvers are developed. [21]

In this study, Roe's upwind flux differencing scheme is used. [22] In this approximate Riemann solver scheme, a piecewise constant distribution of the data is assumed. Figure 1.2 presents the situation for a 1-D problem. The Q_k values represent any conserved property. The main task in this scheme is to find the values of the conserved variables at the cell faces, since the face fluxes should be calculated according to those values. [21]

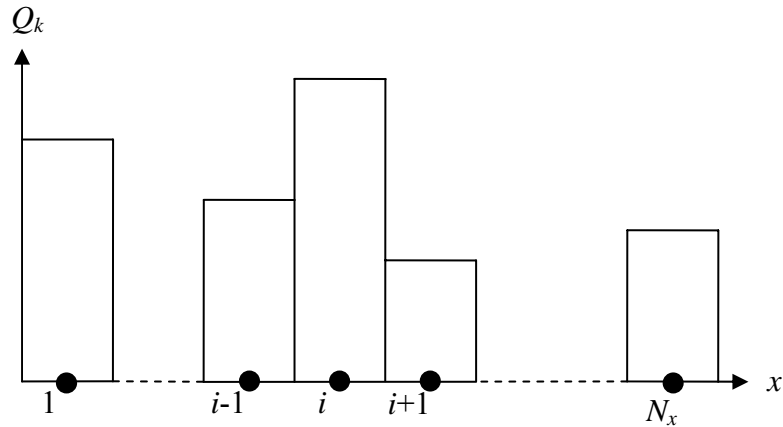


Figure 1.2 1-D representation of Riemann approach

This scheme has gained popularity due to its accuracy and cheapness in computation. The definitions for any conserved variable at the cell faces are given in references [4, 22 and 23] and will be presented in Chapter 3.

In this study, first order schemes are used. These are the simplest type of schemes. Hosseini et al. [24] summarizes the high order schemes briefly. Implementing a higher order scheme into a solver, results generally with a better shock capturing.

1.2.2.6 Temporal Discretization Methods for Solvers

Temporal discretisation can be done by two methods; namely implicit and explicit. Both methods have a wide range of usage. [4]

In explicit methods, all fluxes and sources of the instant t_{n+1} are calculated at the known instant of t_n [3]. The explicit solving technique requires considerably less computer memory. However, the drawback of this method is that the time step Δt is limited with the stability. That means large time steps could yield unstable solutions. *CFL* (Courant-Friedrichs-Lewy) condition helps to overcome this instability problem.

Implicit methods should be used for the prime concern of stability [3]. In implicit approach, the residual term is evaluated at the time step t_{n+1} . This method will bring unfortunately a complicated algorithm and excessive memory requirement.

1.2.3. Postprocessor

Finally, postprocessors are needed to complete any CFD analysis. It is used for data visualization. The outcomes of the CFD code will be sent to the postprocessor and the domain, the grid, any computed vectors or scalars can be displayed. [2]

1.3. Present Study

The aim of this study is to simulate the unsteady flow around rotating or oscillating airfoils. This will help to understand the rotor aerodynamics, which is essential in turbines and propellers.

In this study, a pre-existing Euler solver “Euler2d” with finite volume method that is developed by Prof. Dr. M. Haluk Aksel is improved. This structured pre-existing code was developed for 2-D internal flows with Lax-Wendroff scheme. First of all, first order Roe’s flux splitting scheme is implemented, which requires cell centered approach, rather than vertex centered one of Lax-Wendroff’s. This implementation is done by a joint study with Tahsin Çağrı Şişman. During the development of this code, “Set2D” code developed by Dr. Mehmet Ali Ak [25] is taken as basis.

Then the code is modified to handle external flows. The external Euler code is capable of handling C-type and O-type meshes. The third modification is the implementation of rotation with the help of Arbitrary Lagrangian Eulerian (ALE) method.

All of the grids used in this study are prepared with the GENGRID code version 1.1, [26] which is a 2D structured grid generation program with algebraic and elliptic features. For digitizing the data from different sources, WinDIG Version 2.0 [27] is used. By this way, the experimental or numerical data from papers and technical reports are digitized.

The developed code is run on computers housed in METU Mechanical Engineering Department CFD-HPC Laboratory.

This thesis is divided into five chapters. Chapter 2 defines the assumptions of the flow and gives the governing equations for stationary and moving grid. Chapter 3 deals with numerical methods, which are used in this research. Furthermore, description of the physical and the computational domain and the boundary conditions are also described in this chapter. Chapter 4 aims to verify the present steady and unsteady solver by comparing the results with the literature, either with experimental or computational data. Finally, Chapter 5 presents the conclusion and suggests some future work that could improve this study.

CHAPTER 2

GOVERNING EQUATIONS

"The pure and simple truth is rarely pure and never simple"
Oscar Wilde

*"Natural Science does not simply describe and explain nature;
it is part of the interplay between nature and ourselves;
it describes nature as exposed to our method of questioning."*
Werner Heisenberg

2.1 Euler Equations

Euler equations are a collection of mass, momentum and energy conservation laws. The system of equations below is valid for inviscid, unsteady, adiabatic and two dimensional flows, neglecting gravity. Additionally air is taken as an ideal gas.

Conservation of mass, momentum and energy equations are given in Equations (2.1), (2.2), and (2.3), respectively.

$$\frac{\partial \rho}{\partial t} + \vec{\nabla} \cdot (\rho \vec{V}) = 0 \quad (2.1)$$

$$\frac{\partial(\rho\vec{V})}{\partial t} + \vec{\nabla} \cdot (\rho\vec{V}\vec{V} + p) = 0 \quad (2.2)$$

$$\frac{\partial(\rho e)}{\partial t} + \vec{\nabla} \cdot ((\rho e + p)\vec{V}) = 0 \quad (2.3)$$

where ρ is density, \vec{V} is the velocity vector, p is the static pressure, and e is the total energy per unit mass and defined as

$$\rho e = \frac{p}{(\gamma - 1)} + \rho \left(\frac{|\vec{V}|^2}{2} \right) \quad (2.4)$$

where γ is the specific heat ratio and defined as the ratio of the specific heat at constant pressure, \bar{c}_p to the specific heat at constant volume, \bar{c}_v .

For practical purposes, all of the above equations can be combined in a single set of equations as below:

$$\frac{\partial}{\partial t} \int_{\forall} \vec{Q} d\forall + \oint_s \vec{F} \cdot d\vec{S} = 0 \quad (2.5)$$

where \vec{Q} is the vector of conservative variables and can be given as

$$\vec{Q} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho e \end{bmatrix} \quad (2.6)$$

and $\vec{F} = F_1 \vec{i} + F_2 \vec{j}$ is consisting of flux terms

$$F_1 = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (\rho e + p)u \end{bmatrix} \quad \text{and} \quad F_2 = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (\rho e + p)v \end{bmatrix} \quad (2.7)$$

where $\vec{V} = u\vec{i} + v\vec{j}$

The presented Euler equations contain four equations but five unknowns ρ, u, v, p, e . Therefore, Equation (2.4) is used to close this system of equations. Temperature can be found from ideal gas equation:

$$p = \rho RT \quad (2.8)$$

2.2 Euler Equations for Moving Mesh

Euler equations presented above are using the Eulerian approach, which focuses on a fixed point in the space and traces the changes in the flow variables at that particular point which implies a stationary control volume. On the other hand, for an arbitrary moving control volume, the governing equations will change to include the effect of the grid motion as indicated below.

$$\frac{\partial}{\partial t} \int_{\forall} \vec{Q} d\forall + \oint_S \vec{F}_{ALE} \cdot d\vec{S} = 0 \quad (2.9)$$

where \vec{Q} is the vector of conservative variables and remains the same as in equation (2.6). Furthermore $\vec{F}_{ALE} = F_{ALE_1} \vec{i} + F_{ALE_2} \vec{j}$ is consisting of flux terms as below

$$F_{ALE_1} = \begin{bmatrix} \rho(u - u_{CV}) \\ \rho u(u - u_{CV}) + p \\ \rho(u - u_{CV})v \\ \rho e(u - u_{CV}) + pu \end{bmatrix} \quad \text{and} \quad F_{ALE_2} = \begin{bmatrix} \rho(v - v_{CV}) \\ \rho u(v - v_{CV}) \\ \rho v(v - v_{CV}) + p \\ \rho e(v - v_{CV}) + pv \end{bmatrix} \quad (2.10)$$

where $\vec{V} = u\vec{i} + v\vec{j} = u^n\vec{i}_n + u^t\vec{i}_t$ is the fluid velocity, and

$\vec{W} = u_{CV}\vec{i} + v_{CV}\vec{j} = u_{CV}^n\vec{i}_n + u_{CV}^t\vec{i}_t$ is the grid velocity.

Equations (2.4) and (2.8) are still valid and are required to close the system of equations as in the previous case.

CHAPTER 3

NUMERICAL METHODS

“The formulation of a problem is often more essential than its solution, which may be merely a matter of mathematical or experimental skill.”

Albert Einstein

3.1 Physical and Computational Domain

3.1.1 O-grid solver

The physical domain is selected as a circle with a radius of 15 chord lengths of the airfoil for O-grid solver. Figure 3.1 presents the physical domain and indicates the boundaries. The airfoil is shown as a circle for simplicity.

The physical domain above should be converted to a computational domain, as shown in Figure 3.2. The bottom edge of the rectangle represents the solid boundary and it includes the nodes from $(1,1)$ to $(NX,1)$. The far field boundary is the top edge of the rectangle and the nodes vary from $(1,NY)$ to (NX,NY) . The left and right edges vary from nodes $(1,1)$ to $(1,NY)$ and $(NX,1)$ to (NX,NY) , respectively and they represent the same line. In other words, nodes $(a,1)$ and $(NX+1-a,1)$ are the same node assuming that a is an integer between 1 and NX .

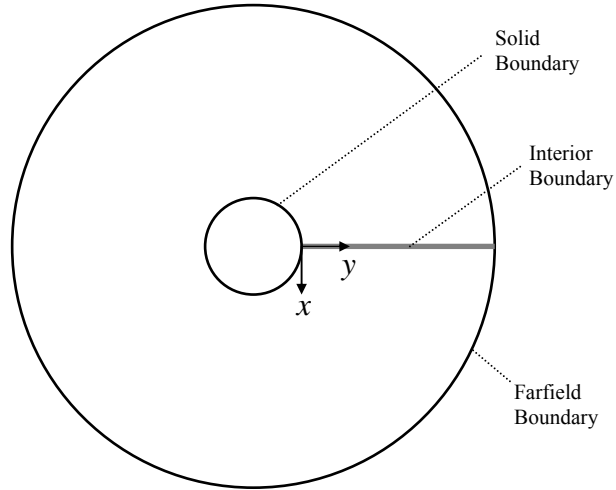


Figure 3.1 Physical domain for O-grid solver

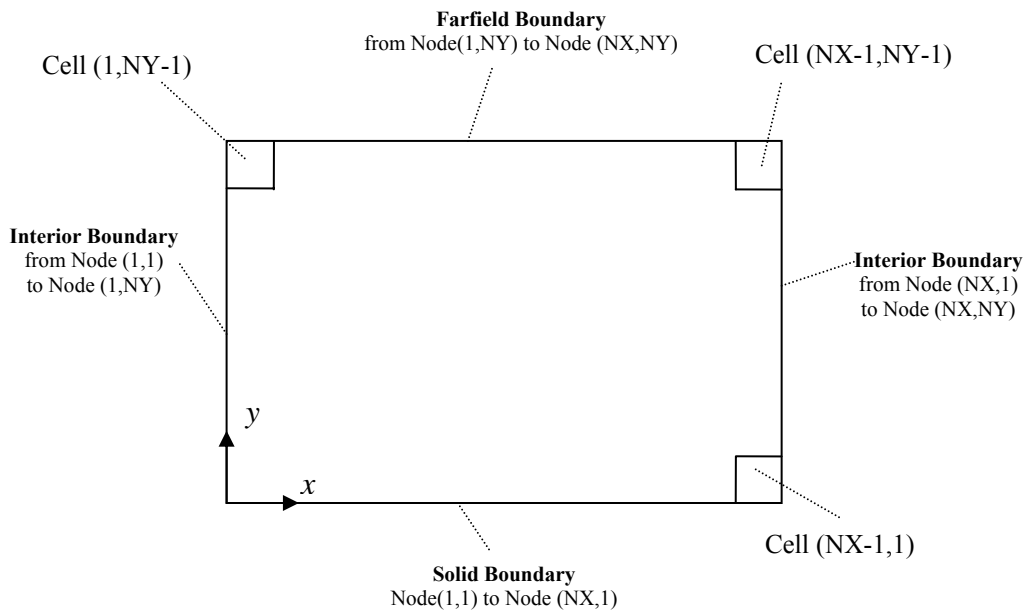


Figure 3.2 Computational domain for O-grid solver

3.1.2 C-grid Solver

The physical domain for a C-grid solver is presented in Figure 3.3. The domain is selected such that it is the combination of a half circle with a radius of 15 chord lengths and a rectangle having a width and height being equal to the diameter of the circle.

Figure 3.4 shows the computational domain for a C-grid solver. The top, left and right edges of the rectangle represent the far field boundary, while both sides of bottom edge represent the interior boundary. The node (SS,1) is the node at the trailing edge of the airfoil. Again, for both interior boundaries the nodes $(a,1)$ and $(NX-a+1,1)$ are the same nodes, providing that a is an integer between 1 and SS. The nodes between (SS,1) and $(NX-SS+1,1)$ are on solid boundary. The node (SP,1) represents the leading edge of the airfoil. The input file for C-grid solver also needs the values of SS and SP, in order to find the location of solid boundary at the bottom edge.

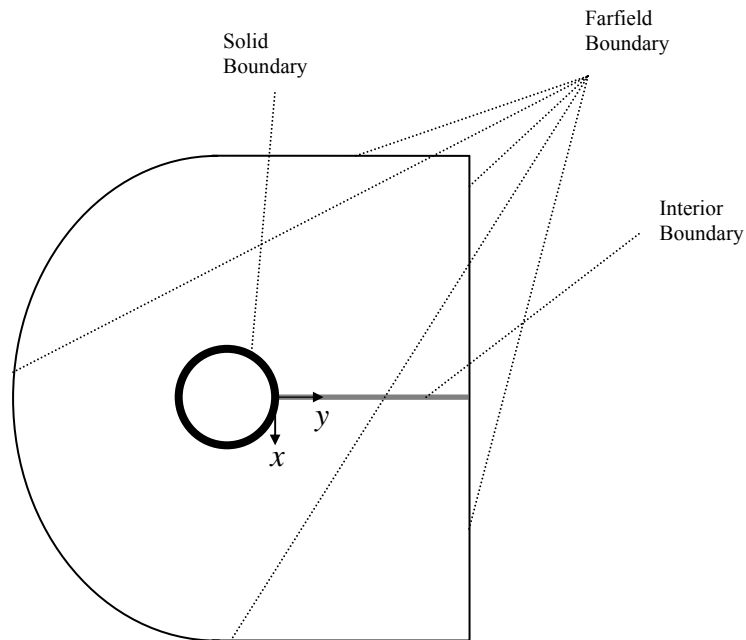


Figure 3.3 Physical domain for C-grid solver

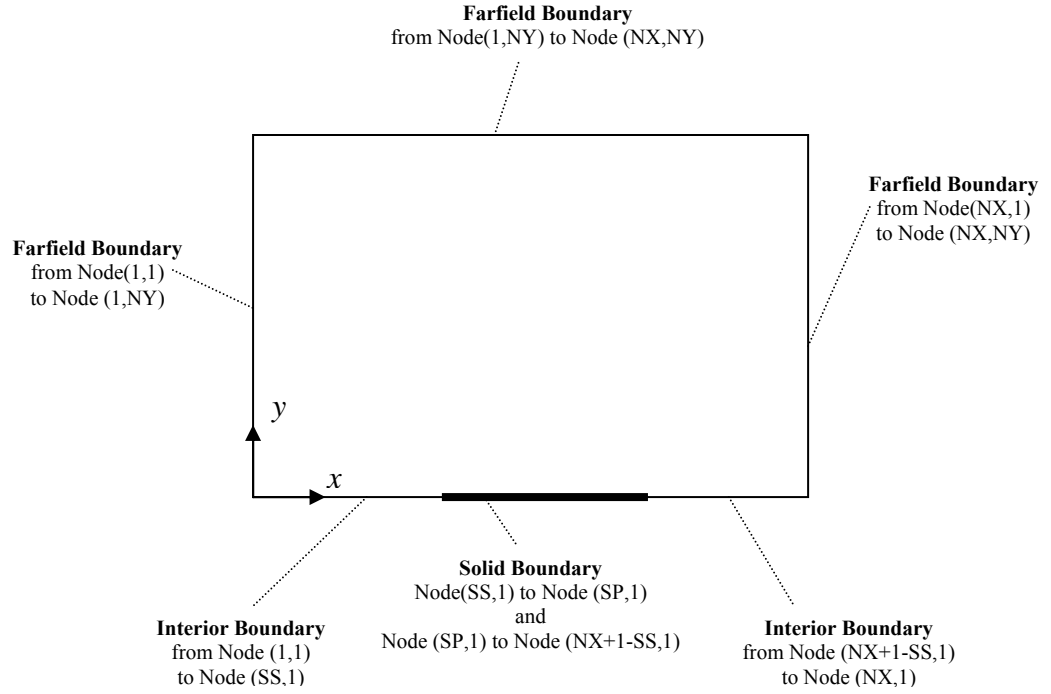


Figure 3.4 Computational domain for C-grid solver

3.2 Data Structure

Structured-boundary fitted grid is used in this study. The flow variables are located and calculated at the geometric centers of the cell. Figure 3.5 shows the structured mesh with cell centered scheme in computational domain.

The cell center location of each cell are found as below:

$$x_{cellcenter(I,J)} = \frac{x_{(i,j)} + x_{(i,j+1)} + x_{(i+1,j)} + x_{(i+1,j+1)}}{4} \quad (3.1)$$

$$y_{cellcenter(I,J)} = \frac{y_{(i,j)} + y_{(i,j+1)} + y_{(i+1,j)} + y_{(i+1,j+1)}}{4} \quad (3.2)$$

Then, the flow variables are stored at the center of the cells and the changes in the cell directly affect the flow variables in the cell.

$$\phi^{n+1}_{Cell(I,J)} = \phi^n_{Cell(I,J)} + \delta\phi^n_{Cell(I,J)} \quad (3.3)$$

After the convergence, the values at the vertices are found by arithmetic mean.

$$\phi^n_{(i,j)} = \frac{\phi^n_{Cell(I,J)} + \phi^n_{Cell(I-1,J)} + \phi^n_{Cell(I,J-1)} + \phi^n_{Cell(I-1,J-1)}}{4} \quad (3.4)$$

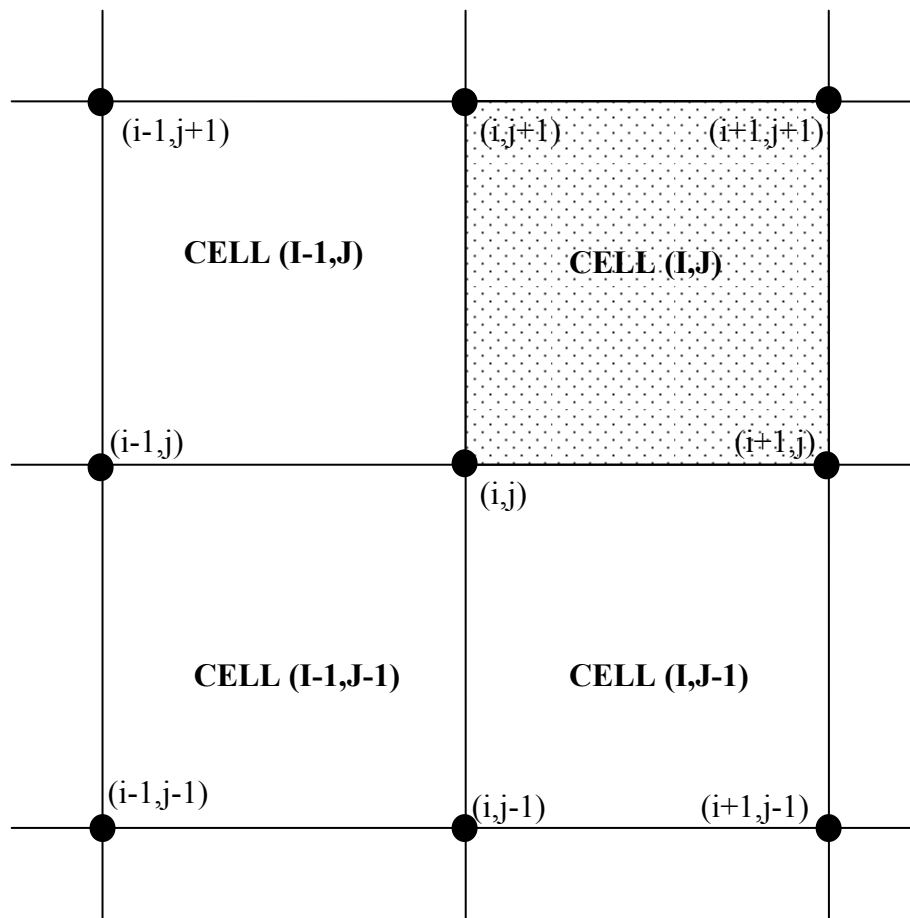


Figure 3.5 The structured mesh with cell centered scheme

The major reason for using a cell centered scheme is that these schemes are more convenient when flux vector splitting is employed.

Mesh input of the code includes the number of nodes for each direction and the position of each node. Sample input files are presented in Appendix A.

3.3 Cell Geometry

Figure 3.6 presents the geometry of an arbitrary cell. The normal vectors for every face of each cell and the area of each cell should be calculated.

