SENSITIVITY ANALYSIS USING FINITE DIFFERENCE AND ANALYTICAL JACOBIANS

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

BY

AHMET ALPER EZERTAS

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

SEPTEMBER 2009

Approval of the thesis:

SENSITIVITY ANALYSIS USING FINITE DIFFERENCE AND ANALYTICAL JACOBIANS

submitted by **AHMET ALPER EZERTAŞ** in partial fulfillment of the requirements for the degree of **Master of Science in Aerospace Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen _____
Dean, Graduate School of **Natural and Applied Science**

Prof. Dr. İsmail H. Tuncer _____
Head of Department, **Aerospace Engineering**

Assoc. Prof. Dr. Sinan Eyi _____
Supervisor, **Aerospace Engineering Dept., METU**

Examining Committee Members:

Prof. Dr. M. Cevdet Çelenligil _____
Aerospace Engineering Dept., METU

Asoc. Prof. Dr. Sinan Eyi _____
Aerospace Engineering Dept., METU

Prof. Dr. İ. Sinan Akmandor _____
Aerospace Engineering Dept., METU

Asst. Prof. Dr. Oğuz Uzol _____
Aerospace Engineering Dept., METU

Prof. Dr. Ünver Kaynak _____
Aerospace Engineering Dept., TOBB ETU

Date _____

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 and results that are not original to this work.

Name, Last Name: Ahmet Alper Ezertaş

Signature:

ABSTRACT

SENSITIVITY ANALYSIS USING FINITE DIFFERENCE AND ANALYTICAL JACOBIANS

Ezertaş, Ahmet Alper
M.S, Department of Aerospace Engineering
Supervisor : Assoc. Prof. Sinan Eyi
September 2009, 150 pages

The Flux Jacobian matrices, the elements of which are the derivatives of the flux vectors with respect to the flow variables, are needed to be evaluated in implicit flow solutions and in analytical sensitivity analyzing methods. The main motivation behind this thesis study is to explore the accuracy of the numerically evaluated flux Jacobian matrices and the effects of the errors in those matrices on the convergence of the flow solver, on the accuracy of the sensitivities and on the performance of the design optimization cycle. To perform these objectives a flow solver, which uses exact Newton's method with direct sparse matrix solution technique, is developed for the Euler flow equations. Flux Jacobian is evaluated both numerically and analytically for different upwind flux

discretization schemes with second order MUSCL face interpolation. Numerical flux Jacobian matrices that are derived with wide range of finite difference perturbation magnitudes were compared with analytically derived ones and the optimum perturbation magnitude, which minimizes the error in the numerical evaluation, is searched. The factors that impede the accuracy are analyzed and a simple formulation for optimum perturbation magnitude is derived. The sensitivity derivatives are evaluated by direct-differentiation method with discrete approach. The reuse of the LU factors of the flux Jacobian that are evaluated in the flow solution enabled efficient sensitivity analysis. The sensitivities calculated by the analytical Jacobian are compared with the ones that are calculated by numerically evaluated Jacobian matrices. Both internal and external flow problems with varying flow speeds, varying grid types and sizes are solved with different discretization schemes. In these problems, when the optimum perturbation magnitude is used for numerical Jacobian evaluation, the errors in Jacobian matrix and the sensitivities are minimized. Finally, the effect of the accuracy of the sensitivities on the design optimization cycle is analyzed for an inverse airfoil design performed with least squares minimization.

Keywords: CFD, Newton's method, Flux Jacobian, Sensitivity Analysis, Inverse Design Optimization

ÖZ

SONLU FARKLAR VE ANALİTİK JACOBİANLAR KULLANARAK DUYARLILIK ANALİZİ

Ezertaş, Ahmet Alper
Yüksek Lisans, Havacılık ve Uzay Mühendisliği Bölümü
Tez Yöneticisi: Doç. Dr. Sinan Eyi
Eylül 2009, 150 sayfa

Elemanları akı vektörlerinin akış değişkenlerine göre türevleri olan akı Jacobian matrisinin hesaplanması, akışın kapalı çözümünde ve analitik duyarlılık analizinde gereklidir. Bu çalışmadaki ana hedefler: sayısal Jacobian matrisinin doğruluğunu araştırmak ve Jacobian matrisindeki hatanın, akış çözücüsünün yakınsamasına, duyarlılık analizinin doğruluğuna ve tasarım en iyileştirilmesi döngüsünün verimliliğine olan etkilerini araştırmaktır. Bu amaçlar doğrultusunda Euler akış denklemleri için, Newton metoduna dayalı akış çözücü, seyrek matris çözücü kullanarak geliştirilmiştir. Akı Jacobian matrisi farklı akı bölme teknikleri ve yüksek dereceli ayrıştırmalar için sayısal ve analitik yöntemlerle hesaplanmıştır. Farklı sonlu farklar değiştirme miktarlarıyla hesaplanan sayısal Jacobian matrisleri analitik Jacabian ile kıyaslanmış, sayısal Jacobian

matrisindeki hatayı en aza indirgeyen en iyi değiştirme miktarı araştırılmıştır. Sayısal Jacobianın doğruluğunu düşüren etkenler analiz edilmiş ve en iyi değiştirme miktarı için basit bir formül türetilmiştir. Duyarlılık türevleri, ayrık yaklaşımla direkt-türev yöntemi kullanarak hesaplanmıştır. Akış çözümünde elde edilen LU çarpanlarının yeniden kullanımıyla duyarlılık analizinin verimliliği arttırılmıştır. Sayısal Jacobian kullanarak hesaplanan duyarlılık türevleri analitik Jacobian ile hesaplanan türevlerle kıyaslanmıştır. İç ve dış akış problemleri; çeşitli akış hızları için, farklı çözüm ağı biçimleri ve boyutlarıyla, farklı ayrışım şemaları kullanarak çözülmüştür. Bu problemlerin çözümünde sayısal Jacobian en iyi değiştirme miktarıyla hesaplandığında, sayısal Jacobian matrisindeki ve duyarlılık analizindeki hata miktarı en küçük değerlere düşmüştür. Son olarak, duyarlılık türevlerinin tasarım en iyileştirme döngüsüne olan etkisini incelemek maksadıyla en küçük kareler yöntemiyle tersine kanat profili tasarımı gerçekleştirilmiştir.

Anahtar Kelimeler: Hesaplamalı Akışkanlar Dinamiği, Newton metodu, Akı Jacobianı, Duyarlılık Analizi, Tersine Tasarım En iyileştirmesi

to my parents

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Assoc. Prof. Dr. Sinan Eyi for his valuable guidance and support throughout my research.

I wish to express my appreciation to my thesis committee members, Prof. Dr. M. Cevdet Çelenligil, Prof. Dr. İ. Sinan Akmandor, Prof. Dr. Ünver Kaynak and Asst. Prof. Dr. Oğuz Uzol for reviewing my work and for their valuable comments.

I would like to thank to the entire academic staff of the Department of Aerospace Engineering; all the instructors for giving their best for my academic education.

Special thanks to my research assistant colleagues, Onur Tarımcı, Tahir Turgut, Emre Altuntas, Deniz Yılmaz and Gökhan Ahmet for their friendship and sharing an enjoyable working environment.

I am thankful to my parents for their encourgement, trust, understanding and patience.

And finally, I am grateful to my sweetheart Oznur for her endless love and the joy she brings to my life. Without her love and support the completion of this thesis would not have been possible.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

LATIN SYMBOLS

A	Jacobian matrix of F flux vector
B	Jacobian matrix of G flux vector
c	Speed of sound
C_p	pressure coefficient
CP	convergence parameter for inverse design
e	Canonical vector
e_t	Total energy per unit volume
E_R	Round-off error
E_C	Condition error
E_{Trun}	Truncation error
E_{total}	Total error
F_c	Inviscid flux vector
F_v	viscous flux vector
F	Inviscid flux vector in i-direction
G	Inviscid flux vector in j-direction
h	Enthalpy
J_c	inviscid flux Jacobian
M	Mach number
p	Pressure
$\mathbb{R}$	Riemann invariant
Q_Λ	Right eigenvector matrix of A, B
r	Ratio of differences
$R(w)$	Residual vector of the system
$\tilde{R}(w)$	Computed value of $R(w)$
s	Entropy
ΔS	length of the surface element on aerodynamic geometry
Δt	Time-like term added to Jacobian matrix diagonal
Δt_0	Initial value for Δt

Δt_f	Removal value for Δt
u,v	Velocity components
U,V	Contravariant velocity components
x,y	Components of Cartesian coordinates
w	Vector of conserved flow variables in Cartesian coordinates

GREEK SYMBOLS

α	angle of attack
β	design variables
Δ	Forward difference operator
δ	high order face interpolation
ε	Finite-difference perturbation magnitude
ε_M	Computer precision or machine epsilon
ε_{OPT}	Optimum value for ε
ϕ	Limiter function
γ	Ratio of specific heats
η	face normal vector
κ	Interpolation order parameter
λ	Eigenvalue
Λ	Diagonal matrices including eigenvalues of A, B
ρ	Density
ζ	A value between the original and perturbed variable
∂	Partial differentiation operator
$\in$	A small number used in limiter functions to prevent division by zero
∇	Backward difference operator
Ω	cell volume

OVERLINES

~	Roe averaged value

SUBSCRIPTS

in	inner domain

i,j	Cell centred grid indices
n	normal direction
max	maximum value
min	minumum value
∞	Free-stream value

SUPERSCRIPTS

L,R	Left and right state value
m	Number of possible highest bits in the binary representation of mantissa
n	Newton's method iteration number
-	Negative (left) value
+	Positive (right) value

CHAPTER 1

INTRODUCTION

1.1 Background

Temporal discretization techniques have always been one of the most popular research areas in Computational Fluid Dynamics (CFD) since the choice of the discretization methodology directly affects the cost and efficiency of the flow analysis. Three decades ago explicit schemes were the main alternatives due to the limitations on computational speeds and memory but by the beginning of the nineties the usage of implicit methods became common with the developments in computational science. Explicit schemes are easy to program with simple algorithms however they lack of stability. On the other hand the implicit schemes provide more stability. Today CFD has much wider range of application area than that is intended earlier. Most of the recent design tools are CFD based and they are coupling the solution of multidiscipline problems in a single analysis. In a design process, multiple flow solutions may be needed therefore the efficiency of the flow solver becomes a greater concern. Although the usage of explicit multigrid schemes can still be more favorable in some class of unsteady problems, today implicit methods based on Newton's method have taken the dominant role in the flow analysis.

Newton's method is a well-known implicit method and has been used extensively for finding the root of the non-linear system of equations. In a steady-state flow solution, the objective is to evaluate the flow variables that result in zero flux residuals. Hence Newton's root finding technique for the nonlinear systems can be applied to the steady state flow solutions. Newton's method is a powerful method due to the quadratic convergence rate it provides. However, for this superior convergence performance, the exact linearization of the residuals is needed. The linearization of the residuals requires the calculation of flux Jacobian matrices, which compose of the derivatives of the flux vectors with respect to the flow variables. The evaluation, factorization and the storage

of the Jacobian matrix are the main concerns in the solution with Newton's method. For example, in three-dimensional problems that require large number of grid points, the size of the Jacobian matrix can be enormous and large computer memory may be needed. The techniques used to deal with these concerns directly influence the efficiency and the accuracy of the Newton's method.

The cost of the Newton's method in solution of the large linear system of the flow equations prevented it from being a widespread technique although; they were robust and providing quadratic convergence. Moreover the promising results obtained with quasi Newton methods encouraged the researchers to completely leave the direct solution method. By the end of the 90's almost all the research on the implicit techniques were shifted into quasi Newton's methods since the exact Newton's method was extremely expensive with available computer technology level. Quasi Newton methods perform the solution of the large linear systems iteratively at each Newton's step. The quasi Newton methods which has approximations only in solutions of linear system is called approximate-Newton methods whereas the methods which also use simplified Jacobian with approximate linearization is called in-exact Newton methods. With the simplifications made in quasi Newton's method the demand on CPU can be much smaller but the penalty will be the drop down to linear convergence rate from quadratic rate. Moreover there is a need on preconditioner, which is derived from simplified Jacobian, and it can be crucial on the performances of the iterative solution of the linear system.

In last decade there was intensive research in applied mathematics focusing on development of efficient algorithms to solve large sparse systems. The motivation of developing faster solver with smaller memory demand resulted with various sparse matrix solvers. UMFPACK, WMSP, MUMPS and PARDISO are the some of the solvers that can perform effectively with various types of sparse matrices. The combination of advances in hardware and algorithms makes it possible to solve those sparse linear systems quickly and easily that might have been considered too large until recently. Hence the usage of direct methods in flow solution can regain attention in next decade although they have not been used since early 90's.

The common point in the Newton's method base flow solution techniques is the need of flux Jacobian evaluation whether the method is exact or quasi. Although the Jacobian matrix is not even constructed in some quasi Newton methods such as Jacobian free GMRES, the elements of the Jacobian is still needed to calculate matrix-vector products. The accuracy of the Jacobian is important since the accuracy may determine the exactness of the linearization. It will be possible to obtain better convergence rates with more accurate Jacobian. One of the main objectives of this study is to analyze the effects of the accuracy of the numerical Jacobian on the convergence of the Newton method. In order to eliminate the effects of the approximation made in Jacobian, an exact Newton method is used.

There are mainly two alternatives for calculating the Jacobian matrix: analytical and numerical methods. Analytical evaluation is more accurate but differentiating by hand or using symbolic manipulators needs effort and it is time consuming. Moreover for each different flux discretization scheme, Jacobian derivation procedure needs to be re-performed. Numerical Jacobian can be evaluated easily by finite differencing the residual vector. Although this method is simple and independent from the complexity of the flux scheme, numerical evaluation has a lack of accuracy. Errors in the numerical Jacobian calculations, which strongly depend on the finite difference perturbation magnitude, may hamper the convergence of Newton's method. The error is large for both small and large perturbation magnitudes. In small magnitudes of perturbation, condition error is dominant whereas truncation error becomes dominant in large magnitudes. In order to evaluate the most accurate numerical Jacobian, the optimum perturbation magnitude should be used.

Gradient-based optimization algorithms are very popular techniques used in the aerodynamic design. Although there are more robust alternatives such as genetic algorithms they are still chosen by many researchers since they are much faster. The efficiency and the accuracy of the design optimization with gradient-based methods mainly depend on the performances of the solution methods used for flow and sensitivity analyses. The derivatives of design objective function with respect to design variables are called sensitivities. The reliability of design results depends on the ability to accurately calculate the flow and sensitivity variables. In a design process, multiple flow

and sensitivity evaluations may be required; therefore, the efficient methods for flow and sensitivity analyses may improve the performance of the design optimization.

In early design applications, sensitivities were evaluated by brute-force method, which was based on finite differencing. Although the brute-force method was easy to implement it suffers from accuracy due to errors rising from finite differencing. Moreover the requirement of multiple flow solution for each design variables makes this method expensive and impractical. The major advance in aerodynamic design optimization area was the introduction of the analytical methods, which provides accurate evaluation of sensitivities in an efficient manner. In analytical methods, sensitivities are calculated by differentiating the governing equations with respect to the design variables. There are two approaches in analytical evaluation. In the first that is called discrete approach, the discretized flow equations are differentiated. Continuum approach is the second one in which the differentiation is performed prior to discretization. Both approaches is subdivided into the direct-differentiation and adjoint methods. In direct differentiation method, the discretized residual equations are differentiated with respect to design variables, and the resulting equations are solved for flow variable sensitivities. In adjoint method, the discretized residual equations are introduced as constraint functions, and the system of equations is solved for adjoint variables. In the design problems where the number of system responses of interest is larger than the number of design variables the direct-differentiation method is favorable in terms of computational work. Otherwise the adjoint method is more advantageous since the solution for adjoint variable is independent from the number of design variables.

In implicit methods, analytical sensitivity calculations require the construction of Jacobian matrix and the solution of linear systems. Therefore, using Newton's method for flow analysis is especially profitable since the Jacobian matrix which is constructed for flow analyses can also be used for sensitivity analyses and the same solution scheme can be used for the flow and sensitivity equations. Resulting from this behavior, the most of the researchers who were dealing with Newton's method for flow solution are also interested in the aerodynamic design optimization today. However, there is not much research on sensitivity analysis with the direct flow solver although it has some

significant benefits. Most of the today's direct sparse solvers stores the LU factors and allows the reuse of them with different right-hand side vectors. Hence sensitivities can be calculated very efficiently by using the LU decomposed Jacobian matrix with different right hand side vectors for various design variables.

In the solution of flow equations with Newton's method the flux Jacobian is used as a driver to the iterative procedure. Even tough the accuracy of flux Jacobian affects the convergence behavior of Newton's method; it does not affect the accuracy of converged solution. However, in calculation of flow variable sensitivities with analytical methods, the flux Jacobian is the part of linear systems of sensitivity equations. Therefore the accuracy of the sensitivities strongly depends on the accuracy of the flux Jacobian.

1.2 Objectives

This thesis study has multiple objectives. One of the objectives is to examine the applicability of the exact Newton's method with the recent direct sparse matrix solvers. UMFPACK multi-frontal solver is used in the study. Another objective is to analyze the errors involved in numerical Jacobian calculations. The entries of the numerically evaluated Jacobian are compared with the ones correspond to analytically evaluated Jacobian. The other objective is to evaluate the optimum finite difference perturbation magnitude that minimizes the error in numerical Jacobian evaluation. The effects of the perturbation magnitude on the accuracy of numerical Jacobian are studied. The other objective of this thesis study is to analyze the effect of the accuracy of the numerically evaluated flux Jacobian on the convergence of the flow solver, on the accuracy of the sensitivities and on the performance of the design optimization cycle.

1.3 Literature Survey

In early nineties with the improvements in computational science the interest in implicit methods and coupling of implicit flow solutions with the design optimization were born.

The most promising technique among the implicit techniques was the Newton's method with its appealing quadratic convergence performance. Moreover the usage of Newton's method was bringing great flexibility for coupling the equations of multidisciplinary problems and also sensitivity analysis could be performed very efficiently with usage of Newton's method.

One of the first implementations of Newton's algorithms for flow solutions was performed by Wigton [1]. In his study Newton's method was applied for transonic flow solution over multielement airfoils since he faced with convergence problems with the conventional techniques. Wigton used symbolic derivation tool for the evaluation of flux Jacobian matrices. In the solution of large linear sparse system he described the nested dissection node reordering technique, which provided efficient results, and this method is still forming the basis for the some of modern sparse matrix solvers.

Bender and Khosla [2] dealt with the initial conditions of Newton's method, which directly affect the convergence behavior. They suggested variational time step method and modified Newton scheme to reduce the sensitivity of Newton iterations to initial condition of the Jacobian matrix.

In 1989 Venkatakrishnan [3] published his research on Newton solution of inviscid and viscous problems. His detailed research guided most of the post researches on Newton's method. Compared to Wigton's study, he used improved sparse matrix solution technique with diagonal term addition to increase the stability of the scheme. Venkatakrishnan's work showed that Newton's method was robust with quadratic convergence rate. However, he stated that the Newton's method was impractical due to its large memory demand as the size of the problem increases for the application such as three-dimensional solutions. This problem firstly stated by Venkatakrishnan and in the following years researchers directed into quasi Newton methods, which were cheaper to use with the convergence rate reduction penalty.

After the Venkatakrishan's study, there are a few researches performed with exact Newton's method. Van Dam et al.[4] used direct solution technique for separating viscous flows. Orkwis [5] developed a Newton's method to solve flow around

axisymetric/planar geometries [6] with inviscid and laminar/turbulent flow equations [7]. In those studies, 2nd order Roe discretization scheme whose corresponding Jacobian matrix is hard to derive was used. Orkwis implemented MACSYMA symbolic differentiation package into his code for analytical evaluation of Jacobian. He reported that the outputs of the symbolic manipulation system should be simplified with care to prevent round-off errors. Orkwis [8] compared the performances of the exact and quasi Newton methods in his later work and showed that quasi Newton methods are much more efficient in terms of CPU time spent although their convergence rate was linear compared to exact method which has quadratic convergence. Moreover in his following work with Kim [9], Orkwis analyzed the feasibility of using modified Newton methods with frozen Jacobian.

Whitfield and Taylor [10] are among the first researchers who use numerically evaluated flux Jacobian in Newton's method solver. They solved 3-D incompressible and compressible flows with high order ROE discretization. Vanden [11], [12] presented 3-D flow solutions with direct and iterative methods and showed that the iterative ones outperformed in terms of time required for solution. Vanden and Orkwis [13] compared the performances of numerical and analytical Jacobian matrices in exact Newton method solution. Analytical Jacobian was derived by MACSYMA as it was done in earlier research of Orkwis and the numerical Jacobian matrices are evaluated by finite differencing. That study showed that the convergence performance of the numerical and analytical Jacobian matrices were almost identical. Authors stated that for simple linear systems analytical evaluation can be favorable whereas for the linearization of complex schemes numerical evaluation will be the best choice.

In quasi Newton's method, approximation is made on the solution of large linear system arising from Newton's formulation. Moreover The Jacobian matrix can also be simplified. Hence the range of options available in the quasi Newton's method is very wide depending on the choice of iterative procedure, choice of the preconditioning technique and the way of simplifying the Jacobian matrix. Hence there are great varieties of publications in the literature relevant to quasi Newton's methods. The most popular class of techniques is the Newton-Krylov methods with GMRES [14] iterative solver.

Venkatakrishnan [15], Mavriplis [16] and Rogers [17] used approximate Newton's method by applying GMRES with first orderly approximated Jacobian. Forstyh and Jiang [18] compared the quasi Newton methods and presented the outcomes of the simplifications made on the Jacobian. They showed that an inexact Newton method, which uses same order of discretization in the residual and the Jacobian, is more effective than approximate methods although the required preconditioner is more expensive to store. Brown and Saad [19] introduced matrix free GMRES method. The method proposed by them does not require the storage of the Jacobian matrix hence it permits the true linearization of the residuals without storage limit. However, the requirement on Jacobian factorization cannot be avoided totally since it is needed by the defined ILU preconditioner. Nielsen [20] and Barth [21] used first order Jacobian in their preconditioner for their matrix free GMRES solvers. Although the studies mentioned in this paragraph are associated with quasi Newton methods and they are not directly related to the objective of this thesis, they present valuable information on the importance of the accuracy of linearization of the residual vector.

In the recent decade there were almost no research using direct solution for Newton's method in CFD area. It was shown that that direct solution was unaffordably expensive in terms of required time and memory. Due to this fact today quasi Newton methods are the common ones that the researchers are focusing on. However in last decade the applied mathematicians have introduced new sparse matrix solvers with advanced algorithms. By the end of nineties Davis [22] introduced his multifrontal sparse matrix solver UMFPACK. Amestoy and Duff [23] introduced MUMPS for distributed memory parallel machines. These two were widely used in various research areas till the introduction of the WSMP and PARDISO developed by Gupta [24] and Schenk [25], respectively. In their publications, Gupta and Schenk reported that WSMP and PARDISO outperform the other solvers in the solutions of systems with variety of sparse matrix patterns. There is an intense research going on for upgrading the available sparse matrix solvers and those studies with recent results are promising that direct solution of Newton's method may regain attention in next decade.

Eyi and Onur [26] used UMFPACK for the exact Newton solution of Euler equations for a supersonic flow problem on a ramp geometry. In their research the analytical Jacobian

is derived from first order Steger-Warming fluxes and compared with the numerically evaluated Jacobian. Recently Gelfgat [27] argued the benefits of using direct solution algorithms in terms of stability and he utilized the MUMPS for the sparse matrix inversion in his study. T'ien and Raju [28] demonstrated the performance of multifrontal solvers for combustion problems. They used UMFPACK to solve the fully coupled linear system and showed that use of direct solvers can significantly reduce the computational time (subject to its memory limitations). They also discussed the feasibility of using multifrontal solvers for three-dimensional problems.

In early design optimization practices, the finite difference method had been widely used for the evaluation of the sensitivities because of its simplicity. However that kind of approach is not only expensive but it is also erroneous. Hence in early nineties some research focused on the analytical calculation of the sensitivities. Analytical sensitivity evaluation was firstly developed by researchers dealing with structural design optimization. The studies presented in [29-32] are well known references in this area. The works presented by Frank and Shubin [33] and later by Narducci et al.[34] are the some of the earliest applications of analytical sensitivity evaluation in fluid flow problems. Rizk [35] published his approach on the simultaneous iterative solutions of flow and design variables. Verhoff and Stookesberry [36] developed an analytical sensitivity calculation for two dimensional Euler equations. Most of the analytical sensitivity evaluation studies were performed by discrete approach which differentiates the discretized flow governing equations to evaluate sensitivities with respect to the design variables. El-banna and Carlson [37] used a discrete method to obtain sensitivities from transonic small disturbance equations. Baysal et al.[38] applied the discrete approach to the two dimensional Euler equations. Korivi et al.[39] used an incremental strategy for calculating discrete sensitivities using the thin-layer Navier-Stokes equations. Jameson [40] introduced and the concept of adjoint method by the usage of control theory for the optimum aerodynamic design. Following the work of Jameson lots of studies performing the aerodynamic design optimization by adjoint method are published. A broad review of sensitivity analysis and shape optimization can be found in [41].

The AGARD report [42] presents many good examples of inverse design optimization applications. Sobieczky [43] published a detailed survey on the methods used in inverse design optimization. Giles and Drela [44] developed an inverse design method based on Euler flow coupled with boundary-layer equations. Malone and Narramore [45] and Brickelbaw [46] published inverse design applications performed with Navier-Stoke's equations. Zingg and Nemec [47],[48] published aerodynamic design applications for airfoils performed with the usage of Newton Krylov approach in flow solutions. Zingg and Leung [49] presented the efficiency gained in design optimization of a wing by the usage of Newton-Krylov approach. Eyi and Lee [50] performed the accuracy study on the inverse design performed by finite difference sensitivities and Eyi [51] investigated the effects of accuracy of finite difference sensitivities on the efficiency of the airfoil design.

1.4 Outline

Chapter 2 introduces the basic theory of the Euler flow equations. The spatial discretization used in the study is explained in detail by the presentation of the formulations used for upwinding schemes and face reconstruction. The ways of Jacobian matrix evaluations are introduced.

Chapter 3 mainly presents the tactics that are followed in the construction of Newton's method. The way of the implementation of boundary and initial conditions are given. Results of the flow solutions performed for the commonly used test cases are presented. Plots which present, the mach contours and the pressure coefficient variation on the geometry are given and compared with reference results. The convergence histories of the solutions performed by the analytically evaluated Jacobian are given. The reaction of the convergence histories to the initial conditioning of the Jacobian and to the deactivation of the limiters in the smooth flow regions, are analyzed.

Chapter 4 presents the detailed error analysis performed for the numerical Jacobian. The source of the errors in the numerical Jacobian is presented. The derivation of the simple formulation, which estimates the optimum finite difference perturbation magnitude and

magnitude of the resulting minim relative error, is given. The variation of the optimum values by the change of different factors such as flow regime, grid resolution, order of discretization etc. is analyzed. The effect of the accuracy of the numerical Jacobian on the convergence of the flow solution is analyzed and the variation of the CPU time spent is tabulated.

In Chapter 5 the objective was to analyze the effect of the accuracy of the numerically evaluated Jacobian on the sensitivity analysis. Both of the numerical and analytical Jacobian matrices are utilized in the sensitivity analysis which is performed by the direct differentiation method. The error plots are given to compare the flow sensitivities calculated by the numerical Jacobian matrices with the one calculated by analytical Jacobian.

Chapter 6 presents the inverse design applied on the airfoil geometry. The sensitivity analysis performed in Chapter5 is used to evaluate the required pressure gradients by the optimization algorithm. The effect of the geometry parameterization on the efficiency of the design is presented by the given results that are calculated using the variety of shape functions. Finally the effect of the finite differencing used in the numerical Jacobian evaluation on the efficiency of the aerodynamic design optimization cycle is presented.

CHAPTER 2

FLOW MODEL

2.1 Introduction

The solution of complex flow physics problems requires advanced flow models with advanced spatial and time discretization techniques. The cost of the analysis amplifies with the increase in the level of complexity of the scheme that is used. Moreover accurate solution for complex flow problems generally demands on advanced computational meshing strategies and highly resolved grids. In this thesis study for the time discretization implicit direct solution technique, which was shown to be computationally expensive by the previous research in literature, is used. Direct solution technique requires the flux Jacobian matrix and as the size of the Jacobian enlarges the requirement on both of the memory and solution time will magnify. The number of flow equations solved and the grid size mainly defines the size of the Jacobian. To prohibit further increase in computational cost of the analysis, relatively simpler flow problems that are modeled with two dimensional Euler flow equations are solved. Usage of inexpensive flow model brought the opportunity for examining the effects of several dependents through the study. Different upwind spatial discretization schemes with varying flux limiters are used for subsonic, transonic and supersonic flows on grids with varying sizes.

The finite volume method with the Steger-Warming [52], Van Leer [53], AUSM [54] and Roe [55] upwind schemes is used for discretization of the Euler equations. The second order accurate face reconstruction of the flow variables are performed by MUSCL [56] interpolation with the utilization of Van Albada's [57] and Venkatakrishnan's [58] continuous limiters. The structured mesh topology is used for both external and internal flow applications at subsonic, transonic and supersonic free-

stream conditions. The usage of structured mesh enabled the simple storage for the elements of the Jacobian matrix and resulted with favorable matrix bandwidth

Besides preventing the high analysis cost, the additional motivation for using Euler flow equations is enabling the analytic hand derivation of flux Jacobian. Therefore it is prevented to use schemes that include discontinuous algebraic expressions, such as those appearing in limiter functions and turbulence models. Although Euler equations are simpler flow model compare to Navier-Stokes equations with turbulence modeling, Euler equations still capable of capturing many high level flow phenomena such as rotational flow and embedded shock waves in transonic flows.

2.2 Governing Equations

The conservation laws of mass, momentum and energy for an arbitrary control volume can be presented by Navier Stokes equations as below:

$$\frac{\partial}{\partial t}\int_{\Omega} \boldsymbol{w}\, dV + \oint_{\partial\Omega} \left(\boldsymbol{F}_c . \boldsymbol{n}\right) dS - \oint_{\partial\Omega} \left(\boldsymbol{F}_V . \boldsymbol{n}\right) dS = 0 \tag{2.1}$$

In Equation 2.1 the bold characters $\boldsymbol{w}, \boldsymbol{F}$ and $\boldsymbol{F}_V$ denotes vector of conservative flow field variables, and convective and dissipative fluxes vectors respectively. In steady flow problems, flow variables inside the control volume do not change with time, hence the first term in Equation 2.1 drops. Moreover in Euler flow modeling, the fluid is assumed to be inviscid and there will be no dissipative flux terms. Therefore, for steady-state Euler equations can be simplified as follow:

$$\oint_{\partial\Omega} \left(\boldsymbol{F}_c . \boldsymbol{n}\right) dS = 0 \tag{2.2}$$

The elements of convective flux vector $\boldsymbol{F}$, which is a function of conservative flow variables $\boldsymbol{w}$, is given in Equation 2.3

$$\boldsymbol{w} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho e_t \end{bmatrix} \qquad \boldsymbol{F}_c = \begin{bmatrix} \rho U \\ \rho u U + \eta_x p \\ \rho v U + \eta_y p \\ (\rho e_t + p) U \end{bmatrix} \tag{2.3}$$

In Equation 2.3 ρ is density, u and v are components of velocity vector, p is pressure, e_t is total energy per unit volume and U is the contravariant velocity aligned in the direction of unit normal vector of the face.

$$\boldsymbol{U} = \eta_x u + \eta_y v \tag{2.4}$$

where η_x and η_y are the components of the unit normal vector.

The system of equations is closed with the introduction of the ideal gas relation for the pressure.

$$p = (\gamma - 1)\rho \left[e_t - \frac{1}{2}(u^2 + v^2) \right] \tag{2.5}$$

Today the most common way of the numerical solution of the flow governing equations is the technique called method of lines [59]. This method introduces the separate discretization in space and in time. In its application, firstly the control volumes are constructed by the proper meshing strategy. Secondly the flux integrals and the spatial derivatives are calculated from the initial conditions. Finally the resulting solution is advanced in time. The solution obtained in last step will be used in the second step to evaluate new fluxes and finally last step will be re-utilized for the updated variables. By this way an iterative procedure will be driven to perform the flow analysis.

2.3 Spatial Discretization

Euler flow equations are discretized by finite volume method by dividing the flow domain into quadrilateral cells. The flow variables are stored at cell centers and for flux

calculation interpolation to the cell faces is applied. The integral formulation given in Equation 2.2 can be written for a mesh cell as :

$$\sum_{n=1}^{\#\,face} \left(\boldsymbol{F}_c . \boldsymbol{dS} \right) = 0 \tag{2.6}$$

The storage of the data corresponding to a structured mesh is simple. Data can be stored according to directions in which cells are aligned. This will be essentially useful in attaining high order discretization accuracy and in ordering the elements of Jacobian matrix. The high order accuracy in spatial discretization can be achieved by the assumption of the varying flow variables in grid cells. Therefore, the gradients of the flow variables are required for the application of high order face interpolation. In structured meshes, these gradients can be evaluated with simple algorithms by using neighboring cells aligned along a continuous mesh lines. The demonstration for the cells and their indices used in this study is given below.

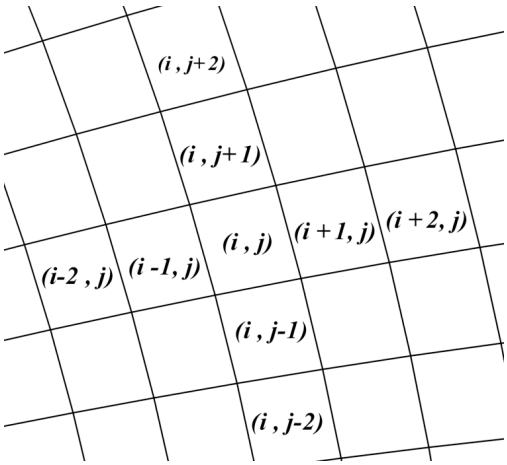

Figure 2.1 ***G****rid cell indices*

The results of the vector product of the fluxes with faces normal vectors, which are aligned in *i* and *j* directions are called as *F* and *G* fluxes, respectively. Using these definitions Equation 2.6 can be re written as:

$$\left(F_{i+1/2,j} - F_{i-1/2,j}\right) + \left(G_{i,j+1/2} - G_{i,j-1/2}\right) = 0 \tag{2.7}$$

The indices ½ denotes the cell interfaces and it is assumed that the variables are constant along the interface. In structured mesh, a typical control volume is given below with the flux and node indices.

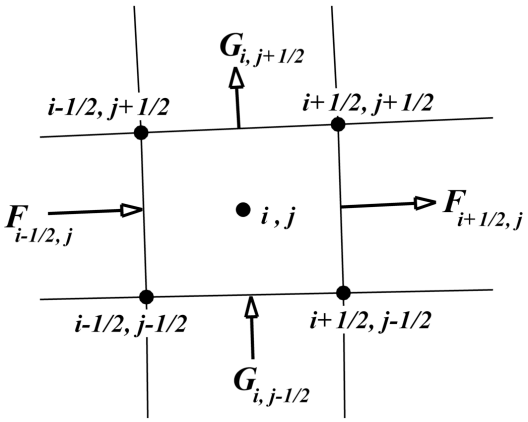

Figure 2.2 C*ontrol volume formed by the mesh and the flux vectors*

In finite volume method, the spatial discretization of the fluxes can be performed in two ways. First alternative is the central schemes. Central schemes are simple and based on averaging the conservative flow variables to the left and to the right in order to evaluate the flux. Although central schemes are simple to implement, they necessitate the usage of artificial dissipation since they cannot recognize and suppress an odd-even decoupling of the solution [60]. Second alternative for the spatial discretization is the usage of upwind schemes which are developed basing on convective physics of the Euler equations. Upwind schemes are capable of capturing the discontinuities such as shocks precisely. However, strong discontinuities cause oscillations in the solutions performed with high order spatial accuracy and to prevent it usage of limiter functions may be required. Limiter functions switch the high order spatial discretization to first order discretization near the discontinuities.

2.3.1 Upwinding Schemes

Upwinding schemes are basically decomposed into two groups:
Flux Vector Splitting schemes
Flux Difference Splitting Schemes

2.3.1.1 Flux Vector Splitting Schemes

The flux vector splitting is based on the fact that convective fluxes are homogenous function of degree one in conservative flow variables.

$$\boldsymbol{F_c} = \frac{\partial \boldsymbol{F_c}}{\partial \boldsymbol{w}} \boldsymbol{w} = \boldsymbol{J_c} \boldsymbol{w} \tag{2.8}$$

where, $\boldsymbol{J_c}$ is the Jacobian matrix evaluated by convective fluxes.

The special property given in Equation 2.8 forms the basis for all the flux vector splitting type methods. The splitting can be performed in terms of the wave speeds [52], in terms of Mach number [53] or alternatively in terms of both of Mach number and pressure after decomposing the flux vector as convective and pressure parts[54].

Considering the wave splitting procedure, convective flux Jacobian matrix can be diagonalized as follows:

$$\boldsymbol{J_c} = \boldsymbol{Q_\Lambda} \boldsymbol{\Lambda} \boldsymbol{Q_\Lambda^{-1}} \tag{2.9}$$

$\boldsymbol{\Lambda}$ is the diagonal matrix that includes the eigenvalues of the Jacobian matrix, $\boldsymbol{J_c}$, and $\boldsymbol{Q_\Lambda}$ is the matrix formed by the corresponding right eigenvectors. Splitting the eigenvalues into two in terms of their signs, Jacobian matrix becomes:

$$\boldsymbol{J_c} = \boldsymbol{Q_\Lambda} \boldsymbol{\Lambda_J} \boldsymbol{Q_\Lambda^{-1}} \quad = \boldsymbol{Q_\Lambda} \boldsymbol{\Lambda_J^+} \boldsymbol{Q_\Lambda^{-1}} + \boldsymbol{Q_\Lambda} \boldsymbol{\Lambda_J^-} \boldsymbol{Q_\Lambda^{-1}} \quad = \boldsymbol{J_c^+} + \boldsymbol{J_c^-} \tag{2.10}$$

Then the flux vector can be splitted as

$$\boldsymbol{F_c} = \boldsymbol{J_c} \boldsymbol{w} = \left\{ \left[\boldsymbol{J_c^+} \right] + \left[\boldsymbol{J_c^-} \right] \right\} \boldsymbol{w} = \boldsymbol{F_c^+} + \boldsymbol{F_c^-} \tag{2.11}$$

F_c^+ and F_c^- are the subvectors associated with the positive and the negative eigenvalues respectively. The F_c^+ carries information from upstream to downstream. Similarly, F_c^- carries information from downstream to upstream. The left-side flow variables, w^L that are located in the upstream of the face are used in the computation of the F_c^+. Consistently the right-side flow variables, w^R that are located in the downstream of the face are used in the computation of the F_c^-. Therefore the formulation of the splitted flux can be written as:

$$\begin{aligned} F_c^+ &= J_c^+ w^L \\ F_c^- &= J_c^- w^R \end{aligned} \tag{2.12}$$

Implementing the splitting procedure into the Equation 2.6 and 2.7 upwind discretized form of the steady, 2-D Euler equations can be written for a quadrilateral cell as follows:

$$\sum_{n=1}^{\#\,face} (\mathbf{F}_c \cdot dS) = \left[F^+(w^L{}_{i+1/2,j}) + F^-(w^R{}_{i+1/2,j})\right] - \left[F^+(w^L{}_{i-1/2,j}) + F^-(w^R{}_{i-1/2,j})\right]$$
$$\left[G^+(w^L{}_{i,j+1/2}) + G^-(w^R{}_{i,j+1/2})\right] - \left[G^+(w^L{}_{i,j-1/2}) + G^-(w^R{}_{i,j-1/2})\right] = 0 \tag{2.13}$$

The resulting flux splitting formulation given in Equation 2.13 is derived by the wave splitting strategy which was introduced by Steger and Warming. Nevertheless this formulation is also valid for any type of flux vector splitting scheme such as Van Leer, AUSM or CUSP schemes. This formulation provides simple analytical derivation strategy for the flux Jacobian which will be presented in next chapter. In this thesis study flux Jacobian corresponding to Steger-Warming, Van-Leer and AUSM flux vector splitting schemes are evaluated by both analytical and numerical methods. In solution of Newton's method, numerically and analytically calculated Steger-Warming, Van-Leer and AUSM flux Jacobian matrices are used.