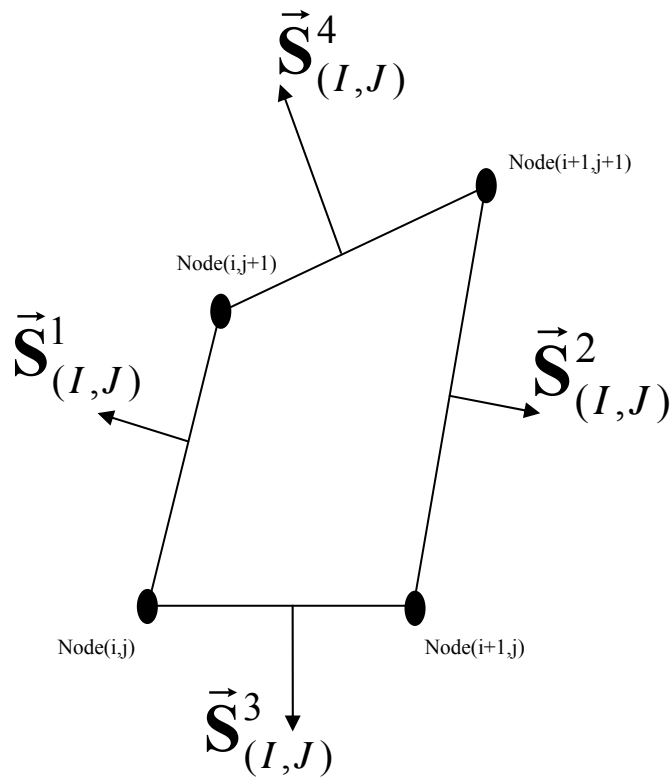


Figure 3.6 The geometry of cell (I,J)

$$\begin{aligned}\bar{\mathbf{S}}_{(I,J)}^k &= S_{(I,J)}^{kx} \bar{\mathbf{i}} + S_{(I,J)}^{ky} \bar{\mathbf{j}} \\ k &= 1, 2, 3, 4\end{aligned}\quad (3.5)$$

The following equations give the $S_{(I,J)}^{kx}$ and $S_{(I,J)}^{ky}$ values for the (I, J) th cell.

$$\begin{aligned}S_{(I,J)}^{1x} &= Y_{(I,J)} - Y_{(I,J+1)} \\ S_{(I,J)}^{1y} &= -X_{(I,J)} + X_{(I,J+1)} \\ S_{(I,J)}^{2x} &= Y_{(I+1,J+1)} - Y_{(I+1,J)} \\ S_{(I,J)}^{2y} &= -X_{(I+1,J+1)} + X_{(I+1,J)} \\ S_{(I,J)}^{3x} &= Y_{(I+1,J)} - Y_{(I,J)} \\ S_{(I,J)}^{3y} &= -X_{(I+1,J)} + X_{(I,J)} \\ S_{(I,J)}^{4x} &= Y_{(I,J+1)} - Y_{(I+1,J+1)} \\ S_{(I,J)}^{4y} &= -X_{(I,J+1)} + X_{(I+1,J+1)}\end{aligned}\quad (3.6)$$

where $X_{(I,J)}$ and $Y_{(I,J)}$ represent the X and Y values of the node (I, J) respectively.

The volume of the cell is computed by taking the cross-product of the diagonal vectors:

$$\begin{aligned}\nabla &= -\frac{1}{2}[(X_{(I+1,J+1)} - X_{(I,J)})(Y_{(I+1,J)} - Y_{(I,J+1)})] + \\ &+ \frac{1}{2}[(X_{(I+1,J)} - X_{(I,J+1)})(Y_{(I+1,J+1)} - Y_{(I,J)})]\end{aligned}\quad (3.7)$$

3.4 Spatial Discretization

For a structured cell-centered FVM, Equations (2.5) or (2.9) can be spatially discretized as follows: [28]

$$\forall_{(I,J)} \frac{\partial \mathbf{Q}_{(I,J)}}{\partial t} + \sum_{k=1}^4 \bar{\mathbf{F}}_{(I,J)}^k \cdot \bar{\mathbf{S}}_{(I,J)}^k = 0 \quad (3.8)$$

Here, k represents the four faces of the relevant cell.

All the variables in Equation (3.8) are explained before, except the face flux term $\bar{\mathbf{F}}_{(I,J)}^k$. Since all the variables are defined at the cell center, the method to calculate the flux terms on faces should be formulated. In the present work, Roe's [4, 22, 23] flux-difference splitting scheme is used.

3.4.1 Inviscid (Convective) Fluxes for Euler Equations

The flux at the face F_{face} is found as follows:

$$F_{face} = \frac{1}{2} [\mathbf{F}(Q_L) + \mathbf{F}(Q_R)] - \frac{1}{2} \sum_{k=1}^4 (|\hat{\mathbf{a}}_k|^*) \Delta \mathbf{V}_k \hat{\mathbf{R}}_k \quad (3.9)$$

where Q_L and Q_R represent the left and right cells of the interested face and they are defined as:

$$\bar{\mathbf{Q}}_L = \begin{bmatrix} \rho_L \\ \rho_L u_L \\ \rho_L v_L \\ \rho_L e_L \end{bmatrix} \quad \text{and} \quad \bar{\mathbf{Q}}_R = \begin{bmatrix} \rho_R \\ \rho_R u_R \\ \rho_R v_R \\ \rho_R e_R \end{bmatrix} \quad (3.10)$$

The subscripts L or R mean that the indicated variables are defined at the cell centers of left and right cells, respectively. Thus, the values $\mathbf{F}(Q_L)$ and $\mathbf{F}(Q_R)$ could be calculated as

$$\mathbf{F}(Q_L) = \begin{bmatrix} \rho_L u_L^n \\ \rho_L u_L^n u_L + p_L \frac{\Delta y}{\Delta s} \\ \rho_L u_L^n v_L - p_L \frac{\Delta x}{\Delta s} \\ \rho_L u_L^n H_L \end{bmatrix} \quad \text{and} \quad \mathbf{F}(Q_R) = \begin{bmatrix} \rho_R u_R^n \\ \rho_R u_R^n u_R + p_R \frac{\Delta y}{\Delta s} \\ \rho_R u_R^n v_R - p_R \frac{\Delta x}{\Delta s} \\ \rho_R u_R^n H_R \end{bmatrix} \quad (3.11)$$

where

$$\begin{aligned} \Delta s &= \sqrt{(\Delta x)^2 + (\Delta y)^2} \\ u^n &= \frac{(u\Delta y - v\Delta x)}{\Delta s} \\ u^t &= \frac{(u\Delta y + v\Delta x)}{\Delta s} \\ H &= (\rho e + p) / \rho \end{aligned} \quad (3.12)$$

u^n and u^t represent the velocities normal and tangent to the face and H is the total enthalpy per unit mass.

To calculate the terms within the summation sign in Equation (3.9), first the weighted average values for some variables at the face should be found as follows:

$$\begin{aligned} \hat{\rho} &= \sqrt{\rho_L \rho_R} \\ \hat{u} &= \frac{\sqrt{\rho_L} u_L + \sqrt{\rho_R} u_R}{\sqrt{\rho_L} + \sqrt{\rho_R}} & \hat{v} &= \frac{\sqrt{\rho_L} v_L + \sqrt{\rho_R} v_R}{\sqrt{\rho_L} + \sqrt{\rho_R}} \\ \hat{u}_n &= \frac{\sqrt{\rho_L} u_L^n + \sqrt{\rho_R} u_R^n}{\sqrt{\rho_L} + \sqrt{\rho_R}} & \hat{u}_t &= \frac{\sqrt{\rho_L} u_L^t + \sqrt{\rho_R} u_R^t}{\sqrt{\rho_L} + \sqrt{\rho_R}} \\ \hat{H} &= \frac{\sqrt{\rho_L} H_L + \sqrt{\rho_R} H_R}{\sqrt{\rho_L} + \sqrt{\rho_R}} & \hat{c} &= \frac{\sqrt{\rho_L} c_L + \sqrt{\rho_R} c_R}{\sqrt{\rho_L} + \sqrt{\rho_R}} \end{aligned} \quad (3.13)$$

where c_L and c_R are speeds of sound for left and right cells, respectively.

For any variable ϕ , the $\Delta\phi$ is defined as $\Delta\phi = \phi_R - \phi_L$.

Finally, the terms in the summation sign are given as follows:

$$\Delta\mathbf{V} = \begin{bmatrix} \frac{\Delta p - \hat{\rho}\hat{c}\Delta u^n}{2\hat{c}^2} \\ \frac{\hat{\rho}\Delta u^t}{\hat{c}} \\ \Delta\rho - \frac{\Delta p}{\hat{c}^2} \\ \frac{\Delta p + \hat{\rho}\hat{c}\Delta u^n}{2\hat{c}^2} \end{bmatrix} \quad (3.14)$$

$$\hat{\mathbf{R}}_1 = \begin{bmatrix} 1 \\ \hat{u} - \hat{c} \frac{\Delta y}{\Delta s} \\ \hat{v} + \hat{c} \frac{\Delta x}{\Delta s} \\ \hat{H} - \hat{u}_n \hat{c} \end{bmatrix} \quad \hat{\mathbf{R}}_2 = \begin{bmatrix} 0 \\ \hat{c} \frac{\Delta x}{\Delta s} \\ \hat{c} \frac{\Delta y}{\Delta s} \\ \hat{u}_t \hat{c} \end{bmatrix} \quad \hat{\mathbf{R}}_3 = \begin{bmatrix} 1 \\ \hat{u} \\ \hat{v} \\ \frac{\hat{u}^2 + \hat{v}^2}{2} \end{bmatrix} \quad \hat{\mathbf{R}}_4 = \begin{bmatrix} 1 \\ \hat{u} + \hat{c} \frac{\Delta y}{\Delta s} \\ \hat{v} - \hat{c} \frac{\Delta x}{\Delta s} \\ \hat{H} + \hat{u}_n \hat{c} \end{bmatrix} \quad (3.15)$$

$$|\hat{\mathbf{a}}_k|^* = \begin{cases} |\hat{a}_k|, & |\hat{a}_k| \geq \frac{1}{2} \delta a_k \\ \frac{(\hat{a}_k)^2}{\delta a_k} + \frac{1}{4} \delta a_k, & |\hat{a}_k| \leq \frac{1}{2} \delta a_k \end{cases} \quad k = 1, 4 \quad (3.16)$$

where

$$\hat{\mathbf{a}} = \begin{bmatrix} \hat{u}_n - \hat{c} \\ \hat{u}_n \\ \hat{u}_n \\ \hat{u}_n + \hat{c} \end{bmatrix} \delta a_k = \begin{cases} 0.0, & 0.0 \geq 4\Delta a_k \\ 4\Delta a_k, & 0.0 \leq 4\Delta a_k \end{cases} \text{ and } \Delta a_k = a_R^k - a_L^k \quad (3.17)$$

Zeeuw and Powell in [23] state that Equation (3.16) is necessary to prevent unphysical expansion shocks. However, it is only applied for the two acoustic waves ($k=1$ and $k=4$).

The four face fluxes are summed up for each cell to obtain the second term in Equation (2.9).

3.4.2 Inviscid (Convective) Fluxes for Euler Equations for Moving Mesh

If Equation (2.11) is implemented for convective fluxes, the u^n , u^t , \hat{u}_n and \hat{u}_t values in Equations from (3.9) to (3.17) should be replaced with the u_{ALE}^n , u_{ALE}^t , \hat{u}_{ALE}^n and \hat{u}_{ALE}^t values, respectively, which are defined as follows:

$$\begin{aligned} u_{ALE}^n &= u^n - u_{CV}^n \\ u_{ALE}^t &= u^t - u_{CV}^t \\ \hat{u}_{ALE}^n &= \frac{\sqrt{\rho_L} u_{ALE_L}^n + \sqrt{\rho_R} u_{ALE_R}^n}{\sqrt{\rho_L} + \sqrt{\rho_R}} \\ \hat{u}_{ALE}^t &= \frac{\sqrt{\rho_L} u_{ALE_L}^t + \sqrt{\rho_R} u_{ALE_R}^t}{\sqrt{\rho_L} + \sqrt{\rho_R}} \end{aligned} \quad (3.18)$$

This would change all the velocity terms that are multiplied by the normal of the face. Furthermore, the eigenvalues vector $\hat{\mathbf{a}}$ would also change.

3.5 Temporal Discretization

Explicit time marching technique is used with third order Runge-Kutta (RK) time integration method. Equations (2.5) or (2.9) could be rewritten as follows for convenience [4]:

$$\forall_{(I,J)} \frac{\partial}{\partial t} \mathbf{Q}_{(I,J)} + \mathbf{R}_{(I,J)} = 0 \quad (3.19)$$

where $\mathbf{R}_{(I,J)}$ is the residual term for the cell (I,J) , which is the sum of all convective face fluxes for each cell.

For the temporal discretization, the third order RK method is used as done by Gönç [4].

$$\begin{aligned} Q_{(I,J)}^0 &= Q_{(I,J)}^{t_n} \\ Q_{(I,J)}^1 &= Q_{(I,J)}^{t_n} - (\alpha_1 \Delta t) R(Q_{(I,J)}^0) & \alpha_1 &= \frac{1}{3} \\ Q_{(I,J)}^2 &= Q_{(I,J)}^{t_n} - (\alpha_2 \Delta t) R(Q_{(I,J)}^1) & \text{where } \alpha_2 &= \frac{1}{2} \\ Q_{(I,J)}^3 &= Q_{(I,J)}^{t_n} - (\alpha_3 \Delta t) R(Q_{(I,J)}^2) & \alpha_3 &= \frac{1}{1} \\ Q_{(I,J)}^{t_{n+1}} &= Q_{(I,J)}^3 \end{aligned} \quad (3.20)$$

At this point, the choice of Δt is critical. If it is too small, the convergence time increases. On the other hand, too large Δt generally results in divergence. This risk of divergence brings a restriction to the selection of Δt . Courant Friedrichs Lewy (CFL) criteria is presented below as [29]

$$\begin{aligned}
\Delta t_{(I,J)}^1 &= \frac{\nabla_{(I,J)}}{|u_{(I,J)}a_{(I,J)}^{1x} - v_{(I,J)}a_{(I,J)}^{1y}| + c_{(I,J)}a_{(I,J)}^1} \\
\Delta t_{(I,J)}^2 &= \frac{\nabla_{(I,J)}}{|u_{(I,J)}a_{(I,J)}^{2x} - v_{(I,J)}a_{(I,J)}^{2y}| + c_{(I,J)}a_{(I,J)}^2} \\
\Delta t_{(I,J)} &= \begin{cases} CFL \cdot \Delta t_{(I,J)}^1 & \Delta t_{(I,J)}^1 \leq \Delta t_{(I,J)}^2 \\ CFL \cdot \Delta t_{(I,J)}^2 & \Delta t_{(I,J)}^1 \geq \Delta t_{(I,J)}^2 \end{cases}
\end{aligned} \tag{3.21}$$

where

$$\begin{aligned}
a_{(I,J)}^{1x} &= \frac{1}{2}(Y_{(I,J+1)} + Y_{(I+1,J+1)}) - \frac{1}{2}(Y_{(I,J)} + Y_{(I+1,J)}) \\
a_{(I,J)}^{1y} &= \frac{1}{2}(X_{(I,J)} + X_{(I+1,J)}) - \frac{1}{2}(X_{(I,J+1)} + X_{(I+1,J+1)}) \\
a_{(I,J)}^{2x} &= \frac{1}{2}(Y_{(I,J)} + Y_{(I,J+1)}) - \frac{1}{2}(Y_{(I+1,J)} + Y_{(I+1,J+1)}) \\
a_{(I,J)}^{2y} &= \frac{1}{2}(X_{(I+1,J)} + X_{(I+1,J+1)}) - \frac{1}{2}(X_{(I,J)} + X_{(I,J+1)}) \\
a_{(I,J)}^1 &= \sqrt{(a_{(I,J)}^{1x})^2 + (a_{(I,J)}^{1y})^2} \\
a_{(I,J)}^2 &= \sqrt{(a_{(I,J)}^{2x})^2 + (a_{(I,J)}^{2y})^2}
\end{aligned} \tag{3.22}$$

The coefficient CFL is between zero and one. As it gets closer to one, the convergence speeds up, but on the other hand, the risk of divergence increases. As CFL goes to zero the code is more stable but the convergence time increases.

As Equation (3.21) implies, all the cells have their own $\Delta t_{(I,J)}$. This would mean that every cell is at a different time level computationally. This feature is commonly used when the desired output is a steady flow; due to the fact that it would accelerate the convergence and the intermediate solutions do not have any physical meaning. However, for an unsteady problem, the intermediate results should have physical meaning so that

all the cells should be at the same time level. This could be achieved with the following formula:

$$\Delta t^{Global} = \text{MIN}(\Delta t_{(I,J)}) \quad (3.23)$$

With Equation (3.23), the stability is ensured because all the cells have the safest (smallest) $\Delta t_{(I,J)}$ value.

3.6 Initial Conditions

For initial conditions, first the freestream values should be specified. These include the inlet Mach number (M_∞), angle of attack (α), stagnation pressure (P_{O_∞}), and stagnation temperature (T_{O_∞}).

$$\begin{aligned} M_{(I,J)} &= M_\infty \\ P_{O_{(I,J)}} &= P_{O_\infty} \\ T_{O_{(I,J)}} &= T_{O_\infty} \\ T_{(I,J)} &= T_\infty = \frac{T_{O_\infty}}{1 + \frac{(\gamma-1)}{2} (M_\infty)^2} \\ p_{(I,J)} &= p_\infty = \frac{P_{O_\infty}}{\left[\frac{T_{O_\infty}}{T_\infty} \right]^{\frac{\gamma}{\gamma-1}}} \\ \rho_{(I,J)} &= \rho_\infty = \frac{P_\infty}{RT_\infty} \\ c_{(I,J)} &= c_\infty = \sqrt{\gamma RT_\infty} \\ u_{(I,J)} &= u_\infty = M_\infty c_\infty \cos(\alpha) \\ v_{(I,J)} &= v_\infty = M_\infty c_\infty \sin(\alpha) \\ e_{(I,J)} &= e_\infty = \frac{P_\infty}{\rho_\infty (\gamma - 1)} + \frac{1}{2} [(u_\infty)^2 + (v_\infty)^2] \end{aligned} \quad (3.24)$$

For steady problems, all the variables within the cells are assigned to their corresponding freestream values as initial condition. However, for the unsteady solver, the input is the converged steady solution. That is to say, the unsteadiness start after the flow reaches the steady state.

3.7 Boundary Conditions

Boundary conditions, in any physical problem, are as important as the governing equations. Since the governing equations do not change for a specific problem, the translation of the physical domain to mathematics is with the help of boundary conditions (BC).

The general idea in this work is to use “ghost” cells for boundary conditions. “Ghost” cells are imaginary cells outside of the boundary with some given cell values. This method is very convenient, since the code also calculates the fluxes at these boundaries like any regular interior cell. Only the appropriate ghost cell values should be defined beforehand.

There are some differences between moving grid BC and stationary BC for the ghost cell values. Thus, it is suitable first to start with the stationary grid BC’s.

3.7.1 Boundary Conditions for a Stationary Grid

3.7.1.1 Far Field Boundary Condition for a Stationary Grid

All variables ϕ are assigned to their corresponding free stream values at all “ghost” cells at the far field so that

$$\phi_{ghost}^{farfield} = \phi_{\infty} \tag{3.25}$$

A better approach could be to implement characteristic boundary conditions. [4, 7, 6 and 9]. However, as this simple method gives successful results, this improvement is left for future studies.

3.7.1.2 Solid Wall Boundary Condition for a Stationary Grid

For an O-grid solver, the I^{th} ghost cell at the solid boundary has the neighbor cell of $(I,1)$, as shown in Figure 3.7. Equation 3.26 presents the values for the ghost cell.

$$\begin{aligned}
 \rho_{ghost(I)}^{solid} &= \rho_{(I,1)} \\
 u_{ghost(I)}^{nsolid} &= -u_{(I,1)}^n \\
 u_{ghost(I)}^{tsolid} &= u_{(I,1)}^t \\
 e_{ghost(I)}^{solid} &= e_{(I,1)}
 \end{aligned}
 \tag{3.26}$$

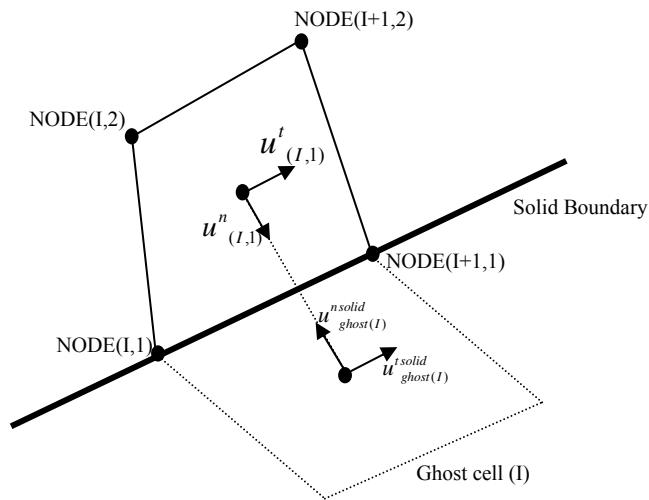


Figure 3.7 Solid wall boundary for a stationary grid (O-grid solver)

The energy and density are taken as equal to their corresponding values at the neighboring cell. The most important requirement on a solid boundary is that the flux

should not cross it. So the u^n values should have opposite sign but the same magnitude, in order to achieve $u^n_{face} = 0$. This will automatically ensure that the flux through the wall is zero. Finally, the tangential velocity is selected equal to its value in the neighboring cell, since Euler equations do not include the frictional effects and there should be a tangential velocity at the solid boundary.

For the C-grid solver, as the solid boundary is only between (SS,1) and (NX-SS+1,1), these conditions could only be applied for this range.

3.7.1.3 Interior Boundary Condition for a Stationary Grid

For interior boundaries, there is no need for a “ghost” cell. However, the neighborhood information is important.

For an O-grid solver, the cell (1, J), which is at the left boundary on computational domain, is neighbor to the cell (NX-1, J), which is at the right boundary of the computational domain, as shown in Figure 3.3.

For a C-grid solver, the neighborhood relationship is between cell (NX-I, 1) and cell (I,1), providing that I is between (1) and (SS-1), as seen in Figure 3.5.