2.3.1.1.a Steger-Warming Scheme

The convective fluxes vectors can be evaluated by Steger-Warming scheme as follows:

$$F_c^{\pm} = \frac{\rho}{2\gamma}\begin{bmatrix} 2(\gamma-1)\lambda_1^{\pm} + \lambda_2^{\pm} + \lambda_3^{\pm} \\ \left(2(\gamma-1)\lambda_1^{\pm} + \lambda_2^{\pm} + \lambda_3^{\pm}\right)u + c\left(\lambda_2^{\pm} - \lambda_3^{\pm}\right)\eta_x \\ \left(2(\gamma-1)\lambda_1^{\pm} + \lambda_2^{\pm} + \lambda_3^{\pm}\right)v + c\left(\lambda_2^{\pm} - \lambda_3^{\pm}\right)\eta_y \\ \left(2(\gamma-1)\lambda_1^{\pm} + \lambda_2^{\pm} + \lambda_3^{\pm}\right)\frac{u^2+v^2}{2} + cU\left(\lambda_2^{\pm} - \lambda_3^{\pm}\right) + c^2\frac{\lambda_2^{\pm} + \lambda_3^{\pm}}{\gamma-1} \end{bmatrix} \tag{2.14}$$

In Equation 2.14 η_x and η_y are the components of the face normal vector. The eigenvalues λ_1 , λ_2 , λ_3 and the speed of sound *c* is defined as:

$$\begin{aligned} c &= \sqrt{\gamma(\gamma-1)\left(e_t - \frac{1}{2}\left(u^2+v^2\right)\right)} \\ \lambda_1 &= U\ , \quad \lambda_2 = U + c\ , \quad \lambda_3 = U - c \end{aligned} \tag{2.15}$$

In the calculation of positive signed fluxes, F^+; velocities, speed of sound and energy will be evaluated from the left state flow variables. Similarly negative signed fluxes are calculated from right state variables. Steger Warming schemes defines the flux splitting with splitting the eigenvalues in terms of their signs with below formulation:

$$\lambda_i^{\pm} = \frac{\lambda_i \pm |\lambda_i|}{2} \tag{2.16}$$

The above formulation can cause some problems in the calculation Jacobian matrices at stagnation and at sonic points. At these points, the derivatives of flux vector with respect to flow variables may be discontinuous. If the proper precautions is not implemented the solution will have oscillations around those singularities and there will be large differences between the numerically and analytically evaluated Jacobian matrices.

2.3.1.1.b Van Leer Scheme

Van Leer proposed a scheme based on Mach number splitting. His scheme does not face the problems that occur in the usage of Steger-Warming scheme at sonic and stagnation points. In the scheme advection Mach number is defined at the cell face:

$$M_n = M_L^+ + M_R^- \tag{2.17}$$

the splitted mach numbers are defined as

$$M_L^+ = \begin{cases} M_L & if\ M_L \geq 1 \\ \frac{1}{4}(M_L+1)^2 & if\ |M_L| < 1 \\ 0 & if\ M_L \leq -1 \end{cases} \qquad M_R^- = \begin{cases} 0 & if\ M_R \geq 1 \\ \frac{1}{4}(M_R-1)^2 & if\ |M_L| < 1 \\ M_R & if\ M_R \leq -1 \end{cases} \tag{2.18}$$

The Mach numbers, M_L and M_R, are calculated by the contravariant velocities that are evaluated with left and right flow variables.

$$M_L = \frac{U_L}{c_L} \quad , \qquad M_R = \frac{U_R}{c_R} \tag{2.19}$$

In the case of subsonic flows where, $|M_n| < 1$ the splitted flux are defined as:

$$F_c^{\pm} = \begin{bmatrix} f_{mass}^{\pm} \\ f_{mass}^{\pm}\left(u + \eta_x \frac{-V \pm 2c}{\gamma}\right) \\ f_{mass}^{\pm}\left(v + \eta_y \frac{-V \pm 2c}{\gamma}\right) \\ f_{energy}^{\pm} \end{bmatrix} \tag{2.20}$$

The mass and energy components are defined as:

$$\begin{aligned} f_{mass}^{+} &= \rho_L c_L \frac{(M_L+1)^2}{4} \\ f_{mass}^{-} &= \rho_R c_R \frac{(M_R-1)^2}{4} \\ f_{energy}^{\pm} &= f_{mass}^{\pm} \left\{ \frac{((\gamma-1)U \pm 2c)^2}{2(\gamma^2-1)} + \frac{u^2+v^2-U^2}{2} \right\}_{L/R} \end{aligned} \tag{2.21}$$

In the case of supersonic flow, where $|M_n| > 1$, the fluxes are evaluated as:

$$\begin{aligned} F_c^+ = F_c \quad , \qquad F_c^- = 0 \qquad if\ M_n \geq 1 \\ F_c^+ = 0 \quad , \qquad F_c^- = F_c \qquad if\ M_n \leq -1 \end{aligned} \tag{2.22}$$

2.3.1.1.c AUSM scheme

The Advection Upstream Splitting Method, AUSM, was introduced by Liou and Steffen. The splitting is performed similar to Van Leer scheme with the additional splitting for the pressure. The AUSM scheme decomposes the flux into two as: the scalar quantities part that convected by contravariant velocity and the pressure part that is governed by acoustic wave speed.

Mach number splitting is same as the Van Leer scheme and the flux splitting formulation is given below:

$$F_c^+ = M_L^+ \begin{bmatrix} \rho c \\ \rho cu \\ \rho cv \\ \rho ch_T \end{bmatrix}_L + \begin{bmatrix} 0 \\ \eta_x p_L^+ \\ \eta_y p_L^+ \\ 0 \end{bmatrix} \qquad F_c^- = M_R^- \begin{bmatrix} \rho c \\ \rho cu \\ \rho cv \\ \rho ch_T \end{bmatrix}_R + \begin{bmatrix} 0 \\ \eta_x p_R^- \\ \eta_y p_R^- \\ 0 \end{bmatrix} \tag{2.23}$$

and pressure is splitted as follows:

$$p_L^+ = \begin{cases} p_L & if\ M_L \geq 1 \\ \dfrac{p_L}{4}\left(M_L + 1\right)^2 \left(2 - M_L\right) & if\ \left|M_L\right| < 1 \\ 0 & if\ M_L \leq -1 \end{cases}$$

$$p_R^- = \begin{cases} 0 & if\ M_R \geq 1 \\ \dfrac{p_R}{4}\left(M_R - 1\right)^2 \left(2 - M_R\right) & if\ \left|M_R\right| < 1 \\ p_R & if\ M_R \leq -1 \end{cases} \tag{2.24}$$

2.3.1.2 Flux Difference Splitting

2.3.1.2.a Roe Scheme

Roe solved an approximate Riemann problem by linearizing the Jacobian matrix by the constant matrix A_{RL}. The Roe's averaged Jacobian matrix also satisfies the homogeneity property. The flux is splitted into the two parts by left and right state flow variables as below:

$$
\begin{aligned}
F_c &= J_{RL}\boldsymbol{w} \\
F_c &= \left|J_{RL}\right|\left(\boldsymbol{w}^R - \boldsymbol{w}^L\right) = F^L - F^R
\end{aligned}
\tag{2.25}
$$

The Roe's Jacobian matrix can be diagonalized as:

$$
\begin{aligned}
\boldsymbol{J}_{RL} &= \tilde{\boldsymbol{Q}}_{\Lambda}\,\tilde{\boldsymbol{\Lambda}}_J\,\tilde{\boldsymbol{Q}}_{\Lambda}^{-1} \\
\left|\boldsymbol{J}_{RL}\right| &= \tilde{\boldsymbol{Q}}_{\Lambda}\left|\,\tilde{\boldsymbol{\Lambda}}_J\,\right|\tilde{\boldsymbol{Q}}_{\Lambda}^{-1}
\end{aligned}
\tag{2.26}
$$

$\tilde{\boldsymbol{\Lambda}}_J$ and $\tilde{\boldsymbol{Q}}_{\Lambda}$ are the diagonal matrix composed of eigenvalues of the Roe's Jacobian and the corresponding matrix of right eigenvectors respectively. Inserting the Equation 2.26 into the Equation 2.25:

$$
F_c = F\left(\boldsymbol{w}^R\right) - F\left(\boldsymbol{w}^L\right) = \sum \tilde{\boldsymbol{Q}}_{\Lambda}\left|\,\tilde{\lambda}_J\,\right|\tilde{\boldsymbol{Q}}_{\Lambda}^{-1}\left(\boldsymbol{w}^R - \boldsymbol{w}^L\right) \tag{2.27}
$$

Considering the sign of the eigenvalues, the flux vector can take the below form:

$$
\begin{aligned}
F_c &= F\left(\boldsymbol{w}^L\right) + \sum_{\lambda_J<0} \tilde{\boldsymbol{Q}}_{\Lambda}\left|\,\tilde{\lambda}_J\,\right|\tilde{\boldsymbol{Q}}_{\Lambda}^{-1}\left(\boldsymbol{w}^R - \boldsymbol{w}^L\right) \\
F_c &= F\left(\boldsymbol{w}^R\right) - \sum_{\lambda_J>0} \tilde{\boldsymbol{Q}}_{\Lambda}\left|\,\tilde{\lambda}_J\,\right|\tilde{\boldsymbol{Q}}_{\Lambda}^{-1}\left(\boldsymbol{w}^R - \boldsymbol{w}^L\right)
\end{aligned}
\tag{2.28}
$$

Averaging the two flux vectors given by the Equation 2.28 the final flux vector proposed by Roe can be written as:

$$
F_c = \frac{1}{2}\left[F\left(\boldsymbol{w}^L\right) + F\left(\boldsymbol{w}^R\right) - \sum \tilde{\boldsymbol{Q}}_{\Lambda}\left|\,\tilde{\lambda}_J\,\right|\tilde{\boldsymbol{Q}}_{\Lambda}^{-1}\left(\boldsymbol{w}^R - \boldsymbol{w}^L\right)\right] \tag{2.29}
$$

Roe's Jacobian matrix and the Roe's averaged variables are given below:

$$
J_{RL} = \begin{bmatrix}
0 & 1 & 0 \\
\dfrac{\gamma-3}{2}u_{RL}^2 & (3-\gamma)u_{RL} & \gamma-1 \\
-h_{T_{RL}}u_{RL} + \dfrac{(\gamma-1)u_{RL}^3}{2} & h_{T_{RL}} - (\gamma-1)u_{RL}^2 & \gamma u_{RL}
\end{bmatrix}
\tag{2.30}
$$

$$u_{RL} = \frac{\sqrt{\rho_R}u_R + \sqrt{\rho_L}u_L}{\sqrt{\rho_R} + \sqrt{\rho_L}}$$

$$h_{T_{RL}} = \frac{\sqrt{\rho_R}h_R + \sqrt{\rho_L}h_L}{\sqrt{\rho_R} + \sqrt{\rho_L}} \tag{2.31}$$

$$\rho_{RL} = \sqrt{\rho_R \rho_L}$$

In this study, flux Jacobian corresponding to Roe's flux difference splitting schemes is evaluated by only numerical methods. In solution of Newton's method, numerically calculated Roe's flux Jacobian matrices are used.

2.3.2 High Order Reconstruction

Flux vector splitting and flux difference splitting schemes which is explained in previous section are first order accurate in space due to the assumption of non-varying flow variables along the cell:

$$\begin{aligned} \boldsymbol{w}^L{}_{i+1/2} &= \boldsymbol{w}_i \quad , \quad \boldsymbol{w}^R{}_{i+1/2} = \boldsymbol{w}_{i+1} \\ \boldsymbol{w}^L{}_{i-1/2} &= \boldsymbol{w}_{i-1} \quad , \quad \boldsymbol{w}^R{}_{i-1/2} = \boldsymbol{w}_i \end{aligned} \tag{2.32}$$

Higher order accuracy can be achieved by the assumption of varying flow variables in each cell. The Monotonic Upstream Centered Scheme, MUSCL, which was introduced by Van Leer [56] is a common technique for the interpolation of the cell center flow variables to the cell faces as left and right states. The interpolation formula is given below:

$$\begin{aligned} \boldsymbol{w}^L{}_{i+1/2} &= \boldsymbol{w}_i + \frac{1}{4}\left\{\phi(r)\left[(1-\kappa)\nabla + (1+\kappa)\Delta\right]\right\}_i \\ \boldsymbol{w}^R{}_{i+1/2} &= \boldsymbol{w}_{i+1} - \frac{1}{4}\left\{\phi(r)\left[(1+\kappa)\nabla + (1-\kappa)\Delta\right]\right\}_{i+1} \end{aligned} \tag{2.33}$$

In above equation, Δ and ∇ are the forward and backward operators and the r is the ratio of these operators.

$$\Delta_i = \boldsymbol{w}_{i+1} - \boldsymbol{w} \quad , \qquad \nabla_i = \boldsymbol{w}_i - \boldsymbol{w}_{i-1}$$
$$r_i = \frac{\Delta_i}{\nabla_i} \tag{2.34}$$

In the Equation 2.33, the parameter κ defines the order of the accuracy of the interpolation. For κ = -1, purely one sided upwind interpolation; for κ = 0, linear interpolation between one upstream and one downstream cell, for κ=1/3 three cells interpolation; are obtained with second order accuracy. For κ = 1 the upwind influence is lost and face values are calculated by the arithmetic mean of the neighboring cells.

Mechanisms such as shock waves produce large gradients in flow. The large gradients can cause significantly diverse left and right interpolated state variables on the same face. This diversity is mainly due to the over and undershoots of the interpolation and generates oscillations in the solution. Those oscillations can be constrained by reducing the slopes that are used in the interpolation. The nonlinear functions that are called limiters are used for this purpose. At strong discontinuities where the gradients are large, the slopes are reduced to zero by the limiters to switch into the first order discretization. In the MUSCL formulation given in Equation 2.33 limiters are denoted by $\phi(r)$. The limiters are generally functions of the forward and backward difference operators and there are various types of limiters in the literature. In this study, the continuous limiter functions are used to enable analytical differentiation of the fluxes for the Jacobian derivation. The two continuous limiter functions used in this study are given below:

$$\phi(r) = \frac{2r}{r^2 + 1} \qquad for \quad \kappa = -1$$
$$\phi(r) = \frac{3r}{2r^2 - r + 2} \qquad for \quad \kappa = \frac{1}{3} \tag{2.35}$$

The usage of above slope limiters leads to Van Albada limiter for κ = 0, and to Hemker-Koren limiter for κ =1/3. Even though the defined limiters are differentiable, it was shown that their utilization results in convergence problems for steady solutions. The reason of this problem is explained as the reaction of the limiter functions to even very small oscillations in smooth regions of the flow domain which introduce high non-linearity. Van Albada[57] tuned his limiter for κ = 0 case to prevent the activation of the limiter in smooth regions by introducing an additional parameter, $\in$. With this

modification interpolation formula reduces to unlimited MUSCL formulation in smoothly varying regions by deactivation of the limiter. The limiter is activated only at high gradient regions to reduce the accuracy of discretization to first order. Similar modification is performed for κ =1/3 scheme by Venkatakrishnan [58]. The formulation of the tuned interpolations is given below:

$$\begin{aligned} \boldsymbol{w}^{L}{}_{i+1/2} &= \boldsymbol{w}_i + \boldsymbol{\delta}^{L}_{i+1/2} \\ \boldsymbol{w}^{R}{}_{i+1/2} &= \boldsymbol{w}_{i+1} - \boldsymbol{\delta}^{R}_{i+1/2} \end{aligned} \tag{2.36}$$

For κ = 0

$$\delta = \frac{(a^2+\in)b_i + (b^2+\in)a}{a^2+b^2+2\in} \tag{2.37}$$

For κ = 1/3

$$\delta = \frac{(2a^2+\in)b + (b^2+2\in)a}{a^2+b^2-ab+3\in} \tag{2.38}$$

where

$$\begin{aligned} a_L &= \Delta_i \quad , \quad b_L = \nabla_i \\ a_R &= \nabla_{i+1} \quad , \quad b_R = \Delta_{i+1} \end{aligned} \tag{2.39}$$

In above equations, $\in$ is a small number which is used to prevent the activation of the limiter in the smooth regions of the flow domain. $\in$ is defined as 0.008 in Van Albada's work and for the Koren's limiter, Venkatakrishnan defined it as the square root of the cell area in 2-D problems.

2.4 Newton's Method

After the spatial discretization is implemented, Euler equations can be written as:

$$\boldsymbol{R}(\boldsymbol{w}) = 0 \tag{2.40}$$

In above equation, $\boldsymbol{R}(\boldsymbol{w})$ is the residual vector of the spatial discretization. Residual is non-linear function of conservative flow variables, $\boldsymbol{w}$.

$$\boldsymbol{R}(\boldsymbol{w}) = \oint_{\partial\Omega} (\boldsymbol{F}_c \,.\boldsymbol{n})\, dS - \oint_{\partial\Omega} (\boldsymbol{F}_V \,.\boldsymbol{n})\, dS \tag{2.41}$$

The system of non-linear equations given in Equation 2.40 is needed to be satisfied. The linearization of the $\boldsymbol{R}(\boldsymbol{w})$ in time, for the implicit solution of Equation 2.40 yields:

$$\boldsymbol{R}^{n+1}(\boldsymbol{w}) = \boldsymbol{R}^{n}(\boldsymbol{w}) + \left(\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{w}}\right)^{n} \Delta \boldsymbol{w}^{n} \tag{2.42}$$

where $\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{w}}$ is the Jacobian matrix.

Assuming that at the $(n+1)^{\text{th}}$ iteration the flow variables exactly satisfies the discretized Euler equations, i.e. $\boldsymbol{R}^{n+1}(\boldsymbol{w}) = 0$, the Newton's Method can be derived as follows:

$$\left(\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{w}}\right)^{n} \Delta \boldsymbol{w}^{n} = -\boldsymbol{R}(\boldsymbol{w}^{n}) \tag{2.43}$$

In above equation, the increment in flow variable vector is calculated. The new values of flow variable vector $\boldsymbol{w}$ at the $(n+1)^{\text{th}}$ iteration can be calculated as:

$$\boldsymbol{w}^{n+1} = \boldsymbol{w}^{n} + \Delta \boldsymbol{w}^{n} \tag{2.44}$$

The above iterative procedure will be repeated until the residual drops below specified tolerance.

2.4.1 Flux Jacobian

In the solution of Euler equations with Newton's method, the evaluation of the flux Jacobian matrix is needed. The entries of Jacobian matrix are the derivatives of the residual vector with respect to the flow variables vector. In the calculation of these derivatives, analytical or numerical derivation methods a can be used, and the resulting matrices are called analytical or numerical Jacobian matrices, respectively.

2.4.1.1 Analytical Jacobian Derivation

Substituting Equation 2.13 into Equation 2.41, the discretized flux residual can be calculated as:

$$R_{i,j}(\boldsymbol{w}) = \left[F^{+}(\boldsymbol{w}^{L}{}_{i+1/2,j}) + F^{-}(\boldsymbol{w}^{R}{}_{i+1/2,j}) \right] - \left[F^{+}(\boldsymbol{w}^{L}{}_{i-1/2,j}) + F^{-}(\boldsymbol{w}^{R}{}_{i-1/2,j}) \right] \\ \left[G^{+}(\boldsymbol{w}^{L}{}_{i,j+1/2}) + G^{-}(\boldsymbol{w}^{R}{}_{i,j+1/2}) \right] - \left[G^{+}(\boldsymbol{w}^{L}{}_{i,j-1/2}) + G^{-}(\boldsymbol{w}^{R}{}_{i,j-1/2}) \right] = 0 \tag{2.45}$$

In above formulation, $R_{i,j}$, presents the residual of the cell at i and j coordinates. The Jacobian matrix contains the derivatives of the residual calculated at each cell with respect to the whole flow variables in the discretized domain. Differentiating the residual $R_{i,j}$ with respect to a flow variable $w_{k,l}$, the residual Jacobian is defined as:

$$\frac{\partial R_{i,j}}{\partial w_{k,l}} = A^{+}{}_{i+1/2,j} \frac{\partial w^{L}_{i+1/2,j}}{\partial w_{k,l}} + A^{-}{}_{i+1/2,j} \frac{\partial w^{R}_{i+1/2,j}}{\partial w_{k,l}} - A^{+}{}_{i-1/2,j} \frac{\partial w^{L}_{i-1/2,j}}{\partial w_{k,l}} - A^{-}{}_{i-1/2,j} \frac{\partial w^{R}_{i-1/2,j}}{\partial w_{k,l}} \\ + B^{+}{}_{i,j+1/2} \frac{\partial w^{L}_{i,j+1/2}}{\partial w_{k,l}} + B^{-}{}_{i,j+1/2} \frac{\partial w^{R}_{i,j+1/2}}{\partial w_{k,l}} - B^{+}{}_{i,j-1/2} \frac{\partial w^{L}_{i,j-1/2}}{\partial w_{k,l}} - B^{-}{}_{i,j-1/2} \frac{\partial w^{R}_{i,j-1/2}}{\partial w_{k,l}} \tag{2.46}$$

where,
$$\hat{A}^{+} = \frac{\partial F^{+}}{\partial \boldsymbol{w}^{L}} \quad , \quad \hat{A}^{-} = \frac{\partial F^{-}}{\partial \boldsymbol{w}^{R}}, \quad \hat{B}^{+} = \frac{\partial G^{+}}{\partial \boldsymbol{w}^{L}} \quad , \quad \hat{B}^{-} = \frac{\partial G^{-}}{\partial \boldsymbol{w}^{R}} \tag{2.47}$$

Analytical derivation of Residual Jacobian needs two sets of derivatives. The first set composes of the derivatives of splitted fluxes with respect to the interpolated flow variables at cell faces, which are $A^{\pm}{}_{i+1/2,j}$, $A^{\pm}{}_{i-1/2,j}$, $B^{\pm}{}_{i,j+1/2}$, $B^{\pm}{}_{i,j-1/2}$. Due to their analytical relations, these derivatives are function of flux schemes but they are independent of order of spatial discretization. In this study; Steger-Warming, Van Leer and AUSM flux schemes are differentiated to evaluate those derivatives.

The second set consists of the derivatives of interpolated cell face flow variables with respect to the main flow variables stored at cell centers. The analytical relation of these derivatives varies according to order of spatial discretization but they are independent of flux schemes. As shown in Equation 2.32, for the first order discretization, right ($\boldsymbol{w}^{R}$) and left ($\boldsymbol{w}^{L}$) flow variables are equal to the values at the cell center of the two cells that are just located at the right and left of the corresponding cell face. Therefore, in first order discretization k and l values in Equation 2.46 changes from $i-1$ to $i+1$ and $j-1$ to $j+1$, respectively.

$$
\begin{aligned}
\boldsymbol{w}^{L}_{i+\frac{1}{2},j} &= \boldsymbol{w}_{i,j} \quad , \quad \boldsymbol{w}^{R}_{i+\frac{1}{2},j} = \boldsymbol{w}_{i+1,j} \\
\boldsymbol{w}^{L}_{i-\frac{1}{2},j} &= \boldsymbol{w}_{i-1,j} \quad , \quad \boldsymbol{w}^{R}_{i-\frac{1}{2},j} = \boldsymbol{w}_{i,j} \\
\boldsymbol{w}^{L}_{i,j+\frac{1}{2}} &= \boldsymbol{w}_{i,j} \quad , \quad \boldsymbol{w}^{R}_{i,j+\frac{1}{2}} = \boldsymbol{w}_{i,j+1} \\
\boldsymbol{w}^{L}_{i,j-\frac{1}{2}} &= \boldsymbol{w}_{i,j-1} \quad , \quad \boldsymbol{w}^{R}_{i,j-\frac{1}{2}} = \boldsymbol{w}_{i,j}
\end{aligned}
\tag{2.48}
$$

Jacobian matrices in first order discretization can be evaluated as:

$$
\begin{aligned}
&\frac{\partial R_{i,j}}{\partial \boldsymbol{w}_{i,j}} = A^{+}_{i+\frac{1}{2},j} - A^{-}_{i-\frac{1}{2},j} + B^{+}_{i,j+\frac{1}{2}} - B^{-}_{i,j-\frac{1}{2}} \\
&\frac{\partial R_{i,j}}{\partial \boldsymbol{w}_{i+1,j}} = A^{-}_{i+\frac{1}{2},j} \quad , \quad \frac{\partial R_{i,j}}{\partial \boldsymbol{w}_{i,j+1}} = B^{-}_{i,j+\frac{1}{2}} \\
&\frac{\partial R_{i,j}}{\partial \boldsymbol{w}_{i-1,j}} = -A^{+}_{i-\frac{1}{2},j} \quad , \quad \frac{\partial R_{i,j}}{\partial \boldsymbol{w}_{i,j-1}} = -B^{+}_{i,j-\frac{1}{2}}
\end{aligned}
\tag{2.49}
$$

where

$$
\begin{aligned}
A^{+}_{i\mp 1/2,j} &= \frac{\partial F^{+}_{i\mp 1/2,j}}{\partial \boldsymbol{w}^{L}_{i\mp 1/2,j}} \quad , \quad A^{-}_{i\mp 1/2,j} = \frac{\partial F^{-}_{i\mp 1/2,j}}{\partial \boldsymbol{w}^{R}_{i\mp 1/2,j}} \\
B^{+}_{i,j\mp 1/2} &= \frac{\partial G^{+}_{i,j\mp 1/2}}{\partial \boldsymbol{w}^{L}_{i,j\mp 1/2}} \quad , \quad B^{-}_{i,j\mp 1/2} = \frac{\partial G^{-}_{i,j\mp 1/2}}{\partial \boldsymbol{w}^{L}_{i,j\mp 1/2}}
\end{aligned}
\tag{2.50}
$$

For the second order spatial discretization, the flow variables at the cell faces are calculated from the interpolation of flow variables at the center of the 4 neighboring cells using MUSCL scheme. Therefore, in second order discretization *k* and *l* values in Equation 2.46 changes from *i-2* to *i+2* and *j-2* to *j+2*, respectively. Thanks to the continuous limiter functions used, the MUSCL scheme is differentiable of flow variables. Hence analytical Jacobian is evaluated without any difficulty for high order schemes.

$$
\begin{aligned}
\boldsymbol{w}^{L/R}_{i-\frac{1}{2},j} &= \boldsymbol{w}^{L/R}_{i-\frac{1}{2},j}\left(\boldsymbol{w}_{i-1,j}, \boldsymbol{w}_{i,j}, \delta^{L/R}_{i-\frac{1}{2},j}\right) \\
\boldsymbol{w}^{L/R}_{i+\frac{1}{2},j} &= \boldsymbol{w}^{L/R}_{i-\frac{1}{2},j}\left(\boldsymbol{w}_{i,j}, \boldsymbol{w}_{i+1,j}, \delta^{L/R}_{i+\frac{1}{2},j}\right) \\
\delta^{L/R}_{i-\frac{1}{2},j} &= \delta^{L/R}_{i-\frac{1}{2},j}\left(\boldsymbol{w}_{i-2,j}, \boldsymbol{w}_{i-1,j}, \boldsymbol{w}_{i,j}, \boldsymbol{w}_{i+1,j}\right) \\
\delta^{L/R}_{i+\frac{1}{2},j} &= \delta^{L/R}_{i+\frac{1}{2},j}\left(\boldsymbol{w}_{i-1,j}, \boldsymbol{w}_{i,j}, \boldsymbol{w}_{i+1,j}, \boldsymbol{w}_{i+2,j}\right)
\end{aligned}
\tag{2.51}
$$

Jacobian matrices in second order discretization can be evaluated as:

$$\begin{aligned}\frac{\partial R_{i,j}}{\partial \boldsymbol{w}_{i,j}} = {} & A^{+}{}_{i,j}\frac{\partial \boldsymbol{w}^{L}_{i+1/2,j}}{\partial \boldsymbol{w}_{i,j}} + A^{-}{}_{i,j}\frac{\partial \boldsymbol{w}^{R}_{i+1/2,j}}{\partial \boldsymbol{w}_{i,j}} - A^{+}{}_{i-1,j}\frac{\partial \boldsymbol{w}^{L}_{i-1/2,j}}{\partial \boldsymbol{w}_{i,j}} - A^{-}{}_{i-1,j}\frac{\partial \boldsymbol{w}^{R}_{i-1/2,j}}{\partial \boldsymbol{w}_{i,j}} \\ & + B^{+}{}_{i,j}\frac{\partial \boldsymbol{w}^{L}_{i,j+1/2}}{\partial \boldsymbol{w}_{i,j}} + B^{-}{}_{i,j}\frac{\partial \boldsymbol{w}^{R}_{i,j+1/2}}{\partial \boldsymbol{w}_{i,j}} - B^{+}{}_{i,j-1}\frac{\partial \boldsymbol{w}^{L}_{i,j-1/2}}{\partial \boldsymbol{w}_{i,j}} - B^{-}{}_{i,j-1}\frac{\partial \boldsymbol{w}^{R}_{i,j-1/2}}{\partial \boldsymbol{w}_{i,j}}\end{aligned} \tag{2.52}$$

$$\frac{\partial R_{i,j}}{\partial \boldsymbol{w}_{i\mp1,j}} = A^{+}{}_{i,j}\frac{\partial \boldsymbol{w}^{L}_{i+1/2,j}}{\partial \boldsymbol{w}_{i\mp1,j}} + A^{-}{}_{i,j}\frac{\partial \boldsymbol{w}^{R}_{i+1/2,j}}{\partial \boldsymbol{w}_{i\mp1,j}} - A^{+}{}_{i-1,j}\frac{\partial \boldsymbol{w}^{L}_{i-1/2,j}}{\partial \boldsymbol{w}_{i\mp1,j}} - A^{-}{}_{i-1,j}\frac{\partial \boldsymbol{w}^{R}_{i-1/2,j}}{\partial \boldsymbol{w}_{i\mp1,j}} \tag{2.53}$$

$$\frac{\partial R_{i,j}}{\partial \boldsymbol{w}_{i,j\mp1}} = B^{+}{}_{i,j}\frac{\partial \boldsymbol{w}^{L}_{i,j+1/2}}{\partial \boldsymbol{w}_{i,j\mp1}} + B^{-}{}_{i,j}\frac{\partial \boldsymbol{w}^{R}_{i,j+1/2}}{\partial \boldsymbol{w}_{i,j\mp1}} - B^{+}{}_{i,j-1}\frac{\partial \boldsymbol{w}^{L}_{i,j-1/2}}{\partial \boldsymbol{w}_{i,j\mp1}} - B^{-}{}_{i,j-1}\frac{\partial \boldsymbol{w}^{R}_{i,j-1/2}}{\partial \boldsymbol{w}_{i,j\mp1}} \tag{2.54}$$

$$\begin{aligned}\frac{\partial R_{i,j}}{\partial \boldsymbol{w}_{i+2,j}} &= A^{+}{}_{i+1/2,j}\frac{\partial \boldsymbol{w}^{L}_{i+1/2,j}}{\partial \boldsymbol{w}_{i+2,j}} + A^{-}{}_{i+1/2,j}\frac{\partial \boldsymbol{w}^{R}_{i+1/2,j}}{\partial \boldsymbol{w}_{i+2,j}} \\ \frac{\partial R_{i,j}}{\partial \boldsymbol{w}_{i-2,j}} &= -A^{+}{}_{i-1/2,j}\frac{\partial \boldsymbol{w}^{L}_{i-1/2,j}}{\partial \boldsymbol{w}_{i-2,j}} - A^{-}{}_{i-1/2,j}\frac{\partial \boldsymbol{w}^{R}_{i-1/2,j}}{\partial \boldsymbol{w}_{i-2,j}} \\ \frac{\partial R_{i,j}}{\partial \boldsymbol{w}_{i,j+2}} &= A^{+}{}_{i,j+1/2}\frac{\partial \boldsymbol{w}^{L}_{i,j+1/2}}{\partial \boldsymbol{w}_{i,j+2}} + A^{-}{}_{i,j+1/2}\frac{\partial \boldsymbol{w}^{R}_{i,j+1/2}}{\partial \boldsymbol{w}_{i,j+2}} \\ \frac{\partial R_{i,j}}{\partial \boldsymbol{w}_{i,j-2}} &= -A^{+}{}_{i,j-1/2}\frac{\partial \boldsymbol{w}^{L}_{i,j-1/2}}{\partial \boldsymbol{w}_{i,j-2}} - A^{-}{}_{i,j-1/2}\frac{\partial \boldsymbol{w}^{R}_{i,j-1/2}}{\partial \boldsymbol{w}_{i,j-2}}\end{aligned} \tag{2.55}$$

The main advantage of the analytical method is that the residual Jacobian can be calculated accurately. The order of error in the analytical method can be as small as the round-off error. Although the analytical method requires code development, run time of an analytical code is short. However, as the complexity of the discretized residual equations increases, the derivation of the analytical Jacobian becomes more complicated and time consuming.

2.4.1.2 Numerical Jacobian Derivation

Another alternative for Jacobian evaluation is to compute the Jacobian numerically. Using a small finite-difference perturbation magnitude,ε the numerical Jacobian can be calculated by the forward-difference as follow [61]

$$\frac{\partial R_m}{\partial \boldsymbol{w}_n} = \frac{\Delta R_m(\boldsymbol{w})}{\Delta \boldsymbol{w}_n} = \frac{R_m(\boldsymbol{w} + e_n\varepsilon) - R_m(\boldsymbol{w})}{\varepsilon}$$

$$where \quad m = 1, mmax \quad and \quad n = 1, (mmax + nbound) \tag{2.56}$$

where, R_m is the m^{th} component of the residual vector and the $\boldsymbol{w}_n$ is the n^{th} component of the flow variable vector, e_n , is the n^{th} unit vector. The value of the n^{th} component of the unit vector e_n is one, and all other components are zero. The size of the residual vector is defined by *mmax,* which is equal to 4 times the number of interior cells in 2D flow problems. The size of the flow variable vector is larger as much as the summation of the number of boundary cells, *nbound* and interior cells, *mmax*. In the numerical method, Jacobian evaluation does not require the large coding effort as needed in the analytical method. The same residual discretization can be used for both the original and perturbed flow variables. This reuse of the same code is one of the important advantages of the numerical approach. Moreover, for cases in which the analytical derivation is difficult, numerical Jacobian can be obtained without any difficulty.

The inaccuracy and long computation time are the two main disadvantages of the numerical Jacobian evaluation. The error in numerical Jacobian is function of the finite-difference perturbation magnitude. The accuracy of the numerical Jacobian can be improved with the usage of an optimum perturbation magnitude that minimizes the total error in the finite difference evaluation. The main reason that causes long computation time is the necessity of the residual vector calculation with each perturbed flow variable in the whole domain. For a given cell, the residual is only a function of flow variables in that cell and the neighboring cells according to the discretization used. In order to reduce the computation time, the perturbed residual is computed only with flow variables in these cells. For first-order discretization, in addition to the cell in which the flow variable is perturbed, four neighboring cells are used. Considering four flow variables in each cell, 20 perturbed residual vector evaluations are required for the given cell. In second-order discretization, using eight neighboring cells in addition to the given cell, 36 perturbed residual vector evaluations are required. Although the speed and the accuracy of the analytical method may not be reached, the numerical Jacobian evaluation method may become faster and more accurate with some precautions.

2.5 Boundary Conditions

The formulations given in previous sections are valid for the interior flow domain. Special treatments are needed for evaluation of the fluxes at the domain boundaries. The most common way to impose boundary conditions is adding ghost cells to the exterior of the physical domain. In order to enforce the desired boundary conditions appropriate flow conditions are specified in the ghost cells. In this thesis study one layer of ghost cells are used and the order of spatial discretization is held at first order at the boundaries. In this section the theory used for the boundary condition definitions will be presented. The technique that is used for the boundary condition implementation is detailed in the Chapter-III.

2.5.1 Far Field Boundary Conditions

Far field boundary conditions are used for enforcing the flow conditions such as flight Mach number or angle of attack to the flow solver. Imposing the free-stream flow conditions directly to the ghost cells will not be a proper implementation, since the physical domain never extends to the infinity where the free stream flow conditions are valid. Therefore, an approach based on the characteristics of the Euler flow equations is used to define flow conditions at the far field. The approach is called characteristic based boundary conditions.

In 2-D flows, two Riemann invariants, $\mathbb{R}^+$ and $\mathbb{R}^-$ travel with two characteristic waves whose velocities are λ_1 and λ_2. The definitions of the Riemann invariants and the characteristic velocities are given below:

$$\begin{aligned} \mathbb{R}^+ &= U + \frac{2c}{\gamma - 1} \\ \mathbb{R}^- &= U - \frac{2c}{\gamma - 1} \\ \lambda_{1,2} &= U \pm c \end{aligned} \tag{2.57}$$

In above formulation U is the contravariant velocity normal to the boundary face and c is the speed of sound. The remaining two characteristic waves convect the tangential velocity, V_t and the entropy, $s = \rho^{\gamma}/P$, with the speed U.

The normal velocity and the speed of sound that are going to be imposed at the boundaries can be evaluated by the addition and subtraction of the two Riemann invariants, respectively.

$$\begin{aligned} U_{boundary} &= \frac{\mathbb{R}^{+} + \mathbb{R}^{-}}{2} \\ c_{boundary} &= \frac{\gamma - 1}{4}(\mathbb{R}^{+} - \mathbb{R}^{-}) \end{aligned} \tag{2.58}$$

The propagation directions of the characteristic waves are defined by their velocities. For 2-D supersonic flows where the flow velocity is greater the speed of sound wave speed all four characteristic waves propagates in the same direction with velocities $U+c$, $U-c$, U and U. If the flow is subsonic the negative Riemann invariant, $\mathbb{R}^{-}$, will propagate in the opposite direction with respect to other three waves since the sign of its speed is negative.

For an inflow with supersonic velocity, all the characteristics will propagate into the flow domain from the outside; hence free-stream conditions can be directly imposed to the ghost cells. Similarly for an outflow with supersonic velocity all the characteristics will propagate from inner flow domain to outside; therefore the flow conditions at the cells adjacent to outflow boundaries will be imposed to the ghost cells.

For the subsonic inflow the three of the characteristics carrying the entropy, tangential velocity and the first Riemann invariant, $\mathbb{R}^{+}$,will propagate from free-stream to flow domain. The second Riemann invariant, $\mathbb{R}^{-}$,will propagate from the inside of the flow domain to free-stream. Conversely for subsonic outflow $\mathbb{R}^{-}$ will propagate from free-stream to flow domains while other three waves propagate in to free-stream direction. The resulting boundary conditions for subsonic inflow and outflow is given below.

Subsonic inflow BC :

$$
\begin{aligned}
&\mathbb{R}^{+} = U_{\infty} + \frac{2c_{\infty}}{\gamma - 1} \quad ; \quad \mathbb{R}^{-} = U_{inner} - \frac{2c_{inner}}{\gamma - 1} \\
&U_{boundary} = \frac{U_{\infty} + U_{inner}}{2} + \frac{c_{\infty} - c_{inner}}{\gamma - 1} \\
&u_{boundary} = \eta_x U_{boundary} \quad ; \quad v_{boundary} = \eta_y U_{boundary} \\
&c_{boundary} = \frac{U_{\infty} - U_{inner}}{2} + \frac{c_{\infty} + c_{inner}}{\gamma - 1} \\
&s_{boundary} = s_{\infty} = \frac{\rho_{\infty}^{\gamma}}{P_{\infty}} = \left(\frac{\rho^{\gamma - 1}}{RT} \right)_{\infty} = \gamma \left(\frac{\rho^{\gamma - 1}}{c^2} \right)_{\infty} \\
&U_{t_{boundary}} = U_{t_{\infty}}
\end{aligned}
\tag{2.59}
$$

Subsonic outflow BC:

$$
\begin{aligned}
&\mathbb{R}^{+} = U_{inner} + \frac{2c_{inner}}{\gamma - 1} \\
&\mathbb{R}^{-} = U_{\infty} - \frac{2c_{\infty}}{\gamma - 1} \\
&U_{boundary} = \frac{U_{\infty} + U_{inner}}{2} + \frac{c_{inner} - c_{\infty}}{\gamma - 1} \\
&u_{boundary} = \eta_x U_{boundary} \quad ; \quad v_{boundary} = \eta_y U_{boundary} \\
&c_{boundary} = \frac{U_{inner} - U_{\infty}}{2} + \frac{c_{\infty} + c_{inner}}{\gamma - 1} \\
&s_{boundary} = s_{inner} \\
&U_{t_{boundary}} = U_{t_{inner}}
\end{aligned}
\tag{2.60}
$$

Using the definitions given in Equations 2.59 and 2.60 the flow conditions at the farfield ghost cells can be imposed as:

$$\begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{bmatrix}_{GHOST} = \begin{bmatrix} \left(\frac{s_{boundary} c_{boundary}^2}{\gamma} \right)^{\frac{1}{\gamma-1}} \\ \rho_{GHOST} u_{boundary} \\ \rho_{GHOST} v_{boundary} \\ \frac{\rho_{GHOST} c_{boundary}^2}{\gamma(\gamma-1)} + \frac{(\rho u)_{GHOST}^2 + (\rho v)_{GHOST}^2}{2\rho_{GHOST}} \end{bmatrix} \tag{2.61}$$

2.5.2 Wall Boundary Conditions

The solid-wall boundary conditions are applied on the geometry surfaces to impose no flow through boundary. That condition is satisfied by the application of the symmetry condition on the velocity. The components of the velocity in the wall normal direction are set to have equal magnitude of velocities but opposite sign in the ghost and interior cells adjacent to wall. The density, tangential component of the velocity and enthalpy is directly extrapolated to the ghost cells from the inner domain cells.

2.5.3 Computational Boundary Conditions

Computational boundary conditions are required to accommodate the use of computational grids. One of the most common computational boundary condition is the symmetry condition. This is used when the flow solution will be symmetric, so that only half of the flow has to be computed. In symmetry boundary conditions density, tangential velocity and enthalpy s directly extrapolated from interior cells to the ghost ones. The normal velocity component is extrapolated by the same magnitude but in reverse direction.

Wake boundary conditions are required to accommodate a grid of C mesh topology. The ghost cells of the lower portion of the wake correspond to interior cells of the upper portion. Similarly the ghost cells of the upper portion of the wake correspond to interior cells of the lower portion of the wake. All the flow variables are extrapolated directly from interior to ghost cell in wake boundary conditions.

CHAPTER 3

FLOW SOLVER

3.1 Introduction

In this chapter, the details of the methodology followed in the development of the Newton method flow solver are presented. The structure of the flux Jacobian matrix, the imposed initial conditions, the assignment of boundary conditions and the technique used for the factorization of the large sparse matrix is explained. In order to examine the performance of Newton's method, internal and external flow problems are solved at different free-stream conditions. The efficiency of the method is demonstrated by the convergence histories and the CPU time spent.

3.2 Structure of Jacobian Matrix

The Jacobian matrix is made up by the partial derivatives of residuals of each grid cells with respect to the each flow variables in the flow domain. The flux Jacobian is a large square matrix with dimensions equal to the total number of flow variables in the system but it is highly sparse. Most of the entries of Jacobian are zero, because the residual of a cell is only dependent on the flow variables stored in its neighboring cells according to order of discretization.

For the solution of 2-D Euler equations, the residual evaluation of a cell requires five-point stencil when first-order upwind discretization is used. Hence in Jacobian evaluation 5 point stencil produces a block diagonal matrix made up of five 4x4-blocks. In second-order discretizations, a nine-point stencil is employed that produces a block diagonal matrix made up of nine 4x4-block bands. Thus, all elements of the Jacobian matrix, except for these block bands and the boundary entries, are zero. The number of nonzero

elements in the Jacobian can be calculated by multiplying number of control volumes created by the mesh with the stencil size and the number of elements in the 4x4 blocks.

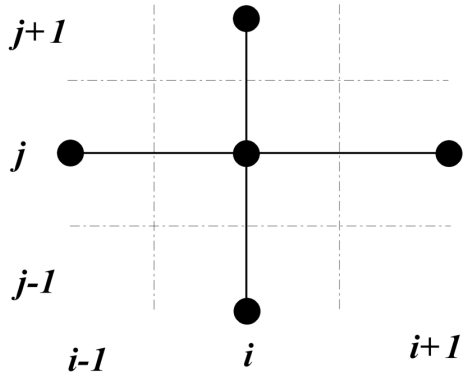

Figure 3.1 *Representation of the first order stencil used in spatial discretization*

Figure 3.1 presents the typical stencil used in the first order spatial discretization.

In the 2-D Euler equations the number of the flow equations and the number of the unknown flow conservative flow variables equal to four. Hence the derivatives of the four residual vector with respect four flow variables are needed, and those derivatives forms the 4x4 blocks. The formulations of the 4x4 blocks on the first and second order stencils are given in Equation 3.1 ad 3.2, respectively.

$$\begin{array}{ccc} & \bar{C} = \dfrac{\partial \boldsymbol{R}_{i,j}}{\partial \boldsymbol{w}_{i,j+1}} & \\ \bar{D} = \dfrac{\partial \boldsymbol{R}_{i,j}}{\partial \boldsymbol{w}_{i-1,j}} & \bar{A} = \dfrac{\partial \boldsymbol{R}_{i,j}}{\partial \boldsymbol{w}_{i,j}} & \bar{E} = \dfrac{\partial \boldsymbol{R}_{i,j}}{\partial \boldsymbol{w}_{i+1,j}} \\ & \bar{B} = \dfrac{\partial \boldsymbol{R}_{i,j}}{\partial \boldsymbol{w}_{i,j-1}} & \end{array} \tag{3.1}$$

$$\bar{G}_{i,j} = \frac{\partial \boldsymbol{R}_{i,j}}{\partial \boldsymbol{w}_{i,j+2}}$$

$$\bar{C}_{i,j} = \frac{\partial \boldsymbol{R}_{i,j}}{\partial \boldsymbol{w}_{i,j+1}}$$

$$\bar{H}_{i,j} = \frac{\partial \boldsymbol{R}_{i,j}}{\partial \boldsymbol{w}_{i-2,j}} \quad \bar{D}_{i,j} = \frac{\partial \boldsymbol{R}_{i,j}}{\partial \boldsymbol{w}_{i-1,j}} \quad \bar{A}_{i,j} = \frac{\partial \boldsymbol{R}_{i,j}}{\partial \boldsymbol{w}_{i,j}} \quad \bar{E}_{i,j} = \frac{\partial \boldsymbol{R}_{i,j}}{\partial \boldsymbol{w}_{i+1,j}} \quad \bar{I}_{i,j} = \frac{\partial \boldsymbol{R}_{i,j}}{\partial \boldsymbol{w}_{i+2,j}} \tag{3.2}$$

$$\bar{B}_{i,j} = \frac{\partial \boldsymbol{R}_{i,j}}{\partial \boldsymbol{w}_{i,j-1}}$$

$$\bar{F}_{i,j} = \frac{\partial \boldsymbol{R}_{i,j}}{\partial \boldsymbol{w}_{i,j-2}}$$

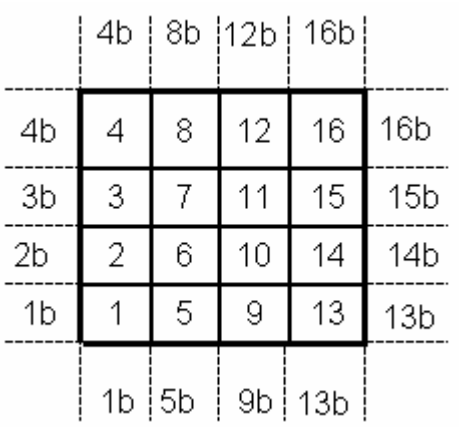

Figure 3.2 *A Typical structured grid which shows the fill in order of the Jacobian matrix*

If the elements of the Jacobian matrix is entered into the Jacobian by the order given in Figure 3.2, the resulting distribution of the 4x4 blocks in the Jacobian matrix can be represented as in Equation 3.3 for the first order spatial discretization and as in Figure 3.3 for the second order discretization.

Figure 3.3 *The distribution of the blocks in the Jacobian matrix row for the 2nd order discretization*

$$
\begin{bmatrix}
\bar{A}_1 & \bar{C}_1 & 0 & 0 & \bar{E}_1 & & & & & & & & & & & \\
\bar{B}_1 & \bar{A}_2 & \bar{C}_2 & 0 & & \bar{E}_2 & & & & & & & & & & \\
0 & \bar{B}_3 & \bar{A}_3 & \bar{C}_3 & & & \bar{E}_3 & & & & & & & & & \\
0 & 0 & \bar{B}_4 & \bar{A}_4 & \bar{C}_4 & & & \bar{E}_4 & & & & & & & & \\
\bar{D}_5 & & & \bar{B}_5 & \bar{A}_5 & \bar{C}_5 & 0 & 0 & \bar{E}_5 & & & & & & & \\
 & \bar{D}_6 & & & \bar{B}_6 & \bar{A}_6 & \bar{C}_6 & 0 & & \bar{E}_6 & & & & & & \\
 & & \bar{D}_7 & & 0 & \bar{B}_7 & \bar{A}_7 & \bar{C}_7 & & & \bar{E}_7 & & & & & \\
 & & & \bar{D}_8 & 0 & 0 & \bar{B}_8 & \bar{A}_8 & \bar{C}_8 & & & \bar{E}_8 & & & & \\
 & & & & \bar{D}_9 & & & \bar{B}_9 & \bar{A}_9 & \bar{C}_9 & 0 & 0 & \bar{E}_9 & & & \\
 & & & & & \bar{D}_{10} & & & \bar{B}_{10} & \bar{A}_{10} & \bar{C}_{10} & 0 & & \bar{E}_{10} & & \\
 & & & & & & \bar{D}_{11} & & 0 & \bar{B}_{11} & \bar{A}_{11} & \bar{C}_{11} & & & \bar{E}_{11} & \\
 & & & & & & & \bar{D}_{12} & 0 & 0 & \bar{B}_{12} & \bar{A}_{12} & \bar{C}_{12} & & & \bar{E}_{12} \\
 & & & & & & & & \bar{D}_{13} & & & \bar{B}_{13} & \bar{A}_{13} & \bar{C}_{13} & 0 & 0 \\
 & & & & & & & & & \bar{D}_{14} & & & \bar{B}_{14} & \bar{A}_{14} & \bar{C}_{14} & 0 \\
 & & & & & & & & & & \bar{D}_{15} & & 0 & \bar{B}_{15} & \bar{A}_{15} & \bar{C}_{15} \\
 & & & & & & & & & & & \bar{D}_{16} & 0 & 0 & \bar{B}_{16} & \bar{A}_{16}
\end{bmatrix}
\begin{bmatrix}
\Delta w_1 \\ \Delta w_2 \\ \Delta w_3 \\ \Delta w_4 \\ \Delta w_5 \\ \Delta w_6 \\ \Delta w_7 \\ \Delta w_8 \\ \Delta w_9 \\ \Delta w_{10} \\ \Delta w_{11} \\ \Delta w_{12} \\ \Delta w_{13} \\ \Delta w_{14} \\ \Delta w_{15} \\ \Delta w_{16}
\end{bmatrix}
=
\begin{bmatrix}
-R_1 \\ -R_2 \\ -R_3 \\ -R_4 \\ -R_5 \\ -R_6 \\ -R_7 \\ -R_8 \\ -R_9 \\ -R_{10} \\ -R_{11} \\ -R_{12} \\ -R_{13} \\ -R_{14} \\ -R_{15} \\ -R_{16}
\end{bmatrix}
\tag{3.3}
$$

Equation 3.3 shows the structure of the non-linear system that is solved by 1st order Jacobian matrix.

3.3 Implementation of Boundary Conditions

In section 2.5 the relations between the ghost cells and interior cells adjacent to boundaries were presented. Those relations can be used to impose the boundary conditions explicitly by using the flow variables of the previous iterations. In this study, boundary conditions are imposed implicitly. The change of the flow variables at the ghost cells and the interior cells are solved simultaneously. To perform the implicit implementation of the boundary conditions, the linearization of the relations that defines the boundary conditions is required. The linearization of the boundary conditions will result in a matrix relation as given below:

$$
[A]\Delta w_{GHOST} = [B]\Delta w_{INTERIOR} \tag{3.4}
$$

The evaluation of the above relation is simple for the supersonic inlet and outlet, wall and symmetry type boundary conditions. For example, in supersonic inlet boundary conditions the relation can be defined simply as:

$$[1]\Delta w_{GHOST} = [1]\Delta w_{INTERIOR} \tag{3.5}$$

However for the application of the far-field boundary conditions in the case of subsonic flow, the coefficients A and B have to be derived from the linearization of Equation 2.61. In the linearization the change of the variables that are interpolated from the free-stream will be equated to zero. For example, at subsonic inflow the entropy is interpolated from the free-stream and it will not change as the Newton iterations proceeds.

$$\begin{aligned} s_{bound} &= \frac{\rho_{\infty}^{\gamma}}{P_{\infty}} \\ \Delta s_{bound} &= 0 \end{aligned} \tag{3.6}$$

To demonstrate the methodology followed in the implicit boundary conditions implementation, the linearization of the density term for the subsonic inflow boundary condition is formulated in below equations:

$$\begin{aligned} \rho_{GHOST} &= \left(\frac{s_{bound} c_{bound}^2}{\gamma} \right)^{\frac{1}{\gamma-1}} \\ \Delta\rho_{GHOST} &= 2\frac{\rho_{GHOST}}{\gamma-1}\left(\frac{\Delta c_{bound}}{c_{bound}} \right) \end{aligned} \tag{3.7}$$

The c_{bound} will be evaluated from the linearization of the Riemann invariants as

$$\begin{aligned} c_{bound} &= \frac{\gamma-1}{4}(\mathbb{R}^{+} - \mathbb{R}^{-}) \\ \Delta c_{bound} &= \frac{\gamma-1}{4}(\Delta\mathbb{R}^{+} - \Delta\mathbb{R}^{-}) \end{aligned} \tag{3.8}$$

where

$$\mathbb{R}^{+} = U_{\infty} + \frac{2c_{\infty}}{\gamma - 1} \qquad (3.9)$$
$$\mathbb{R}^{-} = U_{in} - \frac{2c_{in}}{\gamma - 1}$$

Since $\Delta\mathbb{R}^{+}$ is evaluated from the free stream its change will equal to the zero. Hence $\Delta c_{boundary}$ can be calculated as

$$\Delta c_{bound} = \left[\frac{1-\gamma}{4}(\Delta U_{in}) + \frac{\Delta c_{in}}{2} \right] \qquad (3.10)$$

Using the interior flow variables, the evaluation of the speed of sound and its differentiation can be performed as :

$$\begin{aligned} c_{in} &= \sqrt{\gamma R T_{in}} = \sqrt{\gamma \frac{P_{in}}{\rho_{in}}} = \sqrt{\gamma(\gamma-1)\left[\frac{\rho E}{\rho} - \frac{(\rho u)^2 + (\rho v)^2}{2\rho^2} \right]_{in}} \\ \Delta c_{in} &= \left\{ \frac{\gamma(\gamma-1)}{2c_{in}} \left[\frac{(\rho u)^2 + (\rho v)^2}{\rho^3} - \frac{\rho E}{\rho^2} \right]_{in} \right\} \Delta(\rho)_{in} \\ &+ \left\{ \frac{\gamma(\gamma-1)}{2c_{in}} \left[-\frac{\rho u}{\rho^2} \right]_{in} \right\} \Delta(\rho u)_{in} + \left\{ \frac{\gamma(\gamma-1)}{2c_{in}} \left[-\frac{\rho v}{\rho^2} \right]_{in} \right\} \Delta(\rho v)_{in} \\ &+ \left\{ \frac{\gamma(\gamma-1)}{2\rho_{in} c_{in}} \right\} \Delta(\rho E)_{in} \end{aligned} \qquad (3.11)$$

U_{in} is the contravariant velocity normal to the boundary, and it can be calculated from the inner cells as:

$$U_{in} = \eta_y \left(\frac{\rho u}{\rho} \right)_{in} - \eta_x \left(\frac{\rho v}{\rho} \right)_{in} \qquad (3.12)$$

Finally U_{in} can be linearized as:

$$\Delta U_{in} = \eta_y \left(\frac{\Delta(\rho u)}{\rho} - \frac{\rho u}{\rho^2} \Delta\rho \right)_{in} - \eta_x \left(\frac{\Delta(\rho v)}{\rho} - \frac{\rho v}{\rho^2} \Delta\rho \right)_{in} \qquad (3.13)$$

Inserting Equations 3.8 to 3.13 into the Equation 3.7 the change of the density at the far-field ghost cell with respect to change of flow variables at the interior cells can be defined as:

$$\Delta\rho_{ghost} = 2\frac{\rho_{ghost}}{c_{bound}}\left\{\begin{array}{c}\left(\frac{1}{4\rho^2}\left[\eta_y(\rho u)-\eta_x(\rho v)\right]_{in}+\frac{\gamma}{2c_{in}}\left[\frac{(\rho u)^2+(\rho v)^2}{\rho^3}-\frac{\rho E}{\rho^2}\right]_{in}\right)\Delta(\rho)_{in}\\ +\left(\frac{\eta_y}{(\gamma-1)\rho}+\frac{\gamma}{2c_{in}}\left[-\frac{(\rho u)_{in}}{\rho_{in}^2}\right]\right)\Delta(\rho u)_{in}\\ -\left(\frac{\eta_x}{(\gamma-1)\rho}-\frac{\gamma}{2c_{in}}\left[-\frac{(\rho v)_{in}}{\rho_{in}^2}\right]\right)\Delta(\rho v)_{in}\\ \left(\frac{\gamma}{2\rho_{in}c_{in}}\right)\Delta(\rho E)_{in}\end{array}\right\} \tag{3.14}$$

3.4 Implementation of Initial Conditions

Using the method of lines, the initial solution is required by the spatial discretization for the evaluation of the flux integrals. The residuals calculated by initial solution will be integrated in time by the temporal discretization scheme, which is Newton method in this study. Newton's method requires a good initial guess for convergence. The common way of implementation of initial conditions is usage of the free-stream values for whole domain. Although this kind of implementation is very simple, the initial guess it provides is poor for most of the flow problems. Hence in order to improve the stability of the solution, some modifications are required in initial iterations. Several ideas are available to modify the Newton's method in literature. The most efficient and widely used one is the time-like term addition to the diagonal of Jacobian matrix. This approach is based on the fact that diagonally dominant matrices are more stable in solutions of nonlinear systems. With the implementation of the time like term, the modified Newton's method becomes:

$$\left(\frac{1}{\Delta t}[I]+\frac{\partial R}{\partial \boldsymbol{w}}\right)^n \Delta \boldsymbol{w}^n = -R(\boldsymbol{w}^n) \tag{3.3}$$

For finite Δt values; Newton's method becomes inexact, since the left hand side of implicit operator of Equation 3.3 is not consistent with exact linearization of the residual. As $\Delta t \rightarrow \infty$, the original Newton's method can be reconstructed. The modification to Newton's method is performed by addition of small initial value Δt_0 for the first iteration and Δt values for new iterations are calculated using L_2-norm of the residuals as:

$$\Delta t^n = \Delta t^0 \frac{\left\| R(\boldsymbol{w}^0) \right\|_2}{\left\| R(\boldsymbol{w}^n) \right\|_2} \tag{3.4}$$

The penalty of the modification mentioned above is the reduction of the convergence rate. Pulliam[62] mathematically showed that with the diagonal term addition convergence rate is reduced to linear, whereas quadratic convergence rate can be obtained as time like term approaches to infinity.

The modification in Newton's method is required in the early stages of iterations. As the iterations proceeds the solution gets more accurate and the diagonal term addition may not be needed. Therefore, the withdrawal of the diagonal term from the matrix as it gets satisfactorily large will be favorable to get rid of the convergence rate penalty of the modification. The proper choice for initial time like term, Δt_0, and its withdrawal value, Δt_f will significantly boosts up the convergence performance of the solver. However, the right choices for the values of these terms can only be made by trial and error.

3.5 Solution Method

The flow solution with Newton method requires the construction and the factorization of the Jacobian matrix in order to evaluate the change in the flow variables. The entries of this matrix are non-linear function of the changing flow variables, hence at each Newton's iteration the Jacobian matrix has to be reconstructed and solved again. The size of this matrix can be very big even for small sized problems. The storage of this matrix may be difficult for large sized problems. However, most of the entries of this matrix are zero, and sparse matrix solution and storage techniques may be useful in the solution of Jacobian matrix. There are different methods to solve sparse matrices. In this study, the

Jacobian matrix is solved by using an LU decomposition method. The advantage of this method is that the same matrix can be solved for different right-hand-sides very efficiently. This property is very useful for sensitivity calculations, because the calculation of sensitivities for each design variables require the solution of the same Jacobian matrix with different right-hand-sides.

This study proposes the usage of modern multifrontal sparse matrix solvers for the reduction of the cost of the factorization. The storage cost of the Jacobian can also be reduced by the usage of compressed row storage or compressed column storage formats. UMFPACK (Unsymmetric-pattern MultiFrontal PACKage) sparse matrix solver package [22] is used in order to solve Jacobian matrix. In this method, the full matrix is converted into sparse storage mode and then factorized using a sequence of small dense frontal matrices by LU factorization.

3.6 Solver Performance

In this section flow solution results calculated by the developed solver are presented. Performance of the solver is examined in terms of the accuracy of the solutions and rate of the convergence. CPU time spent in the flow solution is given to present the cost of the direct solution procedure.

3.6.1 Verification of Flow Solver

Commonly used 2-D internal and external flow test cases are used to validate the accuracy of the developed flow solver. For internal flow application, flow over circular arc geometry, which is also known as Ni bump, is used. The solution domain consists of a channel. The width of the channel equals to the length of the circular arc bump. The total length of the channel equals to five times of the bump's length. Results are presented for subsonic, transonic, and supersonic flow conditions at zero angle of attack. For subsonic and transonic calculations, the thickness-to-chord ratio is 10% and for

supersonic flow calculations it is 4%. The solved test cases are summarized in Table 3.1 below.

Table 3.1 *Flow problem cases for bump geometry*

	thickness/chord length	Inlet Mach number
Bump case-1	0.1	0.5
Bump case-2	0.1	0.675
Bump case-3	0.04	1.4

Flow domain is discretized by an H type grid for the solution of flow over bump problem. The mesh is constructed with three different level of resolution. The coarsest mesh has 33x09 nodes, mid sized one has 65x17 and the finest one has 129x33 nodes. The finest grid is presented in the Figure 3.4. Wall boundary conditions are used on the bump geometry. The Riemann invariants are used to apply characteristic type boundary conditions at the inlet and outlet boundaries. The symmetry boundary conditions are applied at the remaining lower and upper boundaries. The mach contour plots evaluated from the solutions of those three cases are given in Figure 3.5. The 2^{nd} order AUSM scheme with Van Albada limiter is used to generate those results.

In the verification of the results obtained for bump geometry the Mach contour plots given in reference [63] is used. In Figures 3.6 to 3.8 the evaluated mach contours are presented in the lower part of the figures and the results given by [63] are presented in the upper part. Figures 3.6 to 3.8 present subsonic, transonic and supersonic solutions respectively. Those figures show that the developed flow solver performs well for predicting the flow over bump test case. The evaluated mach contours are in good agreement with ones presented in [63]

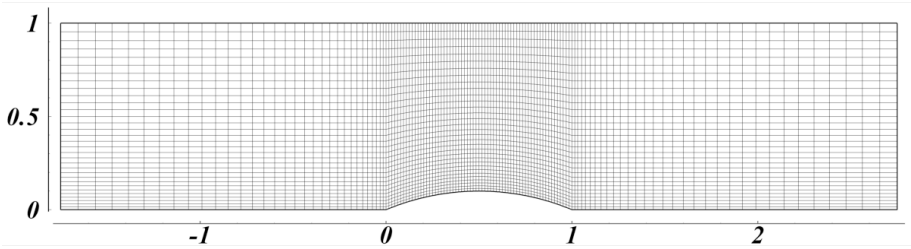

Figure 3.4 *Grid used for bump geometry*

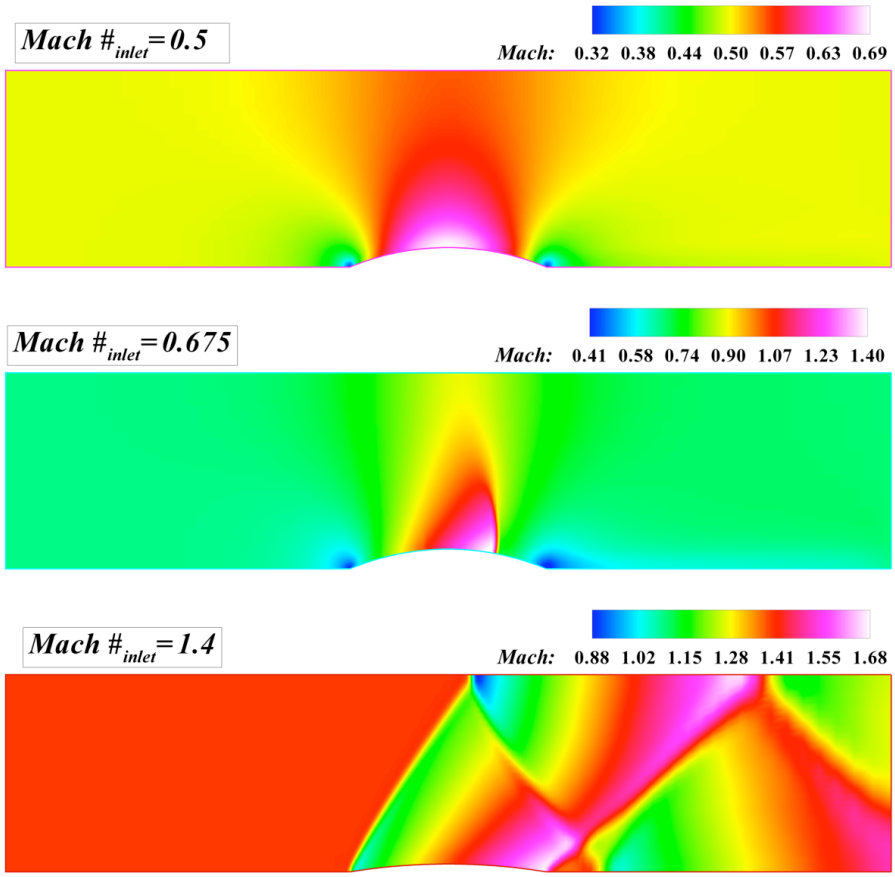

Figure 3.5 *Mach contours of the cases of bump geometry*

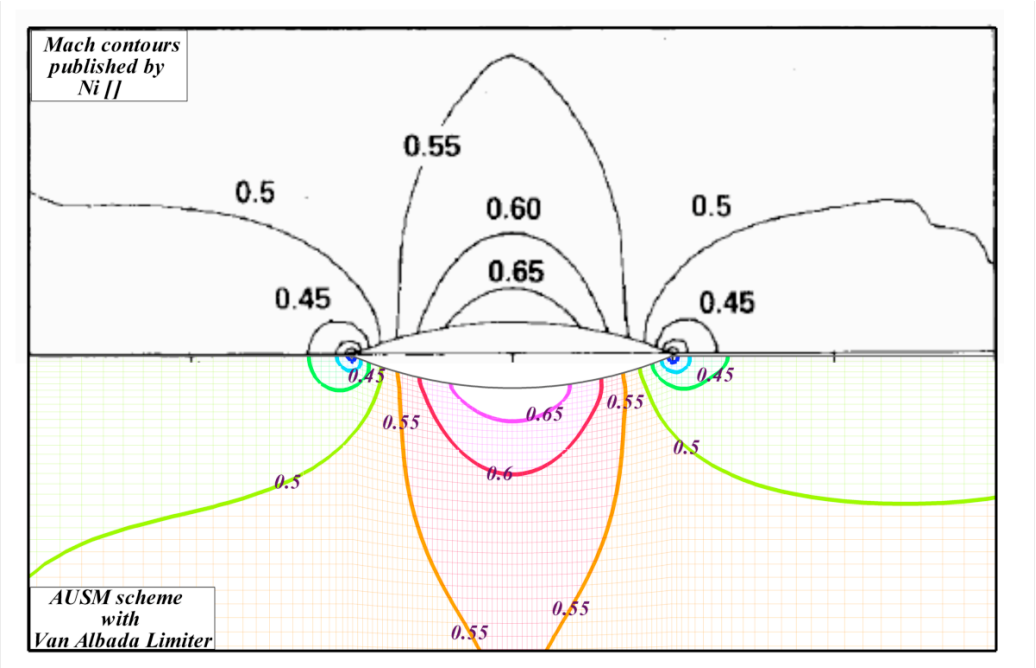

Figure 3.6 *Mach contour comparison for flow with 0.5 Mach inlet velocity*

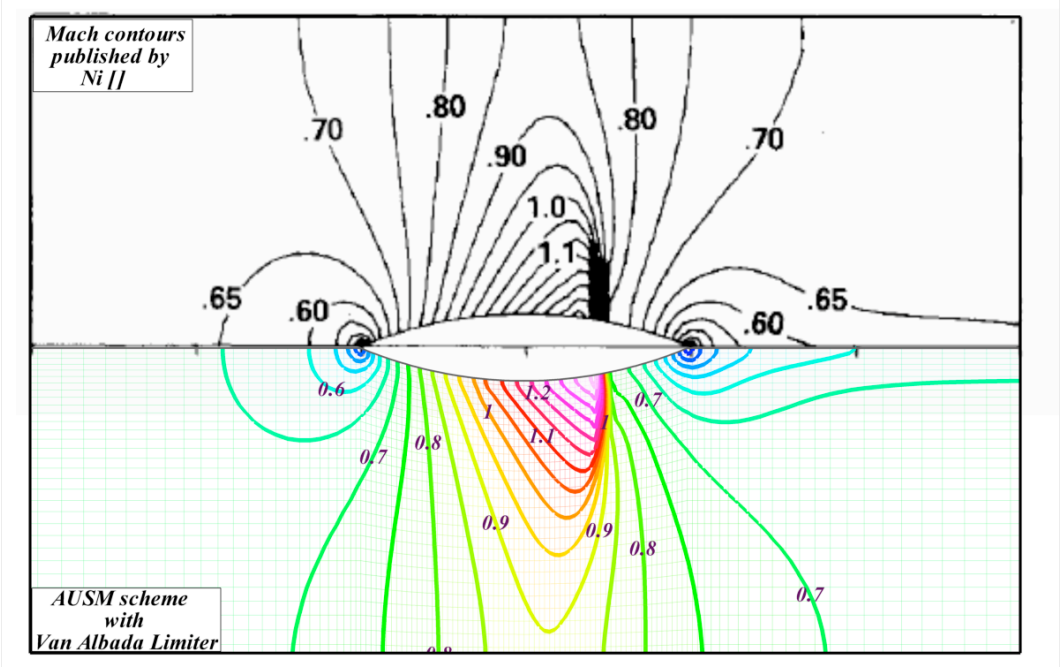

Figure 3.7 *Mach contour comparison for flow with 0.675 Mach inlet velocity*

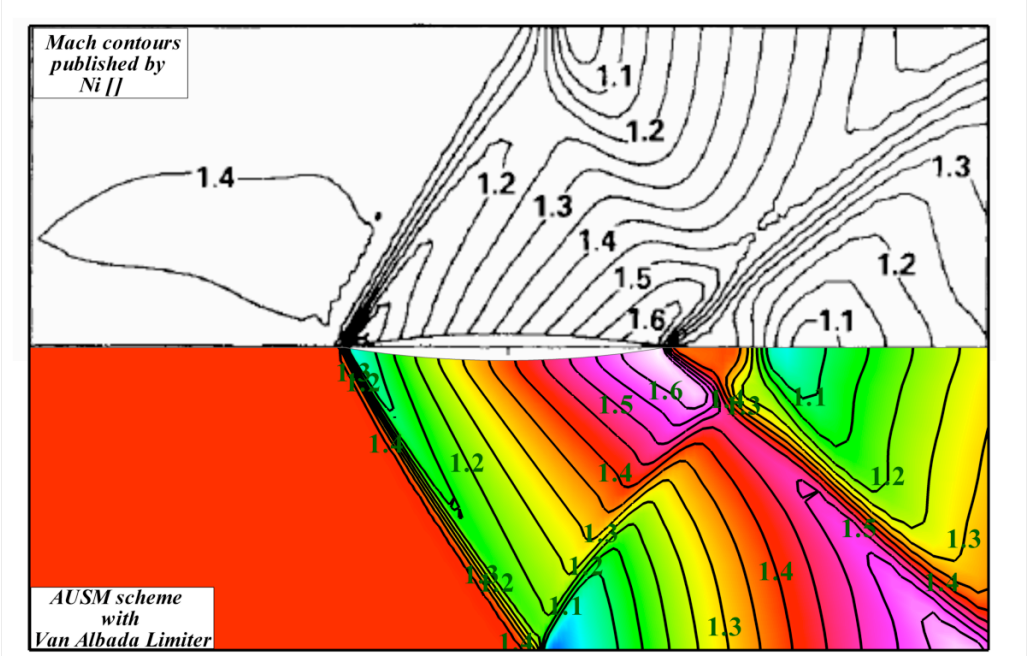

Figure 3.8 *Mach contour comparison for flow with 2.0 Mach inlet velocity*

External flow simulations are performed by the airfoil geometries. The test cases are chosen from the AGARD Advisory Report AR-211 [64]. The operating conditions and the airfoil geometries that are used to generate flow solutions are summarized in Table 3.2.

Table 3.2 *Flow cases for airfoil geometry*

	Airfoil	Angle of attack	Free stream Mach number
case-1	NACA 0012	1°	0.85
case-2	NACA 0012	7°	1.2
case-3	RAE 2822	3°	0.75

Flow domain is discretized by a C type grid for the solutions of flow over airfoil problems. The meshes are constructed with three different level of resolution. The total number of nodes in the finest grid is 257x65, and this grid has 160 nodes on the airfoil surface. The mid size grid has 120 points on the airfoil geometry and 193x49 overall

nodes. The finest one is constructed by 257x65 nodes by placing 160 nodes on the airfoil. Wall boundary conditions are applied on the airfoil geometry and the Riemann invariants are used to define farfield boundary conditions. The symmetry type boundary condition is utilized for the wake cut.

The mach contours given by Reference [64] for the cases listed in Table 3.2 are presented in Figures 3.9, 3.11 and 3.13. The mach contour plots evaluated from the solutions of those three cases are given in Figures 3.10, 3.12 and 3.14. The 2^{nd} order AUSM scheme with Van Albada limiter is used to generate those plots. Results show that, the developed solver generates reliable flow solutions. In all three cases the shock capturing performance of the code is well enough that the evaluated location and the strength of the shock is in good agreement with the ones given in [64].

The given evaluated contour plots show that, similar mach contours are evaluated with respect to reference [64]. Besides examining the mach contour plots, the evaluated pressure coefficient distribution along the chord is compared with the one given in reference [64]. The case; flow over NACA 0012 airfoil with 0.85 Mach, 1 degree angle of attack is used for that comparison. That comparison is given in Figure 3.15 where the evaluated values are represented by lines and the values given in reference [64] is represented by symbols. This case is selected deliberately to examine the shock capturing performance of the developed solver. Figure 3.15 show that both of the limiters used in the study performs pretty well in shock capturing. However the performance of the Van Albada's limiter seems to be slightly better at the region of the lower surface shock in the solved case. More precise shock capturing can be performed by using finer meshes or using adaptive mesh algorithms, however that kind of application is out of scope of this thesis. Moreover the increase in the mesh size will significantly amplify the cost of the direct flow solution.

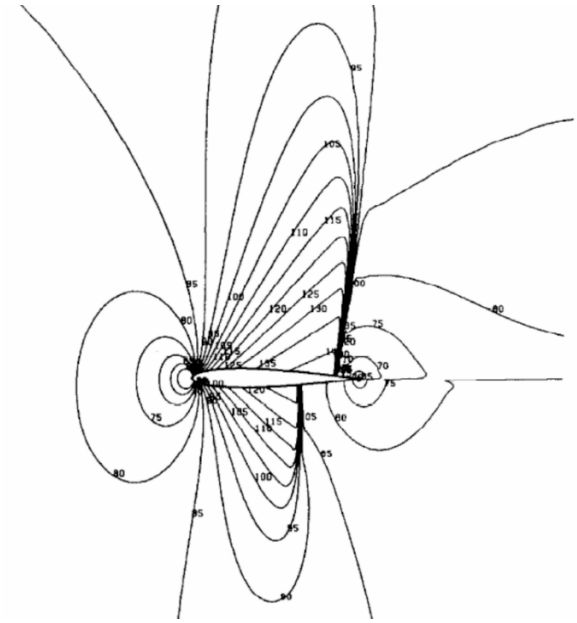

Figure 3.9 *Mach contours given in AGARD AR-211 for NACA0012, M_∞ =0.85, α=1^o*

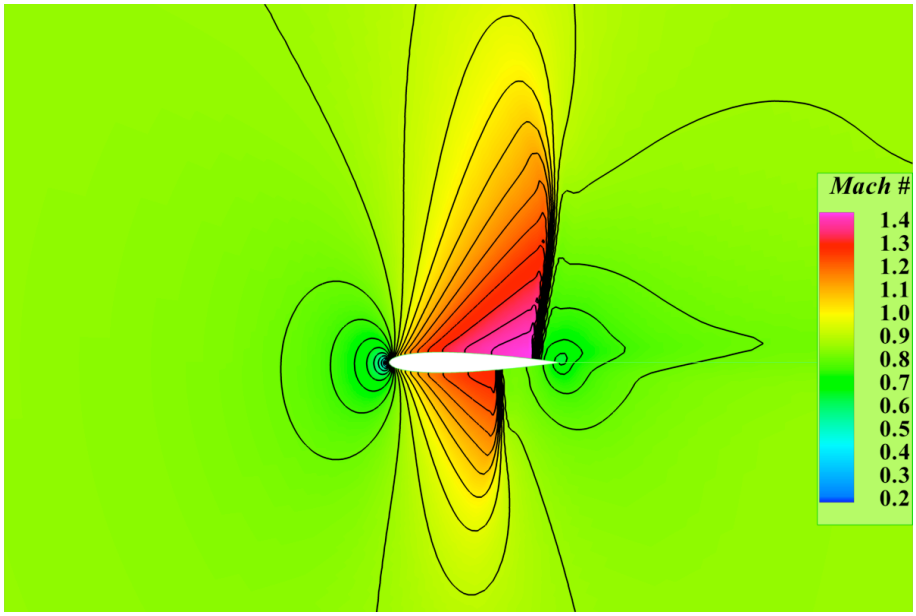

Figure 3.10 *Mach contours evaluated for NACA0012, M_∞ =0.85, α=1^o*

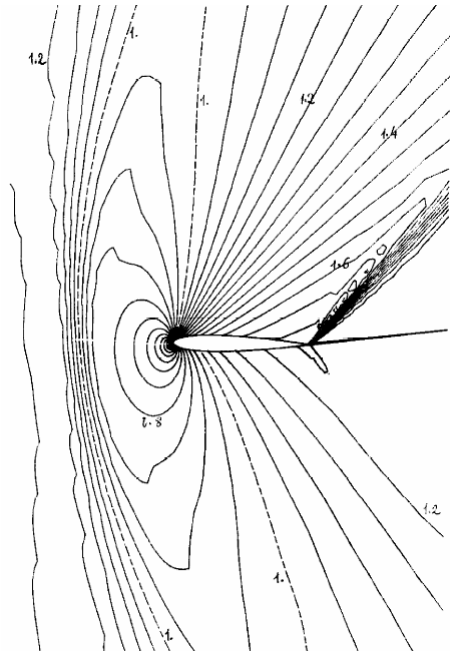

Figure 3.11 *Mach contours given in AGARD AR-211 for NACA0012, M_∞ =1.2, α=7^o*

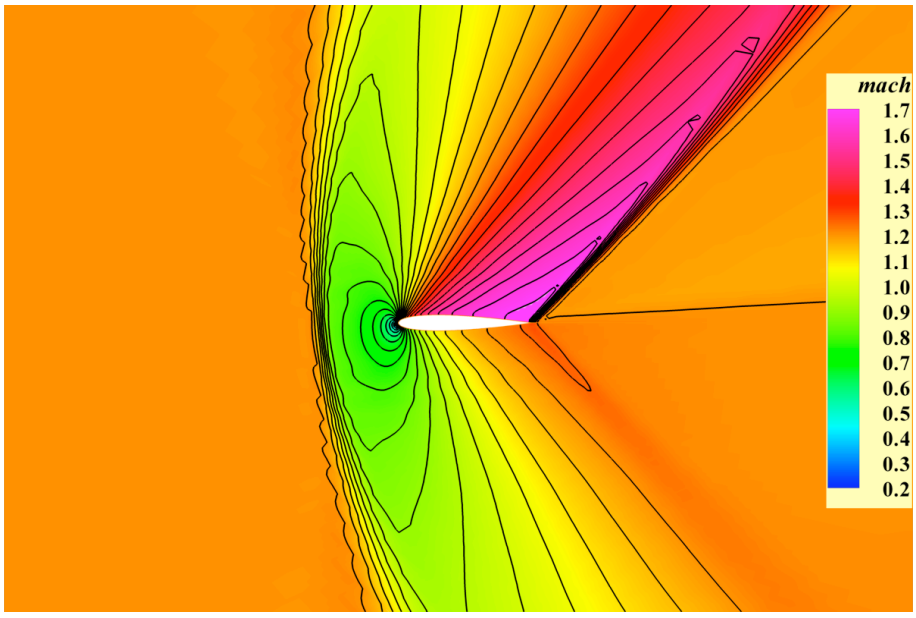

Figure 3.12 *Mach contours evaluated for NACA0012, M_∞ =1.2, α=7^o*

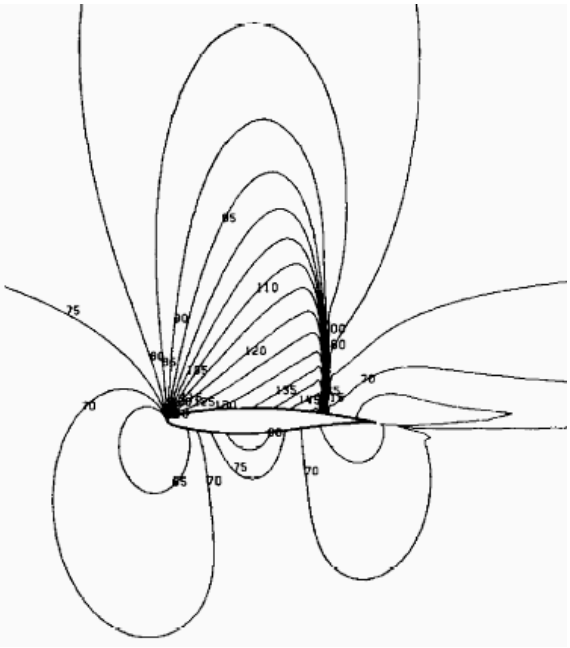

Figure 3.13 *Mach contours given in AGARD AR-211 for RAE2822, M_∞ =0.75, α=3^o*

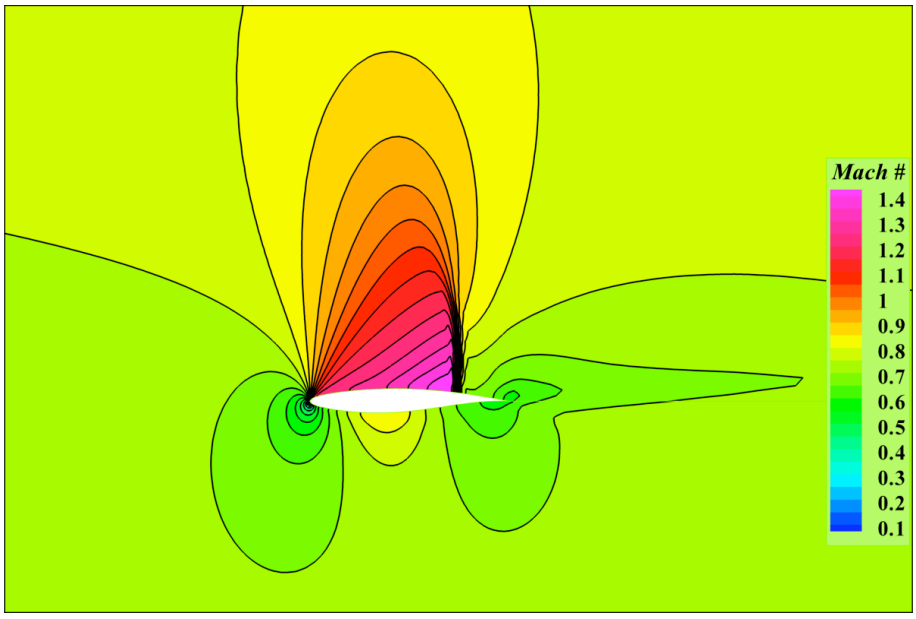

Figure 3.14 *Mach contours evaluated for RAE2822, M_∞ =0.75, α=3^o*

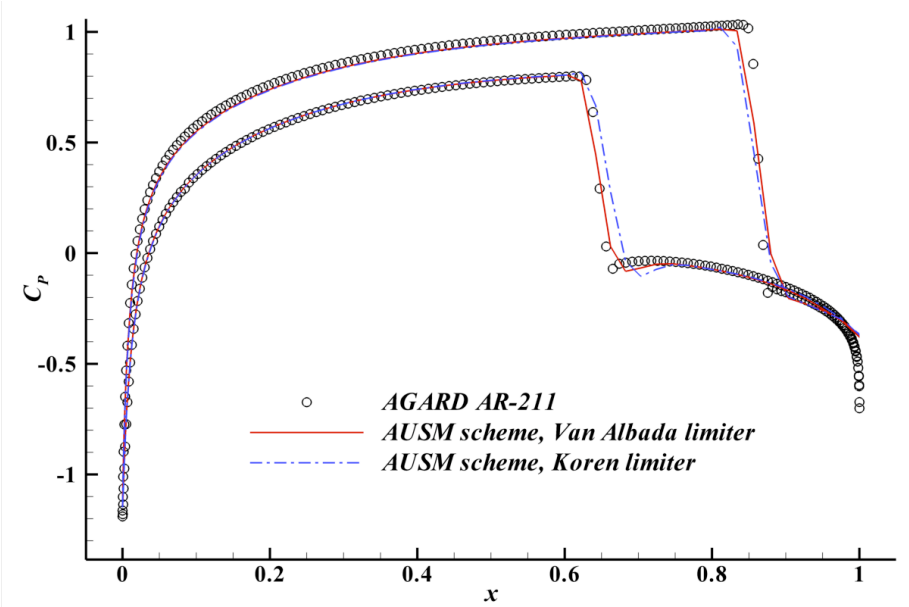

Figure 3.15 *Comparison of the evaluated Cp with the one given Reference[64]*

3.6.2 Convergence Performance

In previous section the developed code is shown to be reliable in the solution of 2-D Euler flow field equations for all flow regimes. In this chapter the convergence performance of the direct solution algorithm is presented and the factors that affect the convergence performance is analyzed.