3.7.2 Boundary Conditions for a Moving Grid

3.7.2.1 Far Field Boundary Condition for a Moving Grid

As the characteristic Riemann invariants are not implied to the code, the effect of rotation should be included in the “ghost” cell fluxes. However, except the grid velocity, again, all the variables at the cell center are taken equal to their corresponding freestream values.

$$\phi_{ghost}^{farfield} = \phi_{\infty} \quad (\text{for all variables except grid velocities}) \quad (3.27)$$

The grid velocity of the “ghost” cell is calculated with linear extrapolation from its neighboring cell and the cell adjacent to it as follows:

$$\begin{aligned} u_{CV_{ghost_I}}^{farfield} &= 2(u_{CV_{(I,NY-1)}}) - u_{CV_{(I,NY-2)}} \\ v_{CV_{ghost_I}}^{farfield} &= 2(v_{CV_{(I,NY-1)}}) - v_{CV_{(I,NY-2)}} \end{aligned} \quad (3.28)$$

3.7.2.2 Solid Wall Boundary for a Moving Grid

For an O-grid solver, the 1th ghost cell at the solid boundary has the neighbor cell (I,1). The “ghost” cell values are presented below [4, 21]:

$$\begin{aligned} \rho_{ghost(I)}^{solid} &= \rho_{(I,1)} \\ u_{ghost(I)}^{nsolid} &= -u_{(I,1)}^n + 2u_{CV_{face}}^n \\ u_{ghost(I)}^{tsolid} &= u_{(I,1)}^t \\ e_{ghost(I)}^{solid} &= e_{(I,1)} \\ u_{CV_{ghost_I}}^{solid} &= 2(u_{CV_{(I,1)}}) - u_{CV_{(I,2)}} \\ v_{CV_{ghost_I}}^{solid} &= 2(v_{CV_{(I,1)}}) - v_{CV_{(I,2)}} \end{aligned} \quad (3.29)$$

where

$$u_{CV_{(I,1)face}}^n = \frac{u_{CV_{(I,1)NODE}}^n + u_{CV_{(I+1,1)NODE}}^n}{2} \quad (3.30)$$

The energy and density are taken equal to their corresponding values at the neighboring cell. When the boundary is moving arbitrarily, there is also tangential velocity on the solid boundary, which is equal to the normal velocity of the solid body ($u_{CV_{face}}^n$). Its

value is found by the arithmetic mean of two closest nodes as explained by Equation (3.30).

The second equation in (3.29) is actually a version of Equation (3.31). It is logical that $u_{CV\ face}^n$ is the arithmetic mean of the normal velocity of $(I,1)$ th cell center ($u_{(I,1)}^n$) and the normal velocity of (I) th ghost cell center ($u_{ghost(I)}^{nsolid}$). [4, 21]

$$\frac{u_{ghost(I)}^{nsolid} + u_{(I,1)}^n}{2} = u_{CV\ face}^n \quad (3.31)$$

Finally, the tangential velocity is selected equal to its corresponding value at the neighboring cell even for the moving grid case, so that they are both equal to the tangential velocity of the solid body ($u_{CV\ face}^t$).

The ALE method is not implemented to the C-grid solver.

3.7.2.3 Interior Boundary Condition for a Moving Grid

As for stationary grid, there is also no need for a “ghost” cell at the interior boundary for a moving grid. The neighborhood relationship is the same for O and C solvers as the stationary grid.

3.8 The Grid Movement

The grid is rotated around the origin with a predefined arbitrary angular velocity. The method for calculating the grid velocities are presented here. The grid velocity definition can be rewritten as follows:

$$\vec{W} = u_{CV} \vec{i} + v_{CV} \vec{j} \quad (3.32)$$

First the magnitude of the velocity can be evaluated as follows:

$$\begin{aligned} |\vec{W}_{(I,J)}| &= \dot{\alpha}(\sqrt{[X_{cellcenter}^{(I,J)}]^2 + [Y_{cellcenter}^{(I,J)}]^2}) \\ |\vec{W}_{NODE(I,J)}| &= \dot{\alpha}(\sqrt{[X_{(I,J)}]^2 + [Y_{(I,J)}]^2}) \end{aligned} \quad (3.33)$$

Here, $\dot{\alpha}$ is the predefined arbitrary angular velocity, which is specific to a problem. The detailed description of the geometry is presented in Figure 3.8.

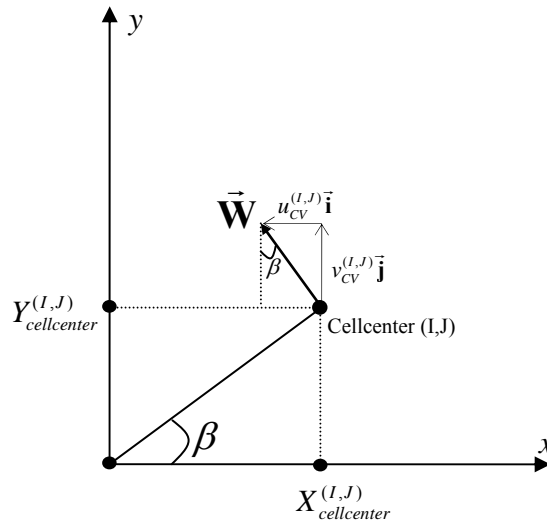


Figure 3.8 Geometry for grid velocities

The next step is to find the base-angle $\beta_{(I,J)}$ to compute the components of this grid velocity.

$$\begin{aligned}\beta_{(I,J)} &= ATAN2\left(\frac{Y_{cellcenter}^{(I,J)}}{X_{cellcenter}^{(I,J)}}\right) \\ \beta_{NODE(I,J)} &= ATAN2\left(\frac{Y_{(I,J)}}{X_{(I,J)}}\right)\end{aligned}\tag{3.34}$$

ATAN2 function gives values in the interval of $[-\pi, \pi]$. Thus, it also finds the quadrant where the cell center is.

Finally, the components of the grid velocity are found as follows:

$$\begin{aligned}u_{CV}^{(I,J)} &= -\sin(\beta_{(I,J)})|\vec{W}_{(I,J)}| \\ v_{CV}^{(I,J)} &= \cos(\beta_{(I,J)})|\vec{W}_{(I,J)}| \\ u_{CVNODE}^{(I,J)} &= -\sin(\beta_{NODE(I,J)})|\vec{W}_{NODE(I,J)}| \\ v_{CVNODE}^{(I,J)} &= \cos(\beta_{NODE(I,J)})|\vec{W}_{NODE(I,J)}|\end{aligned}\tag{3.35}$$

Trepanier et al. [14] argue that, geometric conservation laws (GCL) should be applied for moving grids. The first geometric law, named as Volume Conservation Law (VCL), states that the increase of volume in a cell is equal to total volumetric increases along its faces. The second law is called Surface Conservation Law (SCL) and it makes sure that each cell is closed.

Although the GCL is important to satisfy, it could be argued that if the grid is only moved and not deformed, then this law is satisfied automatically. [4] Because the grid volumes do not alter and all the cells remain as closed.

3.9 Convergence Criteria for Steady Flows

All the variables that contain time are considered as residual terms as in Equation (3.19). If this term is sufficiently small, then it can be assumed that, the solution is converged. The definition of sufficient can be given as follows:

$$\begin{aligned}
 R1^n &= \frac{\sqrt{\sum_{I=1}^{I=NX-1} \sum_{J=1}^{J=NY-1} (\mathbf{R}^n_{(I,J,1)})^2}}{(NX-1)(NY-1)} \\
 R2^n &= \frac{\sqrt{\sum_{I=1}^{I=NX-1} \sum_{J=1}^{J=NY-1} (\mathbf{R}^n_{(I,J,2)})^2}}{(NX-1)(NY-1)} \\
 R3^n &= \frac{\sqrt{\sum_{I=1}^{I=NX-1} \sum_{J=1}^{J=NY-1} (\mathbf{R}^n_{(I,J,3)})^2}}{(NX-1)(NY-1)} \\
 R4^n &= \frac{\sqrt{\sum_{I=1}^{I=NX-1} \sum_{J=1}^{J=NY-1} (\mathbf{R}^n_{(I,J,4)})^2}}{(NX-1)(NY-1)} \tag{3.36}
 \end{aligned}$$

$R1, R2, R3,$ and $R4$ are calculated as shown in Equation (3.36). Here, the first residual corresponds to mass conservation, second and third to momentum conservation for x and y directions respectively, and the last residual corresponds to energy conservation. The superscript n denotes the iteration number.

$$\begin{aligned}
 RS1 &= \log\left(\frac{R1^n}{R1^1}\right) \\
 RS2 &= \log\left(\frac{R2^n}{R2^1}\right) \\
 RS3 &= \log\left(\frac{R3^n}{R3^1}\right) \\
 RS4 &= \log\left(\frac{R4^n}{R4^1}\right)
 \end{aligned}
 \tag{3.37}$$

For the convergence the $RS1$, $RS2$, $RS3$ and $RS4$ values should be less than $E = -6.0$. That is, the error in the converged solution should be less than 10^{-6} times the error in the first iteration for all the conserved variables. Decreasing the variable E will result in increased accuracy, but also in increased run time.

CHAPTER 4

VERIFICATION OF THE CODE, RESULTS AND DISCUSSION

*"When I'm working on a problem,
I never think about beauty.
I think only how to solve the problem.
But when I have finished,
if the solution is not beautiful,
I know it is wrong."
Richard Buckminster Fuller*

At this section, the developed code is tested under various conditions. This chapter is divided into two parts as the discussion of steady and unsteady problems.

NACA 0012 and RAE 2822 airfoil profiles are used to test the code. The coordinates of these profiles can be found in Appendix B. For all problems, the radius of the computation domain is selected as 15 times the chord length, so that the disturbances due to the airfoil do not reach the boundary.

4.1. Steady Solutions

The code is tested under subsonic and transonic conditions. Table 4.1 summarizes the test cases for steady problems. For all cases, the stagnation pressure and temperature are taken as 150 kPa and 350 K, respectively.

Table 4.1 Summary of Steady Test Cases

Test Case	Airfoil Profile	M_∞	α	Reference(s)
4.1.1	NACA 0012	0.600	0.0	[30]
4.1.2	NACA 0012	0.800	0.0	[31]
4.1.3	RAE 2822	0.7290	2.31	[32, 33, 34]

4.1.1. Subsonic Flow over Symmetric Airfoil

The NACA 0012 profile is widely used for 2-D code validation and experimental data for many different cases could easily be found in literature.

The inviscid, steady, subsonic flow solutions around the NACA 0012 airfoil are obtained for an inlet Mach number (M_∞) of 0.60, and for an angle of attack (α) of 0.0° and compared with [30]. As the profile is symmetric, the airfoil does not have any lifting effect for zero angle of attack. Therefore, the expectation is that the pressure coefficient for lower and upper halves of the airfoil should have the same magnitude.

There are different types of grids with different resolution. Table 4.2 summarizes the features of the used grids.

Table 4.2 Grid types for subsonic flow over symmetric airfoil (case 4.1.1.)

Case Number	Grid Type	Node Number
4.1.1.a	O-Grid	121 x 121
4.1.1.b	O-Grid	243 x 257
4.1.1.c	C-Grid	211 x 121

Figures 4.1 and 4.2 give a brief idea of the computational domain for O and C grid solvers, respectively. Figures 4.3-4.5 show close-up views for different mesh resolutions.

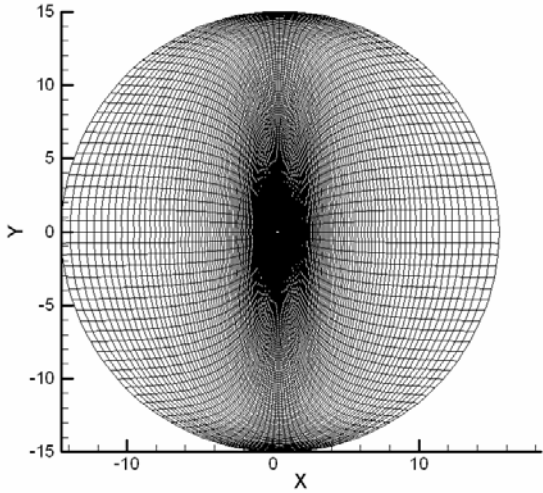


Figure 4.1 Overall view for NACA 0012 profile O-grid with (121x121) nodes

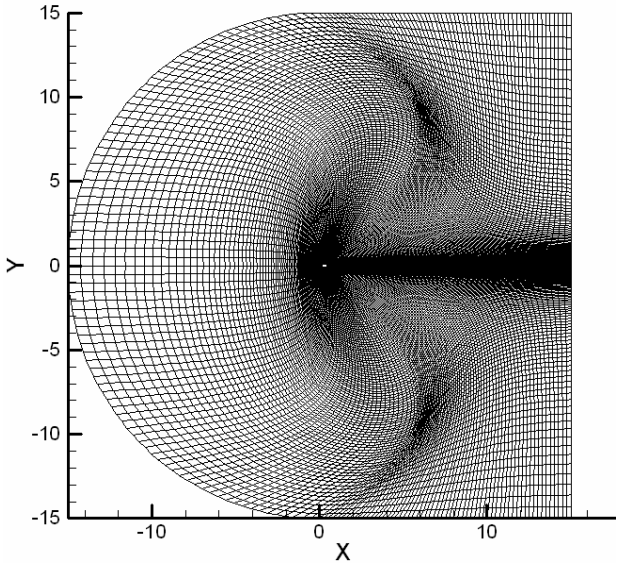


Figure 4.2 Overall view for NACA 0012 profile C-grid with (211x121) nodes

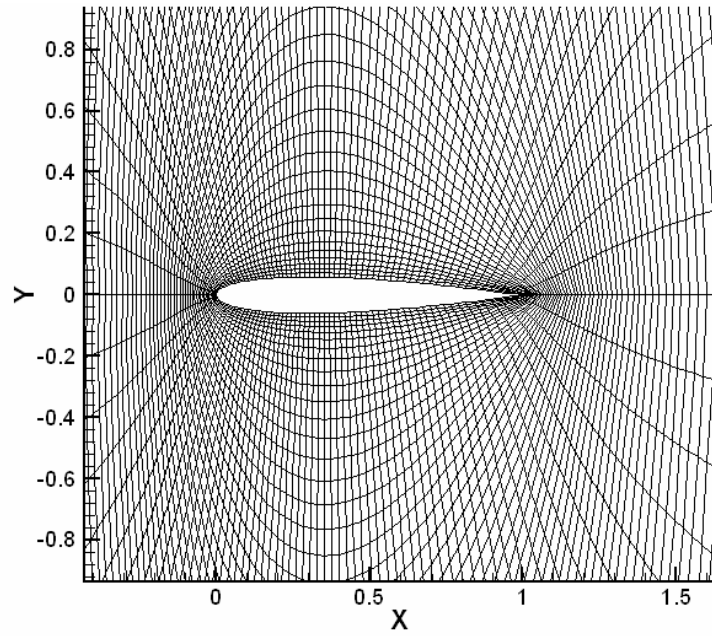


Figure 4.3 Close-up view of NACA 0012 profile O-grid with (121x121) nodes

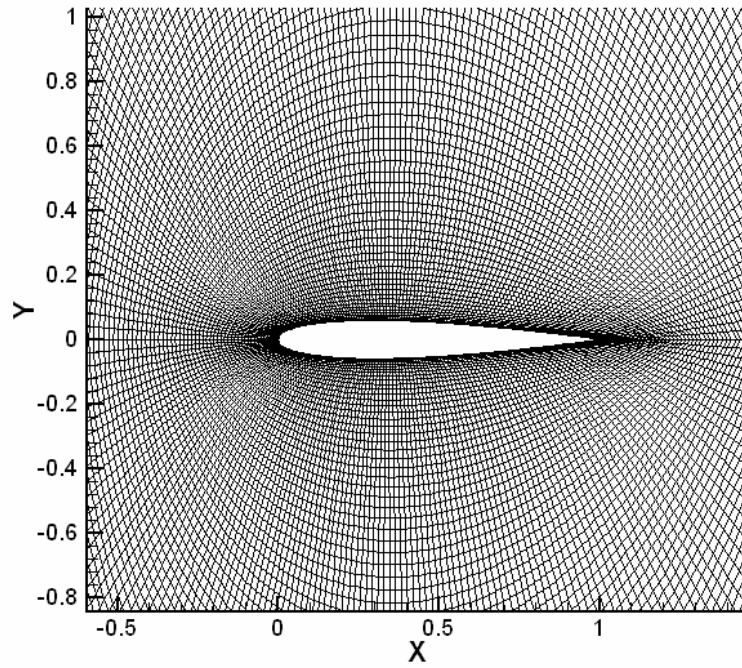


Figure 4.4 Close-up view of NACA 0012 profile O-grid with (243x257) nodes

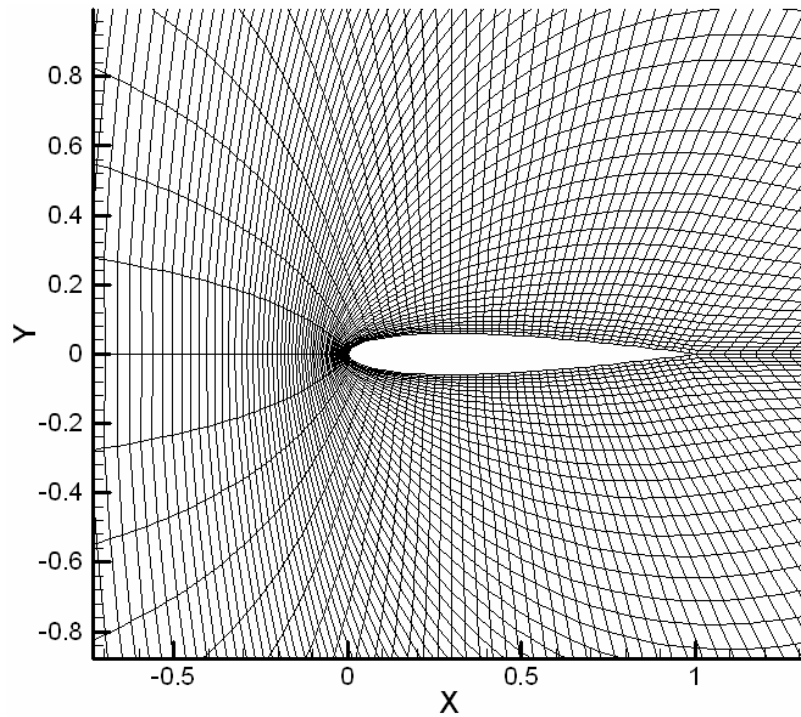


Figure 4.5 Close-up view of NACA 0012 profile C-grid with (211x121) nodes

Figures 4.6 - 4.8 present residual histories for different meshes. As expected, the convergence in coarser O-grid of Figure 4.6 is much faster than the one in fine O-grid of Figure 4.7. The convergence rate in C-grid, which is shown in Figure 4.8, is not as good as the O-grids. This is because of the unnecessary more cells far away from the airfoil. These cells are needed in order to maintain the structure; however they will slow down the convergence.

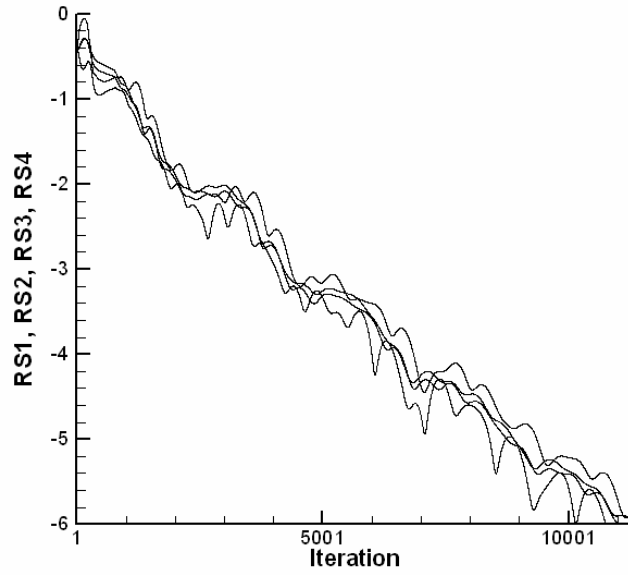


Figure 4.6 Residual history for NACA 0012 (121 x 121) nodes, $M_\infty=0.60$ $\alpha=0.0$

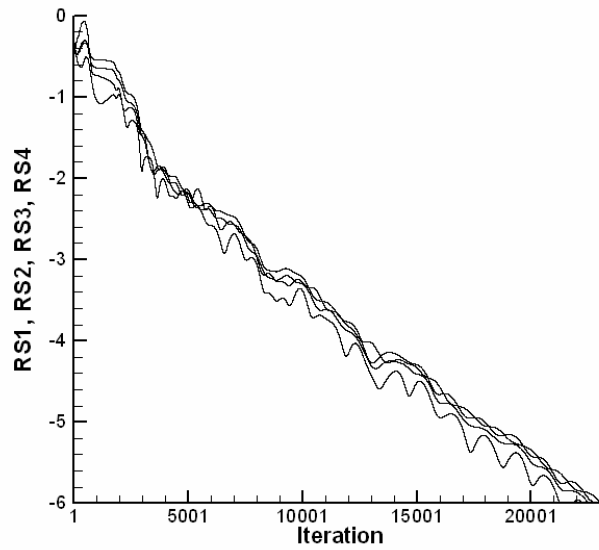


Figure 4.7 Residual history for NACA 0012 (243 x 257) nodes, $M_\infty=0.60$ $\alpha=0.0$

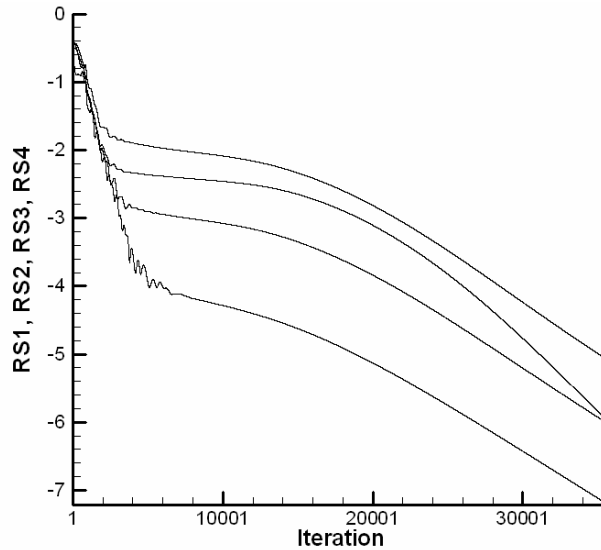


Figure 4.8 Residual history for NACA 0012 C-grid (211 x 121) nodes,
 $M_\infty=0.60$ $\alpha=0.0$

The variation of pressure coefficient along the chord length is shown in Figure 4.9. It can be said that, the computed results of all grids are well fitted to the available experimental data. The coarse O-grid has 14641 nodes and the fine O-grid has 62451 nodes. On the other hand, the C-grid has about 25531 nodes. Although the grid resolution differs so much, the pressure coefficients are in agreement. Thus, it can be said that, this test case also shows the domain independency of the solver. Different grid types and different grid resolutions are resulted in similar solutions, which is a desired feature for solvers.

Here, only the pressure coefficient contours Figure 4.11, and Mach number contours, Figure 4.10, of fine O-grid are presented. However, as stated before, the results of all grids are very similar to each other in Figure 4.9.

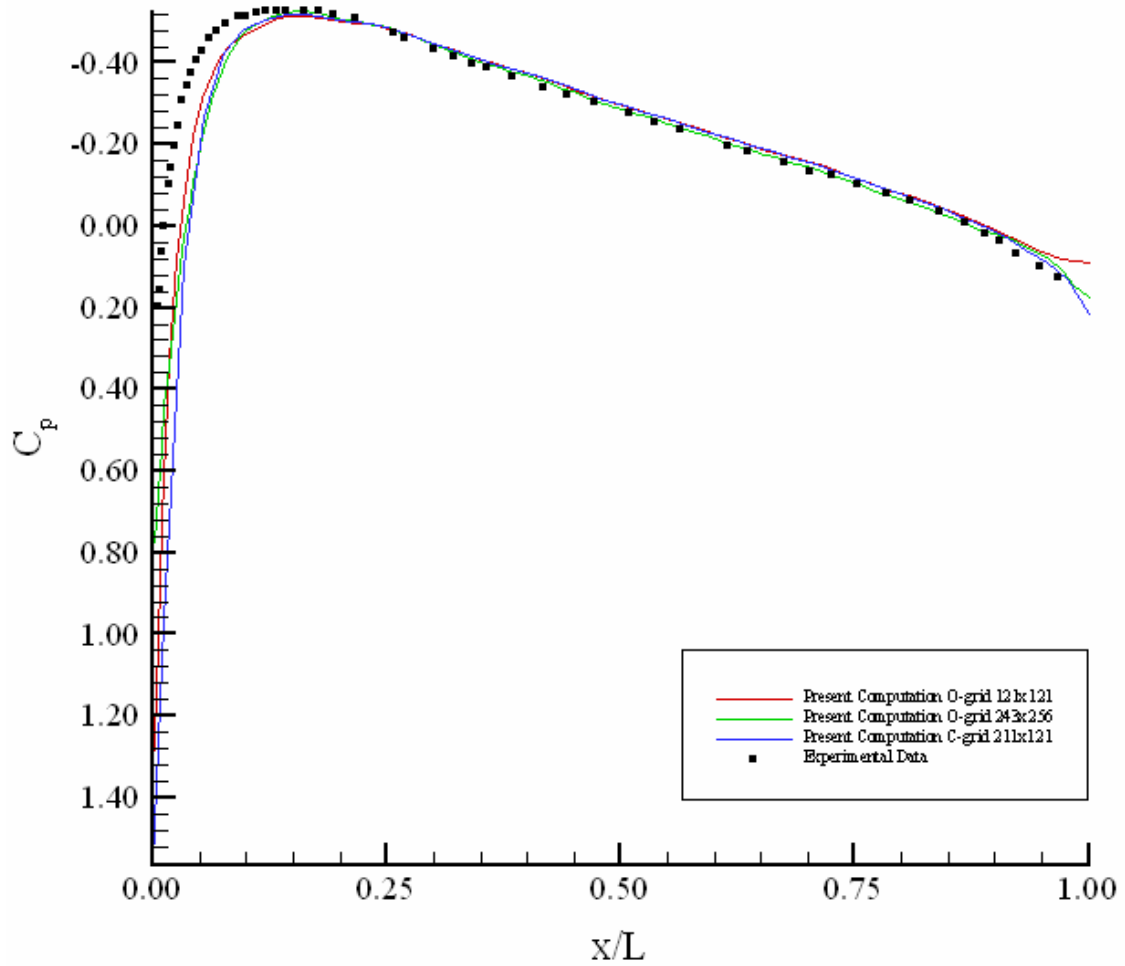


Figure 4.9 Figure 4.0.9 c_p distribution over the NACA 0012 airfoil,
 $M_\infty=0.60$ $\alpha=0.0$

Figure 4.12 enables to see the details of leading edge. Stagnation point is seen clearly at the tip of the airfoil (at point $x = 0.0$ and $y = 0.0$). The velocity vector distribution appears to be correct. Since Euler equations are used in this study, there is velocity on the boundary as seen.

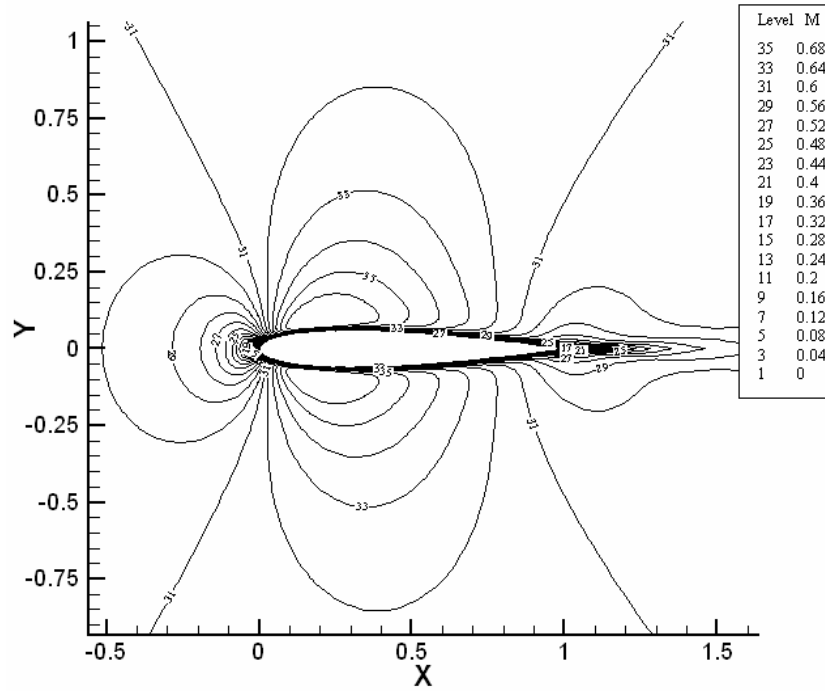


Figure 4.10 Mach Contour, NACA0012 airfoil (243x257) nodes, $M_\infty=0.60$ $\alpha=0.0$

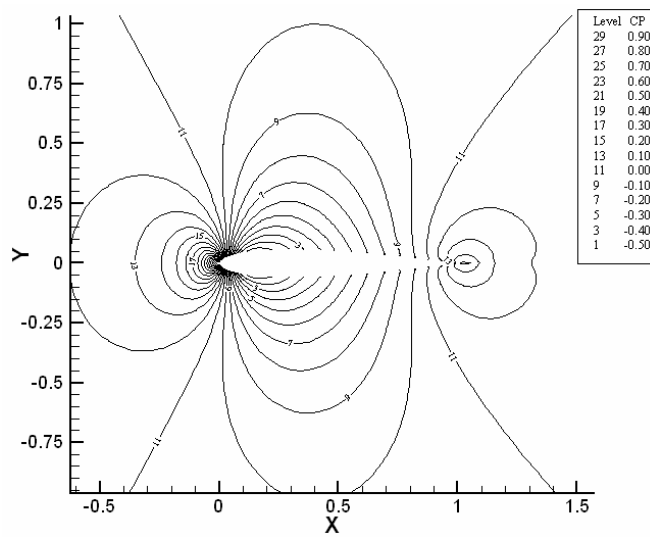


Figure 4.11 c_p contour, NACA0012 airfoil (243x257) nodes, $M_\infty=0.60$ $\alpha=0.0$

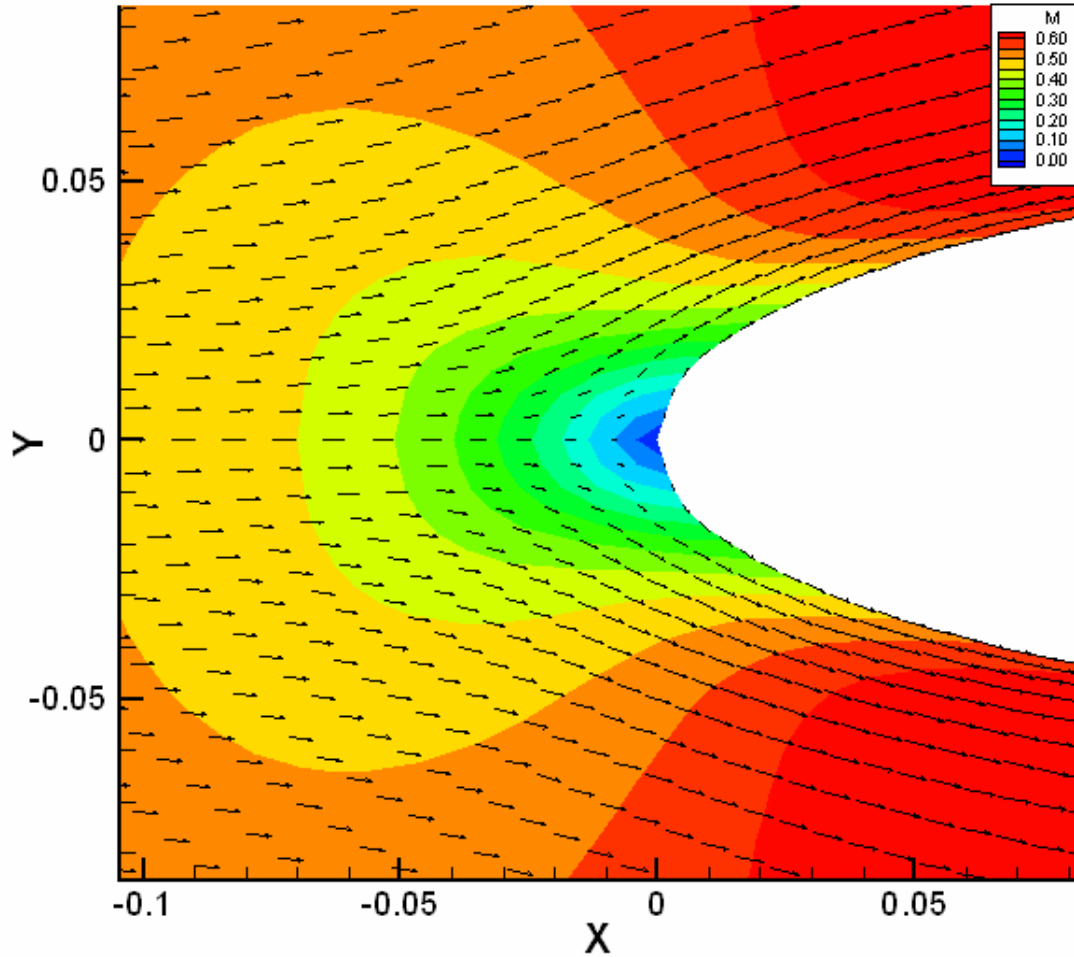


Figure 4.12 Mach Contour and Velocity vectors, NACA0012 airfoil
 Leading edge close-up view (243x257) nodes, $M_\infty=0.60$ $\alpha=0.0$

It can be said that, the results of this test case are acceptable. The grid independency is achieved in this case. Furthermore, all the solutions give symmetric results, as expected from this profile.

4.1.2. Transonic Flow over a Symmetric Airfoil

The inviscid, steady, transonic flow solutions around the NACA 0012 airfoil are obtained for an inlet Mach number (M_∞) of 0.80, and for an angle of attack (α) of 0.0°

and will be compared with [31]. Table 4.3 summarizes the features of the used grids. The close-up view of the mesh for case 4.1.2.a is given previously in Figure 4.3 and the mesh for case 4.1.2.b is given in Figure 4.13. The idea here is that, if one knows approximately the location of the shock and prepares the numerical grid accordingly; the convergence rate and/or the accuracy can be improved. As seen from Figures 4.14 and 4.15, which presents the residual histories for the two grids, the shock adapted mesh needs approximately 3000 less iterations for the same accuracy, even if it has 38% more nodes. Furthermore, it can also be argued that the shock adapted mesh captured shock slightly better as seen in Figure 4.16, which displays the pressure coefficient variation along the airfoil, as expected.

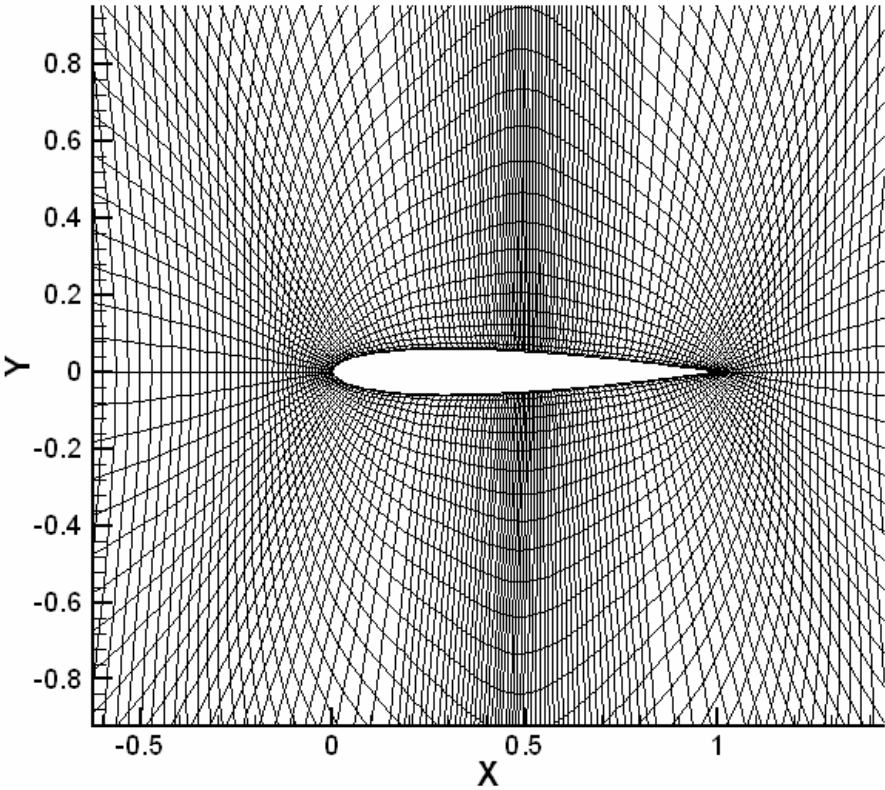


Figure 4.13 Close-up view of NACA0012 profile O-grid with (101x201) nodes

Table 4.3 Grid types for transonic flow over symmetric airfoil (case 4.1.2)

Case Number	Grid Type	Node Number
4.1.2.a	O-Grid	121 x 121
4.1.2.b	O-Grid (shock adapted)	101 x 201

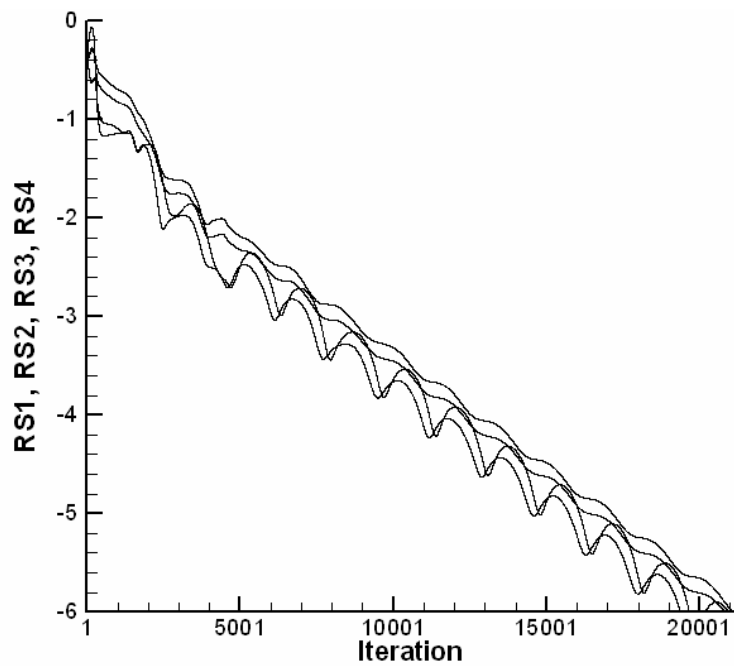


Figure 4.14 Residual history for NACA 0012 (121 x 121) nodes, $M_\infty=0.80$ $\alpha=0.0$

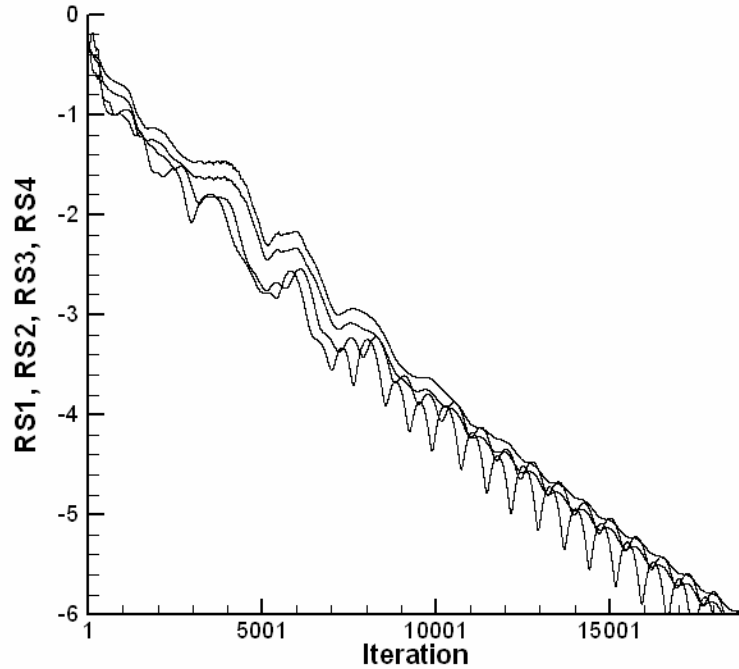


Figure 4.15 Residual history for NACA 0012 (101 x 201) nodes, $M_\infty=0.80$ $\alpha=0.0$

It can be said that, the results of the present computation agree well with the inviscid computation result of Wu and Li [31]. Furthermore, after the comparison of Figures 4.17 and 4.18, it can be also argued that the Mach contours of [31] and present study are in agreement.

As a result, it could be concluded that the present code is also verified for transonic flows over a symmetric airfoil.

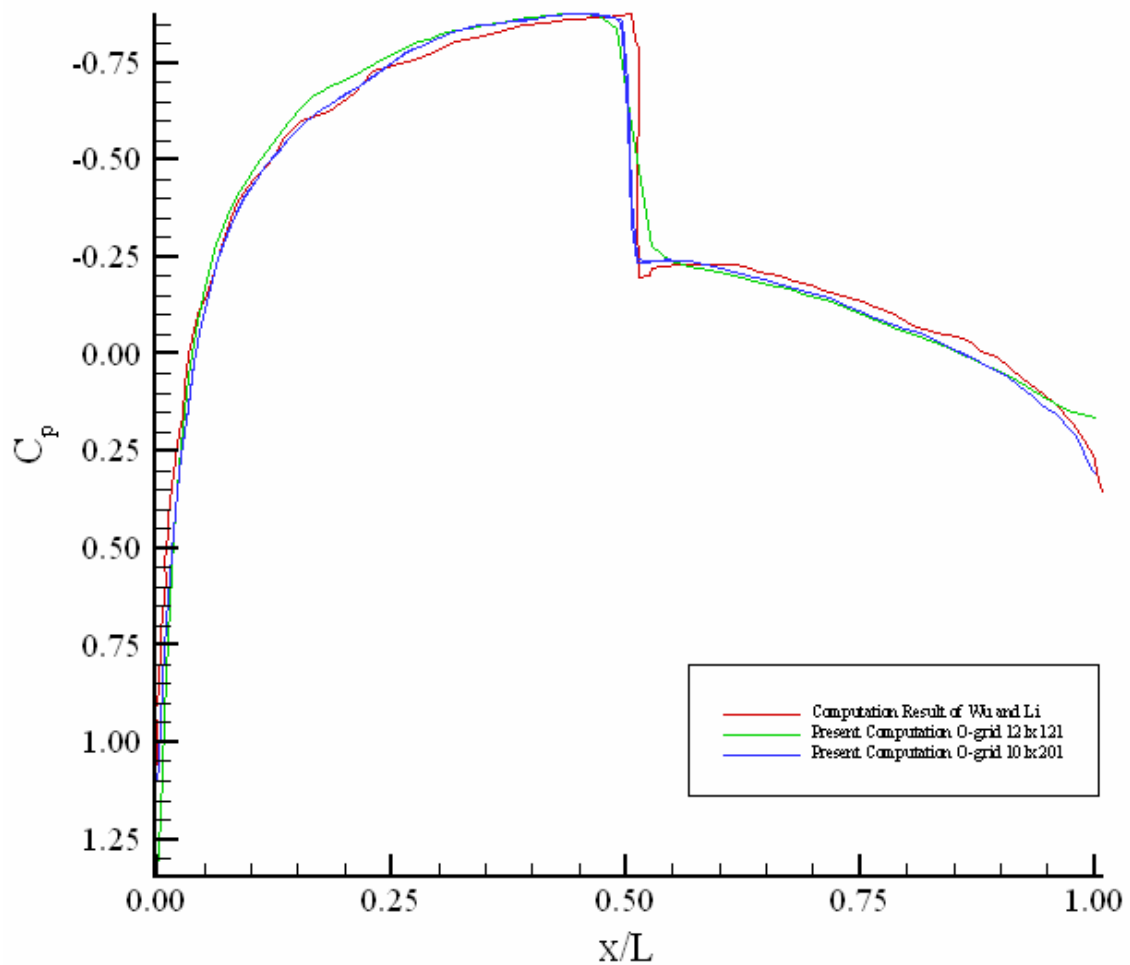


Figure 4.16 c_p distribution over the NACA 0012 airfoil, $M_\infty=0.80$ $\alpha=0.0$

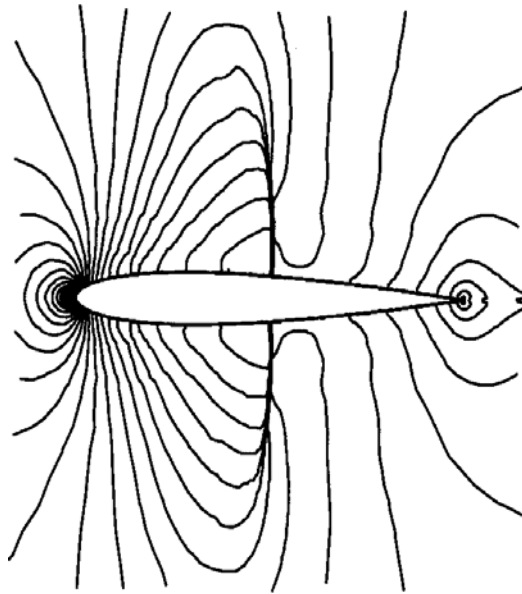


Figure 4.17 Mach Contour, NACA0012 airfoil, $M_\infty=0.80$ $\alpha=0.0$, Solution of Wu et al.