The way of time like term addition to the diagonal of the Jacobian matrix is the most critical factor on the convergence of the flow solution with Newton's method. Actually those terms have almost no affect on the rate of the convergence. Newton scheme will always have quadratic convergence rate if there is no diagonal term addition and by the addition the convergence will reduce to linear rate. The purpose of the diagonal term addition was to improve the initial conditioning of the Jacobian and this term should be withdrawn as the Jacobian is updated with new solutions. To be able to have converged solution within minimum iterations the proper values must be chosen for the initial and the withdrawal values of that diagonal term. However there is no explicit rule for the selection of those values and generally good selection can be made by trial error like procedure.

To illustrate the response of the convergence behavior to diagonal term selection, convergence histories obtained by variety of initial and withdrawal diagonal term values are presented in Table-3.3. The given results are corresponds the flow solution over NACA0012 airfoil with 0.85 Mach freestream velocity. The solutions are performed by AUSM scheme and the Van Albada limiter on the grid which is sized by 129x33 nodes

Table 3.3 ***Effect of*** Δt ***on convergence, NACA0012 airfoil***

Δt_0	Δt_f	iterations required for convergence	
		2^{nd} order discretization	1^{st} order discretization
1	3	Factorization is crashed	105
1	5	867	132
1	10	1097	265
1	25	Factorization is crashed	404
1	100	Factorization is crashed	463
1	1000	1259	478
1	10^6	1267	481
1	10^{12}	1268	483
10	100	98	36
10	75	1230	463
10	50	Factorization is crashed	6
100	1000	Factorization is crashed	13
100	5000	27	14
150	5000	23	12
200	2000	19	11
250	5000	Factorization is crashed	11
300	5000	20	12

The results tabulated above show that there is no explicit rule for the selection of the initial and withdrawal values of the diagonally added term. The values presented in the Table-3.3 are just given to demonstrate the variation of the required iterations for convergence with respect to varying diagonal term addition. The convergence behavior

can be significantly different for different flow solution cases although the same initial and withdrawal time like term values are used.

The residual histories of the transonic flow solutions for the bump geometry and the NACA0012 airfoil are presented in Figure 3.16 and Figure 3.17 respectively. In those figures the affect of the flux evaluation method on the convergence performance are presented for both of the first and second order spatially accurate solutions. The initial and withdrawal values of the diagonally added time like term used in the generation of those results are summarized in Table 3.4. In the figures of residual histories it can be seen that the residual decreases slightly in the initial iterations where the convergence rate is linear due to the added diagonally added time like term. The convergence occurs with the rapid decrement of the residual in the last three to five iterations. In both of the figures it is shown that, it takes less number of iterations to converge when the accuracy of the spatial discretization is first order. It can be also concluded that the all flux vector splitting methods have approximately identical effect on the convergence of the Newton's method.

Figure 3.18 shows the effect of the parameter ϵ, which is defined in the limiter function , on the convergence history of AUSM scheme. In Reference[65] the effects of the limiter functions on the convergence was studied. The results show that, ϵ is directly effective on the convergence performance of the solver. Deeper study on the magnitude of the, ϵ is presented in Chapter-4.

Table 3.4 *The initial and withdrawal values used for the diagonally added term, $1/\Delta t$ in the generation of presented results*

	Δt_0	Δt_f
1st order spatially accurate solution of flow over bump	3	5
2nd order spatially accurate solution of flow over bump	50	1000
1st order spatially accurate solution of flow over airfoil	300	10000
2nd order spatially accurate solution of flow over airfoil	200	20000

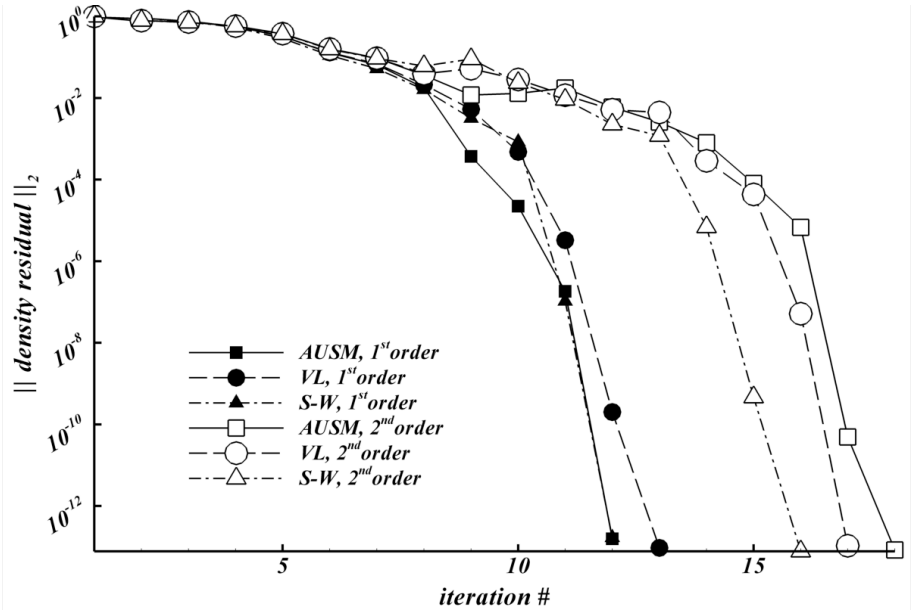

Figure 3.16 *Convergence history for bump geometry, 65x17 grid, 0.675 inlet Mach*

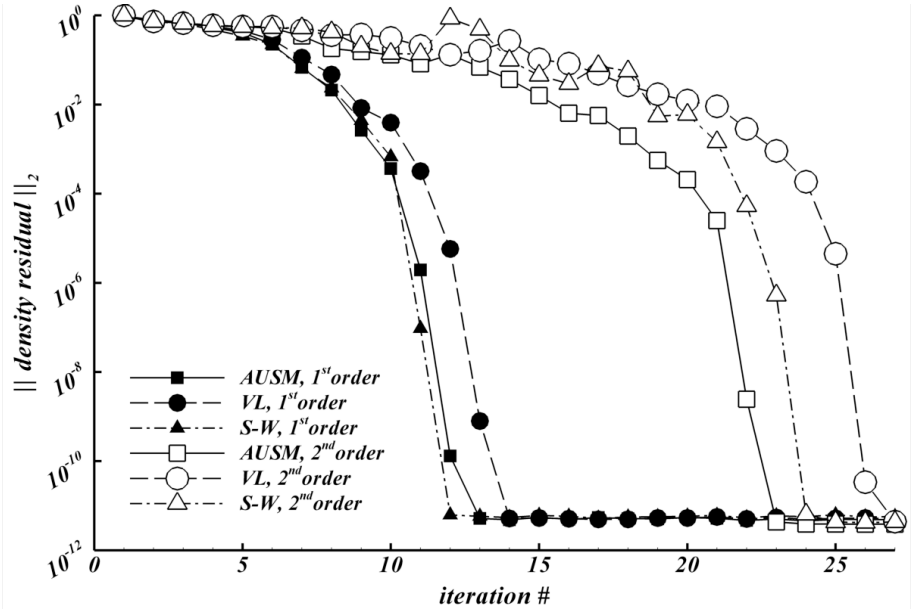

Figure 3.17 *Convergence history for NACA0012, 129x33 grid, 0.85 Mach free-stream flow with 1° angle of attack*

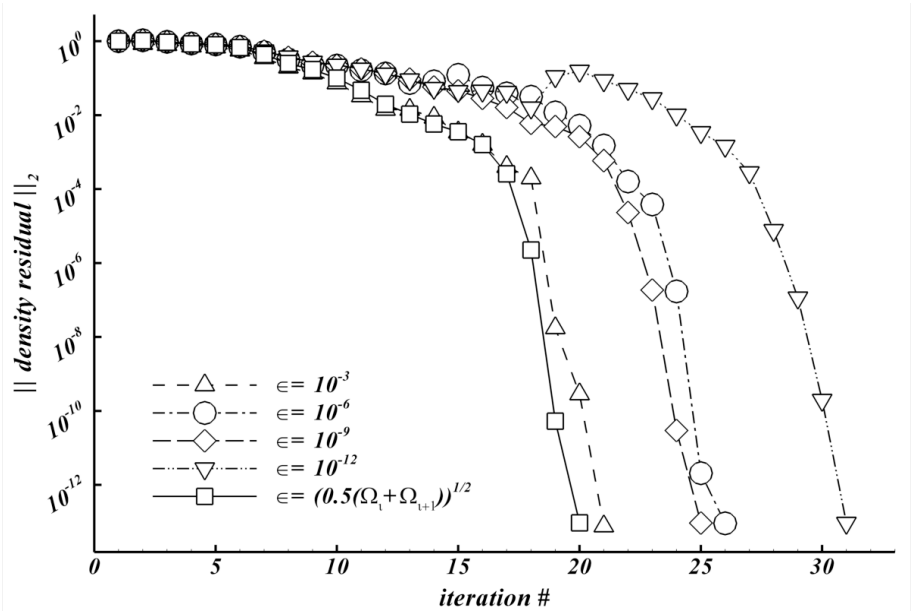

Figure 3.18 *Variation of the Convergence History with ∈ for Van Albada Limiter*

3.6.3 Cost of the Flow Solution in terms of CPU time

The variation of the CPU time spent in the flow solution is presented in Figure 3.19. In Figure 3.19, the upper left plot presents the CPU time spent in each Newton's iterations for the flux and the residual evaluation. Upper right plot gives the total CPU time spent at each iteration. The lower plot presents the ratio of the time spent in Jacobian construction and linear system solution to total time spent in each iterations.

The results presented in Figure 3.19 correspond to flow solution on NACA0012 airfoil by using the grid which has 257x65 nodes. That figure shows that the CPU time spent in Jacobian construction and solution of the linear system is significantly larger than the time spent in flux residual evaluation. In an iteration of Newton's method, approximately 99.9 percent of the total CPU time is spent in the construction of the Jacobian matrix and the solution of the linear system by sparse matrix solver. For first order discretization time spent in solution of linear system and the construction of the Jacobian matrix is found to be close. However for 2nd order spatial discretization, time spent by sparse matrix solver is significantly larger. In the factorization and solution approximately 4 times greater CPU time is spent with respect to the construction of analytical Jacobian matrix.

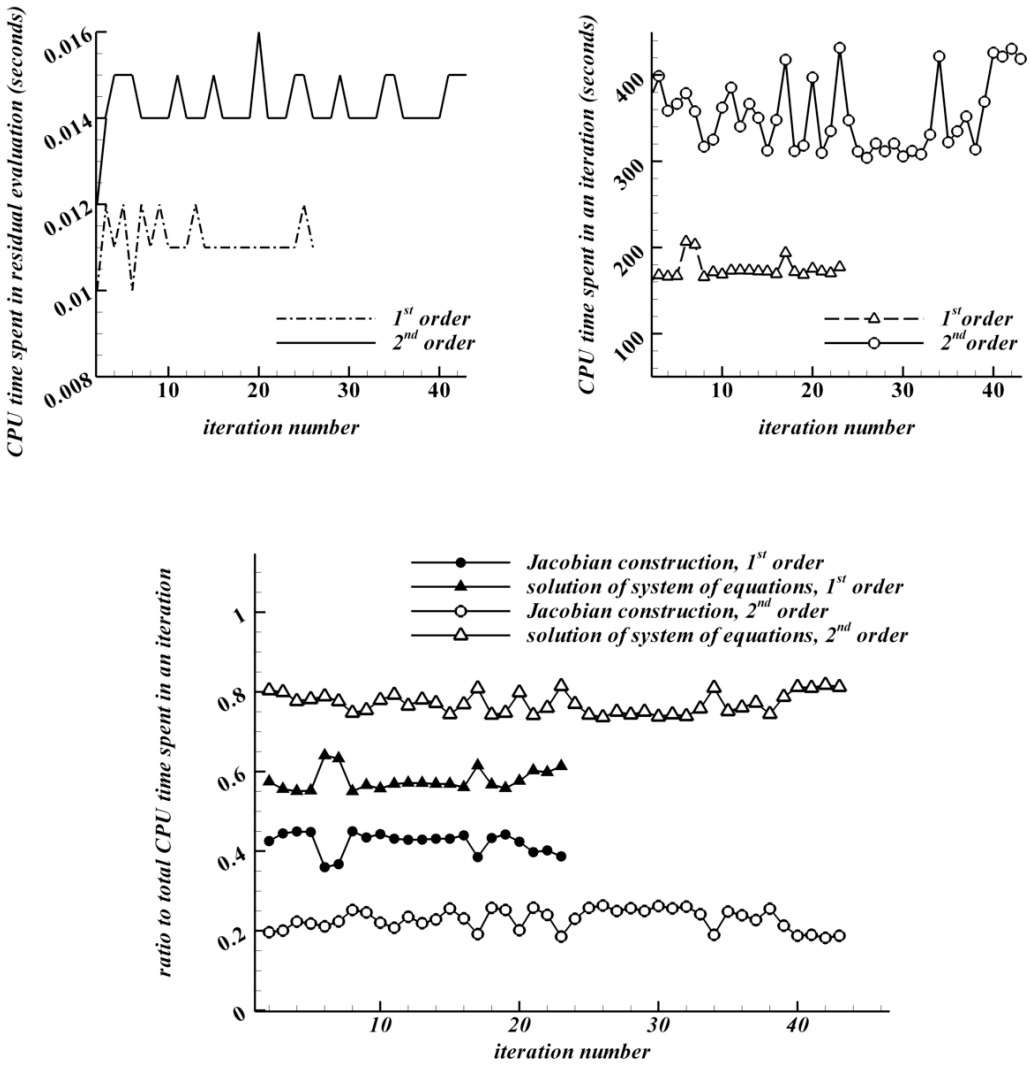

Figure 3.19 *CPU time spent in flow solution by Newton's method*

Tables 3.5 and 3.6 present the variation in CPU time spent in Newton's method by the change in grid resolution. The averages of the fluctuating values shown in above figures are used in those Tables. Finally the effect of the grid resolution on the ratio of CPU time spent in the Jacobian evaluation and factorization to the CPU time spent in flux evaluation is presented in Figure 3.20.

Table 3.5 *Variation of the spent CPU time in an iteration by the grid size (1st order discretization)*

NACA0012 M=0.85 $\alpha = 1^o$ 1st order	CPU time spent for flux and residual evaluation (seconds)	Ratio of time spent in Jacobian construction to time spent in flux evaluation	Ratio of time spent by sparse matrix solver to time spent in flux evaluation
129x33 nodes	0.0025	2900	2190
197x49 nodes	0.009	4140	3100
225x65 nodes	0.011	6110	6270
256x65 nodes	0.012	6770	9010

Table 3.6 *Variation of the spent CPU time in an iteration by the grid size (2nd order discretization)*

NACA0012 M=0.85 $\alpha = 1^o$ 2nd order	CPU time spent for flux and residual evaluation (seconds)	Ratio of time spent in Jacobian construction to time spent in flux evaluation	Ratio of time spent by sparse matrix solver to time spent in flux evaluation
129x33 nodes	0.0035	2350	3130
197x49 nodes	0.012	3135	5950
225x65 nodes	0.0145	5210	16500
256x65 nodes	0.016	5750	18200

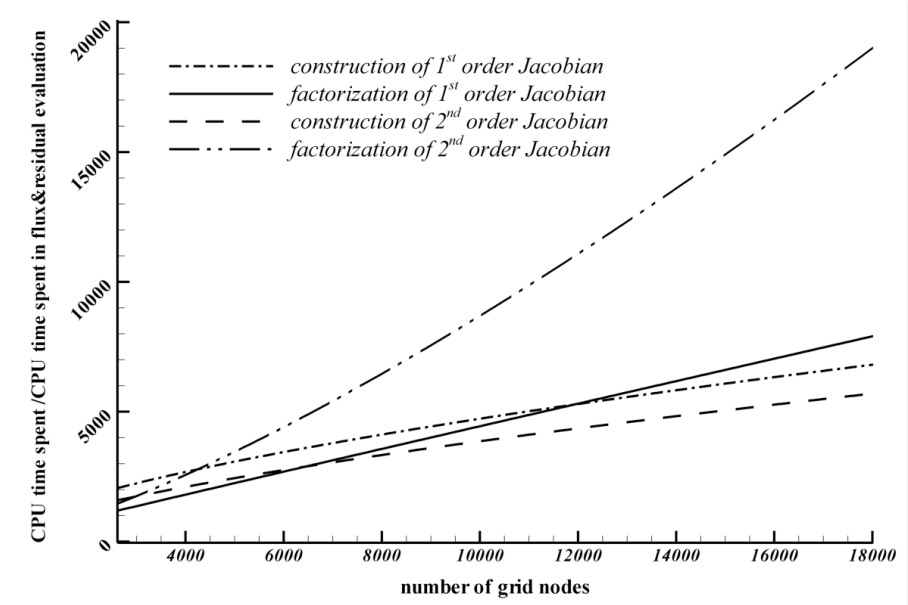

Figure 3.20 *Variation in the ratio of the CPU time spent in factorization and construction of the Jacobian to time spent in entire iteration*

CHAPTER 4

ACCUARCY OF NUMERICAL JACOBIAN

4.1 Introduction

The finite difference approach is the most practical alternative for the derivative evaluation. It is independent from the complexity of the differentiated function. Moreover, the effort needed to develop a numerical differentiation code is negligible with respect to the effort needed for the one that evaluates the derivatives analytically. However, the numerical evaluation is inaccurate and the associated error can be unacceptably large due to the improper choice of the finite differencing step size.

This chapter mainly focuses on the effect of the finite difference perturbation magnitude on the accuracy of numerical flux Jacobian evaluation. First, the sources of the error and their dependencies on the perturbation magnitude are given. Next, the study which intends to derive the formulation for the optimum perturbation magnitude is presented. The optimum perturbation magnitude that minimizes the error in numerical Jacobian is also searched by a trial-error like procedure. The variety of perturbation magnitudes are used to calculate finite difference derivatives and the resulting Jacobian matrices are compared with the analytically evaluated one. The variation of the error in the numerical evaluation with respect to perturbation magnitude is plotted to find the optimum magnitude that minimizes the error. The change of the error and the optimum perturbation magnitude by variety of factors, such as grid size, flow regime, discretization technique etc. is examined. The performance of the numerical Jacobian in the flow solution is compared with the performance of the analytical Jacobian in terms of the CPU time and convergence rate. Finally, the influence of the numerical Jacobian error on the convergence of the flow solution is presented.

4.2 Error Analysis

The derivation of numerical Jacobian by forward finite differencing was given in previous chapter as:

$$\frac{\partial R_m}{\partial \boldsymbol{w}_n} = \frac{\Delta R_m(\boldsymbol{w})}{\Delta \boldsymbol{w}_n} = \frac{R_m(\boldsymbol{w} + e_n\varepsilon) - R_m(\boldsymbol{w})}{\varepsilon} \tag{4.1}$$

In that kind of finite difference approximation of the differentiation, mainly two types of errors occur. These are truncation and condition errors [61]. The truncation error occurs due to neglected terms in the Taylor's series and it grows up linearly with the perturbation magnitude. The truncation error due to the neglected terms in the Taylor series expansion can be written as:

$$E_{Trun}(\varepsilon) = \frac{\partial^2 R_m(\zeta)}{\partial \boldsymbol{w}_n^2}\frac{\varepsilon}{2} \tag{4.2}$$

where ζ=[$\boldsymbol{w}_n$, $\boldsymbol{w}_n + \varepsilon$].

The condition error is caused by inaccuracies in the computed values due to the loss of computer precision. As the magnitude of the perturbation gets smaller the accuracy of the differentiation degrades and the error grows up due to the decrement in the denominator of the Equation 4.1. Because of computational precision, the exact values of the mth components of the vector $R_m(\boldsymbol{w})$ and their computed values $\tilde{R}_m(\boldsymbol{w})$ can be different by the amount of round-off error, $E_{R_m}(\boldsymbol{w})$:

$$\begin{aligned} \tilde{R}_m(\boldsymbol{w}) &= R_m(\boldsymbol{w}) + E_{R_m}(\boldsymbol{w}) \\ \tilde{R}_m(\boldsymbol{w} + \varepsilon) &= R_m(\boldsymbol{w} + \varepsilon) + E_{R_m}(\boldsymbol{w} + \varepsilon) \end{aligned} \tag{4.3}$$

By using the computed function, $\tilde{R}_m(\boldsymbol{w})$ Equation 4.1 can be written as:

$$\frac{\Delta \tilde{R}_m(\boldsymbol{w})}{\Delta \boldsymbol{w}_n} = \frac{\tilde{R}_m(\boldsymbol{w} + e_n\varepsilon) - \tilde{R}_m(\boldsymbol{w})}{\varepsilon}$$
$$\frac{\Delta \tilde{R}_m(\boldsymbol{w})}{\Delta \boldsymbol{w}_n} = \frac{R_m(\boldsymbol{w} + e_n\varepsilon) - R_m(\boldsymbol{w})}{\varepsilon} + \frac{E_{R_m}(\boldsymbol{w} + e_n\varepsilon) - E_{R_m}(\boldsymbol{w})}{\varepsilon} \tag{4.4}$$
$$\frac{\Delta \tilde{R}_m(\boldsymbol{w})}{\Delta \boldsymbol{w}_n} = \frac{\Delta R_m(\boldsymbol{w})}{\Delta \boldsymbol{w}_n} + E_{C_m}(\varepsilon)$$

where, $E_C(\varepsilon)$ is the condition error. Considering an error bound $E_R = \max\{|E_R(\boldsymbol{w})|, |E_R(\boldsymbol{w}+\varepsilon)|\}$, the maximum value of condition error can be approximated as:

$$E_C(\varepsilon) = \frac{2E_R}{\varepsilon} \tag{4.5}$$

The main source of the round-off error is the precision lost and it depends on the computer processor and compiler. For the computations of normalized variables where the magnitude of the computed values are around one, the precision error equals to machine epsilon, ε_M. A reasonable calculation procedure for ε_M can be given as follows:

$$\varepsilon_M = \frac{1}{2^m} \quad \text{such that} \quad 1 + \varepsilon_M > 1 \tag{4.6}$$

where, m is the number of possible highest bits in the binary representation of the mantissa.

4.3 Optimal Perturbation Magnitude Analysis

The total error in numerical Jacobian equals to sum of truncation and condition errors. The total error is highly dependent on perturbation magnitude, ε. For the small values of perturbation magnitude the condition error is large and it dominates the magnitude of the total error. On the other hand as the magnitude of perturbation gets larger, the truncation error becomes larger and the condition error becomes negligible. As a result the total error grows up both for the increasing and decreasing perturbation magnitudes. Hence,

there should be an optimal value for the perturbation magnitude that minimizes the total error in numerical Jacobian.

The total error matrix can be defined as the difference between the numerically and analytically calculated Jacobian matrices.

$$[E_{total}] = \begin{pmatrix} \frac{\Delta\tilde{R}_1}{\Delta w_1} & \frac{\Delta\tilde{R}_1}{\Delta w_2}\cdots & \frac{\Delta\tilde{R}_1}{\Delta w_{mmax}} \\ \frac{\Delta\tilde{R}_2}{\Delta w_1} & \frac{\Delta\tilde{R}_2}{\Delta w_2}\cdots & \frac{\Delta\tilde{R}_2}{\Delta w_{mmax}} \\ \vdots & \vdots & \vdots \\ \frac{\Delta\tilde{R}_{mmax}}{\Delta w_1} & \frac{\Delta\tilde{R}_{mmax}}{\Delta w_1}\cdots & \frac{\Delta\tilde{R}_{mmax}}{\Delta w_{mmax}} \end{pmatrix} - \begin{pmatrix} \frac{\partial R_1}{\partial w_1} & \frac{\partial R_1}{\partial w_2}\cdots & \frac{\partial R_1}{\partial w_{mmax}} \\ \frac{\partial R_2}{\partial w_1} & \frac{\partial R_2}{\partial w_2}\cdots & \frac{\partial R_2}{\partial w_{mmax}} \\ \vdots & \vdots & \vdots \\ \frac{\partial R_{mmax}}{\partial w_1} & \frac{\partial R_{mmax}}{\partial w_2}\cdots & \frac{\partial R_{mmax}}{\partial w_{mmax}} \end{pmatrix} \tag{4.7}$$

To minimize the norm values of the total error matrix, each finite difference computations can be performed with their own optimum perturbation magnitudes. However this kind of approach will be impractical since the calculation procedure for the optimum perturbation value for each element would be costly. Alternative way is to find a single perturbation magnitude which minimizes the global total error arising from the finite differencing of each element. The total error matrix for a single perturbation magnitude, ε, can be approximated with the following matrix:

$$[E_{total}] \approx \begin{pmatrix} \frac{\partial^2 R_1}{\partial w_1^2}\frac{\varepsilon}{2} + \frac{2E_{R_{1,1}}}{\varepsilon} & \frac{\partial^2 R_1}{\partial w_2^2}\frac{\varepsilon}{2} + \frac{2E_{R_{1,2}}}{\varepsilon}\cdots & \frac{\partial^2 R_1}{\partial w_{mmax}^2}\frac{\varepsilon}{2} + \frac{2E_{R_{1,mmax}}}{\varepsilon} \\ \frac{\partial^2 R_2}{\partial w_1^2}\frac{\varepsilon}{2} + \frac{2E_{R_{2,1}}}{\varepsilon} & \frac{\partial^2 R_2}{\partial w_1^2}\frac{\varepsilon}{2} + \frac{2E_{R_{2,2}}}{\varepsilon}\cdots & \frac{\partial^2 R_2}{\partial w_1^2}\frac{\varepsilon}{2} + \frac{2E_{R_{2,mmax}}}{\varepsilon} \\ \vdots & \vdots & \vdots \\ \frac{\partial^2 R_{mmax}}{\partial w_1^2}\frac{\varepsilon}{2} + \frac{2E_{R_{mmax,1}}}{\varepsilon} & \frac{\partial^2 R_{mmax}}{\partial w_2^2}\frac{\varepsilon}{2} + \frac{2E_{R_{mmax,2}}}{\varepsilon}\cdots & \frac{\partial^2 R_{mmax}}{\partial w_{mmax}^2}\frac{\varepsilon}{2} + \frac{2E_{R_{mmax,mmax}}}{\varepsilon} \end{pmatrix} \tag{4.8}$$

Above matrix can also be written as a summation of two matrices

$$[E_{total}] \approx [E_{secder}]\frac{\varepsilon}{2} + [E_{round}]\frac{2}{\varepsilon} \tag{4.9}$$

where

$$[E_{secder}] = \begin{pmatrix} \frac{\partial^2 R_1}{\partial w_1^2} & \frac{\partial^2 R_1}{\partial w_2^2} \cdots & \frac{\partial^2 R_1}{\partial w_{mmax}^2} \\ \frac{\partial^2 R_2}{\partial w_1^2} & \frac{\partial^2 R_2}{\partial w_1^2} \cdots & \frac{\partial^2 R_2}{\partial w_1^2} \\ \vdots & \vdots & \vdots \\ \frac{\partial^2 R_{mmax}}{\partial w_1^2} & \frac{\partial^2 R_{mmax}}{\partial w_2^2} \cdots & \frac{\partial^2 R_{mmax}}{\partial w_{mmax}^2} \end{pmatrix} \tag{4.10}$$

$$[E_{round}] \approx \begin{pmatrix} E_{R_{1,1}} & E_{R_{1,2}} \cdots & E_{R_{1,mmax}} \\ E_{R_{2,1}} & E_{R_{2,2}} \cdots & E_{R_{2,mmax}} \\ \vdots & \vdots & \vdots \\ E_{R_{mmax,1}} & E_{R_{mmax,2}} \cdots & E_{R_{mmax,mmax}} \end{pmatrix} \tag{4.11}$$

The optimum perturbation magnitude can be calculated by minimizing the Frobenius norm of the total error matrix with respect to perturbation magnitude, ε. In the Frobenius norm, error is represented as the summation of the square of the error at each entry of total error matrix. Since the residual vector at a given cell is only function of the flow variables of neighboring cells, most of the entries of numerical and analytical Jacobian matrices are zero. At these entries error becomes zero. Hence, the error matrices are also a sparse. The total error comes from the differences between the nonzero elements of analytical and numerical Jacobian matrices. Neglecting the zero entries of Jacobian matrices, the Frobenius norm of the total error matrix can be written in the following formulation. This formulation can also be achieved from the least square minimization of total error.

$$\left\| E_{total} \right\|_F = \sqrt{\sum_{m=1}^{mmax} \sum_{n=1}^{4(neigh+1)} \left(\frac{2E_{R_m}}{\varepsilon} + \frac{\partial^2 R_m(\zeta)}{\partial w_n^2} \frac{\varepsilon}{2} \right)^2} \tag{4.12}$$

In the Equation 4.12 the outer summation loop is constructed for the whole domain, excluding the ghost boundary cells where the residuals are not computed. The inner loop is constructed for the number of cells that are used in the evaluation of flux residuals. The number of neighboring cells used in the residual calculation is represented by *neigh*; it has a value of 4 in 1st order discretization and 8 in 2nd order discretization. The

minimization of total error is performed by differentiation of the Equation 4.12 with respect to perturbation magnitude.

$$\frac{\partial \left\| E_{total} \right\|_F}{\partial \varepsilon} = \frac{1}{2\left\| E_{total} \right\|_F} \sum_{m=1}^{mmax} \sum_{n=1}^{4(neigh+1)} \left\{ 2\left(\frac{2E_{R_m}}{\varepsilon} + \frac{\partial^2 R_m(\zeta)}{\partial w_n^2} \frac{\varepsilon}{2} \right) \left(-\frac{2E_{R_m}}{\varepsilon^2} + \frac{\partial^2 R_m(\zeta)}{\partial w_n^2} \frac{1}{2} \right) \right\}$$

$$\frac{\partial \left\| E_{total} \right\|_F}{\partial \varepsilon} = -\frac{2}{\varepsilon} \frac{1}{2\left\| E_{total} \right\|_F} \sum_{m=1}^{mmax} \sum_{n=1}^{4(neigh+1)} \left\{ \left(\frac{2E_{R_m}}{\varepsilon} + \frac{\partial^2 R_m(\zeta)}{\partial w_n^2} \frac{\varepsilon}{2} \right) \left(\frac{2E_{R_m}}{\varepsilon} - \frac{\partial^2 R_m(\zeta)}{\partial w_n^2} \frac{\varepsilon}{2} \right) \right\} \quad (4.13)$$

$$\frac{\partial \left\| E_{total} \right\|_F}{\partial \varepsilon} = -\frac{1}{\varepsilon \left\| E_{total} \right\|_F} \sum_{m=1}^{mmax} \sum_{n=1}^{4(neigh+1)} \left\{ \left(\frac{\partial^2 R_m(\zeta)}{\partial w_n^2} \right)^2 \frac{\varepsilon^2}{4} - \frac{4E_{R_m}^2}{\varepsilon^2} \right\}$$

The optimum ε which minimizes the total error is found by equating Equation 4.13 to zero;

$$\left. \frac{\partial \left\| E_{total} \right\|_F}{\partial \varepsilon} \right|_{\varepsilon=\varepsilon_{OPT}} = 0$$

$$\frac{\varepsilon^2}{4} \sum_{m=1}^{m\max} \sum_{nn=1}^{4(neigh+1)} \left(\frac{\partial^2 R_m(\zeta)}{\partial w_{nn}^2} \right)^2 - \frac{4}{\varepsilon^2} \sum_{m=1}^{m\max} \sum_{nn=1}^{4(neigh+1)} \left(E_{R_m} \right)^2 = 0 \quad (4.14)$$

The optimum perturbation magnitude can be evaluated as:

$$\varepsilon_{OPT} = 2 \sqrt{\frac{\sqrt{\sum_{m=1}^{m\max} \sum_{nn=1}^{4(neigh+1)} \left(E_{R_m} \right)^2}}{\sqrt{\sum_{m=1}^{m\max} \sum_{nn=1}^{4(neigh+1)} \left(\frac{\partial^2 R_m(\zeta)}{\partial w_{nn}^2} \right)^2}}} \quad (4.15)$$

Using the definition of the Frobenius norm, above equation can be simplified into the following form:

$$\varepsilon_{OPT} = 2 \sqrt{\frac{\left\| E_R \right\|_F}{\left\| \left(\frac{\partial^2 R(\zeta)}{\partial w^2} \right) \right\|_F}} \quad (4.16)$$

Substituting the optimum perturbation magnitude given by Equation 4.16 into the Equation 4.12 the magnitude of the minimized error is evaluated as follows:

$$E_{total}(\varepsilon_{OPT}) = \sqrt{\left\|E_R\right\|_F \left\|\left(\frac{\partial^2 R(\zeta)}{\partial w^2}\right)\right\|_F + 2\sum_{m=1}^{m\max}\sum_{nn=1}^{4(neigh+1)}\left(E_{R_m}\frac{\partial^2 R_m(\zeta)}{\partial w_{nn}^2}\right)} \tag{4.17}$$

The formulations presented by Equation 4.16 and 4.17 well define the way of accurate evaluation and the source of the errors in the numerical Jacobian. However those formulas are impractical to use for estimating the magnitude of the optimum perturbation since calculations of the round-off error and the second derivatives of the flux vectors are needed. The second derivatives can be approximated by the utilization of central differencing, which requires the proper choice for perturbation magnitude. The round off error in second derivatives can be estimated by proper scaling of the machine epsilon value. The multiplication of the machine epsilon value with the cells face size can give a reasonable estimate. The round-off error in single precision can also be estimated by differencing the residual vectors that are calculated with single and double precisions. Although those approximations can be used to evaluate the second derivatives and the round-off error, performing such sequence of calculations, just to evaluate the optimum perturbation value for numerical Jacobian derivation will be senseless due to its high computational cost. Further assumptions are needed on evaluation of those terms to obtain a straightforward formula for the estimation of optimum perturbation magnitude and the resulting error.

The round-off error in the numerical Jacobian evaluation occurs due to the loss of precision in computation of finite difference operations held on flux vectors. Hence the norm of the round off error in whole Jacobian matrix can be defined as the multiplication of the norm of flux vectors and the machine epsilon:

$$\left\|E_R\right\| = \left\|F\right\|\varepsilon_m \tag{4.18}$$

As another approximation, the order of norms of second derivatives, flux Jacobian and the flux vectors can be estimated to be equal to each other [66]

$$O\left(\left\|F\right\|\right) \cong O\left(\left\|\frac{\partial F}{\partial w}\right\|\right) \cong O\left(\left\|\frac{\partial^2 F}{\partial w^2}\right\|\right) \tag{4.19}$$

Introducing the approximations given by Equations 4.18 and 4.19 into the Equation 4.16 the optimum perturbation magnitude can be estimated as:

$$\varepsilon_{OPT} = 2\sqrt{\frac{\left\|E_R\right\|_F}{\left\|\left(\frac{\partial^2 R(\zeta)}{\partial w^2}\right)\right\|_F}} \cong 2\sqrt{\frac{\left\|F\right\|_F \varepsilon_m}{\left\|F\right\|_F}} = 2\sqrt{\varepsilon_m} \tag{4.20}$$

Following the same assumptions the magnitude of the minimum relative error resulted from usage of optimum perturbation magnitude can be estimated as:

$$\begin{aligned} E_{total}(\varepsilon_{OPT})\big|_{relative} &= \frac{\sqrt{\left\|E_R\right\|\left\|\left(\frac{\partial^2 R(\zeta)}{\partial w^2}\right)\right\| + 2\sum_{m=1}^{m\max}\sum_{nn=1}^{4(neigh+1)}\left(E_{R_m}\frac{\partial^2 R_m(\zeta)}{\partial w_{nn}^2}\right)}}{\left\|\left(\frac{\partial^2 R(\zeta)}{\partial w^2}\right)\right\|} \\ &\cong \frac{\sqrt{2\left\|F\right\|^2 \varepsilon_m + 2\sum_{m=1}^{m\max}\sum_{nn=1}^{4(neigh+1)}\left(E_{R_m}\frac{\partial^2 R_m(\zeta)}{\partial w_{nn}^2}\right)}}{\left\|F\right\|} \cong 2\sqrt{\varepsilon_m} \end{aligned} \tag{4.21}$$

The formulas given in Equations 4.20 and 4.21 present that, both of the optimum perturbation magnitude and the resulting minimum error can be estimated nearly as square root of the machine epsilon. These formulas propose that, the number of correct digits in numerically evaluated Jacobian element will approximately equal to the half number of correct digits in the flux vector if the optimum perturbation magnitude is used.

The formulas presented in Equations 4.20 and 4.21 are derived with some approximations and they do not present the exact values for optimum perturbation and the resulting minimum error. However they are simple compared to Equations 4.16 and 4.17. Machine epsilon can be determined easily by the simple formulation given in 4.6, hence the Equations 4.20 and 4.21 are handy to use.

The computational analyses are performed by using the Linux workstations of the METU Aerospace Engineering Department. The machine epsilon ε_M values of the compiler-computer configuration are calculated by Equation 4.6. Values are found as $\varepsilon_M \cong 1.19\times10^{-7}$ for single precision, and $\varepsilon_M \cong 2.2\times10^{-16}$ for double precision.

4.4 Accuracy Analysis

Numerical Jacobian matrices are calculated for a wide range of perturbation magnitudes and the deviation between the numerical and analytical Jacobian matrices is defined as error. Since the error itself is also a matrix, error in Jacobian is represented by matrix norm definitions. The L_1, L_∞ induced matrix norms and Frobenius entry-wise matrix norm are used through the study, and their definitions are given below:

$$\| A \|_1 = \max_{1\le j\le n} \sum_{i=1}^{m} | a_{ij} | \qquad \| A \|_\infty = \max_{1\le j\le m} \sum_{j=1}^{n} | a_{ij} | \qquad \| A \|_F = \left(\sum_{i=1}^{m} \sum_{j=1}^{n} | a_{ij} |^2 \right)^{1/2} \quad (4.22)$$

The change of the norm values of the error in Jacobian matrix with respect to perturbation magnitude is plotted in Figure 4.1 for the flow over Ni bump case. The given result corresponds to the solution of the transonic flow case, by 1st order AUSM scheme. The computation is performed with double precision. The optimum perturbation magnitude and the minimum relative error read from that figure are compared with the values proposed by Equations 4.20 and 4.21 in Table 4.1.

Table 4.1 *Estimated and evaluated optimum perturbation magnitudes*

	ε_{OPT} (Equation 5.15)	ε_{OPT} (Plotted)	minimum relative error (Equation 5.16)	minimum relative error (Plotted)
L_1	2.97 x 10^{-8}	2.51 x 10^{-8}	2.97 x 10^{-8}	2.10 x 10^{-8}
L_∞	2.97 x 10^{-8}	2.51 x 10^{-8}	2.97 x 10^{-8}	2.37 x 10^{-8}
L_F	2.97 x 10^{-8}	2.51 x 10^{-8}	2.97 x 10^{-8}	1.47 x 10^{-8}

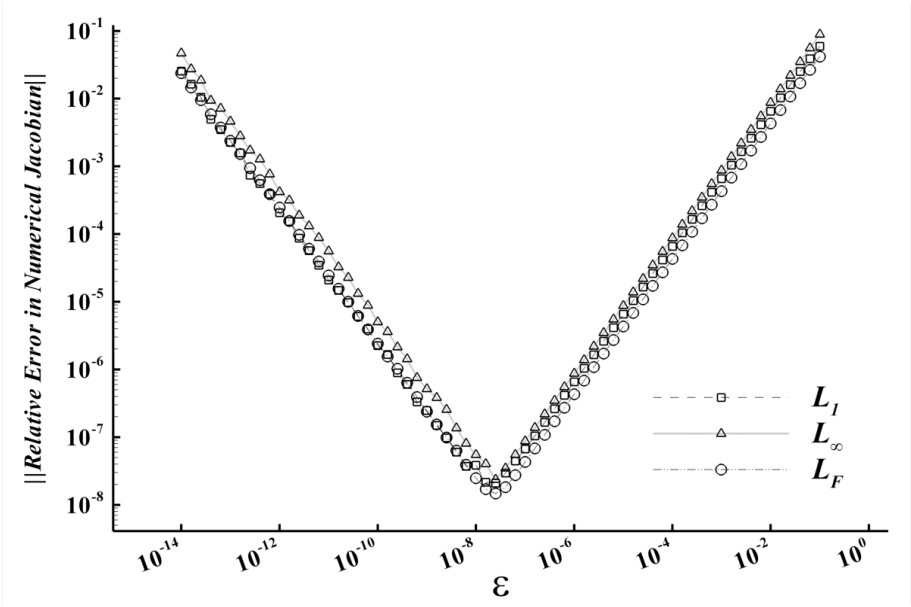

Figure 4.1 *Variation of the error in numerical Jacobian by finite difference perturbation magnitude*

As it can be seen from Table 4.1 the result of the estimation performed for optimum perturbation magnitude is very close to the actual optimum value obtained from trial error procedure. Moreover the estimated magnitude of the minimum relative error also agrees well with the actual relative error.

Table 4.2 *Relative error values read from Figure 4.1 for the smallest and the largest ε*

	$\|\|$ Relative Error $\|\|_1$	$\|\|$ Relative Error $\|\|_\infty$	$\|\|$ Relative Error $\|\|_F$
$\varepsilon = 10^{-14}$	0.0255	0.0237	0.0471
$\varepsilon = 10^{-1}$	0.0593	0.0418	0.0893

The result presented in Figure 4.1 can be also used to check the validity of the assumptions made to simplify the Equations 4.16 and 4.17. As it was stated earlier; for very small magnitudes of perturbation, the truncation error becomes negligible compared to the condition error and for very large magnitudes of perturbation, truncation error becomes dominantly large relative to negligible condition error. Hence in Figure 4.1 the relative error presented for the smallest and largest ε values correspond to condition and

truncation errors respectively. Those error values are tabulated in Table 4.2 and they can be used to detect the norm values of the round-off error and the second derivatives relative to norm value of the Jacobian matrix.

The absolute truncation error was defined in Equation 4.2. That formulation can be rewritten for the relative truncation error of the numerical Jacobian as:

$$\frac{\left\| E_{Trun}(\varepsilon) \right\|}{\left\| \frac{\partial R_m(\zeta)}{\partial \boldsymbol{w}_n} \right\|} = \frac{\left\| \frac{\partial^2 R_m(\zeta)}{\partial \boldsymbol{w}_n^2} \right\|}{\left\| \frac{\partial R_m(\zeta)}{\partial \boldsymbol{w}_n} \right\|} \frac{\varepsilon}{2} \tag{4.23}$$

Inserting the values given for $\varepsilon = 10^{-1}$ in the second row of the Table 4.2 to the left hand side of the above equation, the ratio of the norm value of the second derivatives to the norm value of the flux Jacobian is calculated as; 1.186 for L_1 norm, 0.836 for L_∞ norm and 0.942 for the Frobenius norm. Those calculated ratios shows that the assumption made for the order of the first and second derivatives in Equation 4.19 is reasonable enough.

The definition of the absolute condition error was given in Equation 4.5. That formulation can be rewritten for the relative condition error of the numerical Jacobian as:

$$\frac{\left\| E_C(\varepsilon) \right\|}{\left\| \frac{\partial R_m(\zeta)}{\partial \boldsymbol{w}_n} \right\|} = \frac{2E_R}{\varepsilon \left\| \frac{\partial R_m(\zeta)}{\partial \boldsymbol{w}_n} \right\|} \tag{4.24}$$

Inserting the error values given for $\varepsilon = 10^{-14}$ in the first row of the Table 4.2 to the left hand side of the above equation, the ratio of the round-off error to the norm value of the flux Jacobian elements is calculated as; 1.28×10^{-16} for L_1 norm, 1.19×10^{-16} for L_∞ norm, and 2.36×10^{-16} for the Frobenius norm. In previous section it was stated that the machine epsilon was evaluated as 2.2×10^{-16} for double precision computation. Those results present that assumptions made in Equation 4.18 for the definition of round off error performs very well with the assumption made for the relation in the order of norms of the flux vector, Jacobian elements and the second derivatives.

In previous paragraphs it is shown that assumptions made by Equation 4.19 for simplifying the Equations 4.16 and 4.17 are meaningful. As a last check of the validity of that assumption the second derivatives are calculated by finite differencing and the ratio of their norm values to norm values of the analytical Jacobian is analyzed. In order to find an appropriate finite difference step size for the second derivative evaluation, various perturbation magnitudes are tested. The results are presented in Figure 4.2 and they show that the norm values of second derivatives stayed almost constant in the perturbation magnitude interval of 10^{-8} to 10^{-3}. The estimation of the second derivatives can be done properly with the choice of perturbation magnitude from that interval. The ratio of the norm value of the approximated second derivatives to norm value of the Jacobian matrix elements is found to be close to unity. The ratio is calculated as 0.66 for L_1 norm, 0.69 for L_∞ norm, and 0.54 for the Frobenius norm. These results also verify that the assumption made on ratio of orders of second and first derivatives of the residual vector in Equation 4.19 perform well enough.

In the following sections the performances of the formulation given in Equation 4.19 for the estimation of the optimum perturbation magnitude will be examined for variety of flow solution cases with varying flow regimes, mesh and discretization scheme. The information presented in Equations 4.16 and 4.17 is used to analyze the cause of possible inconsistencies in the calculated and the estimated results.

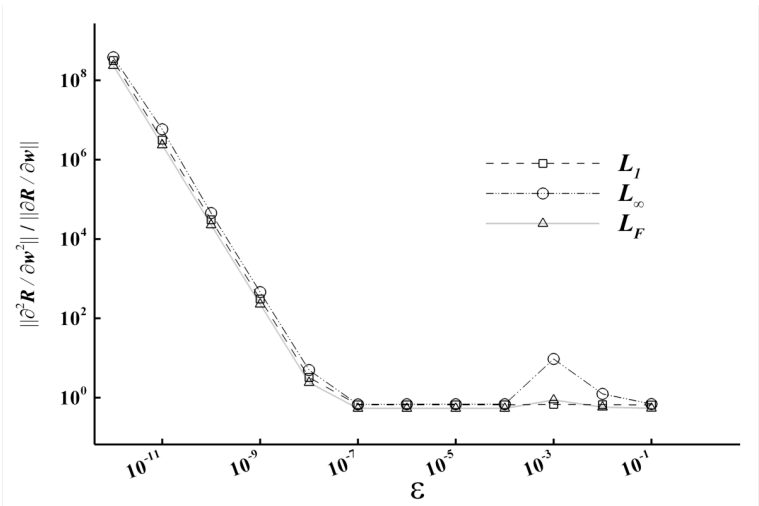

Figure 4.2 *Ratio of the norm of the numerically calculated second derivatives to norm of analytically evaluated Jacobian*

4.4.1 Effect of the Precision on the Numerical Jacobian Error

In this section the variation of the error in the numerical Jacobian with the computer precision is presented. Results are evaluated by using single and the double precision computations for the solution of flow over the Ni-bump geometry with the inlet velocity of 0.675 Mach. The computations are performed by 1st order AUSM scheme.

The change in the precision affects only the round-off error and has a negligible effect on the magnitude of the second derivatives. The round off error in the single precision computation approximately equals to the square of the round-off error in double precision computation due to the change in machine epsilon. The Figure 4.3 shows that with the change of the computation precision from double to the single, the optimum values become the square root of the values obtained for double precision. In single precision computation, the optimum value is read as $6x10^{-4}$ from Figure 4.3 for both of the L_1 and L_F norm definitions. That value agrees very well with the estimated $6.9x10^{-4}$ value by Equation 4.19. The magnitude of the minimum relative error is estimated as $6.9x10^{-4}$ by the Equation 4.20 and it is calculated as $3.2x10^{-4}$ for Frobenius norm and as $4.1x10^{-4}$ for L_1 matrix norm as presented in Figure 4.3.

The norm values of relative error calculated by the smallest perturbation magnitude for the single precision computation are specified in Table-4.3. For that perturbation magnitude the truncation error is small compared to the magnified condition error, hence the computed relative error can be defined as the condition error. Using the Equation 4.24 with the values presented in Table-4.3 the magnitude of the relative round-off error occurred in the single precision computations is calculated as $5x10^{-8}$ for L_1 norm and $2.5x10^{-8}$ for Frobenius norm. Comparing these results with the single precision machine epsilon, which equals to $1.19x10^{-7}$, it can be concluded that the assumptions made in Equation 4.18 for the definition of round-off error is good enough.

Table 4.3 *Calculated relative error for the smallest ε by single precision*

	$\|\|$ Relative Error $\|\|_1$	$\|\|$ Relative Error $\|\|_F$
$\varepsilon = 10^{-7}$	1.01	0.59

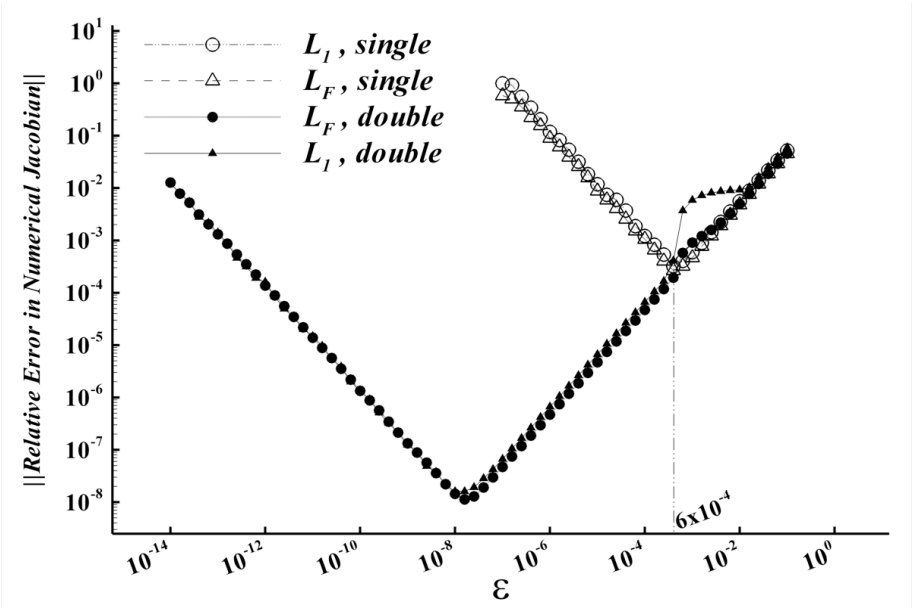

Figure 4.3 *Effect of the precision on the numerical Jacobian error*

4.4.2 Effect of the Norm Definition on the Optimum Perturbation Magnitude

In Equation 4.16, the norm values of round-off error and second derivatives are in division, hence the optimum perturbation magnitude does not vary with different norm definitions. However in Equation 4.17 the norms of round-off error and second derivatives are in multiplication. Hence the absolute error will vary with the norm definition. In previous section, the results were given for the relative error definition, and in this section, the variation of the absolute numerical Jacobian error for the same flow case is given in Figure 4.4 for different norm definitions.

The number of elements used in the L_1 matrix norm is small compared to the number of elements used in the Frobenius norm definition since whole matrix elements are used in the Frobenius norm definition. Therefore, in Figure 4.4 the Frobenius norm of the absolute error is larger compared to the L_1 norm of the absolute error. The optimum perturbation magnitude is same for both norm definitions whether the minimization is performed for the relative or the absolute error of the numerical Jacobian

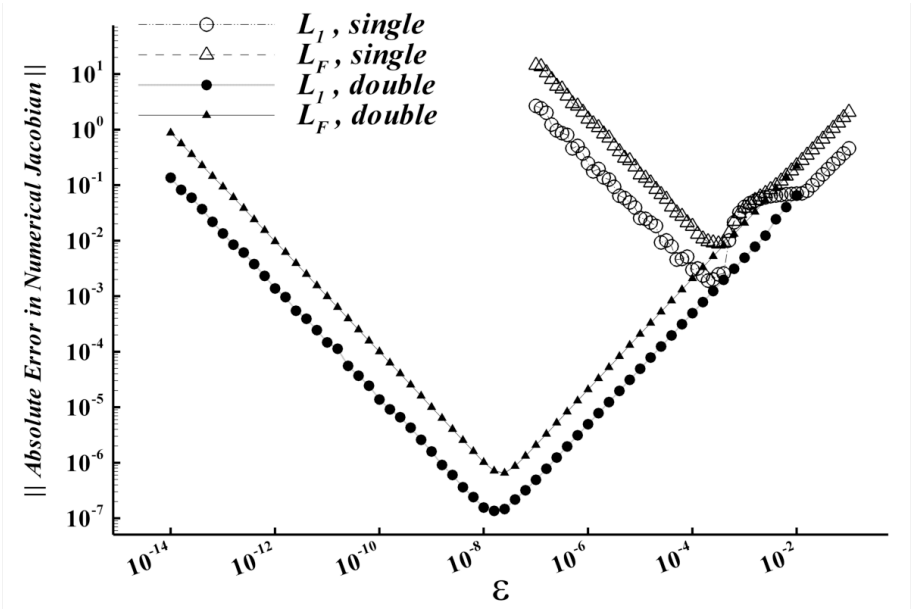

Figure 4.4 *Variation of absolute error in numerical Jacobian with perturbation magnitude*

4.4.3 Effect of the Size of the Grid Cells on the Numerical Jacobian Error

The geometries, grid topology and the boundary conditions used in the internal and external flow computations generally vary greatly. Therefore, to analyze the effect the size of the grid cells on the optimum perturbation magnitude, the accuracy of numerical Jacobian is studied in both internal and external flow solutions. In both of the flow solutions over bump and airfoil geometries, approximately similar transonic flow conditions are used. The main difference between the two cases was the sizes of the cell faces of the mesh.

For internal flow problem, the flow over Ni-bump is analyzed on an H-type grid, whose inlet and outlet are located 1.75 chord length away from the leading and trailing edges. For external flow problem, a C-type grid, in which distance between far-field boundary and the airfoil geometry varies 20 to 60 chord lengths, is used. To be able to resolve the gradients, nodes are clustered near to airfoil geometry and were expanded at the far-field. The size of the cell faces of the grid used in flow over airfoil problem is significantly larger than the ones used in H-type grid, except at the regions near to wall geometries. At far field of the C-type grid, cell faces are approximately larger by two orders of

magnitude than the ones used at the inlet and the outlet of the H-type grid. Due to the larger cell faces it will be reasonable to expect larger flux values in the solution performed for airfoil case compared to solution performed for bump geometry case. The enlargement in the flux magnitudes may amplify the round-off error and will cause the second and first derivatives of the flux vectors to grow. In Figure 4.5 the variation of the absolute error in the numerical Jacobian for the flow over airfoil and flow over bump computations is presented. In the computations of the both cases 1st order AUSM discretization scheme is used.

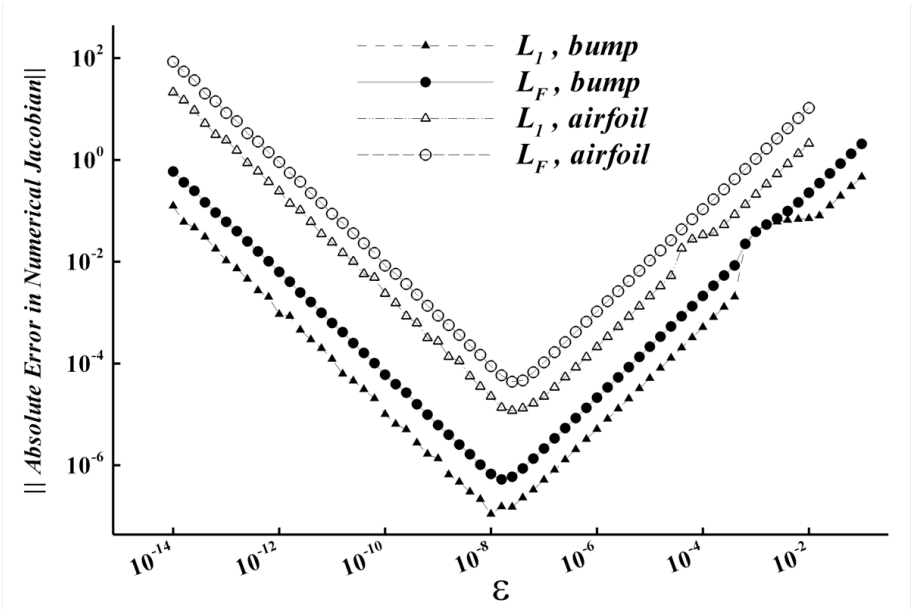

Figure 4.5 *Variation of the absolute numerical Jacobian error for flow solutions with different geometries and grids*

In Figure 4.5 the absolute error is shown on the vertical axis and the error plots are drawn for both L_1 and Frobenius norms. In both of the flow solutions on H-type and C-type grids, the optimum perturbation magnitudes are found to be same and agree well with the proposed value by Equation 4.19. One of the important conclusions from this figure is that, although the grid type and size almost have no effect on the magnitude of optimum perturbation, they significantly affect the magnitude of the absolute error. The absolute error in the airfoil case is approximately larger by two order of magnitudes compared to the bump case. The cause of this difference can be extracted from the

Equations 4.16 and 4.17. In formulation which gives optimum perturbation magnitude, the round off error and second derivatives are in division whereas in the formulation which gives the total error, those two terms are in multiplication. An increase in the total error without a change in the optimum perturbation magnitude can only be possible with the proportional enlargement in both of round-off error and the second derivatives.

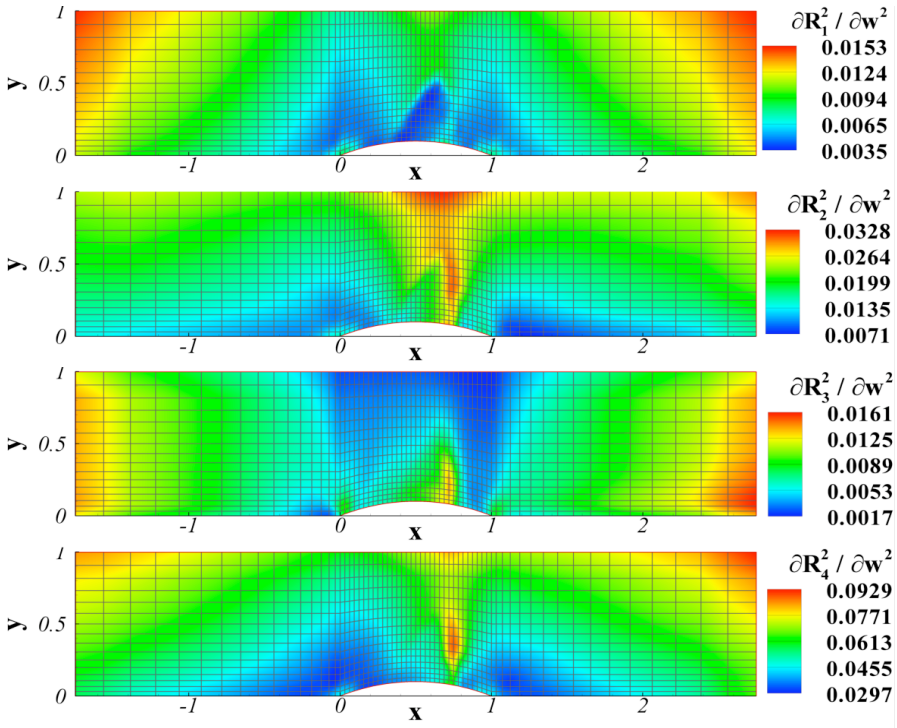

Figure 4.6 *Variation of second derivatives on grid used for Ni bump geometry*

The second derivatives of the residual at a given cell depends on the flow variable at that cell plus the 4 or 8 neighboring cells according to the order of discretization. The contour plots of the averaged second derivatives of the residual vector are given in Figure 4.6 and 4.7. Those figures show that the second derivatives of the residual vector get larger as the size of the cell face gets larger. The comparison of the values of second derivatives for those two cases shows that in the regions closer to the wall geometries the size of the cells are approximately equal and the second derivatives are in the same order of magnitudes. However in the far-field regions of the C-grid the cells are approximately larger by two orders of magnitude compared to the grid used on bump geometry and the

resulting second derivatives of the residual vector are also larger by approximately 2 orders of magnitude.

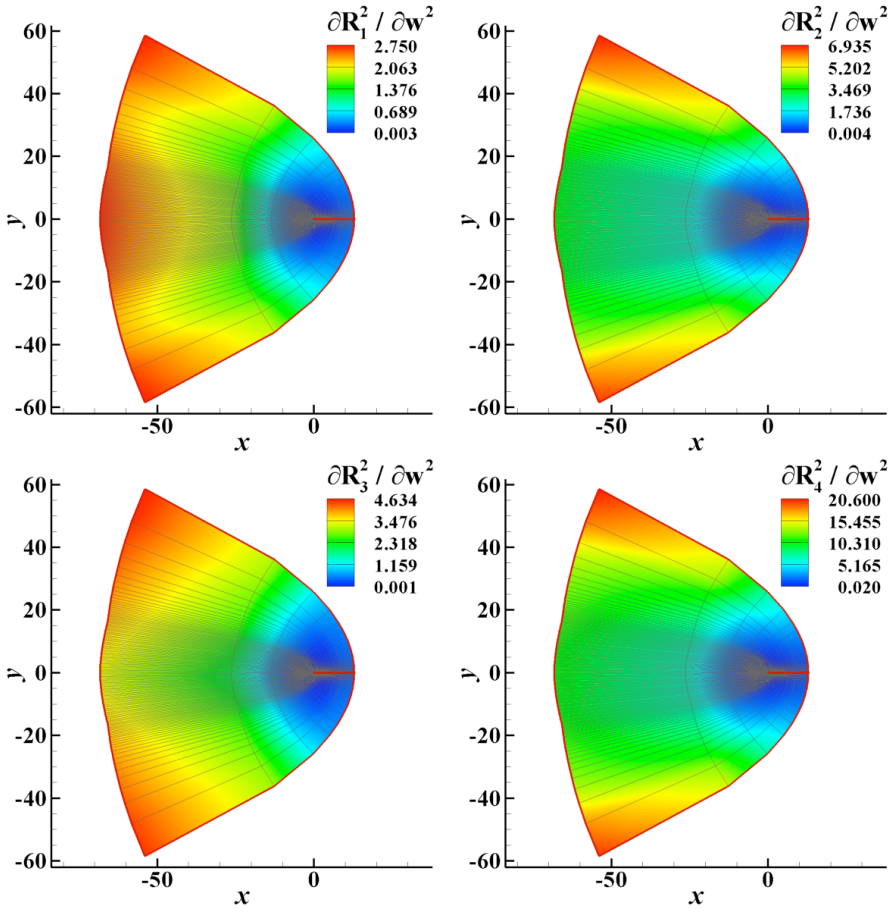

Figure 4.7 *Variation of second derivatives of residual vector on grid used for NACA0012 airfoil geometry*

The effects of the cell size on the second derivatives and on the total absolute error in numerical Jacobian are also shown by two other alternative ways. Firstly, comparison of the absolute error resulted from flow solution over bump geometry and the flow over airfoil geometry is performed by only using the small sized cells of C-grid that are close to the airfoil. Secondly, the same flow over airfoil problem is re-solved with a C-grid whose farfield is closer to the airfoil geometry with the same number of nodes. The grids used in those two alternative approaches are given in Figures 4.8 and 4.9.

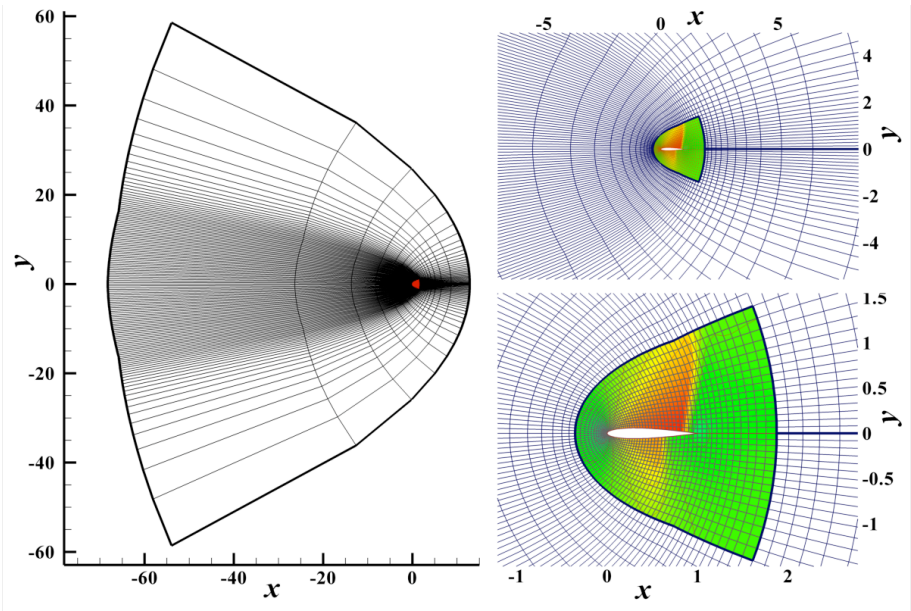

Figure 4.8 *Alternative 1: The small sized cells closer to the airfoil geometry*

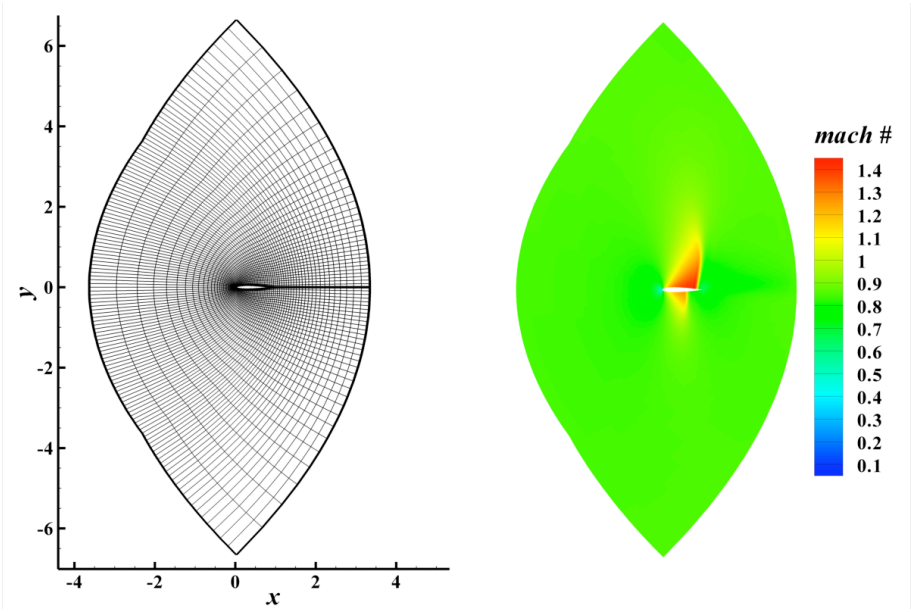

Figure 4.9 *Alternative 2: The c-type grid with closer far-field*

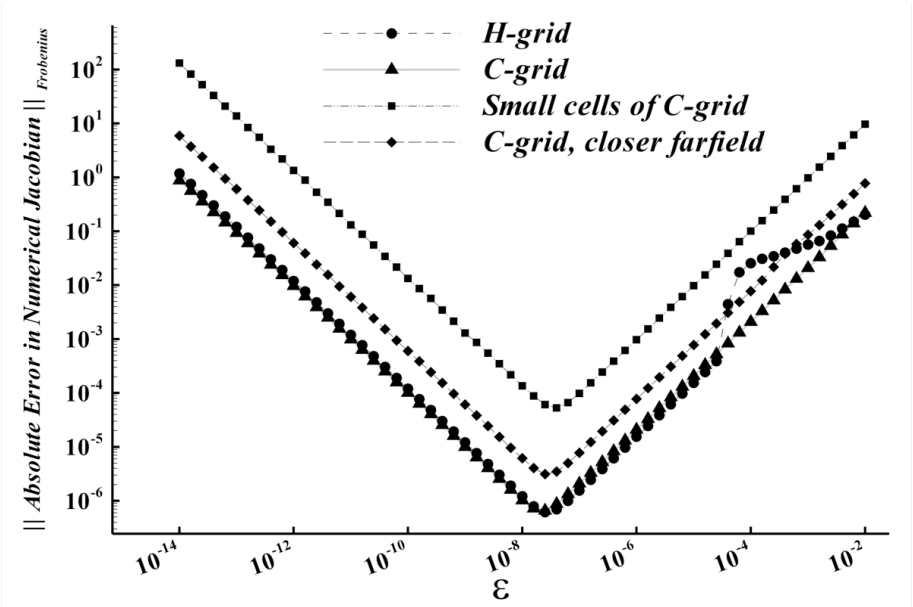

Figure 4.10 *Variation of the absolute error in numerical Jacobian by the size of the cell faces*

The absolute error plots of the numerical Jacobian for those two alternative cases are given in Figure 4.10 with the error plot of the H-grid case. The small region chosen from the C-grid has approximately equal sized cells compared to the grid used for bump geometry. Hence the absolute numerical Jacobian error computed for that region is found to be close to the absolute error calculated for the H-grid case. The results also show that by using a grid with closer farfield boundaries, uniformly sized grid cells can be constructed and the absolute error may become smaller due to the decrease in the size of the cell faces.

Since there is no practical way of round-off error representation for double precision computations, the variation of round-off error according to the size of the cell face is checked for single precision computation. To estimate the off error in single precision; the flux residuals are computed both by single and double precision. The difference of the residuals is defined as the round off error. Although the absolute error plots given in Figures 4.5 and 4.10 were presented for double precision, the single precision computation can also give satisfactory clues for the variation of round-off errors by grid type. The round-off errors calculated by single precision are tabulated in Table 4.4 for both the H-type and C-type grids with different norm definitions.

Table 4.4 *Single precision round-off error*

	$\sum_{i=2}^{imax-1}\sum_{j=2}^{jmax-1}\left\|R_{i,j}^{double}-R_{i,j}^{single}\right\|$	$\sqrt{\sum_{i=2}^{imax-1}\sum_{j=2}^{jmax-1}\left(R_{i,j}^{double}-R_{i,j}^{single}\right)^2}$
65x17 H-grid	$7.41\text{x}10^{-9}$	$1.3\text{x}10^{-8}$
129x33 C-grid	$1.4\text{x}10^{-7}$	$4\text{x}10^{-6}$

If the assumptions made in Equation 4.18 were valid the resulting relative errors in both bump and airfoil case would be approximately the same. Therefore to examine the validity of that assumption the variation of the relative error in numerical Jacobian is plotted in Figure 4.11 for the same case analyzed in Figure 4.5 for absolute error variation.

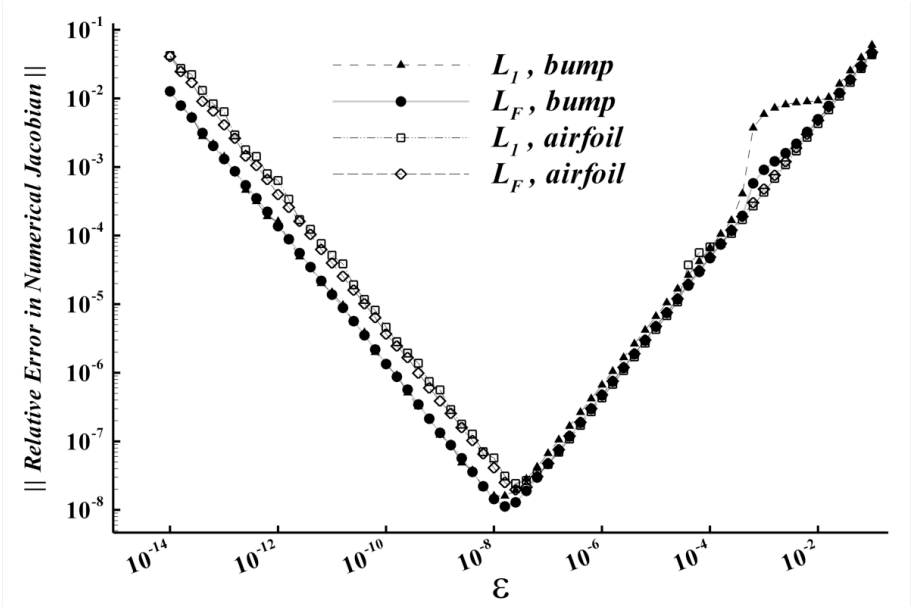

Figure 4.11 *Variation of the relative numerical error in Jacobian for the flow solutions with different geometries and grids*

Figure 4.11 shows that the relative error in the numerical Jacobian is approximately same for the flow solutions performed with bump and airfoil geometries except the small discrepancy occurred in the usage of small perturbation magnitudes where the condition

error is dominant. Comparing the results presented in Figure 4.11 with the estimated values, it can be said that the approximation formulas given in Equations 4.20 and 4.21 produce very accurate results for the optimum perturbation magnitude and the resulting minimum relative error is also guessed successfully for each type of grid and the geometry.

4.4.4 Effect of the Flux Scheme on the Numerical Jacobian Error

The variation of the magnitude of the second derivatives of flux vector and the round-off error with different flux evaluation method is analyzed to get the dependency of the optimum perturbation magnitude on the flux evaluation method. The formulation for the second derivatives of the flux vector that is obtained by differentiating the Equation 2.46 is given below:

$$\begin{aligned}\frac{\partial^2 R_{i,j}}{\partial \boldsymbol{w}_{k,l}^2} = &\frac{\partial^2 F^+_{i+1/2,j}}{\partial\left(W^L_{i+1/2,j}\right)^2}\frac{\partial \boldsymbol{w}^L_{i+1/2,j}}{\partial \boldsymbol{w}_{k,l}} + A^+_{i+1/2,j}\frac{\partial^2 \boldsymbol{w}^L_{i+1/2,j}}{\partial \boldsymbol{w}^2_{k,l}} + \frac{\partial^2 F^-_{i+1/2,j}}{\partial\left(\boldsymbol{w}^R_{i+1/2,j}\right)^2}\frac{\partial \boldsymbol{w}^R_{i+1/2,j}}{\partial \boldsymbol{w}_{k,l}} + A^-_{i+1/2,j}\frac{\partial^2 \boldsymbol{w}^R_{i+1/2,j}}{\partial \boldsymbol{w}^2_{k,l}} \\ &- \frac{\partial^2 F^+_{i-1/2,j}}{\partial\left(W^L_{i-1/2,j}\right)^2}\frac{\partial \boldsymbol{w}^L_{i-1/2,j}}{\partial \boldsymbol{w}_{k,l}} - A^+_{i-1/2,j}\frac{\partial^2 \boldsymbol{w}^L_{i-1/2,j}}{\partial \boldsymbol{w}^2_{k,l}} - \frac{\partial^2 F^-_{i-1/2,j}}{\partial\left(\boldsymbol{w}^R_{i-1/2,j}\right)^2}\frac{\partial \boldsymbol{w}^R_{i-1/2,j}}{\partial \boldsymbol{w}_{k,l}} - A^+_{i-1/2,j}\frac{\partial^2 \boldsymbol{w}^R_{i-1/2,j}}{\partial \boldsymbol{w}^2_{k,l}} \\ &+ \frac{\partial^2 G^+_{i,j+1/2}}{\partial\left(W^L_{i,j+1/2}\right)^2}\frac{\partial \boldsymbol{w}^L_{i,j+1/2}}{\partial \boldsymbol{w}_{k,l}} + B^+_{i,j+1/2}\frac{\partial^2 \boldsymbol{w}^L_{i,j+1/2}}{\partial \boldsymbol{w}^2_{k,l}} + \frac{\partial^2 G^-_{i,j+1/2}}{\partial\left(\boldsymbol{w}^R_{i,j+1/2}\right)^2}\frac{\partial \boldsymbol{w}^R_{i,j+1/2}}{\partial \boldsymbol{w}_{k,l}} + B^-_{i,j+1/2}\frac{\partial^2 \boldsymbol{w}^R_{i,j+1/2}}{\partial \boldsymbol{w}^2_{k,l}} \\ &- \frac{\partial^2 G^+_{i,j-1/2}}{\partial\left(\boldsymbol{w}^L_{i,j-1/2}\right)^2}\frac{\partial \boldsymbol{w}^L_{i,j-1/2}}{\partial \boldsymbol{w}_{k,l}} - B^+_{i,j-1/2}\frac{\partial^2 \boldsymbol{w}^L_{i,j-1/2}}{\partial \boldsymbol{w}^2_{k,l}} - \frac{\partial^2 G^-_{i,j-1/2}}{\partial\left(\boldsymbol{w}^R_{i,j-1/2}\right)^2}\frac{\partial \boldsymbol{w}^R_{i,j-1/2}}{\partial \boldsymbol{w}_{k,l}} - B^-_{i,j-1/2}\frac{\partial^2 \boldsymbol{w}^R_{i,j-1/2}}{\partial \boldsymbol{w}^2_{k,l}}\end{aligned} \tag{4.25}$$

In first order accurate spatial discretization, the second derivatives of interpolated face variables with respect to the cell center flow variables equal to zero. Hence the magnitudes of the second derivatives of the fluxes depend only on the second derivatives of the splitted fluxes. Applying the first order spatial discretization, the variation of the second derivatives by the fluxes evaluation method is investigated. Second derivatives are approximated by the central differencing and for each flux evaluation method the magnitude of them are found to be approximately equal. The L_2 norm values of the second derivatives of the residual vectors are given in Table-2 for the transonic flow over bump case. The magnitudes of the flow variables and the fluxes will not change

significantly as long as the solved flow problem or the used grid does not change. Hence the amount of round-off error will also be identical in each flux calculation methods

Table 4.5 *Second derivatives of the residual, calculated for the transonic flow over the bump*

	$\left\| \frac{\partial^2 R(w)}{\partial w^2} \right\|_{Frobenius}$
1st order Steger Warming	23.859
1st order Van Leer	25.186
1st order AUSM	24.405

The effect of flux evaluation methods on the relative numerical Jacobian error is presented in Figure 4.12 for the first order accurate spatial discretization. The figure shows that the magnitude of the error and the optimum perturbation magnitude do not change with 1st order Steger-Warming, Van-Leer and AUSM flux evaluation methods.