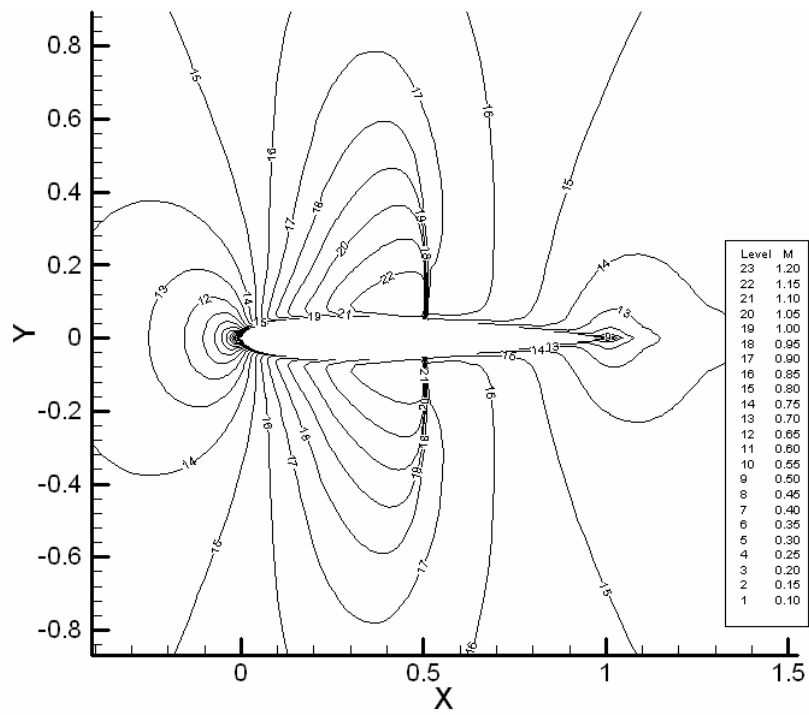


Figure 4.18 Mach Contour, NACA0012 airfoil (101x212) nodes, $M_\infty=0.80$ $\alpha=0.0$

4.1.3. Transonic Flow over an Unsymmetrical Airfoil with a Non-zero Angle of Attack

The inviscid, steady, transonic flow solutions around the RAE 2822 airfoil are obtained for an inlet Mach number (M_∞) of 0.7290, and for an angle of attack (α) of 2.310° and will be compared with [32, 33 and 34]. The computational results of Kudinov [32, 33] are viscous solutions with 6.5 million Reynolds number. The experimental results of Cook et al. [34] are obtained from [32].

The summary of used grids can be found in Table 4.4. These numerical grids are presented in Figures 4.19 and 4.20.

Table 4.4 Grid types for test case 4.1.3

Case Number	Grid Type	Node Number
4.1.3.a	O-Grid	801 x 301
4.1.3.b	C-Grid	601 x 121

The residual histories for different mesh types are seen in Figures 4.21 and 4.22. Furthermore, pressure coefficient variation over the airfoil is shown in Figure 4.23. The results of present solver are quite successful at the lower surface. However, at the upper surface there are problems at the locations of shock and the pressure maximum at the leading edge. The prediction of the shock location could be enhanced by using fine mesh or (if the location is approximately known) a shock adapted mesh could be used. In this part, a very fine mesh is used in order to catch the shock location. On the other hand, the pressure maximum at the leading edge could not be predicted. The main reason may be the omission of viscous effects in the present solver. The maximum error in computation is less than 20%.

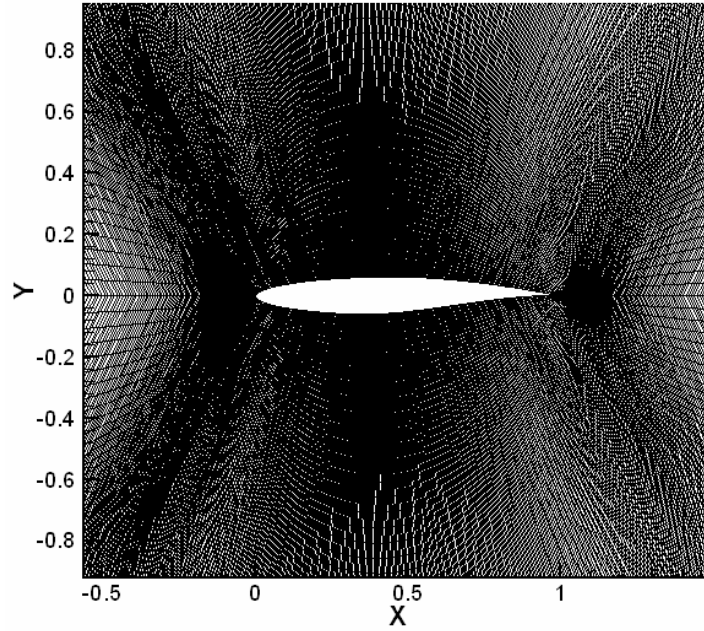


Figure 4.19 Close-up view of RAE 2822 profile O-grid with (801x301) nodes

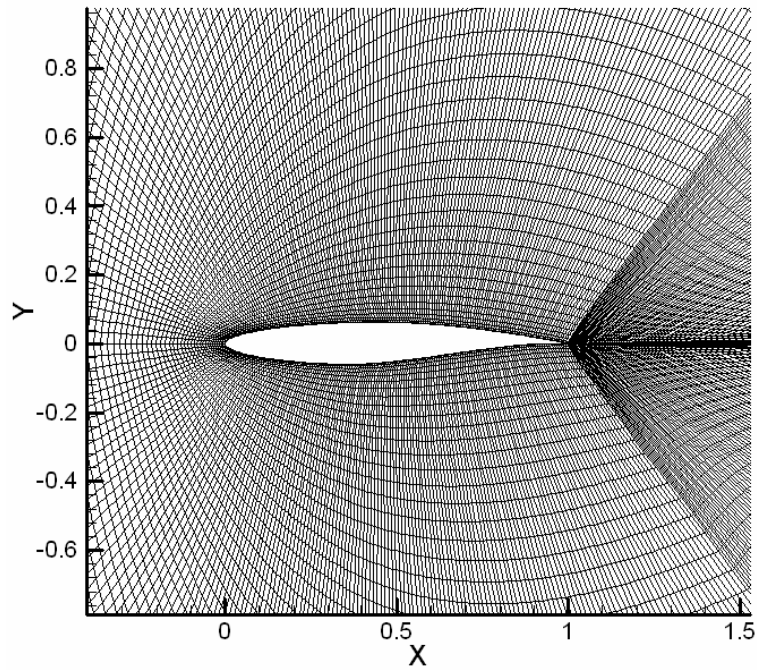


Figure 4.20 Close-up view of RAE 2822 profile C-grid with (601x121) nodes

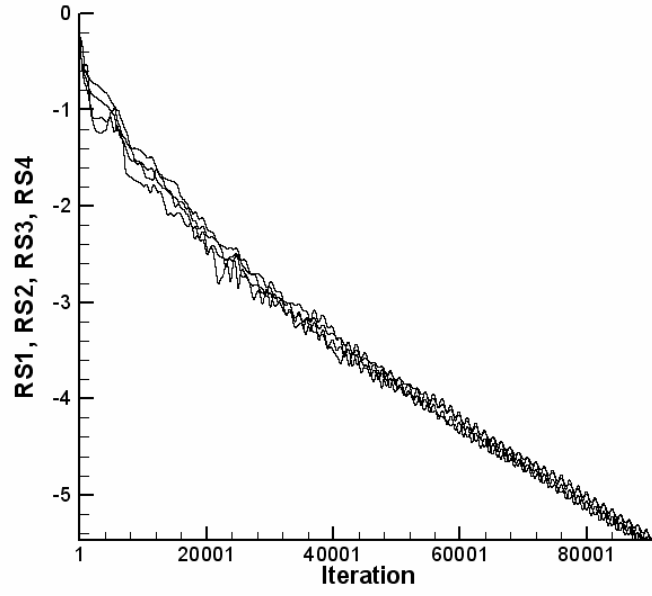


Figure 4.21 Residual history for RAE 2822 profile O-grid with (801x301),
 $M_\infty=0.7290$ $\alpha=2.310$

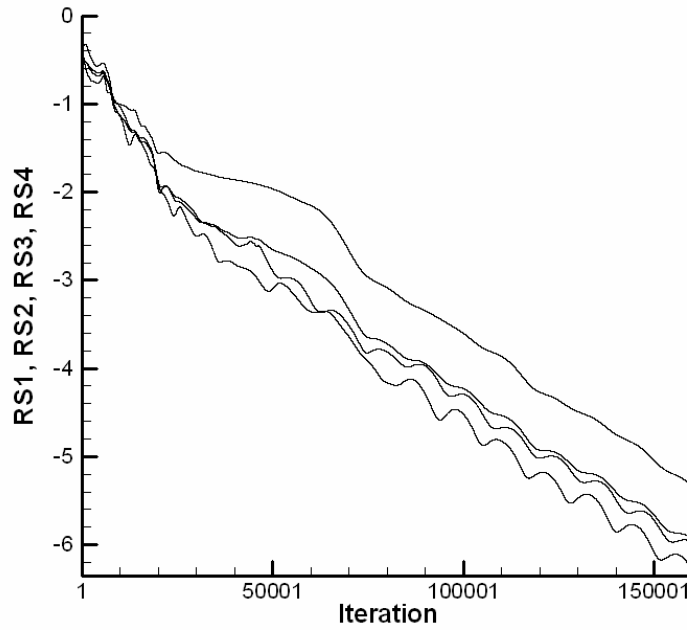


Figure 4.22 Residual history for RAE 2822 profile C-grid with (601x121),
 $M_\infty=0.7290$ $\alpha=2.310$

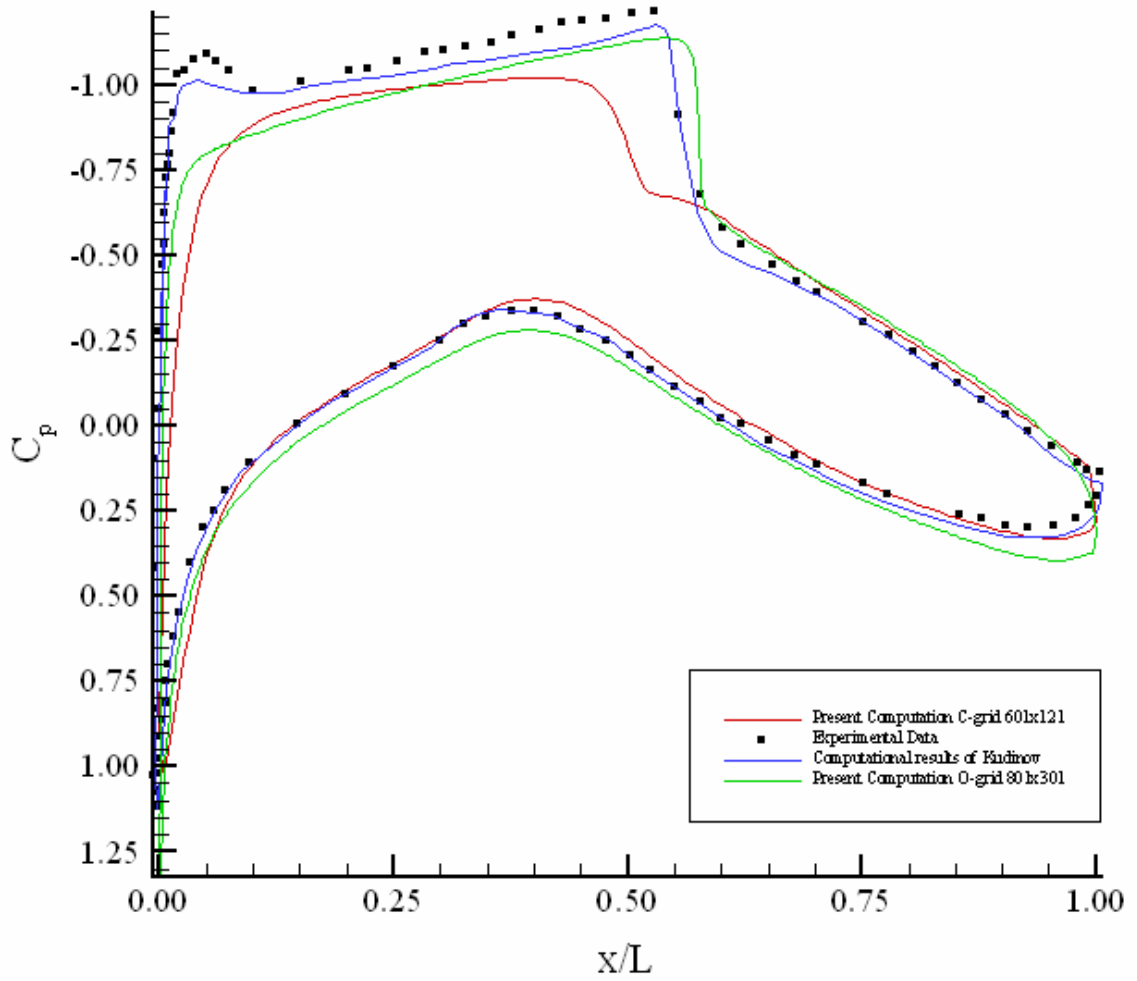


Figure 4.23 c_p distribution over RAE 2822 profile, $M_\infty=0.7290$ $\alpha=2.310$

Figures 4.24 and 4.25 compare the Mach contours of Kudinov [32] and present study. The contours of both solutions are very much alike.

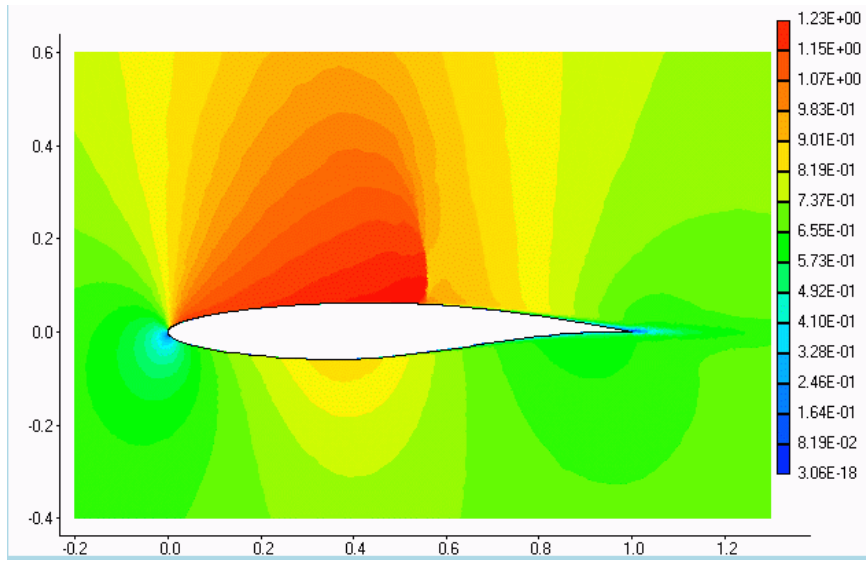


Figure 4.24 Mach Contour, RAE 2822 profile, $M_\infty=0.7290$ $\alpha=2.310$,
Solution of Kudinov [32]

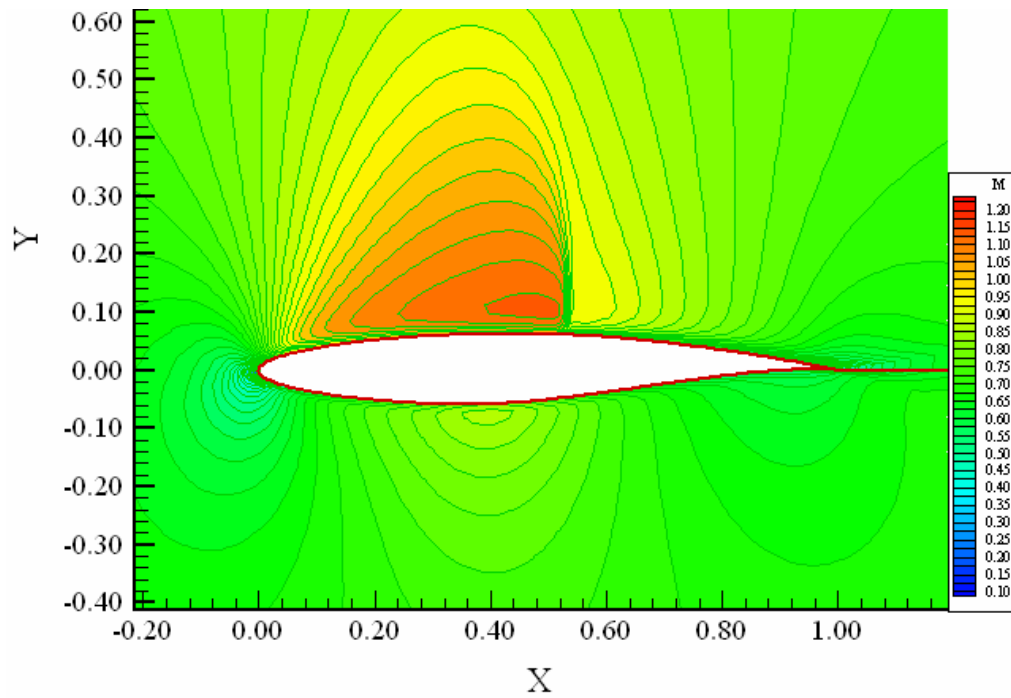


Figure 4.25 Mach Contour, RAE 2822 profile, O-grid
801x301 nodes, $M_\infty=0.7290$ $\alpha=2.310$, present computation

4.2 Unsteady Solution - Oscillating Airfoil

This problem is widely used to validate unsteady CFD codes. The experiment is done by AGARD [36]. The results will also be compared with inviscid computational results of [18, 35]. Table 4.5 gives brief information about the test case.

The code is tested for oscillating airfoil problem at 8 different test cases. For this problem as the steady problems, the stagnation pressure and temperature are taken as 150 kPa and 350 K, respectively.

For unsteady problems, the grid movement start after the steady part of the problem converges. For the steady part, CFL is selected as 0.5 in order to prevent divergence whereas it could be as high as 1.0 for the unsteady part. Furthermore, M_∞ remain the same for steady and unsteady solvers.

Table 4.5 Properties of Unsteady Test Case

Case	Airfoil Profile	M_∞	α	References
4.2	NACA 0012	0.7550	$0.016^\circ + 2.51^\circ \sin(\omega t)$	[18, 35 and 36]

In this problem, the NACA 0012 profile airfoil is first brought to steady state with a Mach number (M_∞) of 0.7550 and angle of attack (α) of 0.016° . After steady state is reached, the airfoil starts to oscillate about a hinge located at $0.25L$ (quarter chord length). The oscillation amplitude is 2.51° . In other words, the oscillation is between $(+2.526^\circ)$ and (-2.494°) . The relationship of the angle of attack and the time is given below:

$$\alpha = 0.016^\circ + 2.51^\circ \sin(\omega t) \quad (4.1)$$

$$k = \frac{\omega L}{2M_\infty c_\infty} = 0.0814 \quad (4.2)$$

where ω [rad/s] is the circular pitch frequency, L [m] is the chord length, k [rad] is the reduced frequency based on chord length. The angular velocity ($\dot{\alpha}$) is the time derivative of angular position function and is presented below:

$$\dot{\alpha} = 2.51^\circ \frac{\pi}{180^\circ} \omega \cos(\omega t) \quad (4.3)$$

There are eight test cases. They are summarized in Table 4.6. In these test cases instantaneous values are taken and compared to the experimental and computational data. The reference [35] has the data on all the 8 test cases and reference [18] has only on 2 of them (case 4.2.c and case 4.2.e).

Michler et al. [18] states that the periodicity is achieved at the third cycle of oscillation. Therefore, the data is taken after the airfoil has undergone two complete cycles. Figure 4.26 presents the test cases (as dots) in the sinusoidal oscillation cycle.

Table 4.6 Summary of oscillating airfoil test cases

Test Case No	Angle of Attack [deg]	Direction of Movement
4.2.a	1.090	UP
4.2.b	2.340	DOWN
4.2.c	2.010	DOWN
4.2.d	0.520	DOWN
4.2.e	-1.250	DOWN
4.2.f	-2.410	DOWN
4.2.g	-2.000	UP
4.2.h	-0.540	UP

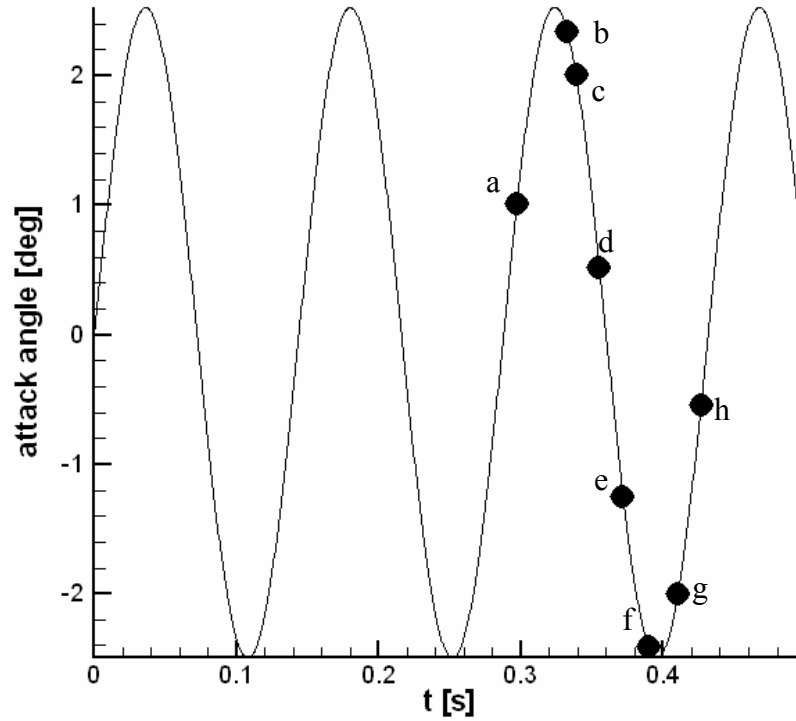


Figure 4.26 Oscillating Airfoil Test cases in the sinusoidal oscillation cycle

This problem is solved with two different grids. Table 4.7 shows the features of these grids. The close-up views of these grids are given in Figures 4.27.a and 4.27.b.