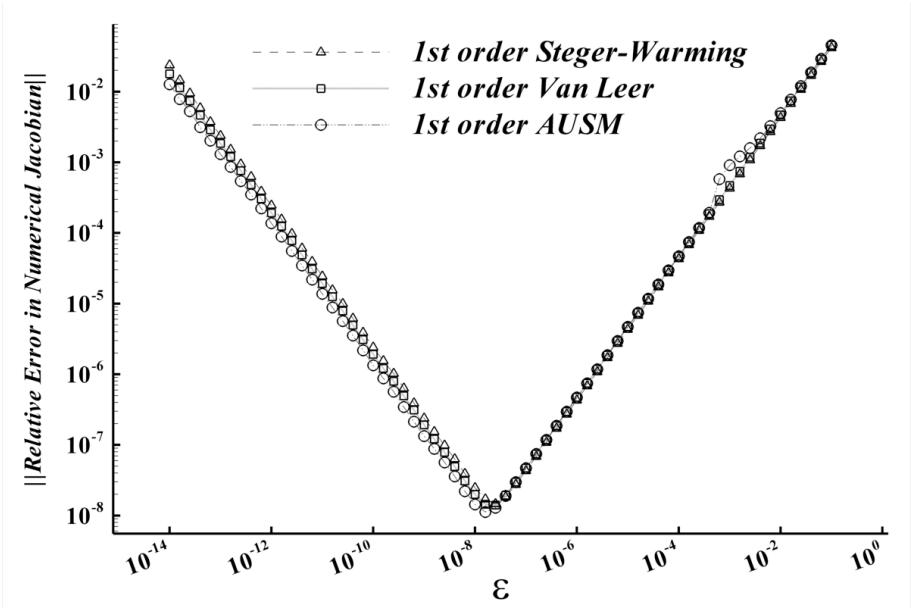

Figure 4.12 *Effect of the flux evaluation method on the relative numerical Jacobian error*

4.4.5 Effect of Order of Discretization on Numerical Jacobian Error

In second order accurate spatial discretization, the second derivatives of interpolated face variables with respect to the flow variables are nonzero. Hence these terms will affect the second derivatives of the residual vector. The formulation of the interpolated face variables was given in Equation 2.36. Differentiating this equation with respect to the flow variables twice gives:

$$\frac{\partial^2 \boldsymbol{w}^L{}_{i\mp 1/2}}{\partial \boldsymbol{w}^2_{k,l}} = \frac{\partial^2 \delta^L_{i\mp 1/2}}{\partial \boldsymbol{w}^2_{k,l}}, \quad \frac{\partial^2 \boldsymbol{w}^R{}_{i\mp 1/2}}{\partial \boldsymbol{w}^2_{k,l}} = -\frac{\partial^2 \delta^R_{i\mp 1/2}}{\partial \boldsymbol{w}^2_{k,l}} \tag{4.26}$$

The second derivatives of the interpolated face variables directly depend on the second derivatives of the limiter functions. In smooth flow regions, difference of the flow variables at the neighboring cells is nearly zero. Therefore backward and forward differencing of flow variables (a and b in Equation 2.39) can be reasonably assumed to be equal to each other. The formulation for the second derivatives of the limiter function with this assumption will become very simple and it is given in Equation 4.27.

$$\left.\frac{\partial^2 \delta}{\partial \boldsymbol{w}^2}\right|_{a=b} = \frac{a}{a+\frac{\in}{a}} \tag{4.27}$$

Limiters are known to stall the convergence of an iterative scheme, because of accidental switching in smooth flow regions [59], [65], [67]. The remedy proposed by most of the researchers dealing with this issue is the addition of a small number, $\in$ to the numerator and denominator of the interpolation formulation to control the activation of the limiter. There are different studies performed to define the proper values of $\in$. [68], [69]. The formulation given in Equation 4.27 presents that in smooth flow regions, the second derivatives of the limiter functions will be large if the value of $\in$ is small. The effects of the magnitude of $\in$ on the accuracy of numerical Jacobians are studied. In Table 4.6, the change of Frobenius norm of the second derivatives of the residual vector with the variation of the size of $\in$ is presented. The results given in Table 4.6 are generated by the transonic flow solution over the bump geometry using 65x17 grid and 2nd order AUSM scheme with Van Albada Limiter. Consistent to Equations 4.25, 4.26 and 4.27, the second derivatives of residual vector get larger as the size of the $\in$ gets smaller. The

variation of the Frobenius norm of the relative error in numerical Jacobian matrix with respect to the size of $\in$ is given in Figure 4.13 and 4.14 for the Koren and Van Albada limiters respectively.

Table 4.6 *Variation of the second derivatives of residuals by the $\in$ value used in limiters*

	$\left\|\frac{\partial^2 R(w)}{\partial w^2}\right\|_F$ (Koren Limiter)	$\left\|\frac{\partial^2 R(w)}{\partial w^2}\right\|_F$ (Van Albada Limiter)
$\in=\sqrt{0.5(\Omega_i+\Omega_{i+1})}$	21.03	24.11
$\in = 10^{-3}$	209.7	156.9
$\in = 10^{-6}$	10211.7	7705.6
$\in = 10^{-9}$	162762.3	124771.2
$\in = 10^{-12}$	365644.2	355329.1

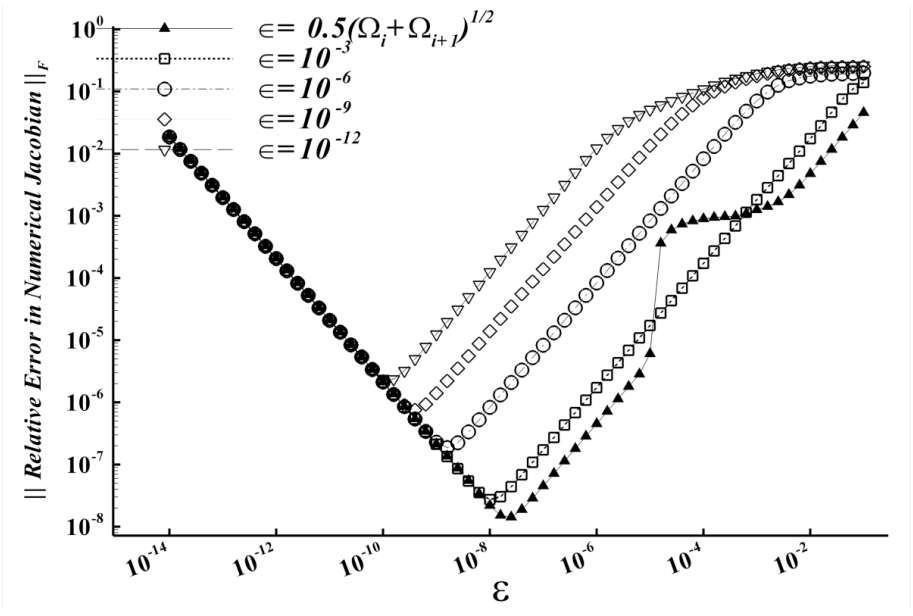

Figure 4.13 *Variation of the numerical Jacobian error for Koren limiter*

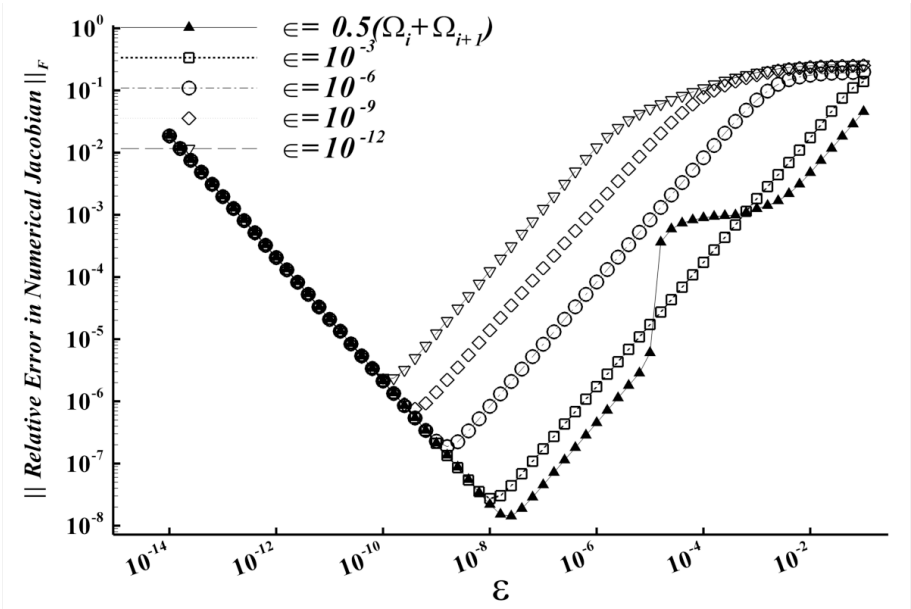

Figure 4.14 *Variation of the numerical Jacobian error for Van Albada limiter*

In second order discretization, the size of the second derivatives of the residual vector is dominated by the second derivatives of the limiter functions for small $\in$ values. However the value of $\in$ has no affect on the round-off error. Therefore, as the values of $\in$ become smaller the magnitude of second derivatives increases while the round-off error stays constant. As a result, according to Equation 4.17 the usage of small values of $\in$ will result in magnified total error and smaller optimum perturbation magnitude. The results show that limiter functions calculated with small $\in$ values significantly increased the magnitudes of the second derivative of residual vector and the validity of the assumption made in Equation 4.18 is lost. Hence the optimum perturbation magnitude deviates from the one proposed by Equation 4.20 significantly.

The usage of very small $\in$ values is shown to be prohibitive due to the amplifying effect on second derivatives. However the selection of large values for the $\in$ is also not reasonable. In the case of using very large values for the $\in$, the second derivatives of interpolated face variables will become negligibly smaller compared to the second derivatives of splitted flux vectors. Hence, the size of the second derivatives of the residual vector will be significantly dominated by the second derivatives of the splitted

flux vectors. The further enlargement in $\in$ will not affect the magnitude of the second derivatives of the residual vector nor the accuracy of the Jacobian. Moreover, the usage of very large $\in$ values in limiter function will be improper since the definition of the interpolation with limiter function will be degraded. As a summary, the accuracy of the second order numerical Jacobian is bounded by that of the first order numerical Jacobian. In this study the most accurate evaluation for the second order numerical Jacobian is achieved with the usage of the $\in$ values as proposed in Equation 4.28. $\in$ value that will be used in the face variable interpolation is defined as the square root of the average of the cell areas of the neighboring cells located at upstream and downstream of the face.

$$\in = \sqrt{\frac{1}{2}\left(\Omega_i + \Omega_{i+1}\right)} \tag{4.28}$$

4.4.6 Variation of Error with Resolution of Grid

Completing the search for the optimum perturbation magnitude for different flux evaluation methods and for different flow problems with different grid types, the variation of the error with grid resolution is studied. The analysis is performed both for the H-type grid and C-type grid cases. The results are presented in Figures 4.15 and 4.16. Calculations are performed using AUSM scheme for first and second order spatial discretizations. In second order discretization, limiter function of the Van Albada is used with $\in$ value defined in Equation 4.28. For the bump geometry an inlet Mach number of 0.675 and for the airfoil geometry the free stream Mach number of 0.85 and 1 degree of angle of attack is used. Like in the previous cases, for different grid resolutions, the optimum perturbation locations for bump and airfoil geometries coincide with each others, and they are in good agreement with value proposed by Equation 4.20. To make the grid finer or coarser changes the magnitudes of the flux variables slightly but has no significant effect on the ratios of magnitudes of the round-off error and the second derivatives to magnitude of fluxes. As a result the optimum perturbation magnitude minimizing the total error is identical for all coarse and fine grids. The minimum relative error is approximately equal for all cases and in good agreement with Equation 4.21.

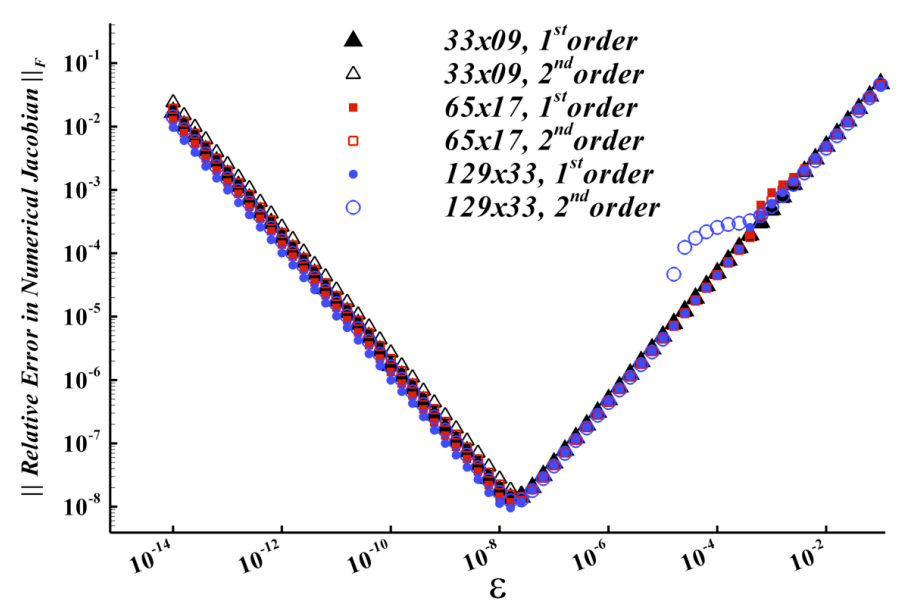

Figure 4.15 *Variation of the relative error with varying grid resolution for the solution of the flow over Ni bump*

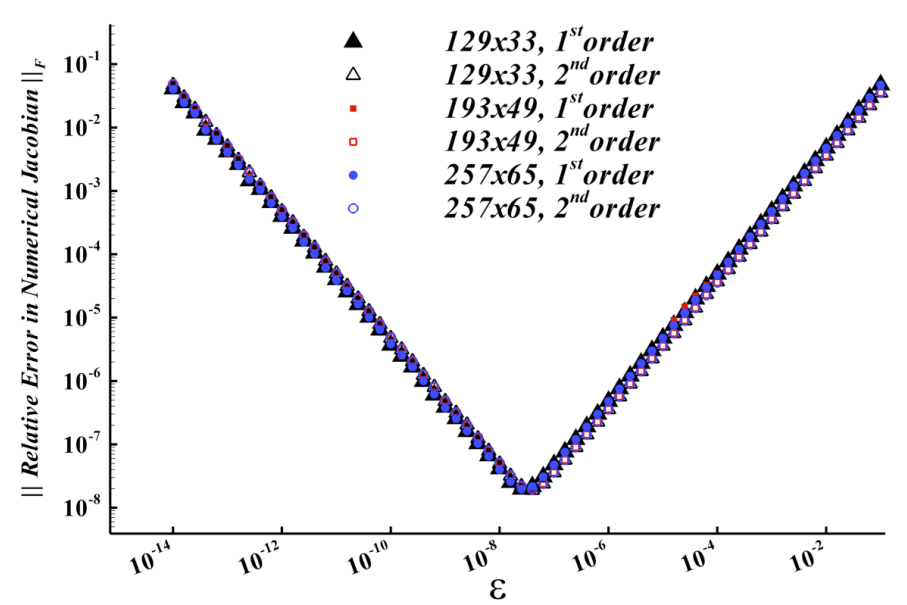

Figure 4.16 *Variation of the relative error with varying grid resolution for the solution of the flow over NACA0012 airfoil*

4.4.7 Variation of Error with Flow Regime

As the last factor affecting the accuracy, the effect of the free-stream flow condition is analyzed.

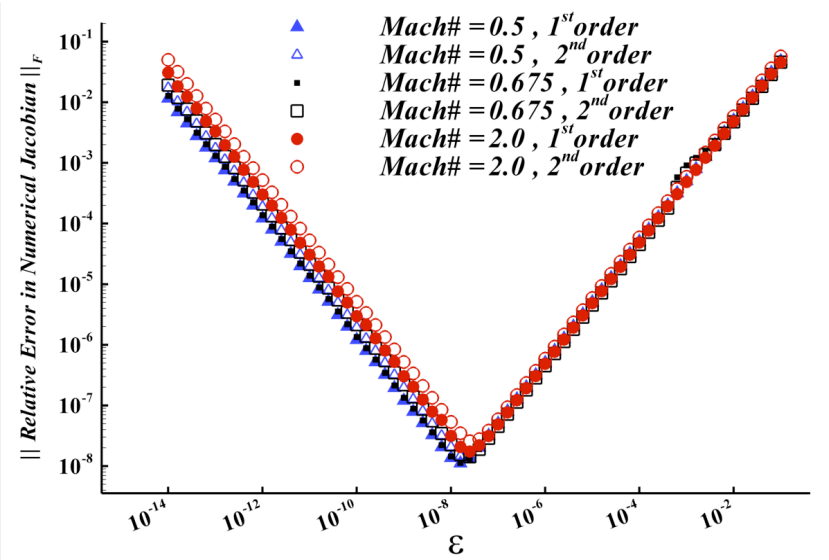

Figure 4.17 *Variation of the relative error with varying inlet Mach # (Ni bump)*

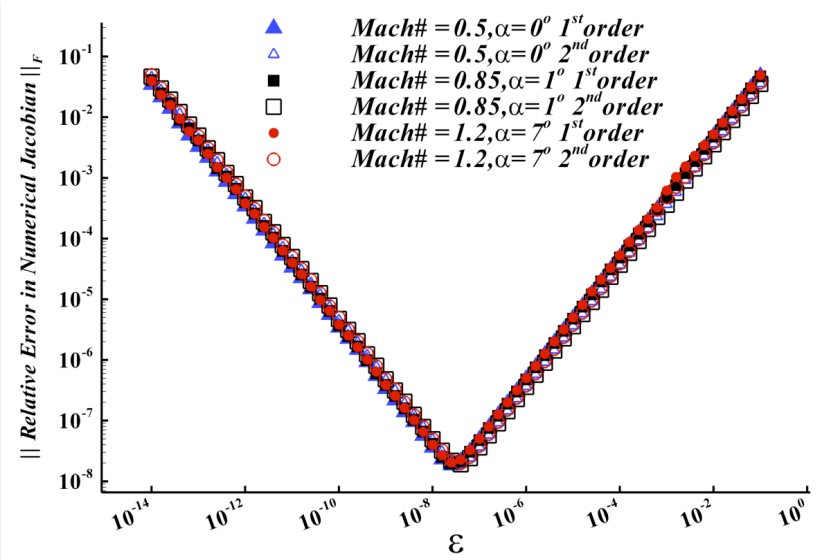

Figure 4.18 *Variation of the relative error with varying free-stream Mach# (airfoil)*

For the Ni-bump geometry the inlet flow Mach numbers of 0.5, 0.675 and 2.0 are tested. For the NACA0012 airfoil geometry, free stream flows of, 0.3 Mach, 0.85 Mach with 1 degree of angle of attack, and 1.2 Mach with 7 degrees of angle of attack are analyzed. Figure 4.16 shows the change of relative error with perturbation magnitude for three different flow conditions over Ni bump geometry. Similar results are shown in Figure 4.17 for three different flow conditions analyzed for NACA0012 airfoil. The presented results are calculated by AUSM scheme with first and second order spatial discretizations. Results show that the optimum perturbation magnitude is not sensible to the free stream or inlet conditions of the flow problem. The optimum magnitude did not varied significantly for the cases with varying flow regimes and it is in well agreement with the magnitude proposed by Equation 4.20.

4.5 Performance of the Numerical Jacobian on Flow Solution

In previous sections, different factors which may affect the accuracy of the numerical Jacobian are analyzed in detail. All the results show that usage of the finite difference perturbation magnitude prescribed by Equation 4.20 is satisfactory enough to construct accurate numerical Jacobian matrices compared to analytical ones. The Equation 4.20 was the simplification of the Equation 4.16 by assuming that norms of second derivatives, Jacobian matrix and flux vectors equal to each others. Magnitude of the round off error was assumed as the machine epsilon times the norm of the flux vector. Those assumptions performed well for all the cases tried except the cases where the proper tuning of the interpolation formulation is not done in second order accurate spatial discretization. It is realized that the proper scaling of the small number, which is added to face interpolation formulation in order to inactivate the limiter in smooth flow domain, is needed. The addition of very small values of $\in$, resulted in amplified magnitude of second derivatives relative to norm of flux vector. The scaling of $\in$ by the neighboring cell areas produced satisfactory results and the retained the reasonability of the assumptions made in derivation of simple formula of optimum perturbation magnitude. The proposed optimum perturbation magnitude is $3x10^{-8}$ for double precision computations and $6.9x10^{-4}$ for single precision computations. Analyzing the ways of

accurate computation of numerical Jacobian, the effect of the accuracy of the Jacobian on the convergence of the flow solution is studied.

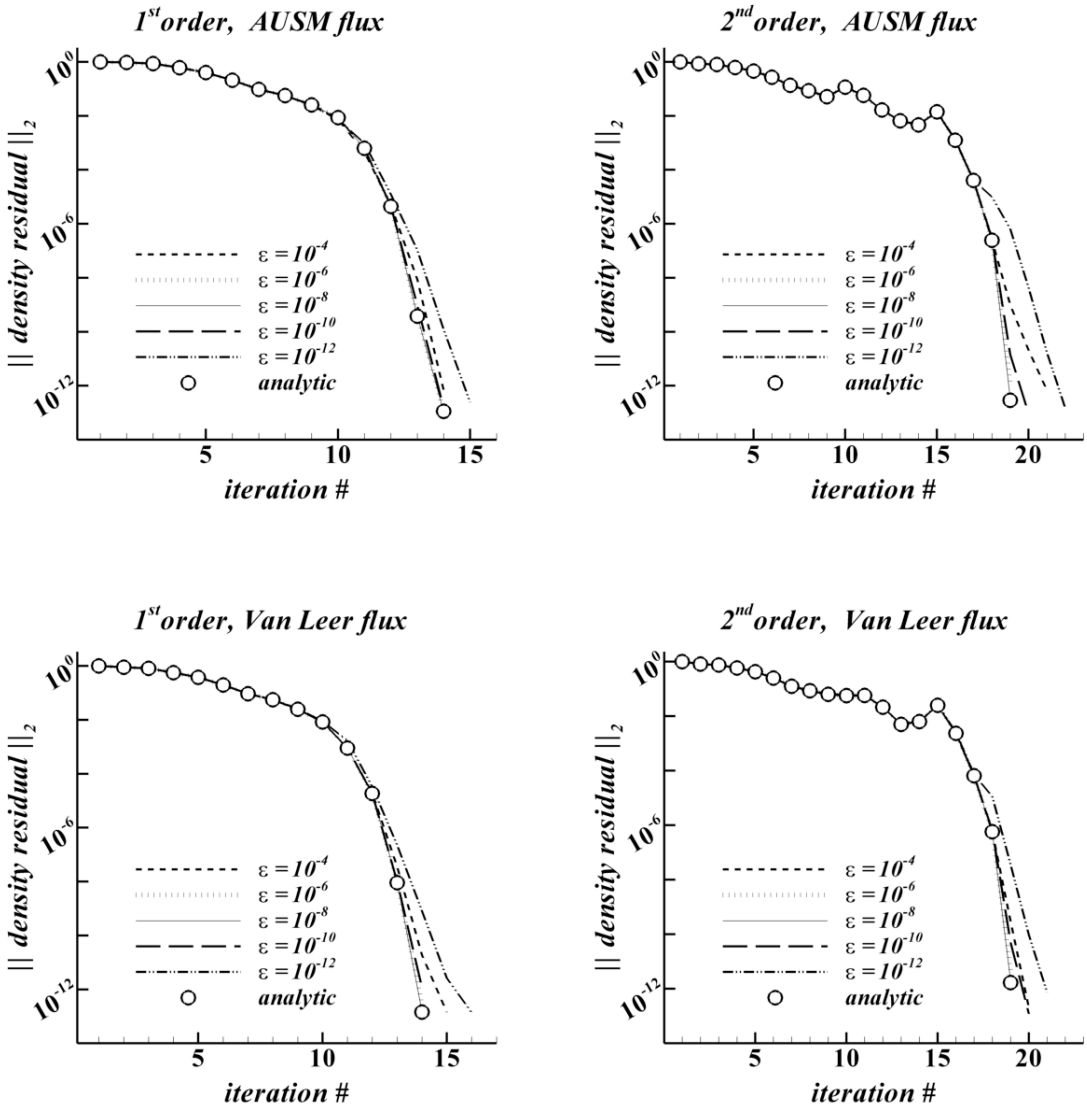

Figure 4.19 *Residual histories for AUSM and Van Leer schemes in the solution of flow over Ni-bump. M=0.675*

The convergence performance of the flow solver using the analytically evaluated Jacobian was given in Chapter 3. The convergence performance obtained by the utilization of the numerical Jacobian, which is evaluated by optimum perturbation magnitude, is compared with those results given for analytical Jacobian. Similar to the

strategy followed in accuracy analysis change of convergence performance by numerical Jacobian matrices that are derived by the variety of the perturbation magnitude is presented.

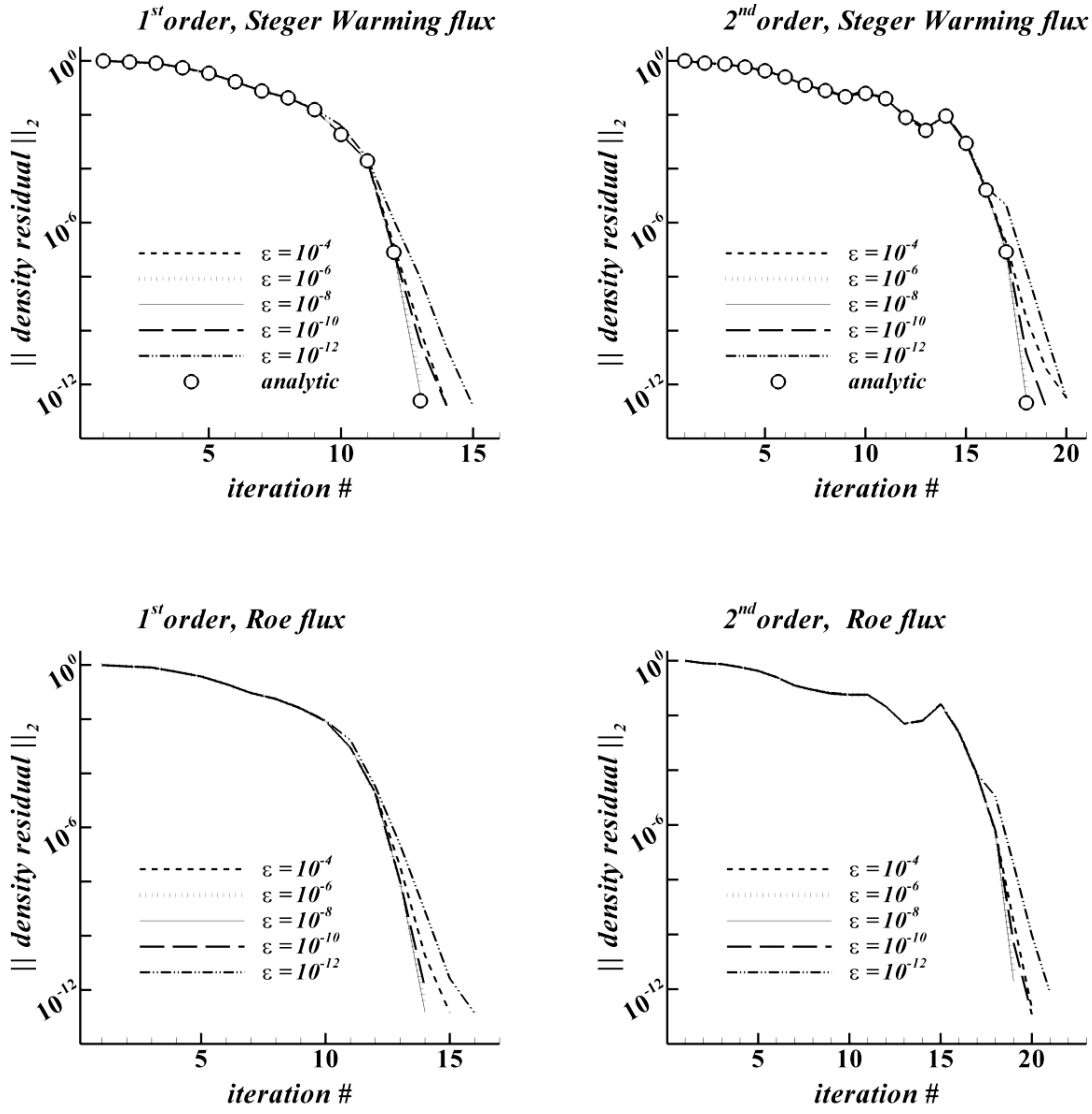

Figure 4.20 *Residual histories for AUSM and Van Leer schemes in the solution of flow over Ni-bump. M=0.675*

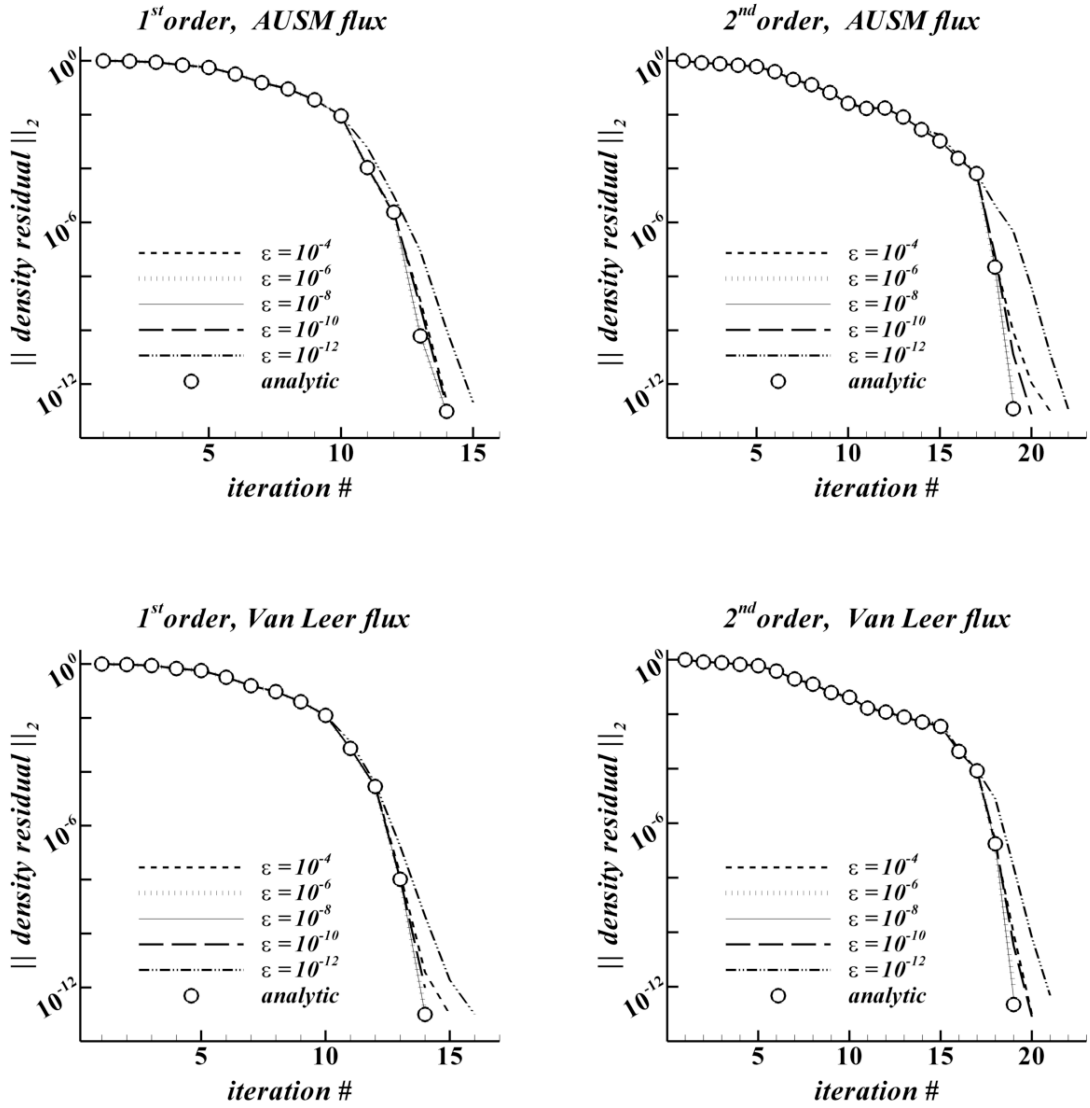

Figure 4.21 *Residual histories for AUSM and Van Leer schemes in the solution of flow over NACA0012. M=0.85 α = 1°*

Figure 4.19 gives the history of the averaged density residual for the ASUM and Van Leer schemes for both of the first and second order accurate spatial discretization. Similar results are given for the Steger warming scheme in Figure 4.20. In those plots the convergence history attained by the analytical Jacobian is represented by symbols and the results correspond to the numerical Jacobian matrices are presented by lines. In Figure 4.20 the residual histories resulted from the usage of Roe's scheme are also given. Although the analytical differentiation of the Roe scheme is not performed for Jacobian evaluation, the results corresponding to numerical Jacobian matrices are given to present

the effect of the perturbation magnitude on the convergence rate. Figure 4.19 and 4.20 presents the convergence histories correspond to solution of transonic flow solution over bump geometry. Results correspond to the solution of transonic flow over NACA0012 airfoil is presented in Figure 4.21 and 4.22

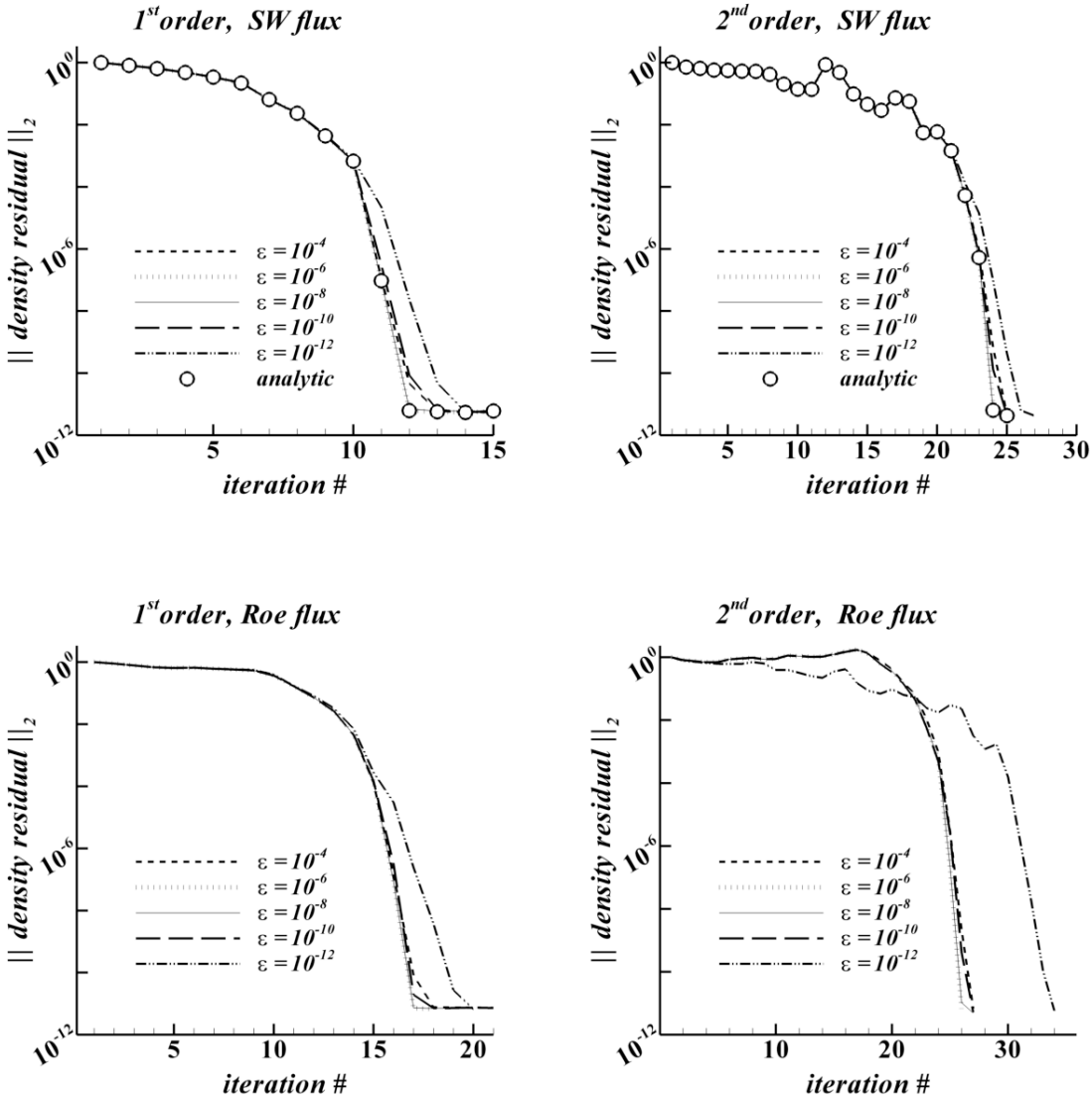

Figure 4.22 *Residual histories for Steger Warming and Roe schemes in the solution of flow over NACA0012. M=0.85 α = 1°*

The residual history plots presented for the bump geometry are evaluated by the mesh that has 65x17 nodes and the plots correspond to NACA0012 airfoil are evaluated by the

mesh that has 129x33 nodes. To eliminate the effect of diagonally added time like term on the convergence, its initial and withdrawal values are kept constant for all numerical Jacobian matrices evaluated with different finite difference perturbation magnitude. Those fixed values for the flow over bump and airfoil geometries are listed in Table 4.7.

Table 4.7 *The initial and withdrawal values used for the diagonally added term, $1/\Delta t$ in the generation of presented results*

	Δt_0	Δt_f
1st order spatially accurate solution of flow over bump	3	5
2nd order spatially accurate solution of flow over bump	50	1000
1st order spatially accurate solution of flow over NACA0012	300	10000
2nd order spatially accurate solution of flow over NACA0012	200	20000

In Chapter 3 it was shown that usage of smaller values of ϵ, which is used to inactivate the limiter in smooth flow regions, degraded the convergence performance of the flow solver. In the section 4.4.5 it is shown that magnitude of the second derivatives of the residual becomes significantly larger compared to the norm of the flux vector. The large values of second derivatives of the residual may degrade the validity of the assumption made in the linearization of the residual by neglecting the high order terms. The inaccurate linearization will be caused due to the large values of second derivatives and it may result in as a fall of convergence performance.

Figure 4.23 presents the change of the residual histories of the flow solutions evaluated by variety of ϵ used in MUSCL interpolation. The results evaluated by analytical Jacobian is represented by the symbols and the residual histories resulted from the utilization of the numerical Jacobian matrices are represented by the lines. The results are generated by the transonic flow solution over bump geometry. Second order spatial discretization is used with the Van Albada's limiter. In Figure 4.14 it was shown that the optimum perturbation magnitude for the numerical Jacobian evaluation changes depending on the magnitude of ϵ used. In the generation of the Figure 4.23 each

numerical Jacobian is constructed by its own optimum finite difference perturbation magnitude that is read from Figure 4.14.

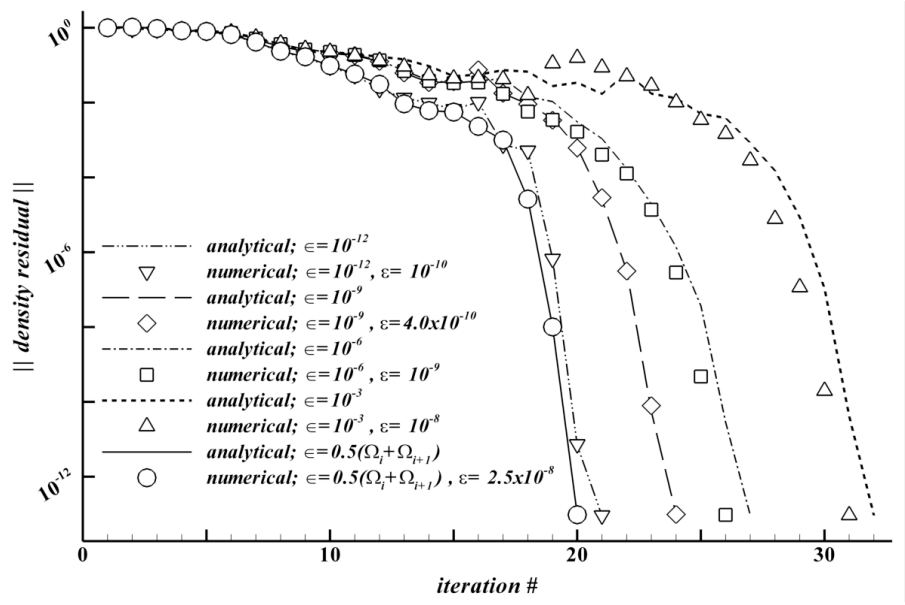

Figure 4.23 *Variation of the residual histories by the value of the ∈ that is used in MUSCL interpolation*

The last residual history type plot is given to present the effect of the exactness of the Jacobian matrix linearization on the convergence performance. In references [70], [71], [72] it was stated that for the best convergence performance the scheme used in the linearization of the residual for Jacobian evaluation should be consistent with the one used in the spatial discretization. In this study Roe scheme is implemented into the developed code for spatial discretization but the analytical Jacobian derivation of the Roe flux scheme was not performed. A trial case, where the Jacobian is evaluated by AUSM scheme and the fluxes are calculated by the Roe scheme, is solved to observe the effect of the inexact linearization on the convergence performance. The residual history given in Figure 4.24 presents the significant reduction in the convergence rate due to inexact linearization.

Results given in Figures 4.19 to 4.22 presents that using the optimum perturbation magnitude given by Equation 4.20 for numerical Jacobian evaluation, convergence

performance identical to one resulted from the usage of analytical Jacobian can be obtained. However it is also shown that convergence performance of the Newton's method in the flow solution is not very sensitive to the accuracy of the Jacobian. The reduction in the accuracy of the numerical Jacobian only caused a slight increment in the number of iterations required for the converged solution. Moreover it is also shown by Figures 4.23 and 4.24 that, the convergence performance degraded significantly in the case of improper MUSCL limiter definition and in the case of inexact linearization.

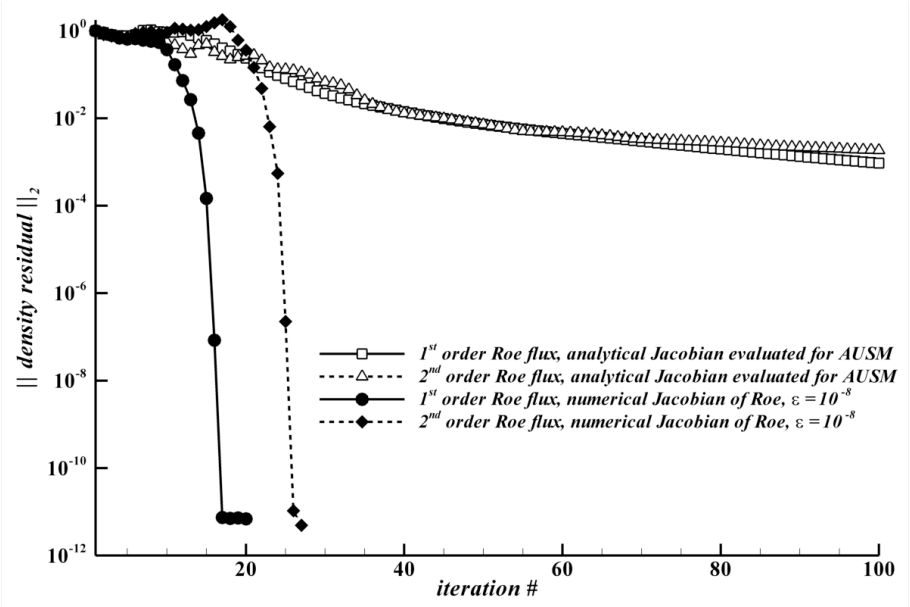

Figure 4.24 *Effect of the inexact linearization of the Jacobian on the convergence rate*

It can be concluded that as long as exact linearization is performed; the Newton's method is very robust scheme in terms of convergence and it is insensitive to the small errors in the Jacobian matrix. Results also showed that numerical evaluation of the Jacobian matrix can be very beneficial since Jacobian of complex schemes can be constructed simply without a loss in convergence performance.

This chapter is concluded by the cost analysis of the choice made on analytical and numerical Jacobian in the flow solution. The comparison of the cost is made in terms of the CPU time spent.

Figure 4.25 presents the ratio of the CPU time spent in evaluation and factorization of the numerical Jacobian to total CPU time spent in an iteration of the Newton's method. As it can be seen from the plots almost entire computational effort in a Newton's method is demanded by Jacobian evaluation and the factorization. As the grid gets finer, the percentage of the time spent in the factorization step increases.

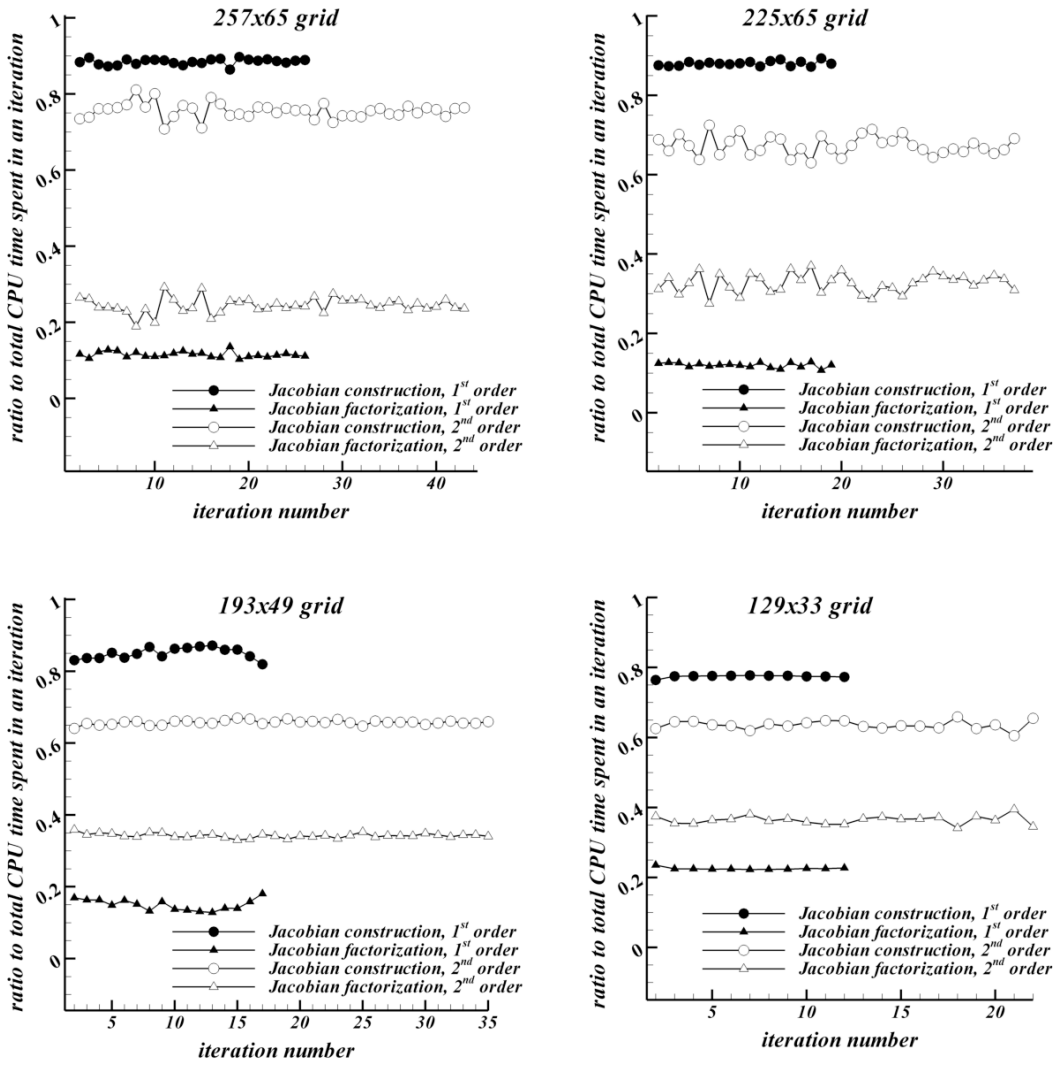

Figure 4.25 *CPU time spent in numerical Jacobian evaluation and factorization in terms of the percentage of the total time spent in iteration*

Figure 4.26 presents the variation of the CPU time spent in evaluation and factorization of both of the numerical and analytical Jacobian matrices by the change of grid

resolution. Both of the cost of first and second order discretization is presented in that figure. Figure 4.26 shows that the both of first and second order numerical Jacobian evaluation require 5 times more CPU time than the one needed for the evaluation of analytical Jacobian. Another conclusion is that, if the numerical Jacobian is used in flow solution by a fine grid, great percentage of the computational effort will be spent on the finite differencing. On the other hand if the analytical Jacobian is used the CPU time spent in matrix factorization will be larger than the one required for the evaluation of the analytical differentiation.

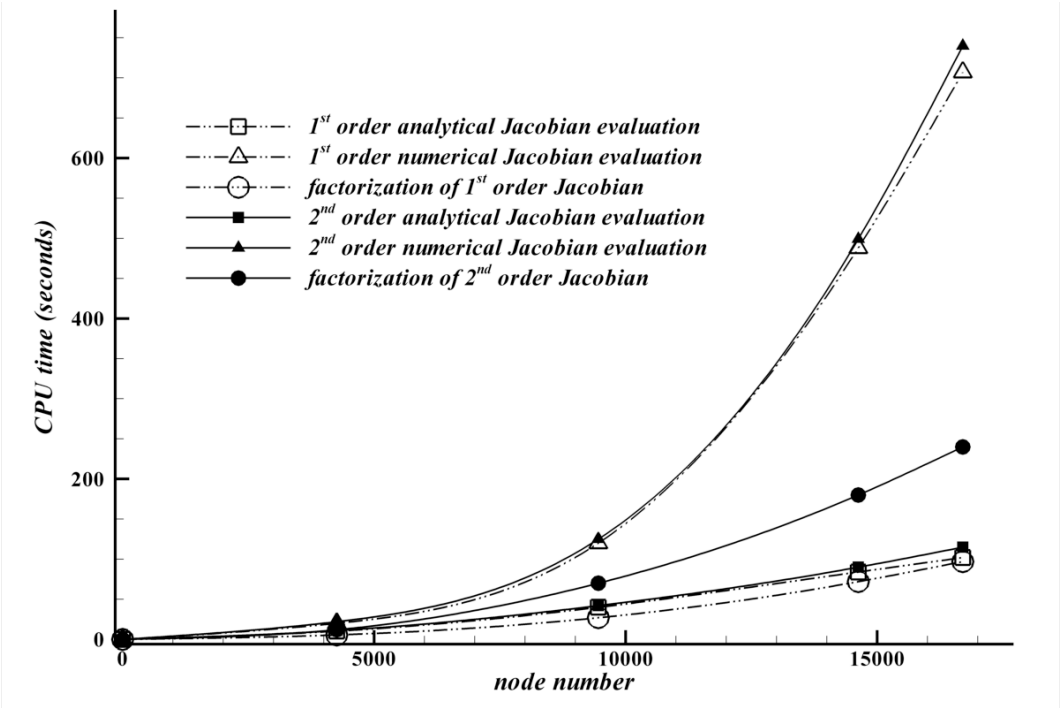

Figure 4.26 *Variation of the CPU time spent in one Newton iteration by the grid resolution*

CHAPTER 5

SENSITIVITY ANALYSIS

5.1 Introduction

In recent years, the promising developments accomplished in the computational science and enabled the usage of CFD as an efficient design tool. Today very complex aircraft configuration can be designed effectively by the utilization of the CFD methods. Design techniques applied by the researchers have continuously evolved, and resulted in variety of available design techniques. Amongst the widely used design techniques, gradient based optimization methods are the most commonly applied ones. The efficiency and the effectiveness of the gradient based optimization methods are dependent on the technique known as the sensitivity analysis.

Sensitivity analysis was a well understood technique in the field of computational structural mechanics prior to its applications in the fluid dynamics discipline. At the most fundamental level sensitivity analysis is developed on the principle that, information about the behavior of an unknown function in the neighborhood of a known point can be approximated if the slopes in the neighborhood of the known point are defined. Roughly speaking, a sensitivity analysis involves the calculation of slopes, known as sensitivity derivatives. Those slopes are replaced by the gradients in the gradient based design optimization applications.

Application fields of the of sensitivity analysis are not limited to design optimization, but they are also effectively used; in the generation of better initial guesses for analysis, in function approximation to predict trends in the response of a system as a consequence of changes in the independent variables, and in error estimation. In this thesis performed study is focused on the application of sensitivity analysis in the aerodynamic design optimization.

5.2 Aerodynamic Design Optimization

Today the coupled efficient optimization algorithms with CFD solvers became a handy tool to be used both in the aerodynamic design and the modernization of existing designs for further performance enhancement. Generally the numerical optimization problem is defined as

$$\begin{aligned} &\textit{Minimize (Maximize)}: \quad F(\boldsymbol{\beta}) \\ &\textit{subject to}: \quad g_j(\boldsymbol{\beta}) \le 0 \qquad\qquad j = 1, J \\ &\qquad\qquad \beta_i^L \le \beta_i \le \beta_i^U \qquad i = 1, I \end{aligned} \quad . \tag{5.1}$$

where, F is the objective function to be minimized (maximized), and g_j are the J inequality constraints. $\boldsymbol{\beta}$ is the vector of design variables which has I β_i components. β_i^L and β_i^U are the side constraints which determine the lower and upper limits of the design variables.

The gradient based optimization method is one of the most widely used techniques in aerodynamic design optimization. The objective of the optimization is generally defined as maximization or minimization of the aerodynamic loads. To be able satisfy this objective, gradients of objective functions in design variable space is required. Those gradients of aerodynamic loads are called as aerodynamic sensitivities. The efficient and accurate computation of the sensitivities are vital since the procedure followed in the sensitivity evaluation and the accuracy of them directly affect the quality of the design and the performance of the design in terms of computational requirements. Sensitivity analysis is also required in inverse design optimization where the objective is defined as obtaining the aerodynamic geometry which produces the specified pressure distribution, under specified flight conditions. In the iterative design process of the inverse design, which starts from an initial baseline geometry and end up with target geometry, the sensitivities of the surface pressure to the design variables are needed.

5.3 Aerodynamic Sensitivity Analysis

Aerodynamic loads are functions of state flow variables, geometrical variables and the design variables [73].

$$C_j = C_j\left\{\boldsymbol{w}(\beta), \boldsymbol{X}(\beta), \beta\right\} \tag{5.2}$$

The sensitivities of the aerodynamic loads to the design variables can be defined by the utilization of the chain rule as below:

$$\frac{dC_j}{d\beta_k} = \left[\frac{\partial C_j}{\partial \boldsymbol{w}}\right]\left\{\frac{\partial \boldsymbol{w}}{\partial \beta_k}\right\} + \left[\frac{\partial C_j}{\partial \boldsymbol{X}}\right]\left\{\frac{\partial \boldsymbol{X}}{\partial \beta_k}\right\} + \left\{\frac{\partial C_j}{\partial \beta_k}\right\} \tag{5.3}$$

Equation 5.3 shows that three group of derivatives are needed to evaluate the sensitivities of the aerodynamic loads. In Equation 5.3, the second term represents the sensitivity of the aerodynamic loads on the geometric variables. Geometric variables consist of the coordinates of the analyzed geometry and the mesh used to discretize the flow domain. If the aerodynamic optimization is performed with non-geometric design variables with constant kept geometry and mesh, the second term in the right side of the Equation 5.3 becomes zero. If the geometric design variables are used to control the geometry and the mesh, the third term in the right hand side of the equality will be zero equated to zero, since the effect of the geometrical variables on the total derivative will already be stated by the second term.

In the aerodynamic shape optimization geometry will be parameterized and the design variables would be geometrical and directly related to the parameterization. There is wide range of parameterization techniques available in literature and the most common ones are the parameterization by shape functions and the method of B-spline control points. During the shape optimization process to satisfy the targeted objective, parameterized baseline geometry is distorted by applied perturbations. The amount of those perturbations is controlled by design variables. For aerodynamic shape optimization applications, the Equation 5.4 will be simplified to Equation 5.5

$$\frac{dC_j}{d\beta_k} = \left[\frac{\partial C_j}{\partial \boldsymbol{w}}\right]\left\{\frac{\partial \boldsymbol{w}}{\partial \beta_k}\right\} + \left[\frac{\partial C_j}{\partial \boldsymbol{X}}\right]\left\{\frac{\partial \boldsymbol{X}}{\partial \beta_k}\right\} \tag{5.5}$$

where C_j, $\boldsymbol{w}$, $\boldsymbol{X}$, β_k are aerodynamic loads, flow variables vector, grid coordinates and design variables, respectively. The aerodynamic loads have explicit dependence on the flow variables and on the coordinates of the geometry. As an example, for Euler equations, the lift coefficient, C_L, and drag coefficient, C_D will be evaluated from the pressure coefficient, C_p, which explicitly depends on the flow variables and the geometry coordinates. The explicit relations between the aerodynamic loads, pressure coefficient, flow variables and the mesh coordinates are presented in Equations 5.6 to 5.10.

$$\begin{aligned} C_L &= C_y \cos\alpha - C_x \sin\alpha \\ C_D &= C_y \sin\alpha + C_x \cos\alpha \end{aligned} \tag{5.6}$$

C_x and C_y are the force coefficients in x and y directions and can be evaluated as:

$$C_x = \sum_{j=1}^{NE} C_{xj} \qquad C_y = \sum_{j=1}^{NE} C_{yj} \tag{5.7}$$

In above equation, NE denotes the number of elements on the geometry constructed by the mesh. The evaluation of the C_{xj} and C_{yj} will be performed by using C_p as follow:

$$\begin{aligned} C_{xj} &= C_{pj}\left(y_{b_{j+1}} - y_{b_j}\right) \\ C_{yj} &= C_{pj}\left(x_{b_j} - x_{b_{j+1}}\right) \end{aligned} \tag{5.8}$$

Pressure coefficient can be calculated as:

$$C_{p_j} = \frac{P_{wall\,j}}{\frac{1}{2}\rho_\infty V_{\infty l}^2} \tag{5.9}$$

Finally, the pressure on the geometry can be written in terms of the conservative flow variables as below:

$$P_{wall\,j} = (\gamma - 1)\left((\rho E)_{wall_j} - \frac{(\rho u)^2_{wall_j} + (\rho v)^2_{wall\,j}}{2\rho_{wall\,j}} \right) \tag{5.10}$$

Following the relations given in above equations the analytical evaluations of the sensitivity of the aerodynamic load to the flow variables and to grid coordinates ($\partial C_j/\partial \boldsymbol{w}$ and $\partial C_j/\partial \boldsymbol{X}$ derivatives) can be performed effortlessly as follow:

$$\frac{\partial C_j}{\partial \boldsymbol{w}} = \frac{\partial C_j}{\partial C_x}\frac{\partial C_x}{\partial \boldsymbol{C}_p}\frac{\partial C_p}{\partial \boldsymbol{w}} + \frac{\partial C_j}{\partial C_y}\frac{\partial C_y}{\partial \boldsymbol{C}_p}\frac{\partial C_p}{\partial \boldsymbol{w}} \tag{5.11}$$

$$\frac{\partial C_j}{\partial \boldsymbol{X}} = \frac{\partial C_j}{\partial C_x}\frac{\partial C_x}{\partial X} + \frac{\partial C_j}{\partial C_y}\frac{\partial C_y}{\partial X} \tag{5.12}$$

The evaluation of the grid sensitivity term is also straightforward for most of the practical cases. Considering the shape function usage in the aerodynamic shape optimization, the geometry modification will be performed by the formulation given below:

$$\boldsymbol{X}_{new} = \boldsymbol{X}_{old} + \beta_k \left(f_k(x)\frac{j_{\max} - j}{j_{\max} - 1} \right) \tag{5.13}$$

where $f_k(x)$ is the shape function used for geometric parameterization, j is the grid point index which varies along the direction vertical to the geometry, β_k is the weighting of the shape function(design variable) , $\boldsymbol{X}_{new}$ and $\boldsymbol{X}_{old}$ are the coordinates of the grid points prior to and after the perturbation respectively. Generally the modification to geometry is applied such that the domain close to the wall geometry will be perturbed mostly and effect of perturbation will be diminished away from the wall. The grid points on the boundaries that are furthest to the wall are generally prevented from the modification to keep the bounds of the flow solution domain constant.

Equation 5.13 shows that the grid coordinates have an explicit dependence on the design variables; hence the grid sensitivities can be evaluated simply as:

$$\frac{\partial \boldsymbol{X}}{\partial \beta_k} = f_k(x)\frac{j_{\max} - j}{j_{\max} - 1} \tag{5.14}$$

Although, the evaluation of the gradients; $\partial C_j / \partial \boldsymbol{w}$, $\partial C_j / \partial \boldsymbol{X}$, $\partial \boldsymbol{X} / \partial \beta_k$ is straightforward thanks to the explicit relations presented above, the evaluation of the derivative $\partial \boldsymbol{w} / \partial \beta_k$ is not simple since the relation between flow variables and design variables is implicit. This derivative can be evaluated with a brute approach using the finite differencing, however the evaluated sensitivities would be erroneous. More accurate evaluation can be performed analytically by one of two common methods. First alternative is the direct differentiation method where, the solution of linear system is performed for the evaluation of flow state sensitivities. Adjoint method is the other alternative and it also requires a solution of similar linear system for the evaluation of the adjoint variable. Adjoint method bypasses the requirement for the evaluation of the flow sate sensitivities, $\partial w / \partial \beta_k$, by introduction of Lagrange multipliers. The computational advantage of the introduction of the adjoint method can be summarized as; requirement on multiple linear system solutions for the evaluation of the sensitivity of flow variables with respect to each design variables will be reduced to a single linear system solution for the adjoint variable. Both of the adjoint and the direct differentiation method necessitate the calculation of the flux Jacobian matrices in the solution of the linear systems. By utilization of those two techniques the sensitivity of the aerodynamic loads can be evaluated fully analytically, hence sensitivity derivatives can be obtained accurately compared to the evaluated ones by finite difference. The choice between those two analytical methods is made according to the size of the objective functions and the design variables. If the number of design variables is larger than the number of objective functions, the adjoint method is computationally more desirable compared to the direct differentiation method. However the usage of direct solvers in flow solution will remove the computational difference between those two methods. The Jacobian matrix which was factored in the flow solution can be reused in sensitivity analysis; hence cost of the application of both techniques will reduce considerably. In this study the direct differentiation method is applied for sensitivity analysis and its formulation is given in next sub-section. Detailed formulations for both of the direct differentiation method and the adjoint methods can be found in References [78] and [79].

5.3.1 Direct Differentiation Method

Flux residual vector, which is dependent on flow variables and grid coordinates, equals to zero at the steady state conditions.

$$\left\{ R\left(\boldsymbol{w}\left(\beta_k\right), \boldsymbol{X}\left(\beta_k\right)\right)\right\} = \{0\} \tag{5.15}$$

Following the fact given in Equation 5.15 the flow state sensitivity, $\partial w/\partial \beta_k$ will be evaluated from direct differentiation of steady state flow governing equations as follow [75], [76].

$$\left[\frac{\partial R}{\partial \boldsymbol{w}}\right]\left\{\frac{\partial \boldsymbol{w}}{\partial \beta_k}\right\} + \left[\frac{\partial R}{\partial X}\right]\left\{\frac{\partial X}{\partial \beta_k}\right\} = 0 \tag{5.16}$$

$$\frac{\partial \boldsymbol{w}}{\partial \beta_k} = -\left[\frac{\partial R}{\partial \boldsymbol{w}}\right]^{-1}\left[\frac{\partial R}{\partial X}\right]\left\{\frac{\partial X}{\partial \beta_k}\right\} \tag{5.17}$$

The formulation given in Equation 5.14, shows the straightforward evaluation of the $\partial X / \partial \beta_k$ term. The $\partial R / \partial X$ term can be performed analytically or numerically. The analytical evaluation is favorable and it can be generally performed simply. For most of the cases the dependency of residual on the grid coordinates comes from the explicit relations used in the definitions of the face normal vectors.

The solution for the systems of linear equations given by Equation 5.16 can be performed with the same algorithm used in the flow solution. As it is stated earlier; using the LU factors of the Jacobian that are computed in the flow solution, the sensitivity vector can be evaluated efficiently. Simple backward and forward substitution operations will be sufficient to obtain the solution vector.

5.4 Analytical Sensitivity Analysis

This section briefly summarizes the results obtained from the analytical sensitivity analysis. To generate the results presented in this chapter the geometry was firstly

parameterized by the utilization of Hicks-Henne shape functions. Application of smooth perturbations on the geometry by adding set of Hicks-Henne functions is a commonly used technique in geometry modification. The peak value of the each function in the set occurs in different chord-wise locations, hence the geometry modification can be controlled by adjustment of the weighting of each functions in the set. The weighting of each Hicks-Henne functions are defined as the design variables in the generation of results presented in this chapter.

Throughout this chapter the direct differentiation technique is used to evaluate the sensitivities of flow state variables. Both of the numerically and analytically evaluated Jacobian matrices are used in the solution of the system given in equation 5.16. In the following sections the sensitivities evaluated by numerical Jacobian is compared with the one that is evaluated using the analytical Jacobian. Before proceeding into such kind of error analysis, the verification of the sensitivities evaluated by the direct differentiation method is performed, in order to examine whether the developed code produces reasonable analytical sensitivities or not.

Firstly, verification is performed by the comparison of the analytically evaluated flow variables sensitivities with the ones evaluated simply by finite differencing. Although finite difference results are erroneous, using the proper step size their results will not differ from the exact results tremendously and they can be used to examine the reasonability of analytical results. In the evaluation of the sensitivity derivatives by finite differencing the magnitude of the step size is determined by the approach presented in reference [74].

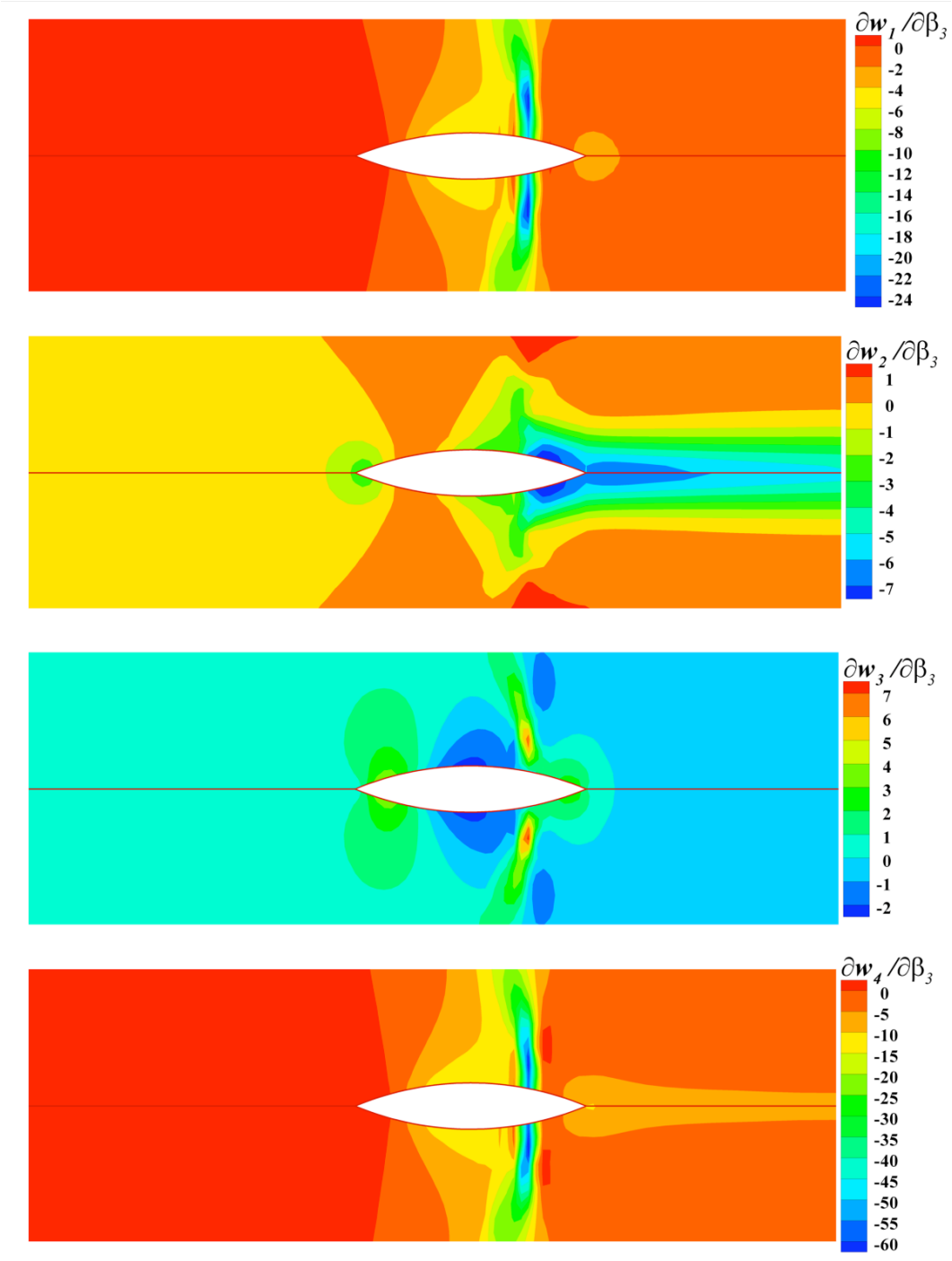

Figure 5.1 *The comparison of the sensitivities evaluated by direct differentiation method and finite differencing (Upper halves: finite difference, lower half: analytical sensitivity analysis)*

Figure 5.1 presents the comparison of the contours of the analytically evaluated flow variable sensitivities and the ones that are evaluated by finite differencing. The analytically evaluated sensitivity contours are presented on the upper half of the plots whereas on the lower halves the sensitivities evaluated by brute method are presented. The presented contour plots correspond to the sensitivities of the flow variables with respect to the randomly selected Hicks-Henne's function, whose peak value occurs at the thirty percent of the chord. The contour patterns obtained from both of the analytical and brute sensitivity evaluation looks similar with approximately equal magnitudes, hence it can be referred from that figure that the developed solver works fine for the analytical sensitivity computation.

Secondly, the cost of the application of direct differentiation method for sensitivity analysis is presented. The utilization of the Newton's method in the flow solution brought out the efficient sensitivity evaluation. Using the LU factors obtained in flow solution, flow state sensitivities formulated by Equation 5.17 are calculated in single iteration by simple backward and forward substitution. The computational time spent in the evaluation of the sensitivities is presented in Table 5.1. To present the cost of the sensitivity evaluation relative to flow solution, the CPU time spent in the flow solution is also given in Table 5.1. Results are given for the evaluation of sensitivities with respect to one design variable.

Table 5.1 *Comparison of time CPU time spent in flow and sensitivity analysis*

	CPU time spent in flow solution (seconds)	*CPU time spent in the evaluation of flow variable sensitivities to one design variable (seconds)*
65x17 grid, 1st order	9.23	0.11
65x17 grid, 2nd order	18.87	0.21

The advantage gained by the re-use of the LU factors is independent from the evaluation method followed in the Jacobian evaluation. Although the utilization of the Newton's

method is expensive for flow solution in terms of computation time, the sensitivity analyses can be handled very efficiently both by the numerical and the analytical Jacobian matrices.

5.5 Accuracy of Sensitivity Analyses

In previous section it was explained that, the flow state sensitivities are the dominant components on the accuracy of the objective function sensitivities. Hence, the accuracy of the flow state sensitivities are analyzed in detail and results are presented in this section. Equation 5.17 shows that, the calculation of flow state sensitivities requires the evaluation of flux Jacobian matrix, $\partial R/\partial w$. In Chapters 3 and 4, the analytical and numerical evaluations of flux Jacobian were presented, and the accuracy of the numerical Jacobian was analyzed. The main interest of this section is to investigate the effect of accuracy of the numerical flux Jacobian on the accuracy of flow state sensitivities.

5.5.1 Condition Number

In linear algebra to find out the effect of small changes, which occurred in the right-hand side vector or in left-hand side matrix, on the solution of the system the condition number is defined. The linear system

$$Ax = b \tag{5.18}$$

has a unique solution for every right-hand side only if the matrix, A is square and non-singular. The exact solution of the system is defined as:

$$x = A^{-1}b \tag{5.19}$$

Suppose that the left hand-side matrix is changed to $A+\delta A$ by a perturbation δA, and the exact solution of the system with the perturbed matrix is calculated as $x+\delta x$, i.e.

$$(A + \delta A)(x + \delta x) = b \tag{5.20}$$

The deviation, δx, resulted in the solution x due to the small change, δA, in the matrix is formulated as:

$$\delta x = -A^{1}\delta A(x+\delta x) \tag{5.21}$$

Applying the vector and matrix norm definitions, the measure of the deviation, δx, can be estimated by

$$\|\delta x\| \le \| A^{-1}\| \| \delta A \| \| (x+\delta x) \| \tag{5.22}$$

Representing the amount of change $\|\delta A\|$ relative to the $\|A\|$, the relative change in the exact solution is given as:

$$\frac{\|\delta x\|}{\|(x+\delta x)\|} \le \|A\| \|A^{-1}\| \frac{\|\delta A\|}{\|A\|} \tag{5.23}$$

Equation 5.23 represents that, for a linear system with non varying right hand side vector, the relative change in the exact solution due to the relative change in the left hand side matrix is bounded by a factor $\| A \| \| A^{-1}\|$. This factor is defined as condition number, *cond(A)*.The condition number is always greater than one, and it indicates the maximum effect of the perturbation occurred in matrix on the exact solution. If the condition number is large the matrix is called as ill conditioned. For ill conditioned matrices, even a small amount of change relative to matrix can cause substantial variations on the exact solution.

In this study the condition number definition is used to obtain an initial consideration on the effect of the error occurred in numerical Jacobian on the analytically calculated sensitivities. Following the above equations, the relative change in the left hand side matrix is defined as relative Jacobian error in this study, and the relative change in the solution is defined as the relative sensitivity error. The resulting formula derived from Equation 5.23 to define the relation between the relative sensitivity error and the relative Jacobian error is given below:

$$\frac{\left\|\frac{\partial w}{\partial \beta}\right\|_{numeric} - \left\|\frac{\partial w}{\partial \beta}\right\|_{analytic}}{\left\|\frac{\partial w}{\partial \beta}\right\|_{analytic}} \leq cond\left(\frac{\partial R}{\partial w}\right) \frac{\left\|\frac{\partial R}{\partial w}\right\|_{numeric} - \left\|\frac{\partial R}{\partial w}\right\|_{analytic}}{\left\|\frac{\partial R}{\partial w}\right\|_{analytic}} \tag{5.24}$$

where

$$cond\left(\frac{\partial R}{\partial w}\right) = \left\|\frac{\partial R}{\partial w}\right\| \left\|\frac{\partial R}{\partial w}^{-1}\right\| \tag{5.25}$$

The condition number evaluation is not straightforward since the norm value of the inverse of the large sparse Jacobian matrix is needed. In literature there are some routines developed for the estimation of the condition number without any matrix inversion algebra [77]. In this study the approximate condition number provided by the UMFPACK sparse matrix solver is used.

The condition number alters slightly during the flow solution, since the Jacobian matrix is updated at each Newton's iteration. The average of the condition number evaluated in the last three Newton's iteration of the flow solution is used to estimate the maximum bound of the sensitivity error. The variation of the condition number by the grid size, order of spatial discretization and flow regime is presented in the Table 5.2 for the flow over NACA0012 cases. The approximation to the optimum perturbation magnitude and the resulting minimum error was given by Equation 4.20 and Equation 4.21 in Chapter 4. Inserting those condition numbers and estimated minimum numerical Jacobian error values into the Equation 5.24; the maximum bounds of the relative sensitivity error, which correspond to the numerical Jacobian evaluated by optimum perturbation magnitude, are estimated. Estimated error bounds are given in Table 5.2.

The results presented in Table 5.2 propose that, the condition number varies significantly by the change of order of the spatial discretization. Due to the relation between stencil size and the order of discretization, the change in order of discretization will significantly affect the structure of the Jacobian matrix. Hence it is reasonable to expect shifting

condition number by that kind of change. Table 5.2 also presents that condition number is sensitive to change in free-stream/inlet Mach number of the dealt flow case. In supersonic flow all the eigenvalues of the Jacobian matrix would be positively signed; hence the flux vectors would be calculated by the utilization of the downstream cells only. However for subsonic case cells located both in downstream and the upstream would be used in the flux evaluation. Therefore the structure of the Jacobian matrix may slightly vary with the change in the flow regime. That kind of variation could be the main reason for the change of condition number by the flow regime.

Table 5.2

NACA 0012	*Estimated condition number*	*Estimated maximum bound for the relative sensitivity error*
129x33 grid, 0.85Mach 1st order	2.0×10^{4}	6.3×10^{-4}
129x33 grid, 0.85Mach 2nd order	7.1×10^{5}	2.1×10^{-2}
257x65 grid, 0.85Mach 2nd order	8.3×10^{5}	2.5×10^{-2}
129x33 grid, 0.5Mach 2nd order	2.4×10^{5}	7.2×10^{-3}
129x33 grid, 1.20Mach 2nd order	1.1×10^{6}	3.3×10^{-2}

5.5.2 Effect of the Numerical Jacobian Accuracy on the Sensitivity Accuracy

Expressing the effect of the condition number on the accuracy of the analytical sensitivity evaluation the detailed error analysis is performed. Approach similar to one followed in Chapter 4 is followed to examine the effect of the numerical Jacobian evaluation on the accuracy of the sensitivity analysis. Numerical Jacobian matrices are evaluated by variety of finite difference perturbation magnitudes and they are used in sensitivity evaluation to calculate sensitivity vectors and they are referred as numerical sensitivities in the following sections. The numerical sensitivities evaluated with those numerical Jacobian matrices are compared with the ones that are evaluated by the analytical Jacobian. The difference between the sensitivities that are evaluated from

numerical and analytical Jacobian is defined as the sensitivity error. To be able to present the total error in the sensitivity vector, vector norm definitions are used.

$$\left(Error_{\text{sensitivity}}\right)_i = \left(\frac{\partial W}{\partial \beta_k}_{\ numeric\ \ jacobian}\right)_i - \left(\frac{\partial W}{\partial \beta_k}_{\ analytic\ \ jacobian}\right)_i \tag{5.26}$$

$$\begin{aligned} \left\|Error_{\text{sensitivity}}\right\|_1 &= \sum_{i=1}^{4(imax)(jmax)} \left| \left(\frac{\partial W}{\partial \beta_k}_{\ numeric\ \ jacobian}\right)_i - \left(\frac{\partial W}{\partial \beta_k}_{\ analytic\ \ jacobian}\right)_i \right| \\ \left\|Error_{\text{sensitivity}}\right\|_2 &= \sqrt{\sum_{i=1}^{4(imax)(jmax)} \left(\left(\frac{\partial W}{\partial \beta_k}_{\ numeric\ \ jacobian}\right)_i - \left(\frac{\partial W}{\partial \beta_k}_{\ analytic\ \ jacobian}\right)_i \right)^2} \end{aligned} \tag{5.27}$$

The relative error in the numerical sensitivity vector is defined as:

$$\left(Error_{\text{sensitivity}}\right)_{relative} = \frac{\left\|Error_{\text{sensitivity}}\right\|}{\left\|\frac{\partial W}{\partial \beta_k}_{\ \substack{by\ analytic \\ jacobian}}\right\|} \tag{5.28}$$

To be able to perform the accuracy analysis investigating the affect of the finite difference perturbation magnitude used in Jacobian evaluation on the accuracy of sensitivity evaluation, following procedure is followed by the developed flow solver.

Firstly, the flow equations are solved with Newton's method for given operating conditions. In the last Newton iteration, the LU factors of the analytical Jacobian evaluated with converged flow solution is stored in the disk (UMFPACK provides an option for this). The right hand side of the Equation 5.17 is evaluated analytically for each design variable. Using the stored factors of analytical Jacobian, analytical sensitivities are evaluated. A loop is constructed to vary the finite difference perturbation magnitude. In the loop for the each perturbation magnitudes, numerical Jacobian matrices are calculated and using those ones, numerical sensitivity vectors are computed. For the each perturbation magnitude in the loop, calculated numerical Jacobian and sensitivity are compared with the previously stored analytical Jacobian and the sensitivity to generate the error plots given in following sections.

5.5.3 Effect of the Spatial Discretization Method on the Variation of the Sensitivity Error

In this section, the effects of the flux splitting methods and the order of discretization on the variation of the error in the numerical sensitivities are presented. In Figures 5.2 and 5.3 sensitivity error plots showing the variation of the error for different flux splitting schemes are given. The plots are generated for the flow over Ni-bump and the NACA0012 airfoil cases respectively. In those plots, the identical cases used in the generation of the Figures 4.11 and 4.12 are used. Comparison of the Figures 5.2 and 5.3 with the Figures 4.11 and 4.12 shows that; same perturbation magnitude minimizes both of the error in the numerical Jacobian and sensitivity evaluations. The optimum perturbation magnitudes presented by those plots are in good agreement with the one approximated by Equation 4.11.