Table 4.7 Grid types for the oscillating airfoil

Grid type	Node Number
O-grid / Fine	451x201
O-grid / Coarse	301x151

In Figures 4.28 - 4.35, the pressure coefficient distributions along the airfoil for different test cases are plotted. For all test cases except 4.2.c and 4.2.e, the present solver results are compared with available experimental data from AGARD [36] and the inviscid

computational results of Ramesh et al. [35]. For cases 4.2.c and 4.2.e, in addition to those results, inviscid computational results from Michler et al. [18] are also available.

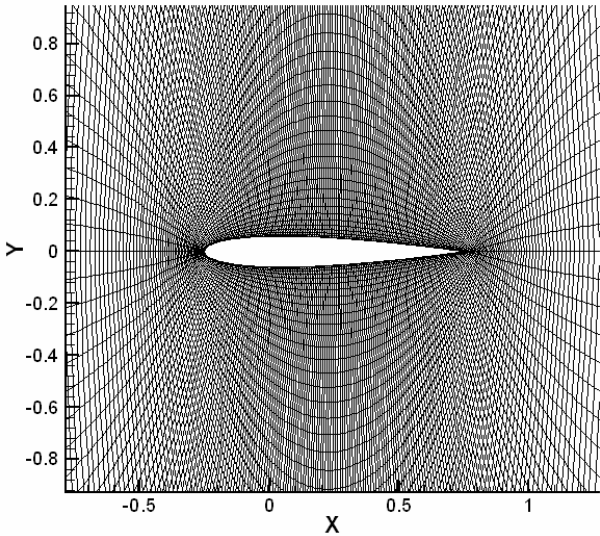


Figure 4.27.a Numerical mesh of NACA0012 for oscillating airfoil problem
301x151 nodes

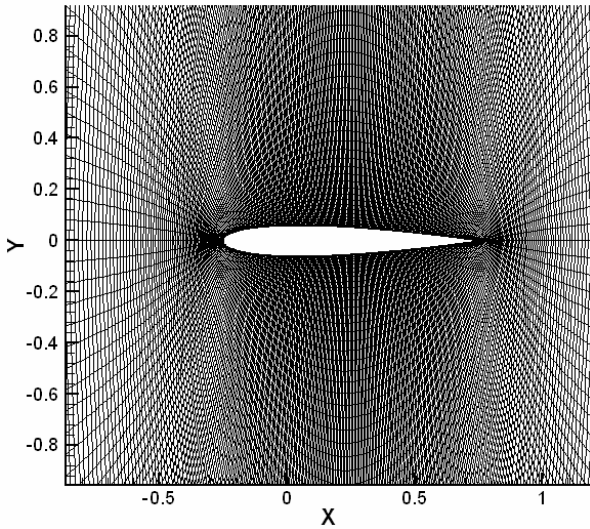


Figure 4.27.b Numerical mesh of NACA0012 for oscillating airfoil problem
451x201 nodes

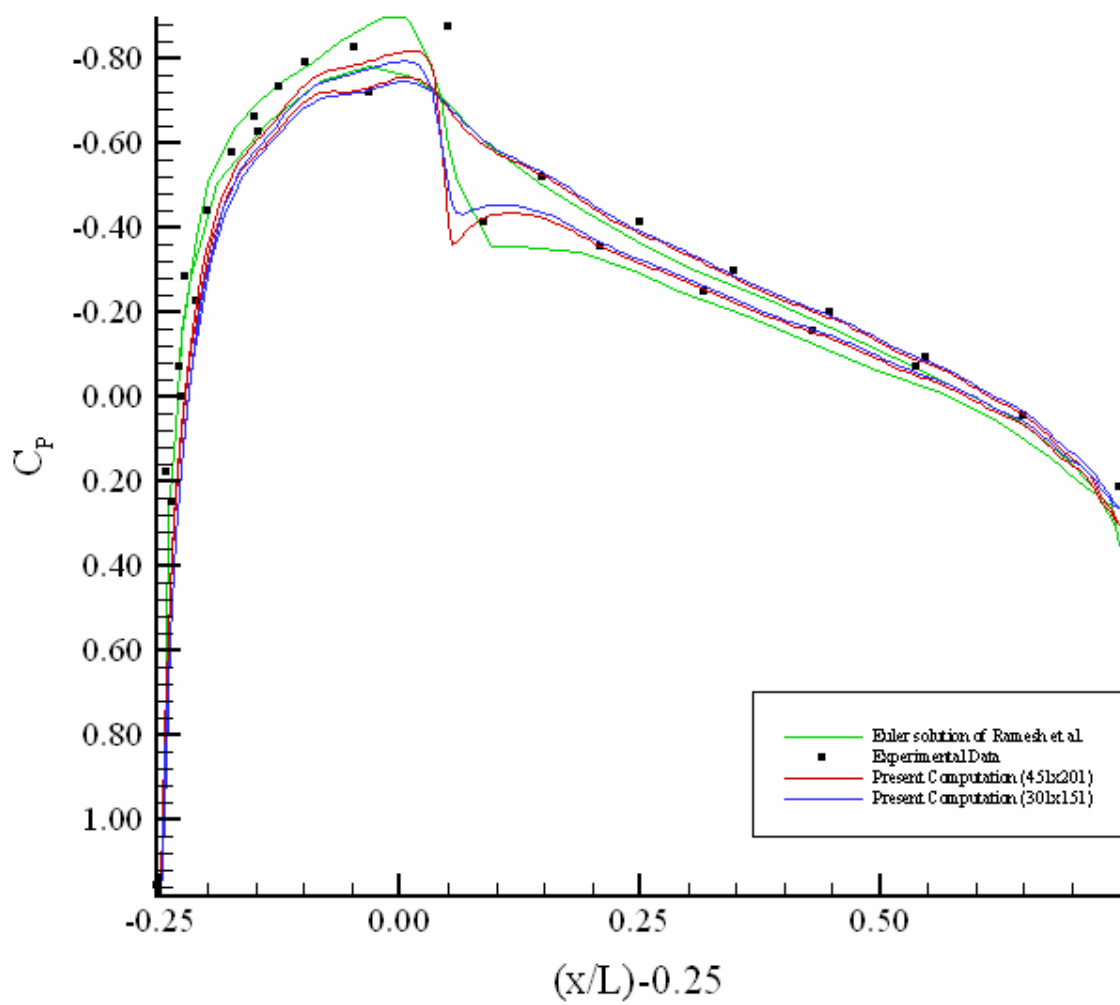


Figure 4.28 Instantaneous c_p distribution, $\alpha = 1.090^\circ$, case 4.2.a

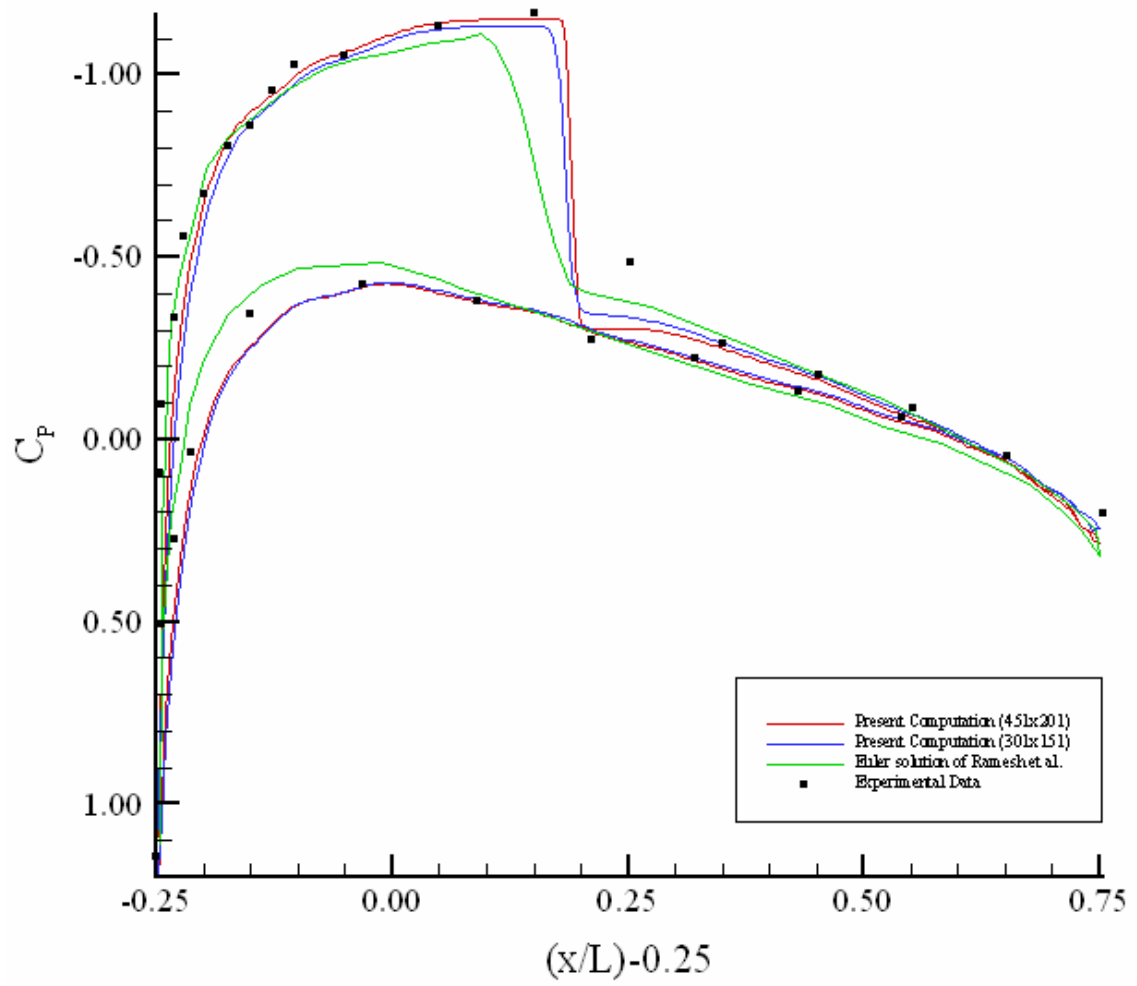


Figure 4.29 Instantaneous c_p distribution, $\alpha = 2.340^\circ$, case 4.2.b

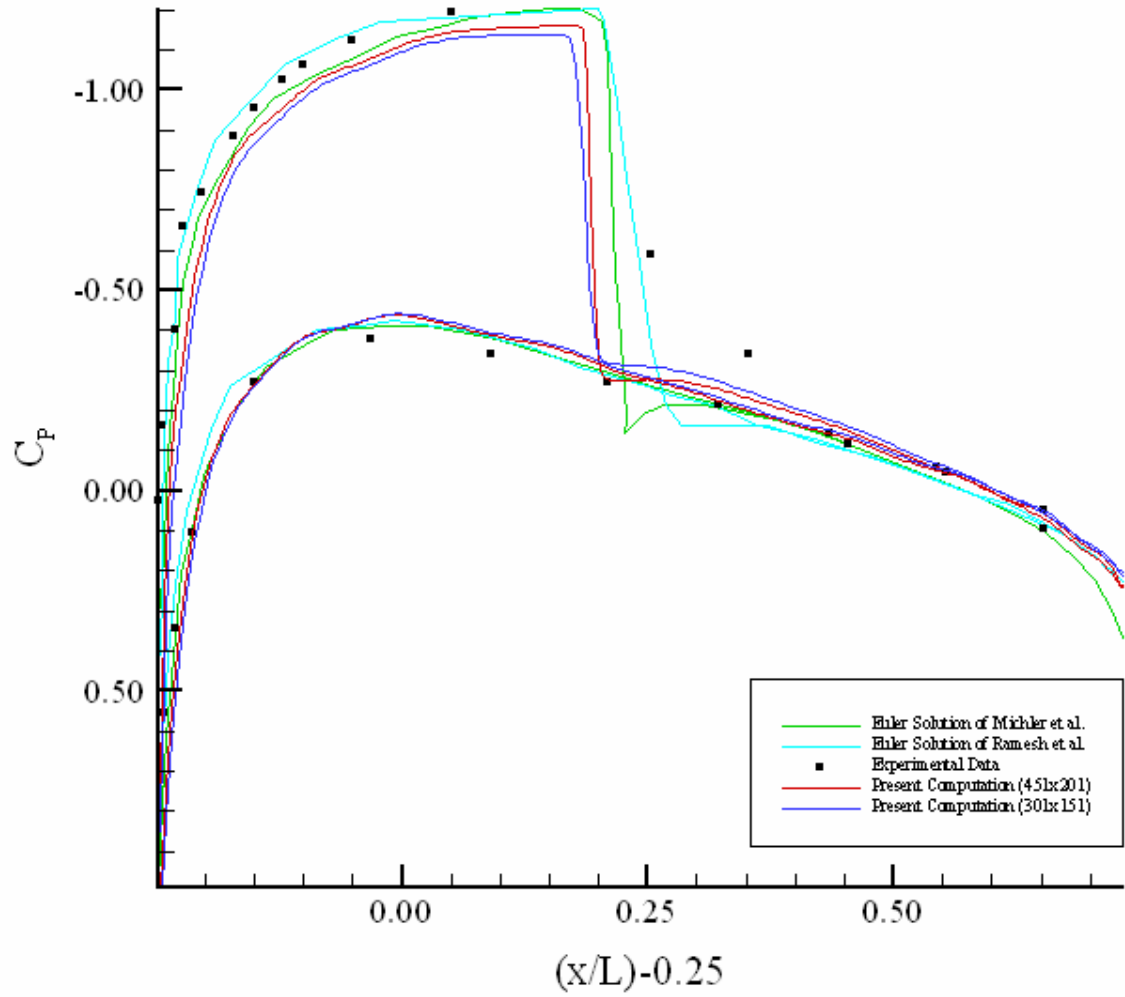


Figure 4.30 Instantaneous c_p distribution, $\alpha = 2.010^\circ$, case 4.2.c

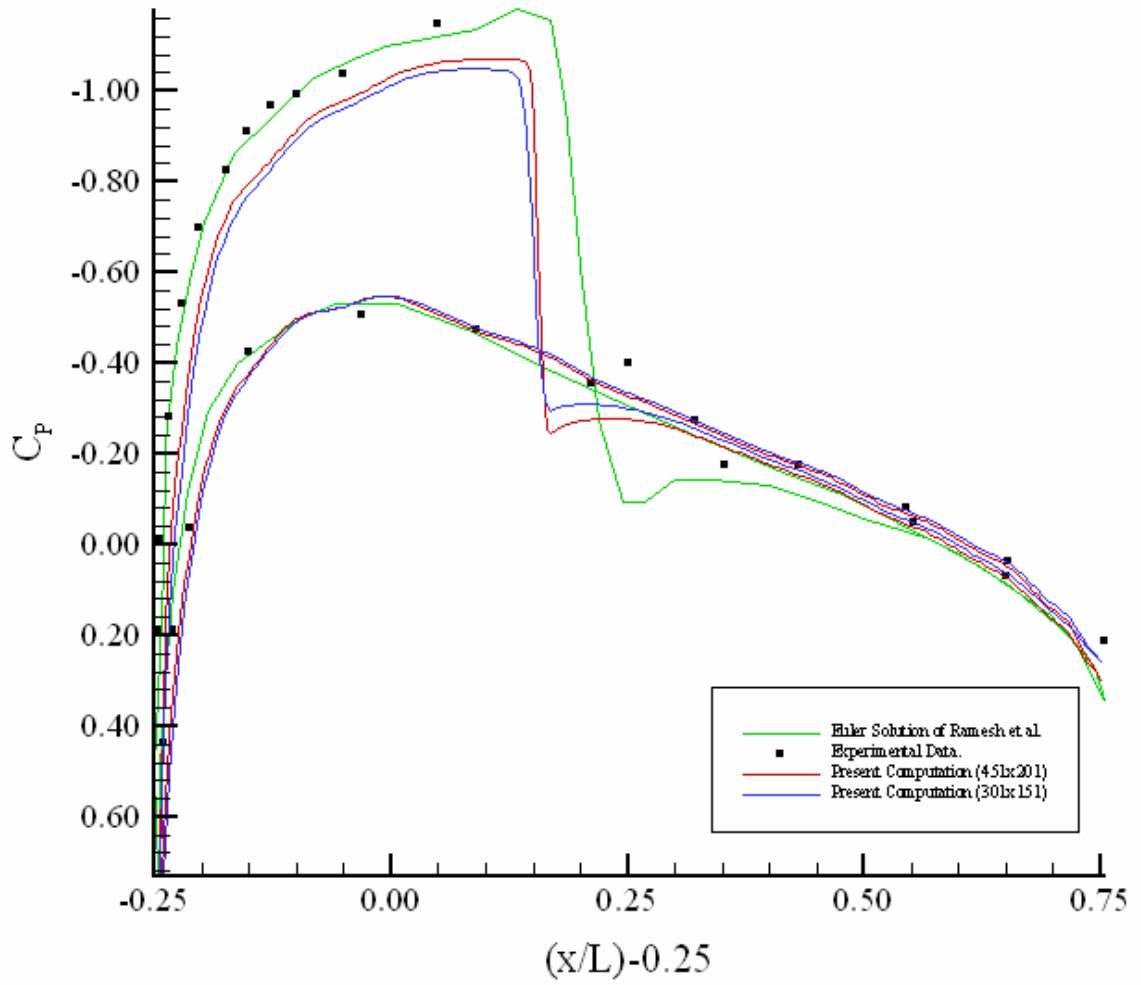


Figure 4.31 Instantaneous c_p distribution, $\alpha = 0.520^\circ$, case 4.2.d

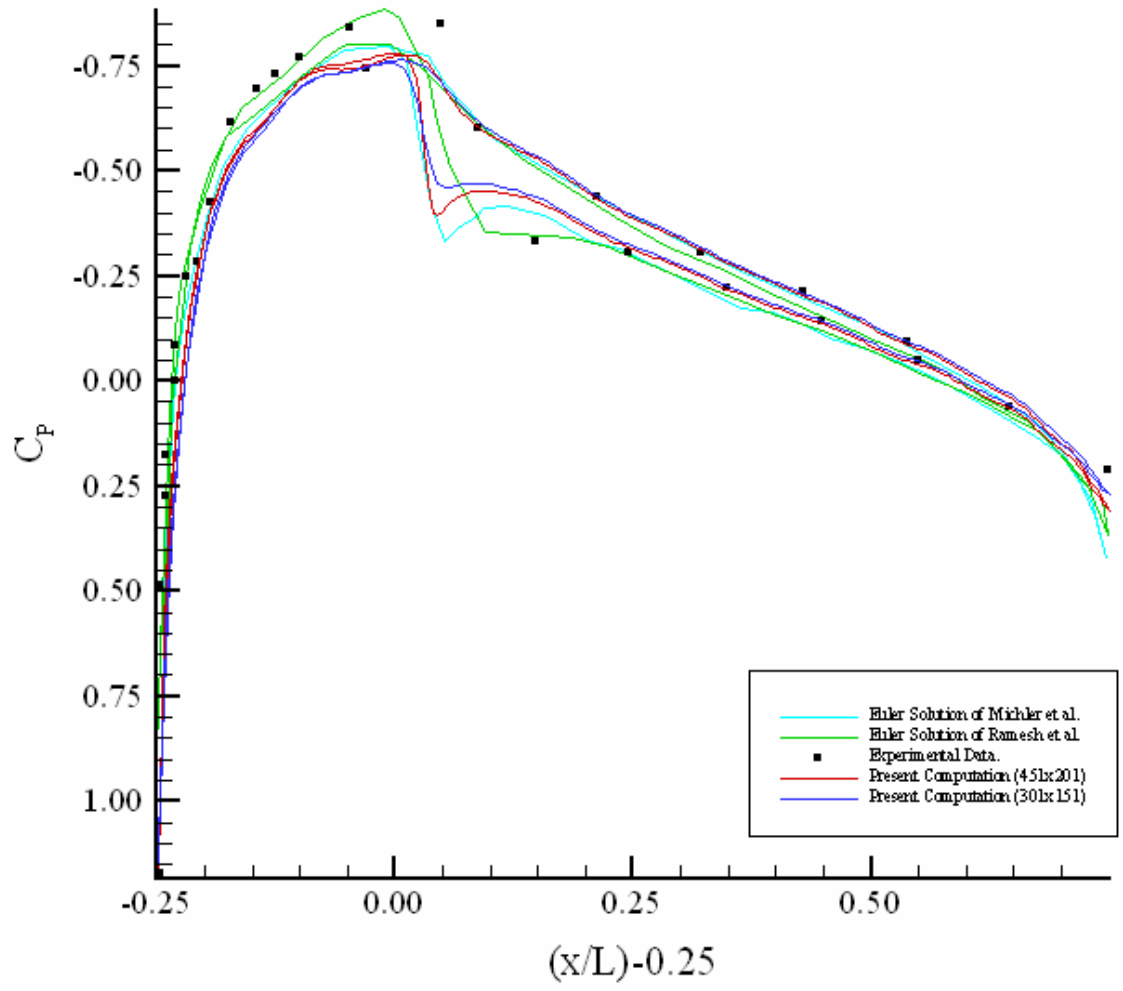


Figure 4.32 Instantaneous c_p distribution, $\alpha = -1.250^\circ$, case 4.2.e

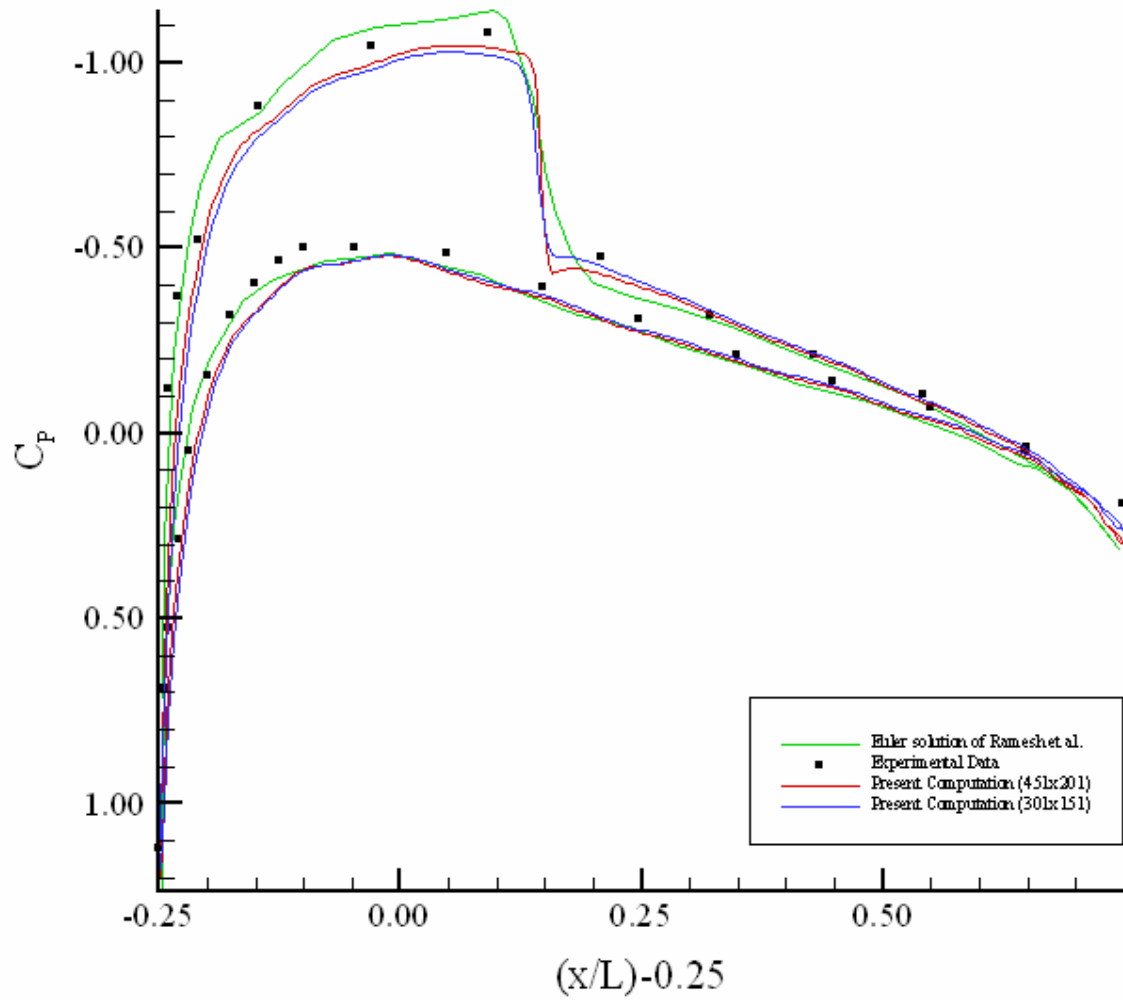


Figure 4.33 Instantaneous c_p distribution, $\alpha = -2.410^\circ$, case 4.2.f

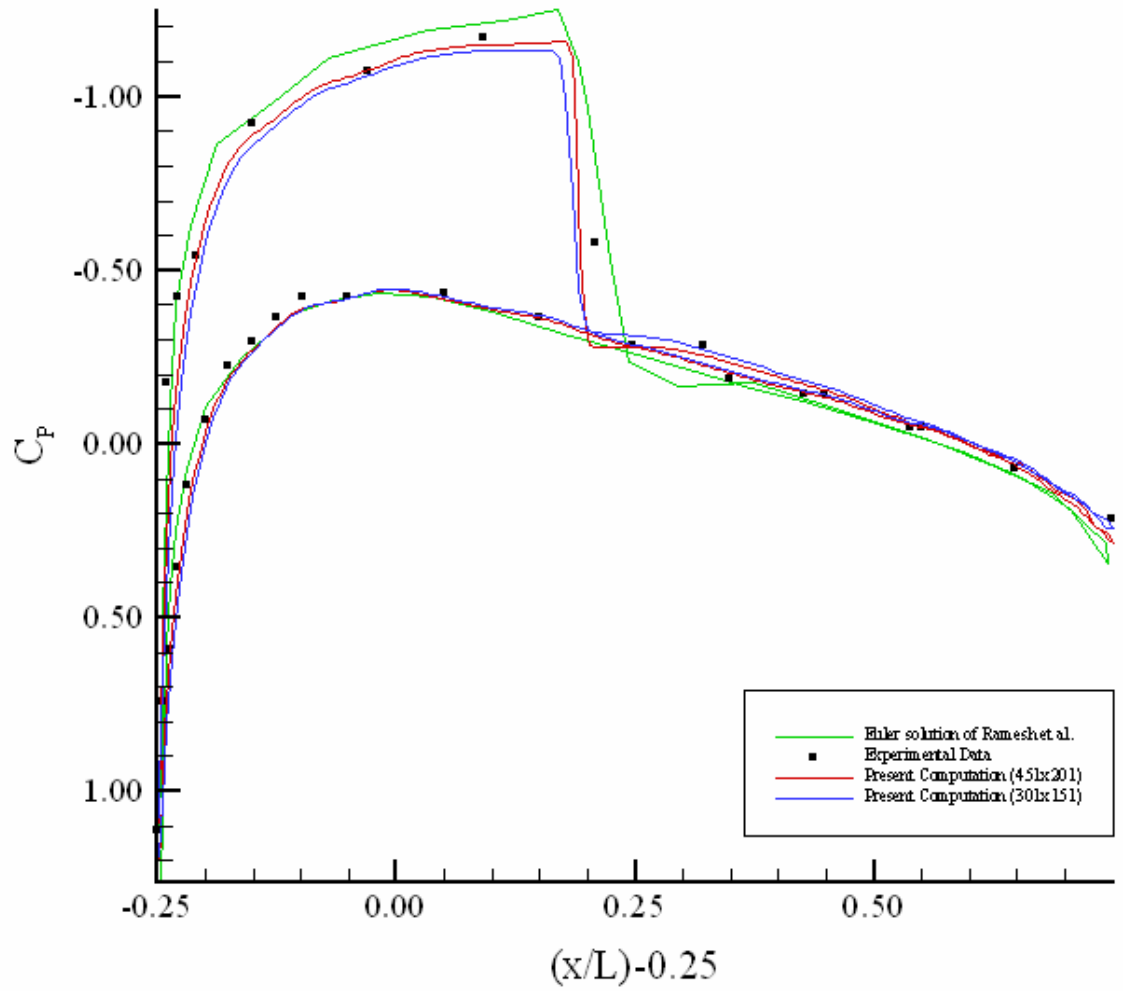


Figure 4.34 Instantaneous c_p distribution, $\alpha = -2.000^\circ$, case 4.2.g

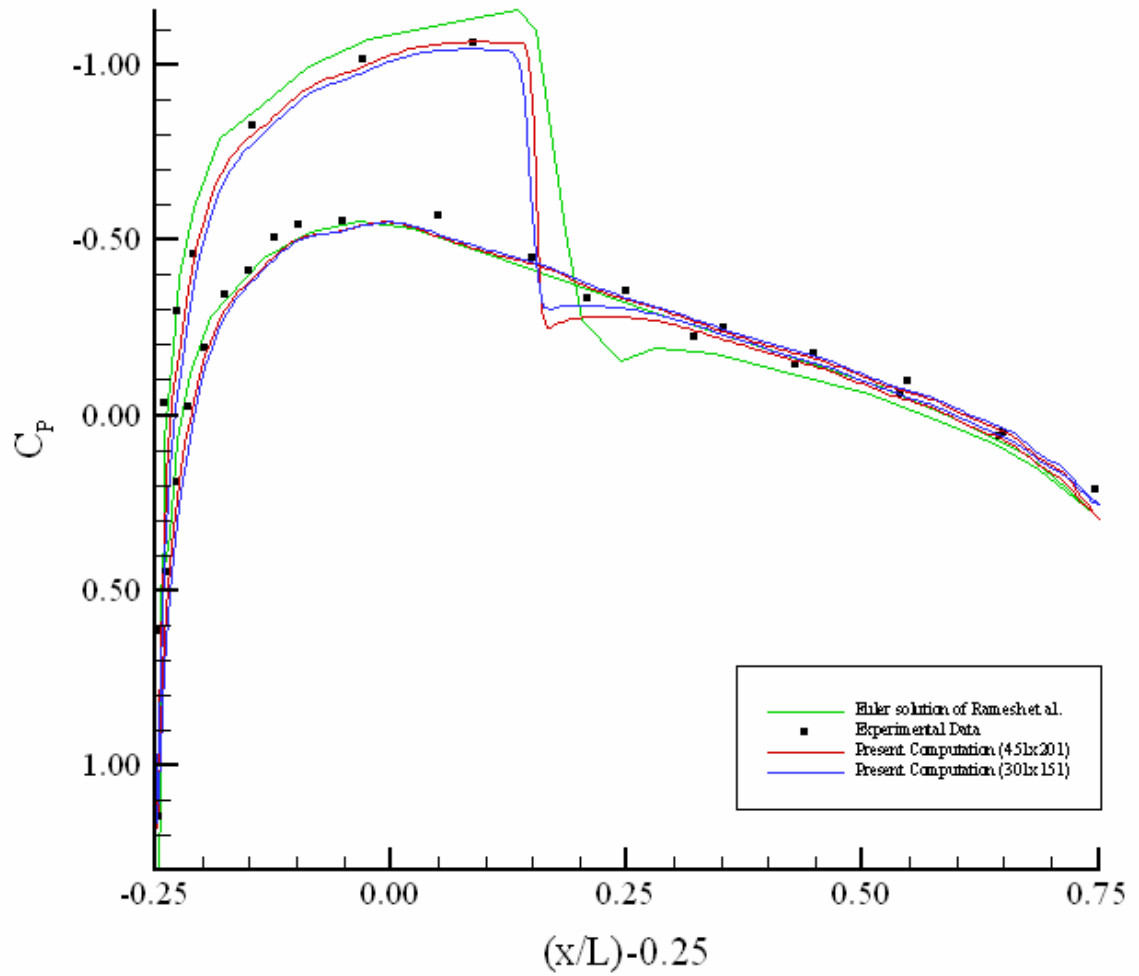
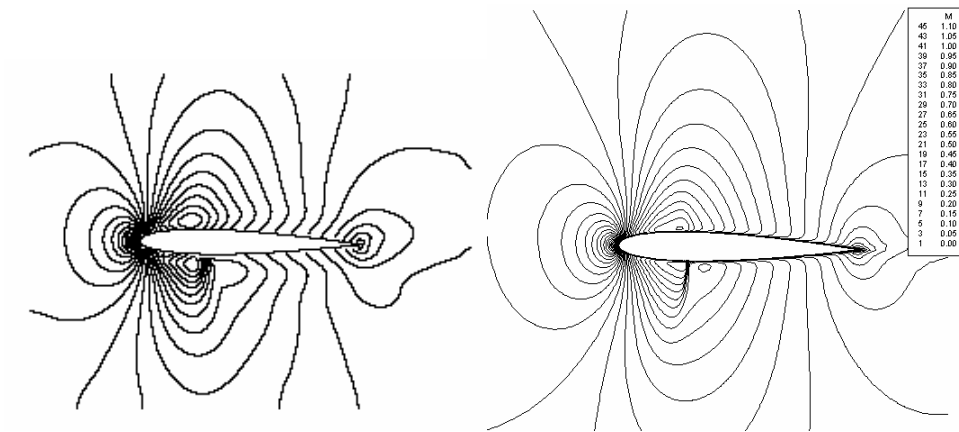
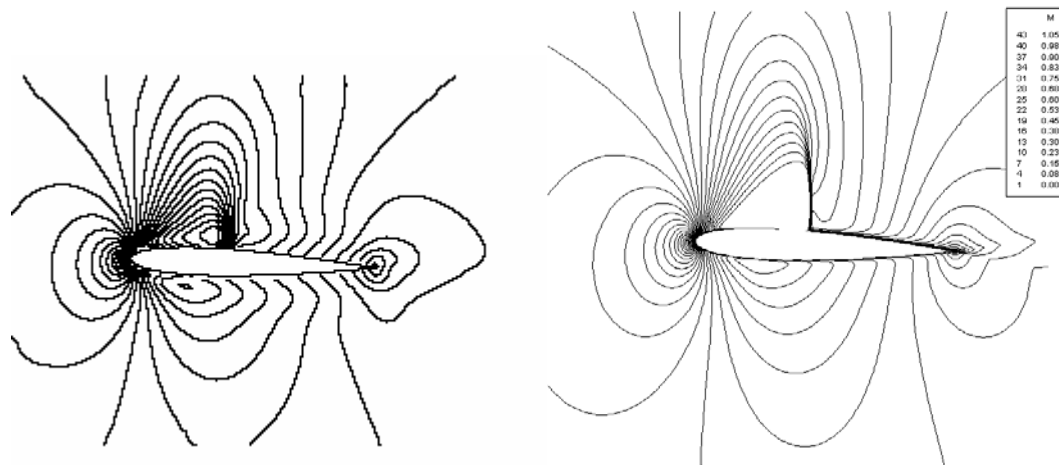


Figure 4.35 Instantaneous c_p distribution, $\alpha = -0.540^\circ$, case 4.2.h

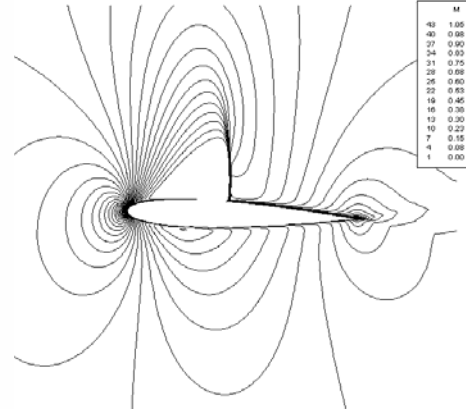
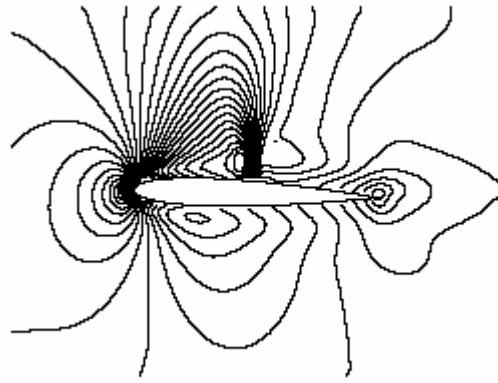
Figures 4.36 – 4.43 presents the Mach contours of Ramesh et al. [35] and the present solver.



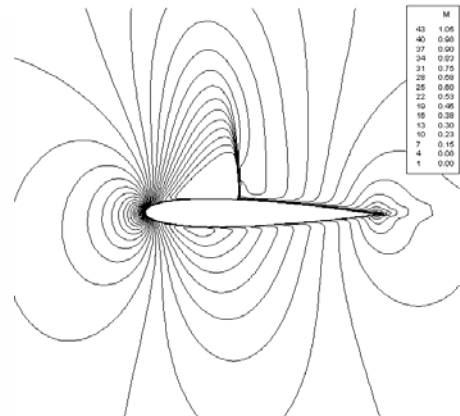
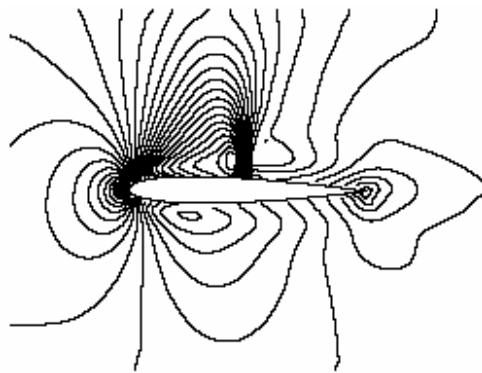
(a) Ramesh et al. (b) Current computation (451x201)
 Figure 4.36 Instantaneous Mach contours, $\alpha = 1.090^\circ$, case 4.2.a



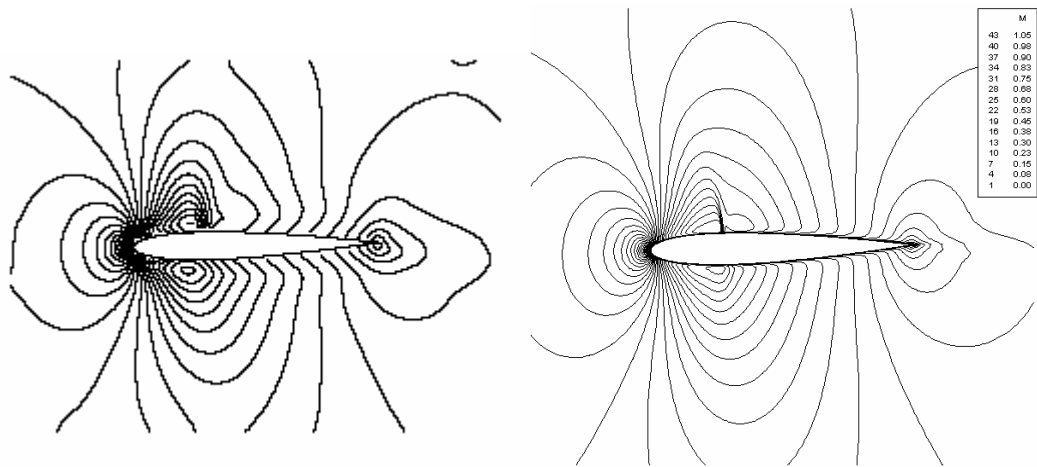
(a) Ramesh et al. (b) Current computation (451x201)
 Figure 4.37 Instantaneous Mach contours, $\alpha = 2.340^\circ$, case 4.2.b



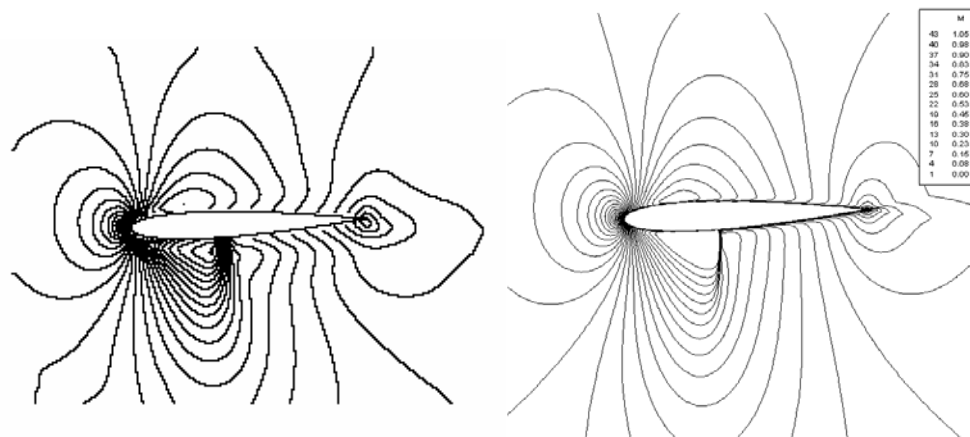
(a) Ramesh et al. (b) Current computation (451x201)
 Figure 4.38 Instantaneous Mach contours, $\alpha = 2.010^\circ$, case 4.2.c



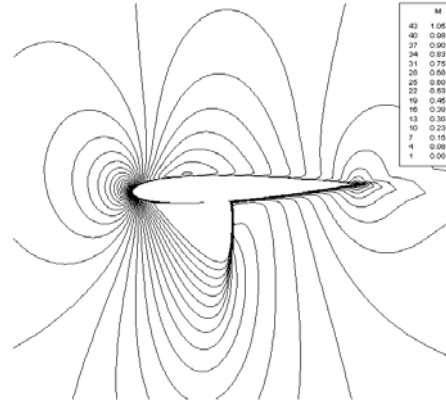
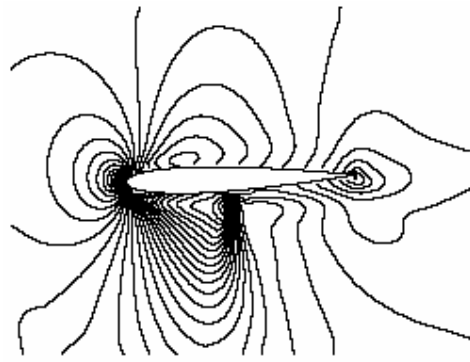
(a) Ramesh et al. (b) Current computation (451x201)
 Figure 4.39 Instantaneous Mach contours, $\alpha = 0.520^\circ$, case 4.2.d



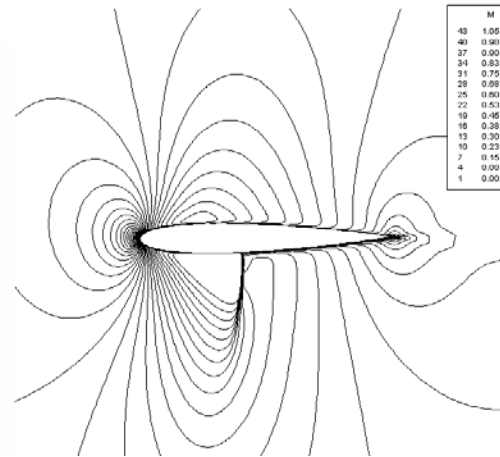
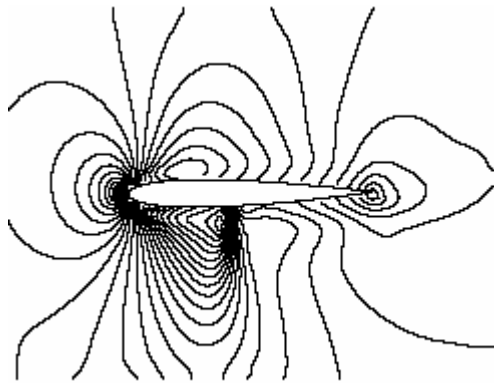
(a) Ramesh et al. (b) Current computation (451x201)
 Figure 4.40 Instantaneous Mach contours, $\alpha = -1.250^\circ$, case 4.2.e



(a) Ramesh et al. (b) Current computation (451x201)
 Figure 4.41 Instantaneous Mach contours, $\alpha = -2.410^\circ$, case 4.2.f



(a) Ramesh et al. (b) Current computation (451x201)
 Figure 4.42 Instantaneous Mach contours, $\alpha = -2.000^\circ$, case 4.2.g



(a) Ramesh et al. (b) Current computation (451x201)
 Figure 4.43 Instantaneous Mach contours, $\alpha = -0.540^\circ$, case 4.2.h

In Figure 4.44, the unsteady lift coefficient versus angle of attack is plotted. The experimental results, which are again from Landon [36], are taken after two complete cycles. Thus, the experimental results are only a function of angle of attack.

It could be said that, the general trend matches with the inviscid computational results of Michler et al.; that is to say, the periodicity is reached after two cycles and the general characteristics of the curves are the same. Furthermore, the trend matches also with the experimental data.

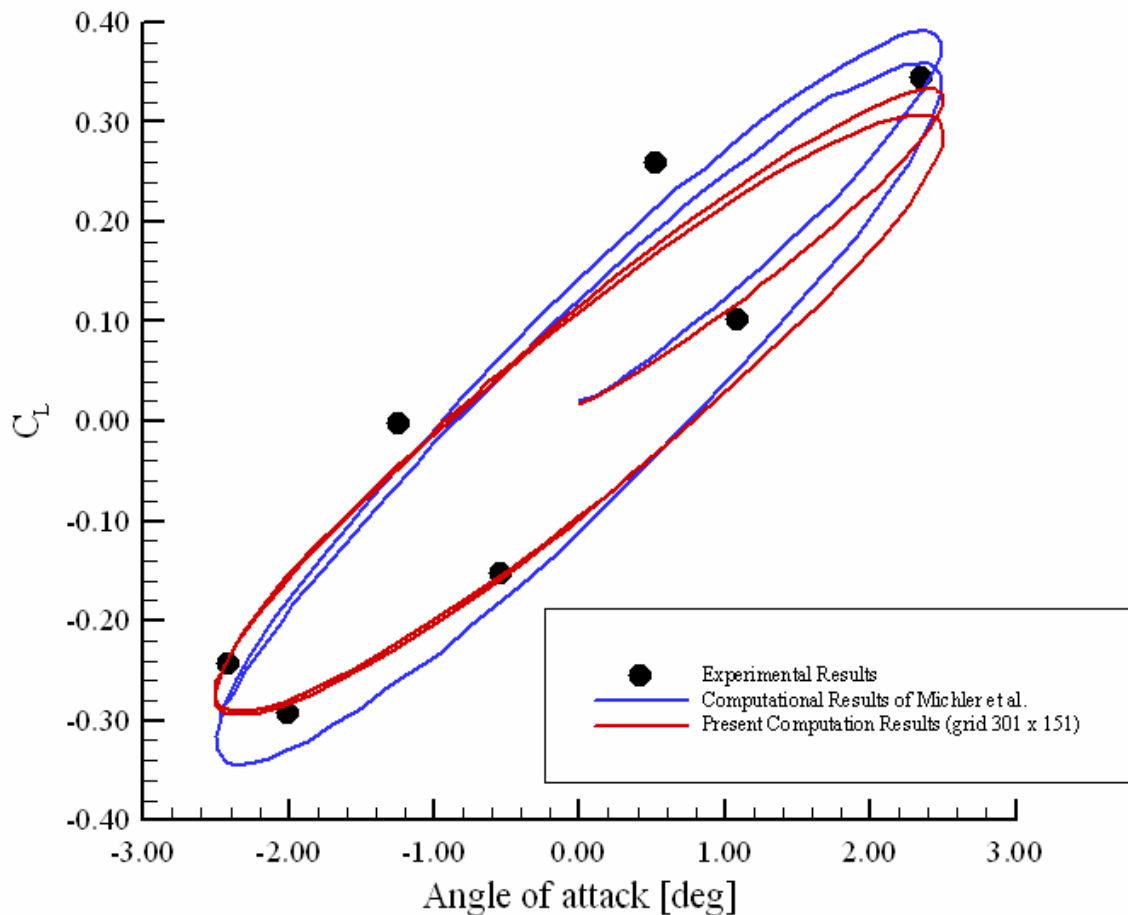


Figure 4.44 Comparison of C_L versus Angle of Attack values

To conclude this problem, Figures from 4.28 to 4.44 present the results of present computation and compare it with experimental data and/or with computational work in the literature. After investigating these test cases, it can be concluded that, the present computations have acceptable results. Some shock locations does not exactly match, which is the direct result of insufficient grid resolution. To capture the shock locations better, shock adaptive solution strategy can be used, so that the intensity of the grids will be increased at the locations where the probability of shock increases. However, this is beyond the scope of this thesis, and will be listed for future improvements.

CHAPTER 5

CONCLUSION

*“It is good to have an end to journey toward,
but it is the journey that matters in the end.”*

Ursula K. LeGuin

*“I may not have gone where I intended to go,
but I think I have ended up where I intended to be.”*

Douglas Adams

This thesis contains numerical simulations of steady and unsteady flows around 2-D airfoils. The unsteady flows consist of flows around oscillating airfoils. The solutions have been acquired using a FORTRAN code, which solves Euler equations. The solver uses the cell centered finite volume method with first order Roe’s flux splitting scheme for spatial discretisation, and a third order Runge-Kutta explicit time stepping scheme for temporal discretisation. The unsteady behavior is simulated with the help of Arbitrary Lagrangian Eulerian method, where the computational grid could be moved but could not be deformed.

The assumptions of the flow are listed previously as inviscid, unsteady, adiabatic, and two dimensional. The gravity is neglected and the air is taken as ideal gas.

The obtained results throughout the study are in well agreement with the literature. It can be argued that mismatches with the results of literature in steady problems could be eliminated by using better computational grids. However, even very fine meshes could not predict the pressure maximum for RAE 2822 airfoil, which may be due to neglecting of viscous effects in the solver. On the other hand, to obtain better results for the unsteady solver future improvements, which are listed below, should be done.

First improvement for the unsteady solver as for the steady one should be the replacement of Euler equations by Navier-Stokes (NS) equations. This will include the effect of viscosity. A further improvement could be to use one of the turbulence models to include the effect of turbulence, which could be useful at high angle of attacks, where the viscous effects dominate the flow.

Second modification should be the generalization of the code to 3-D. This will enable to simulate the flow more realistically and the flow around propellers or turbines could be solved.

Third enhancement of the code could be the implementation of shock adaptive solution strategy into the solver. This enhancement directly requires the satisfaction of Geometric Conservation Law, since the computational grid will not be rotated only but also be deformed.

Another modification could be implementation of characteristic boundary conditions. It is known that this method is superior to the method used in this study. However, the method used in this study gives satisfactory results.

The last improvement would be implementing higher order spatial discretisation, which generally helps for a better prediction of shock waves.

The outcome of this thesis could be useful in understanding the behavior of rotor aerodynamics. Furthermore, if the previously mentioned improvements are done, the obtained solver could be used for analysis of turbines or propellers.

REFERENCES

- [1] Aksel, M. H., “ME 485 Computational Fluid Dynamics Using Finite Volume Method Lecture notes”, Mechanical Engineering Department, METU, 2003
- [2] Versteeg, H.K., Malalasekera W., “An introduction to computational fluid dynamics - The finite volume method”, Longman, 1995
- [3] Ferziger, J.H., Perić, M., “Computational Methods for Fluid Dynamics”, Springer, 1996
- [4] Gönç, L. O., “Computation of External Flow Around Rotating Bodies”, Ph.D. Thesis, Mechanical Engineering Department, METU, 2005
- [5] Hirsch, C., “Numerical Computation of Internal and External Flows – Volume 1: Fundamentals of Numerical Discretization”, John Wiley & Sons, 1988
- [6] Thompson, K. W., “Time Dependent Boundary Conditions for Hyperbolic Systems, II”, Journal of Computational Physics, 89, pp. 439-461, 1990
- [7] Poinso, T. J., Lele, S. K., “Boundary Conditions for Direct Simulations of Compressible Viscous Flows”, Journal of Computational Physics, 101, pp. 104-129, 1992
- [8] Dadone, A., Grossman, B., “Surface Boundary Conditions for the Numerical Solution of the Euler Equations”, AIAA Journal, Vol. 32, No. 2, pp.285-293, 1994
- [9] Thompson, K.W., “Time Dependent Boundary Conditions for Hyperbolic Systems II”, Journal of Computational Physics, 89, pp. 439-461, 1990
- [10] Alpman, E., “Numerical Simulation of Rotary Wing Flow Fields on Parallel Computers”, M.Sc. Thesis, Aeronautical Engineering Department, METU, 2001
- [11] Hirt, C.W., Amsden, A., Cook, J. L., “An Arbitrary Lagrangian-Eulerian Computing Method for All Flow Speeds”, Journal of Computational Physics, 14, pp. 227-253, 1974
- [12] Aksel, M. H., “Fluid Mechanics” Vol. 1, METU Press, 1999

- [13] White, F. M., "Fluid Mechanics", Mc Graw Hill, 1994
- [14] Trepanier, J. Y., Reggio, M., Zhang, H., Camarero, R., "A Finite Volume Method for the Euler Equations on Arbitrary Lagrangian-Eulerian Grids", *Computers & Fluids*, Vol.20, No.4, pp. 399-409, 1991
- [15] Uzun, A., "Parallel Computations of Unsteady Euler Equations on Dynamically Deforming Unstructured Grids", Ph.D. Thesis, Mechanical Engineering Department, Purdue University, 1999
- [16] Darlington, R. M., "Large Eddy Simulation of Rayleigh-Taylor Instability using the Arbitrary Lagrangian-Eulerian Method", Ph.D. Thesis, Engineering - Applied Science, University of California, 1999
- [17] Batina, J. T., "Unsteady Euler Algorithm with Unstructured Dynamic Mesh for Complex-Aircraft Aerodynamic Analysis", *AIAA Journal*, Vol.29, No.3, pp. 327-333, 1991
- [18] Michler, C., Sterck, H. D., Deconnick, H., "An Arbitrary Lagrangian Eulerian Formulation for Residual Distribution Schemes on Moving Grids", *Computers & Fluids*, 32, pp. 59-71, 2003
- [19] Uzun, A., Akay, H.U., Bronnenberg, C.E., "Parallel Computations of Unsteady Euler Equations on Dynamically Deforming Unstructured Grids", *Proceedings of the Parallel CFD'99' Conference*, May 23-26, 1999, Williamsburg, VA USA
- [20] Thomas, P. D., Lombard, C. K., "Geometric Conservation Law and Its Application to Flow Computations on Moving Grids", *AIAA Journal*, Vol.17, No.10, pp. 1030-1037, 1979
- [21] Toro, E.F., "Riemann Solvers and Numerical Methods for Fluid Dynamics", Springer, 1999
- [22] Roe, P.L., "Characteristic Based Schemes For The Euler Equations" *Ann. Rev. Fluid Mech.*, 18, pp. 337-365, 1986
- [23] De Zeeuw D., Powell K. G., "An Adaptive Cartesian Mesh Method for the Euler Equations", *Journal of Computational Physics*, Vol. 104, pp. 56-68, 1993
- [24] Hosseini, R., Rahimian, M.H., Mirzaei, M., "Performance of High-Accuracy Schemes in Inviscid Fluxes Calculation", unpublished paper, Mechanical Eng. Department, University of Tehran

- [25] Ak, M. A., "Analysis of Transient Regimes in Solid Rocket Propulsion", Ph.D. Thesis, Mechanical Engineering Department, METU, 2001
- [26] Sah, J. Y., Chae, E. M., CAF2D/GENGRID version 1.1, Computer Applied Fluid Laboratory, Yeuugnam University, Korea
- [27] Lovy, D., WinDIG Version 2.0, Department of Physical Chemistry, University of Geneva, Switzerland
- [28] Tuncer İ. H., "Lecture Notes of AE546 - CFD on Unstructured Grids", Aerospace Engineering Department, METU, 2003
- [29] Dođru, K., "Parallel Processing of Two Dimensional Euler Equations for Compressible Flows", Msc thesis, Mechanical Engineering Department, METU, 2000
- [30] Amick, J. L., "Comparison of the Experimental Pressure Distribution on an NACA 0012 Profile at high Speeds with that calculated by the relaxation method", National Advisory Committee for Aeronautics, Technical note 2174, 1950
- [31] Wu, Z. N., Li, K., "Anisotropic Cartesian grid method for steady inviscid shocked flow computation", International Journal for Numerical Methods in Fluids, 41, pp. 1053-1084, 2003
- [32] Kudinov, P. I., "Compressible flows on RAE 2822 transonic airfoil", <http://xoptimum.narod.ru/eng/results/compressible/rae2822/rae2822.htm>, Last date accessed: 16/09/2005, 2004
- [33] Kudinov, P.I., "Comparative testing of turbulence models of Spalart-Allmaras and Menter on the problem of transsonic flow about of single aerofoil RAE 2822", Bulletin of Dnepropetrovsk University-Mechanics, issue 8 vol.1, pp.34-42, 2004
- [34] Cook, P.H., McDonald, M. A., Firmin, M. C. P., "Aerofoil RAE 2822 - Pressure Distributions, and Boundary Layer and Wake Measurements - Experimental Data Base for Computer Program Assessment", AGARD Report AR 138, 1979
- [35] Ramesh, V., Deshpande, S.M., "Least Squares Kinetic Upwind method on moving grids for Unsteady Euler Computations", Computers & Fluids, 30, pp. 621-641, 2001
- [36] Landon, R.H., "Compendium of unsteady aerodynamic measurements", AGARD, AGARD-R-702, 1982

[37] “ME-410 Experiment 6, Characteristics of an Airfoil”, Laboratory Instruction Sheet, Mechanical Engineering Department, METU, 2005

APPENDIX A

SAMPLE INPUT FILES

Sample input file for O-grid solver

```
301 151
0.7500000000000000    0.0000000000000000
0.752879590139900    0.0000000000000000
0.755859965934698    0.0000000000000000
.....
.....
14.047057139317422    0.0000000000000000
14.515333729333431    0.0000000000000000
15.0000000000000000    0.0000000000000000
```

Key for sample input file for O-grid solver

```
NX NY
X(1,1)          Y(1,1)
X(1,2)          Y(1,2)
X(1,3)          Y(1,3)
.....
.....
X(NX,NY-2)     Y(NX,NY-2)
X(NX,NY-1)     Y(NX,NY-1)
X(NX,NY)       Y(NX,NY)
```

Sample input file for C-grid solver

```

601 121 151 301
  15.000000000000000      0.000000000000000
  14.999999999999998     -0.000117050412220
  15.000000000000000     -0.000243464857418
  .....
  .....
  15.000000000000000     12.859873572653100
  15.000000000000000     13.888780508877574
  15.000000000000000     15.000000000000000
  
```

Key for sample input file for C-grid solver

```

NX NY NSS NSP
      X(1,1)                Y(1,1)
      X(1,2)                Y(1,2)
      X(1,3)                Y(1,3)
      .....
      .....
      X(NX,NY-2)           Y(NX,NY-2)
      X(NX,NY-1)           Y(NX,NY-1)
      X(NX,NY)              Y(NX,NY)
  
```

Sample input file for O-grid solver (moving mesh)

```

301 151
0.7500000    0.0000000    1.1177106    157.7848782    0.0365164    295416.7704199
0.7432783    -0.0010051    1.1241846    175.9842514    24.8984617    297154.1586574
0.7365415    -0.0019041    1.1294371    200.4848514    26.8676770    298941.2852078
  .....
  .....
  .....
  
```

Key for sample input file for O-grid solver (moving mesh)

NX NY

X(1,1) Y(1,1) $\rho(1,1)$ $\rho(1,1)u(1,1)$ $\rho(1,1)v(1,1)$ $\rho(1,1)e(1,1)$

X(1,2) Y(1,2) $\rho(1,2)$ $\rho(1,2)u(1,2)$ $\rho(1,2)v(1,2)$ $\rho(1,2)e(1,2)$

X(1,3) Y(1,3) $\rho(1,3)$ $\rho(1,3)u(1,3)$ $\rho(1,3)v(1,3)$ $\rho(1,3)e(1,3)$

.....

.....

.....

APPENDIX B

AIRFOIL PROFILES

As the NACA 0012 geometry is symmetric, the lower and upper surface values have the same magnitude but the inverse sign. The second column in Table B.1 indicates the upper halve, and the third indicates the lower halve of the NACA 0012 profile.

Table B.1 The Geometry of NACA 0012 profile

x/L	y/L	y/L
0.00000	0.00000	0.00000
0.00390	0.01090	-0.01090
0.01570	0.02110	-0.02110
0.03510	0.03040	-0.03040
0.06180	0.03880	-0.03880
0.09550	0.04590	-0.04590
0.13550	0.05170	-0.05170
0.18130	0.05590	-0.05590
0.23210	0.05860	-0.05860
0.28710	0.05960	-0.05960
0.34550	0.05910	-0.05910
0.40630	0.05730	-0.05730
0.46860	0.05420	-0.05420
0.53140	0.05020	-0.05020
0.59370	0.04540	-0.04540
0.65450	0.04010	-0.04010
0.71290	0.03450	-0.03450
0.76790	0.02870	-0.02870
0.81870	0.02310	-0.02310
0.86450	0.01770	-0.01770
0.90450	0.01280	-0.01280
0.93820	0.00840	-0.00840
0.96490	0.00490	-0.00490
0.98430	0.00220	-0.00220
0.99610	0.00060	-0.00060
1.00000	0.00000	0.00000

The second column in Table B.2 indicates the lower halve and the fourth column indicates the upper halve of the RAE 2822 profile.

Table B.2 The Geometry of RAE 2822 profile

x/L	y/L	x/L	y/L
1.00000	0.00000	0.00000	0.00000
0.99940	0.00004	0.00060	0.00316
0.99759	0.00014	0.00241	0.00631
0.99459	0.00030	0.00541	0.00942
0.99039	0.00050	0.00961	0.01248
0.98502	0.00072	0.01498	0.01549
0.97847	0.00093	0.02153	0.01844
0.97077	0.00111	0.02923	0.02135
0.96194	0.00121	0.03806	0.02422
0.95200	0.00120	0.04801	0.02706
0.94096	0.00103	0.05904	0.02987
0.92886	0.00069	0.07114	0.03264
0.91574	0.00016	0.08427	0.03536
0.90160	-0.00060	0.09840	0.03801
0.88651	-0.00159	0.11349	0.04059
0.87048	-0.00283	0.12952	0.04307
0.85355	-0.00431	0.14645	0.04546
0.83578	-0.00605	0.16422	0.04773
0.81720	-0.00803	0.18280	0.04987
0.79785	-0.01024	0.20215	0.05188
0.77778	-0.01269	0.22221	0.05375
0.75705	-0.01536	0.24295	0.05547
0.73570	-0.01823	0.26430	0.05703
0.71378	-0.02129	0.28622	0.05841
0.69134	-0.02449	0.30866	0.05963
0.66845	-0.02781	0.33156	0.06066
0.64514	-0.03121	0.35486	0.06150
0.62149	-0.03463	0.37851	0.06213
0.59754	-0.03804	0.40245	0.06256
0.57336	-0.04140	0.42663	0.06278
0.54901	-0.04464	0.45099	0.06277
0.52453	-0.04772	0.47547	0.06253
0.50000	-0.05056	0.50000	0.06203
0.47547	-0.05310	0.52453	0.06125
0.45099	-0.05526	0.54901	0.06019

Continues at the next page

x/L	y/L
0.42663	-0.05698
0.40245	-0.05822
0.37851	-0.05897
0.35486	-0.05924
0.33156	-0.05905
0.30866	-0.05846
0.28622	-0.05755
0.26430	-0.05638
0.24295	-0.05499
0.22221	-0.05343
0.20215	-0.05169
0.18280	-0.04981
0.16422	-0.04777
0.14645	-0.04561
0.12952	-0.04333
0.11349	-0.04093
0.09840	-0.03843
0.08427	-0.03584
0.07114	-0.03317
0.05904	-0.03042
0.04801	-0.02759
0.03806	-0.02469
0.02923	-0.02172
0.02153	-0.01871
0.01498	-0.01565
0.00961	-0.01256
0.00541	-0.00944
0.00241	-0.00631
0.00060	-0.00316
0.00000	0.00000

x/L	y/L
0.57336	0.05885
0.59754	0.05722
0.62149	0.05534
0.64514	0.05326
0.66845	0.05099
0.69134	0.04857
0.71378	0.04603
0.73570	0.04338
0.75705	0.04064
0.77778	0.03785
0.79785	0.03502
0.81720	0.03218
0.83578	0.02935
0.85355	0.02655
0.87048	0.02382
0.88651	0.02115
0.90160	0.01858
0.91574	0.01611
0.92886	0.01377
0.94096	0.01156
0.95200	0.00951
0.96194	0.00762
0.97077	0.00592
0.97847	0.00440
0.98502	0.00309
0.99039	0.00200
0.99459	0.00114
0.99759	0.00051
0.99940	0.00013
1.00000	0.00000

APPENDIX C

CALCULATION OF PRESSURE, LIFT AND DRAG COEFFICIENT

The pressure coefficient will be calculated in each node and cell center as below:

$$C_p^{(I,J)} = \frac{\frac{P_{(I,J)} - P_\infty}{(M_\infty)^2 \frac{\gamma}{2}}}{-1} \quad (\text{C.1})$$

For numerical integration of drag and lift coefficients the simple trapezoidal rule is used. The pressure coefficient average of successive nodes are taken and assumed that that value is valid for the whole interval. The drag and lift coefficients on the solid body are calculated as below [37]:

$$C_L = \sum_{i=1}^{NX-1} \frac{(C_p^{(i,1)} + C_p^{(i+1,1)})(X(i+1,1) - X(i,1))}{(-2)} \quad (\text{C.2})$$

$$C_D = \sum_{i=1}^{NX-1} \frac{(C_p^{(i,1)} + C_p^{(i+1,1)})(Y(i+1,1) - Y(i,1))}{(2)} \quad (\text{C.3})$$