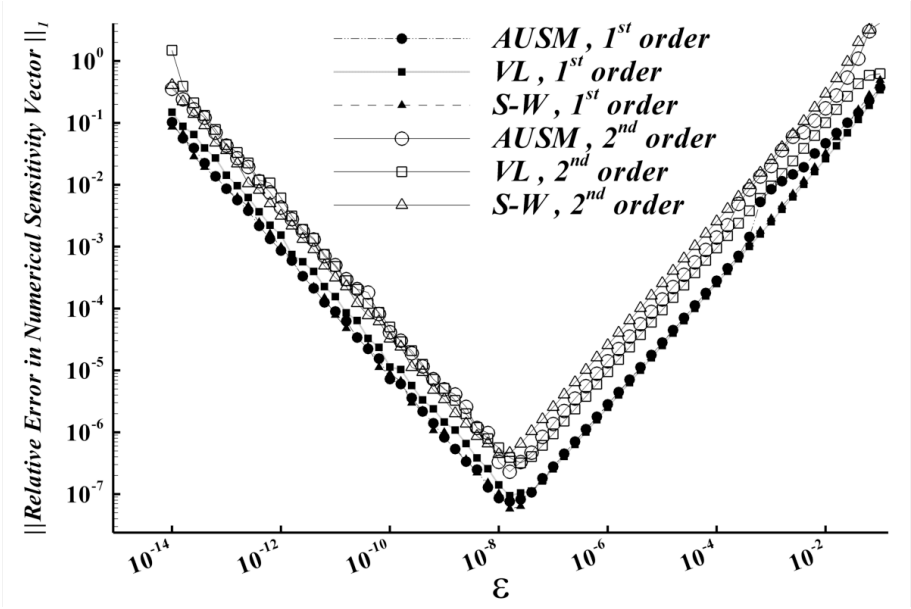

Figure 5.2 *Effect of the flux evaluation method on the variation of the sensitivity error with respect to the finite difference perturbation magnitude used in numerical Jacobian, bump geometry*

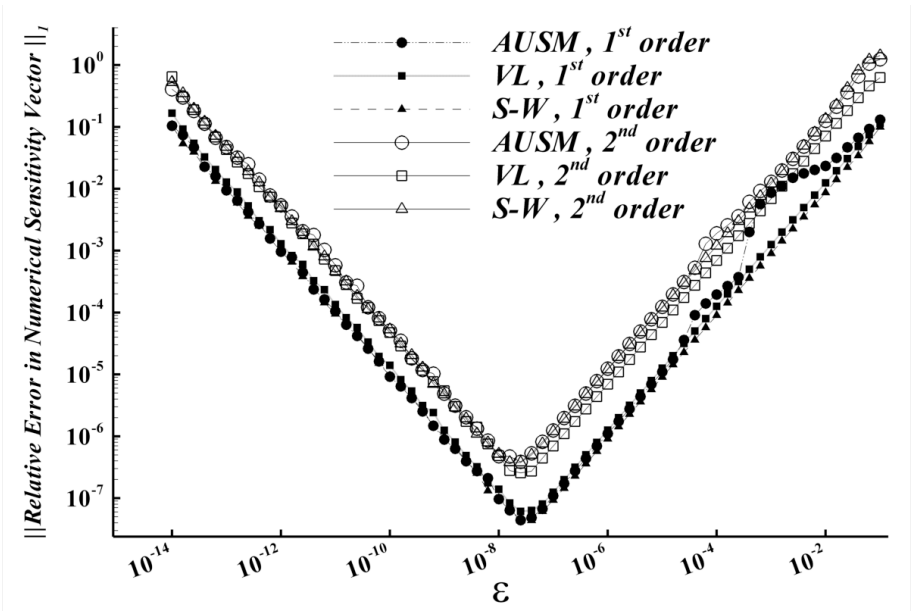

Figure 5.3 *Effect of the flux evaluation method on the variation of the sensitivity error with respect to the finite difference perturbation magnitude used in numerical Jacobian, airfoil*

In the Table 5.2 it was presented that; the condition number of the matrix does not vary significantly with the change in the upwind method whereas it magnifies approximately by an order of magnitude when the second order discretization is used. Basing on that change in the condition number, relevant increase in the maximum bound of sensitivity error was approximated. The results plotted in Figures 5.2 and 5.3 are also in consistent behavior with those given in Tables 5.2. The magnitude of the error in the sensitivity vector amplified for the higher order spatial accuracy.

5.5.4 Variation of Error with Grid Resolution

The effect of the grid resolution on the accuracy of the sensitivity analysis is studied. Figures 5.4 and 5.5 present sensitivity accuracy analysis for bump and airfoil geometries, respectively. In the flow solutions over both of the presented geometries the relative error does not vary by the change of the grid resolution. Similar behavior was presented in Table 5.2 where there was no variation in condition number by the change of the grid resolution. The increment in the magnitude of the relative sensitivity error by the usage of second order of spatial discretization can also be detected from those figures

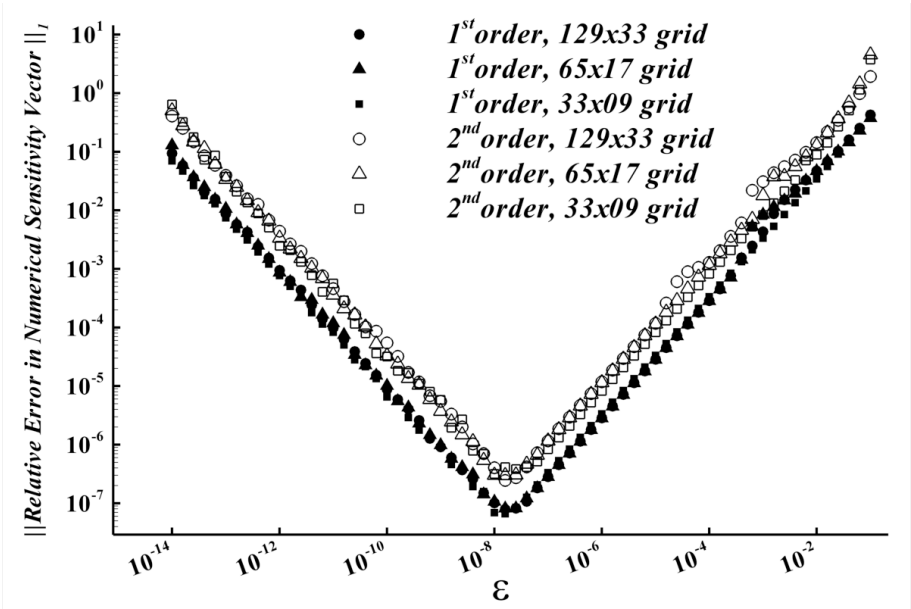

Figure 5.4 *Variation of the relative error in sensitivity by grid resolution, bump geometry*

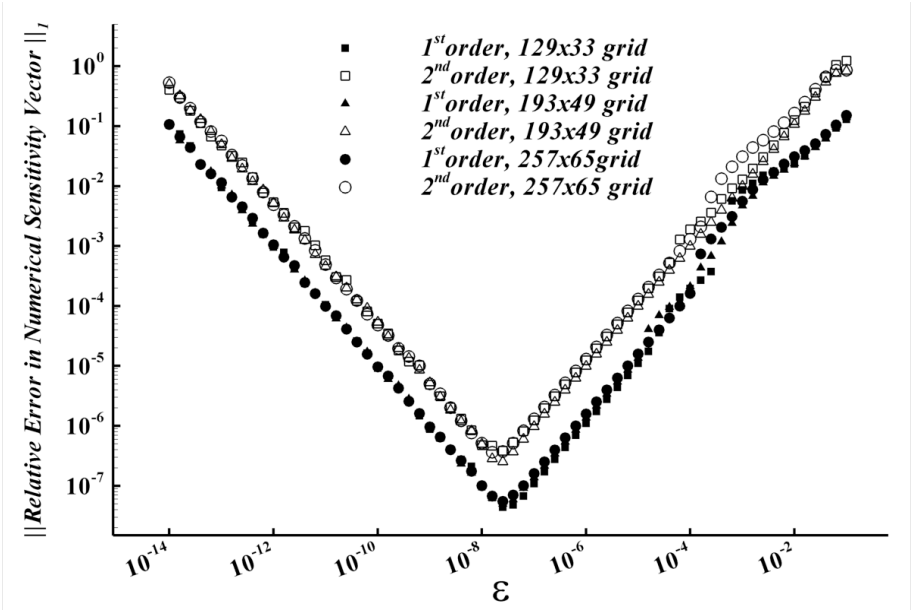

Figure 5.5 *Variation of the relative error in sensitivity by grid resolution, NACA0012 geometry*

5.5.5 Variation of Error with Flow Regime

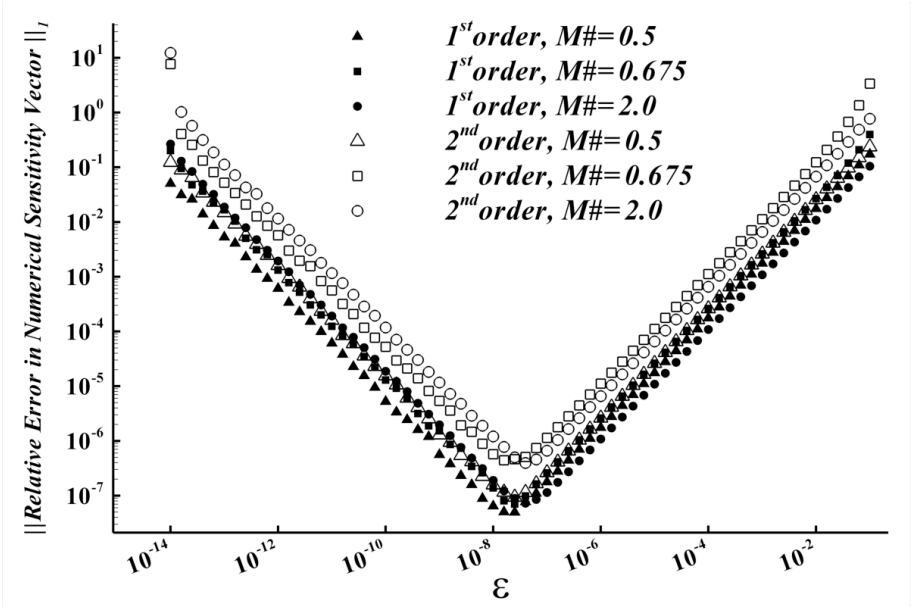

Figure 5.6 *Variation of the relative error in sensitivity by inlet Mach number, Ni bump geometry*

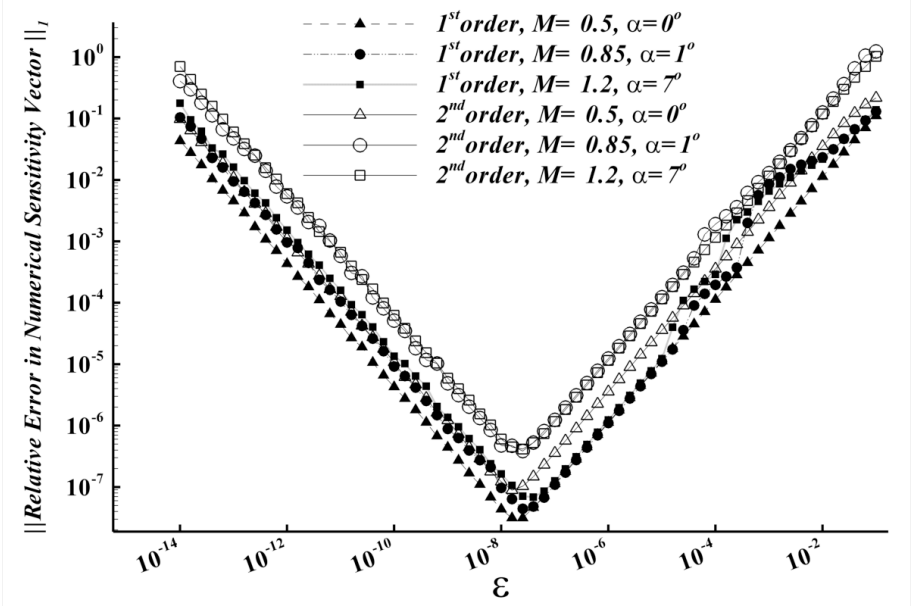

Figure 5.7 *Variation of the relative error in sensitivity by free stream Mach number, NACA0012 geometry*

The effect of the flow regime on the relative error of sensitivities is investigated both for the flow over bump and flow over airfoil cases. For flow over Ni-bump case, the inlet Mach number and for the flow over NACA0012 airfoil case the free stream Mach number is altered. The resulting changes in the relative error of the sensitivities are presented in Figure 5.6 and 5.7. The optimum finite difference perturbation magnitude which results in minimum relative sensitivity error is found to be identical in all cases. Moreover the calculated relative error almost equals to the value that is estimated by Equation 4.20. In Table 5.1, the condition number was presented to be smaller for subsonic flow regime. Similar results were also shown by the Figures 5.6 and 5.7.

For all the cases investigated above, it is seen that magnitude of the relative sensitivity error varies by the finite differencing perturbation magnitude used in Jacobian evaluation. In all cases, relative sensitivity error is minimized by the identical perturbation magnitude, which can be efficiently estimated by equation 4.20. Analyzing the effects of the grid resolution, flux evaluation technique, order of discretization and the flow regime on the variation of the sensitivity error it is found that, the increase in the order of discretization amplified the error in all cases nearly by an order of magnitude, whereas no significant outcome was observed from the other factors. For all cases the magnitude of the upper bound of relative error was estimated by the condition number definition. The variation in the both of the condition number and the relative sensitivity error was found to be consistent. However, the magnitude of the calculated relative sensitivity error is found to be significantly smaller relative to the estimated upper bound. Those results show that although the error in the Jacobian matrix affects the sensitivity error and it is possible to minimize the sensitivity error by the utilization of the most accurate numerical Jacobian, the magnitude of the resulting relative error is still small even for the usage of perturbation magnitudes significantly larger or smaller than the optimum one.

CHAPTER 6

INVERSE DESIGN

6.1 Introduction

The efficiency of the aerodynamic design optimization depends on many factors. The choice of the design parameterization technique and number of the design variables used in the parameterization significantly affects the quality of the design and the efficiency of the optimization procedure. However to perform efficient design optimization, focusing on the quality of the parameterization is not sufficient alone. If the optimization is applied via gradient based methods, accurate evaluation of sensitivities and the accurate modeling of the flow domain are also critical as much as the effort made on the parameterization. In previous chapters the accuracy study performed both for the flow solver and the sensitivity analysis was presented. The main focus of this chapter is the investigation of the effect of the errors in the numerical Jacobian evaluation on the efficiency of the aerodynamic design optimization. To perform this objective inverse design of an airfoil is practiced. Various shape functions are used in the geometry parameterization. The weightings of the components of the shape function are defined as the design variables. Both of the analytical and numerical Jacobian matrices are used to evaluate the sensitivities of the surface pressure that are required in the inverse design optimization. The variation of the performance of the inverse design by the Jacobian evaluation method is analyzed. The comparison of the design results evaluated with the numerical Jacobian matrices that are evaluated by various finite difference perturbation magnitudes is presented.

6.2 Inverse Design Optimization

The objective of an inverse design optimization is to obtain the geometry which produces a specified pressure distribution, under specified operating conditions. The design process starts from initial baseline geometry and perturbations will be applied on it iteratively till the desired pressure distribution is obtained. The iterative design optimization process will be speeded up by the usage of an initial guess which has pressure distribution already close to the targeted one. Figure 6.1 presents the numerical procedure followed for the inverse design optimization.

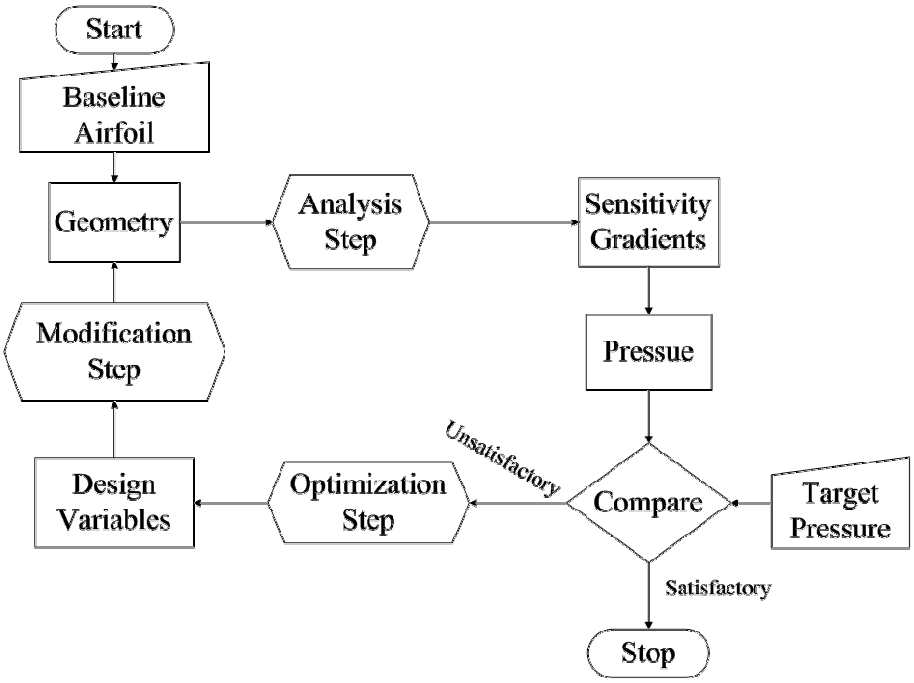

Figure 6.1 *Flow chart of inverse design optimization process*

In the design process the flow analysis step will evaluate the pressure distribution of the latest geometry and it will be followed by the comparison step to check the closeness to the targeted pressure distribution. If this check is failed new design cycle will start with sensitivity analysis which gives the variation of the pressure distribution due to each geometry parameter perturbation. The geometry is modified by addition of smooth perturbations and the weighting of the each perturbation is determined through an optimization process. The design cycle will be completed by the flow analysis step

applied on the geometry which was modified according to the information obtained from the optimization step.

6.2.1 Design Variables

The aerodynamic geometry is modified by adding a smooth perturbation, Δy, which is defined as linear combinations of the shape functions, f_k as below:

$$\Delta y(x) = \sum_{k=1}^{K} \beta_k f_k(x) \tag{6.1}$$

where x is the normalized chordwise position, f_k is the shape function and K is the number of the shape functions. The design variables are defined as the weightings of the each shape functions and they are symbolized as β_k in the Equation (6.1). The design variables mainly control the amount of the perturbation that will be applied on the geometry by each shape functions. Variety of shape functions are used to examine the affect of the geometry parameterization on the efficiency of the design cycle. Hicks-Henne functions, Wagner functions, Legendre polynomials and the patched polynomials are used in the study. Detailed explanations and formulations on the functions and polynomials used in the shape parameterization are given in Appendix-B.

6.2.2 Least Squares Minimization Optimization

In this study, a least-square optimization method is used in the optimization step to minimize the discrepancy between the pressure of the target and the designed airfoil. The formulation for that objective function is given below:

$$F_{OBJ}(\beta^n) = \sum_{j=1}^{J} \left[P_j^T - P_j(\beta^n) \right]^2 \Delta S_j(\beta^{n-1}) \tag{6.2}$$

In equation 6.2 j is the location where the pressure is evaluated and ΔS_j is the length of the surface element of the designed airfoil that provides a proper scaling for the pressure values which is necessary due to the varying size of computational cells. ΔS_j is evaluated

at design cycle n-1 hence, it is viewed as a constant in the differentiation of $F(\beta^n)$ with respect to design variables β_i^n .There are total of *J* elements on the airfoil. The pressure distribution of the target airfoil and the pressure distribution of designed airfoil at design cycle n are denoted by P_j^T and $P_j(\beta^n)$ respectively.

The pressure distribution at design cycle n can be evaluated by the Taylor series expansion as below:

$$P_j(\beta^n) = P_j(\beta^{n-1}) + \sum_{i=1}^{I} \frac{\partial P_j(\beta^{n-1})}{\partial \beta_i} \left(\beta_i^n - \beta_i^{n-1} \right) \tag{6.3}$$

where, the $\partial P_j / \partial \beta_i$ are the sensitivities. If Equation 6.3 is substituted into the Equation 6.2 the objective function can be approximated as

$$F(\beta^n) = \sum_{j=1}^{J} \left[P_j^T - P_j(\beta^{n-1}) - \sum_{i=1}^{I} \frac{\partial P_j(\beta^{n-1})}{\partial \beta_i} \left(\beta_i^n - \beta_i^{n-1} \right) \right]^2 \Delta S_j(\beta^{n-1}) \tag{6.4}$$

The minimization condition with respect to β_i^n yields

$$\frac{\partial F}{\partial \beta_k^n}(\beta^n) = 0 \tag{6.5}$$

Substituting Equation(6.4) into equation(6.5) gives:

$$2\sum_{j=1}^{I} \left[P_j^T - P_j(\beta^{n-1}) - \sum_{i=1}^{I} \frac{\partial P_j(\beta^{n-1})}{\partial \beta_i} \left(\beta_i^n - \beta_i^{n-1} \right) \right] \frac{\partial P_j(\beta^{n-1})}{\partial \beta_k} \Delta S_j(\beta^{n-1}) = 0 \tag{6.6}$$

for *j=1,K*. This formulation can be rewritten as:

$$\sum_{i=1}^{I} \left[\sum_{j=1}^{J} \frac{\partial P_j(\beta^{n-1})}{\partial \beta_k} \frac{\partial P_j(\beta^{n-1})}{\partial \beta_i} \Delta S_j(\beta^{n-1}) \right] \left(\beta_i^n - \beta_i^{n-1} \right) = \sum_{j=1}^{J} \frac{\partial P_j(\beta^{n-1})}{\partial \beta_k} \left[P_j^T - P_j(\beta^{n-1}) \right] \Delta S_j(\beta^{n-1}) \tag{6.7}$$

Solving the Equation 6.7 the weighting of the shape functions, β_i will be determined.

In order to judge the design quality and to monitor the convergence of the design cycle, a convergence parameter CP is defined. This parameter is based on the root-mean square of length weighted pressure discrepancies between the target pressure and the pressure of the designed geometry.

$$CP = \left(\frac{\sum_{i=1}^{I} \left(P_i^T - P_i \right)^2}{\sum_{i=1}^{I} \Delta S_i} \right)^{\frac{1}{2}} \Delta S_i \tag{6.8}$$

where P_i^T is the targeted pressure the P_i is evaluated pressure on the point i. ΔS_i is the length of the surface element I is the number elements on the geometry.

In previous chapter the technique applied for the evaluation of the flow variable sensitivities were given. Those sensitivities will be used to derive the surface pressure sensitivities required by the aerodynamic design optimization, which is formulated above by Equations 6.1 to 6.8.

$$\begin{aligned} P &= (\gamma - 1) w_4 - \frac{w_2^2 + w_3^2}{2 w_1} \\ \frac{\partial P}{\partial \beta} &= (\gamma - 1) \left[\frac{\partial w_4}{\partial \beta} - \frac{w_2 \frac{\partial w_2}{\partial \beta} + w_2 \frac{\partial w_2}{\partial \beta}}{w_1} + \frac{w_2^2 + w_3^2}{2 w_1^2} \frac{\partial w_1}{\partial \beta} \right] \end{aligned} \tag{6.9}$$

6.3 Results

The pressure distribution generated by the RAE 2822 airfoil was chosen as target pressure and the inverse design is applied by using the NACA 0012 airfoil as the initial geometry. The operating conditions are chosen as free-stream flow by 0.73 Mach with

2.78 degrees angle of attack. The flow analysis is performed by 2nd order AUSM scheme with the usage of Van Albada limiter. Flow domain is discretized by the C-type grid which has 129x33 node points. Each shape functions presented in the previous section is examined in the same inverse design problem to observe the effect of the geometry parameterization on the design.

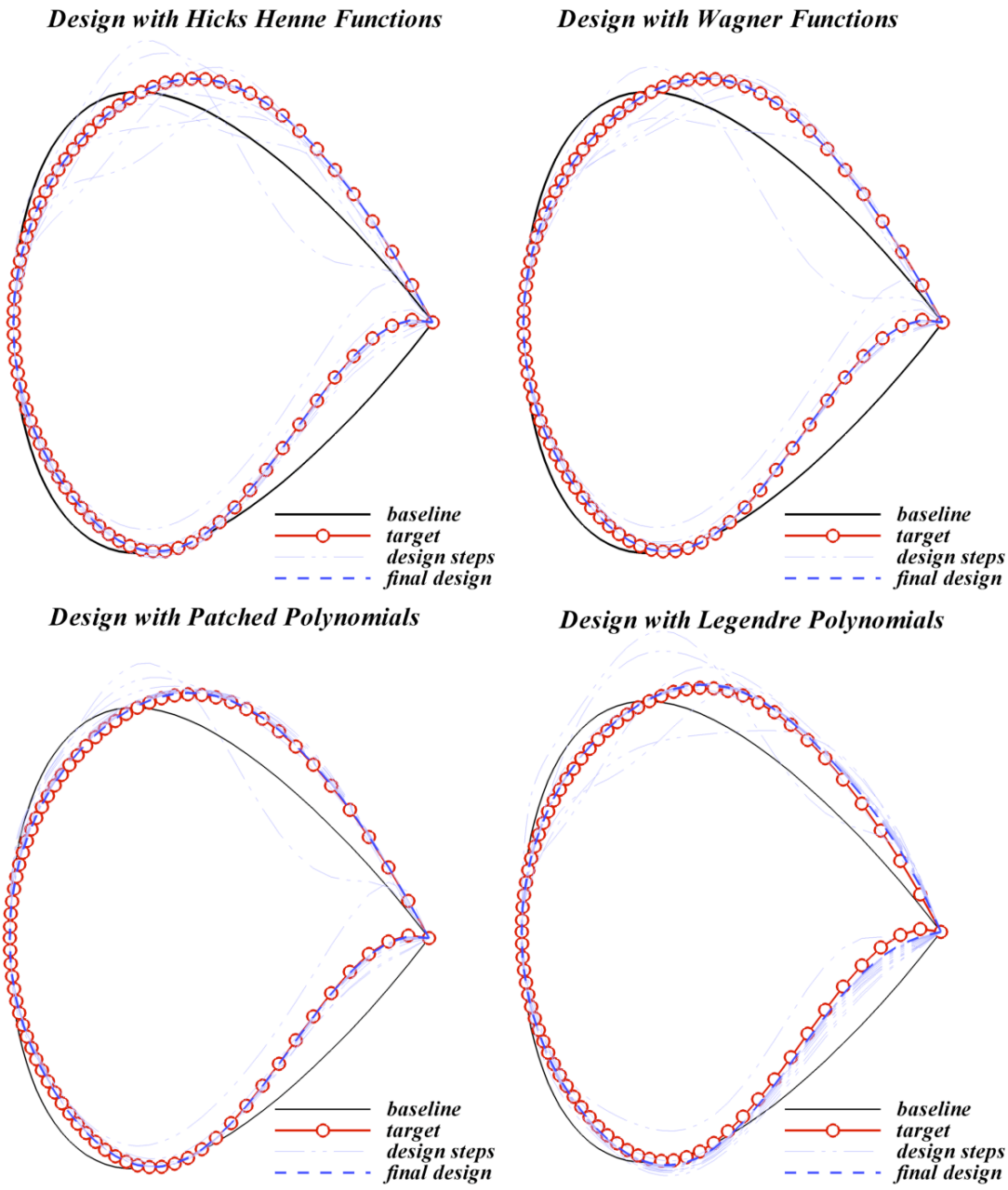

Figure 6.2 *Inverse Design of RAE-2822 airfoil by using NACA0012 as baseline*

Figure 6.2 presents the evolution of the airfoil geometry throughout the inverse design cycle. The targeted geometry is presented by red solid line, initial geometry is presented by black solid line and the geometry attained by design is presented by green dashed lines.

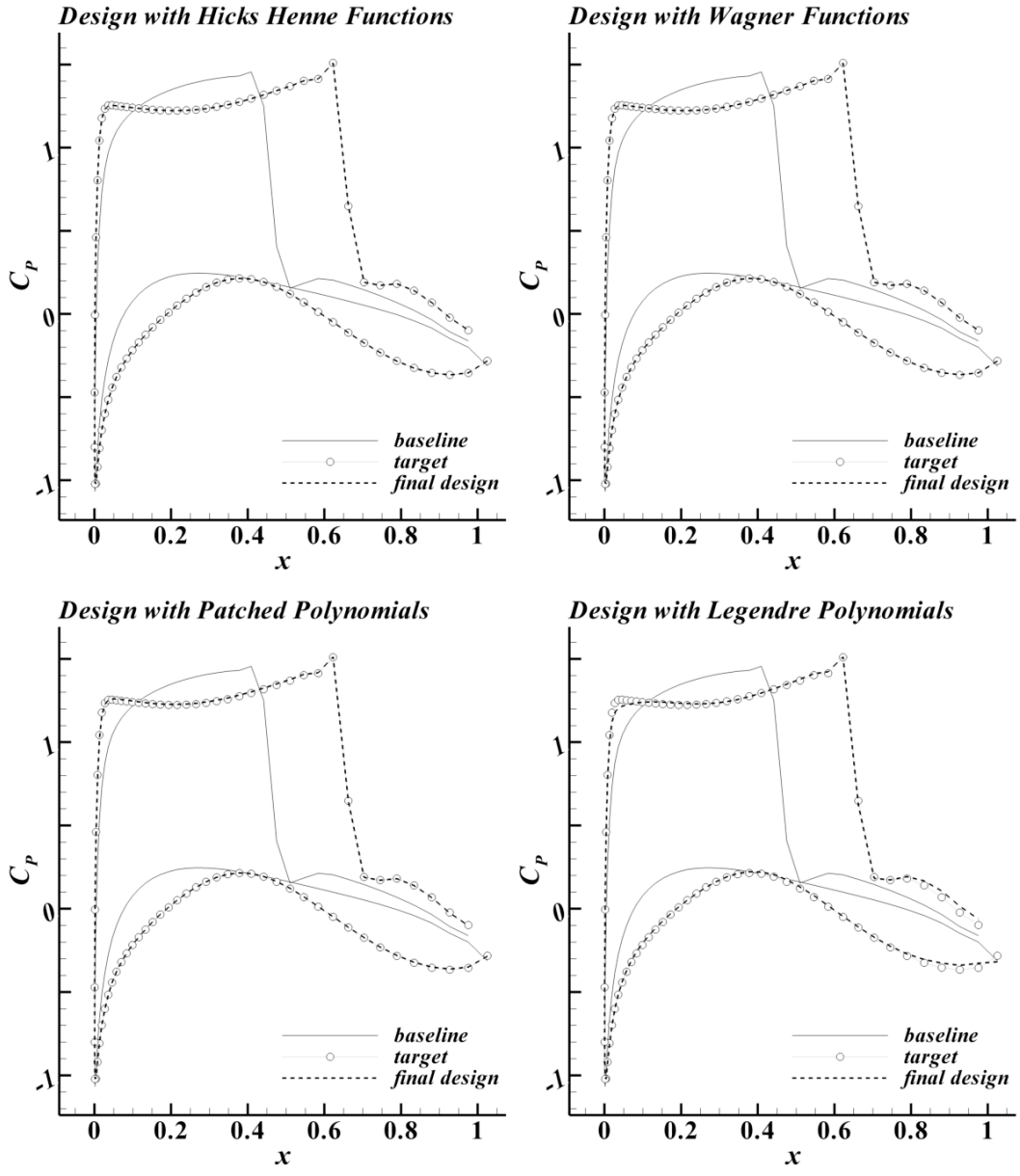

Figure 6.3 *Change of pressure coefficient distribution throughout the inverse design*

Figure 6.3 presents the evolution of the pressure coefficient distribution throughout the inverse design cycle. The targeted C_p is presented by symbols, initial C_p is presented by solid line and the C_p attained by the design is presented by dashed lines

Figures 6.2 and 6.3 shows that, performing the geometry parameterization by each of the patched polynomials, Hicks-Henne's and Wagner's shape functions, the objective of the design is successfully accomplished. However the utilization of Legendre polynomials is found to be insufficient to capture the targeted pressure distribution at the vicinity of trailing edge. The chordwise distribution of the Legendre polynomials causes that insufficiency. As it can be seen by the formulations and figures given in Appendix-B the Legendre polynomials have their peak values in the vicinity of leading edge and their values get smaller near the trailing edge. Hence the smooth perturbations defined by Legendre polynomials are ineffective in the vicinity of the trailing edge.

The inverse design results given in Figures 6.2 and 6.3 are evaluated by using analytical Jacobian matrix in the sensitivity evaluation. The same inverse design is also performed by the usage of numerical Jacobian matrices, to investigate the accuracy of the numerical Jacobian matrix on the design cycle.

The variation of the surface pressure sensitivities relative to finite difference perturbation magnitude is given in Figures 6.4 to 6.7 for each geometry parameterization technique. In those figures, variations correspond to the 2^{nd}, 5^{th}, 9^{th} and 12^{th} design variables are given. The design variables 2 and 5 modify the upper surface whereas 9 and12 are effective on lower surface. The results presented in those figures shows that finite difference perturbation magnitude used in the Jacobian evaluation almost have no effect on the sensitivities of the surface pressure. Almost identical sensitivities are attained for the wide range of perturbation magnitudes used. The reason behind that can be extracted from the error plots given in previous chapter. In those plots it can be seen that the relative error in the flow variable sensitivities are very small even for the perturbation magnitudes significantly larger or smaller than the optimum one. For example in Figure 5.1, usage of any perturbation magnitude from 10^{-4} to 10^{-12} resulted in a relative error lower than 10^{-2}.

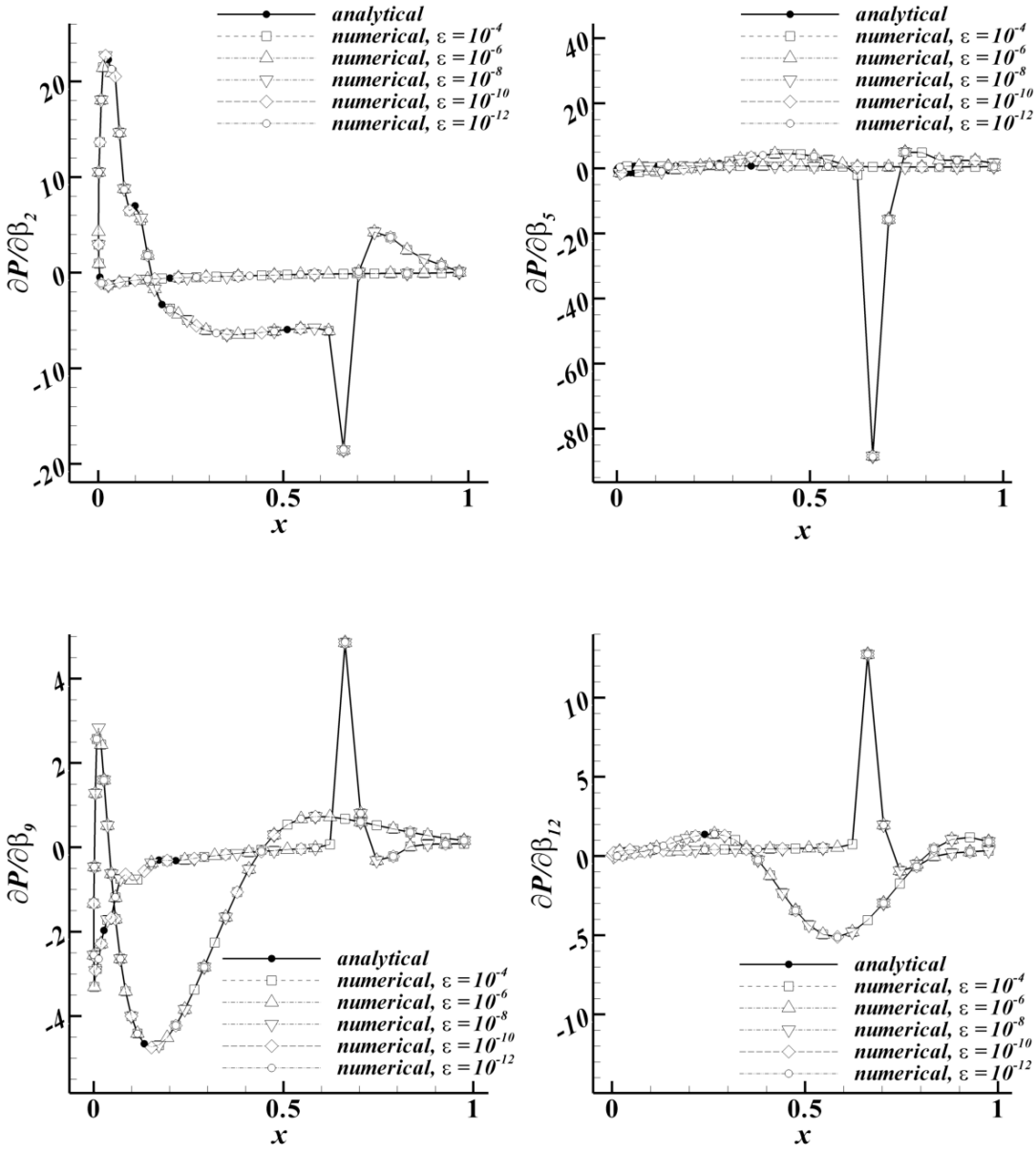

Figure 6.4 *Comparison of pressure sensitivities evaluated by analytical and numerical Jacobian matrices (Hicks Henne functions)*

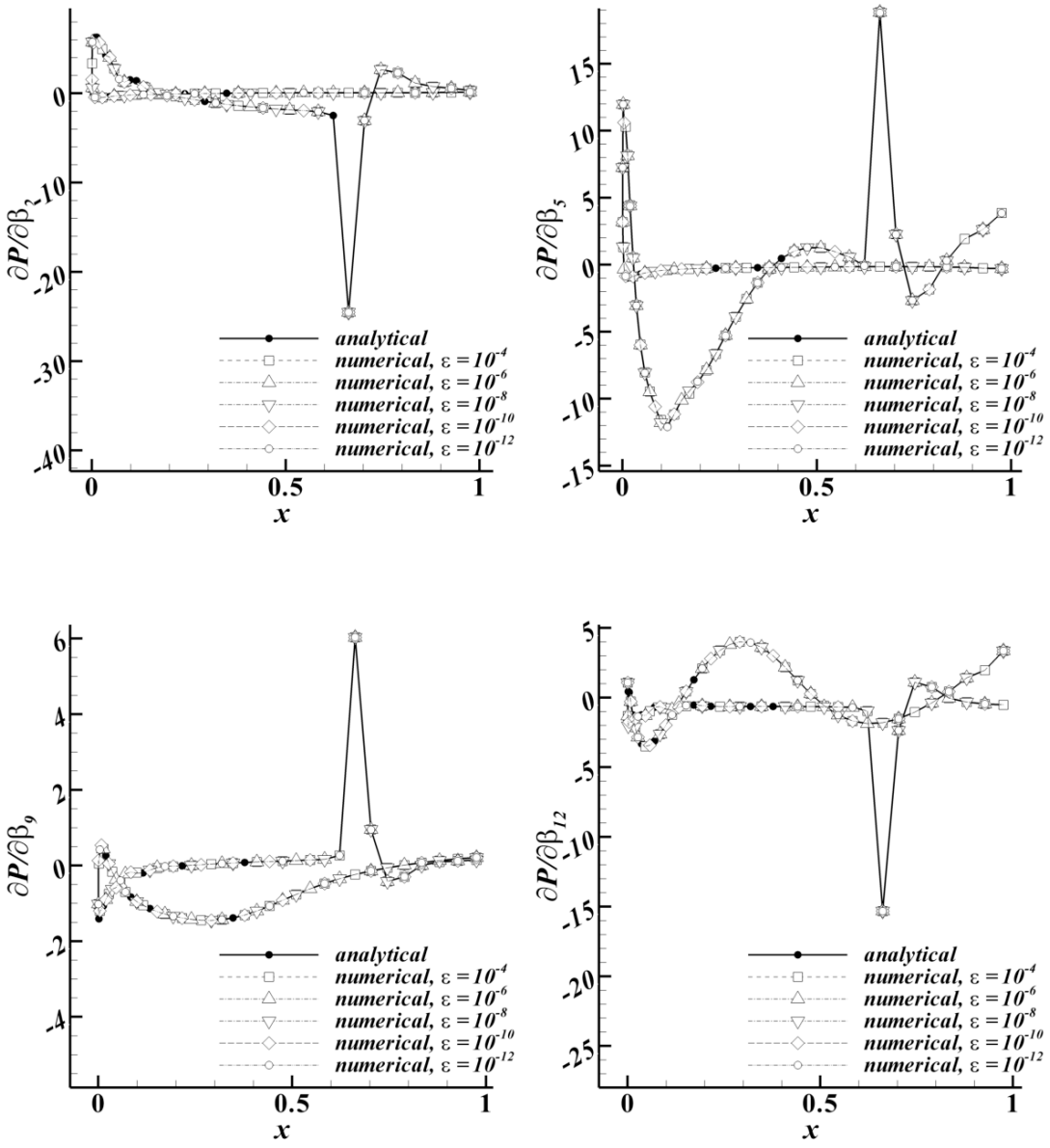

Figure 6.5 *Comparison of pressure sensitivities evaluated by analytical and numerical Jacobian matrices (Wagner functions)*

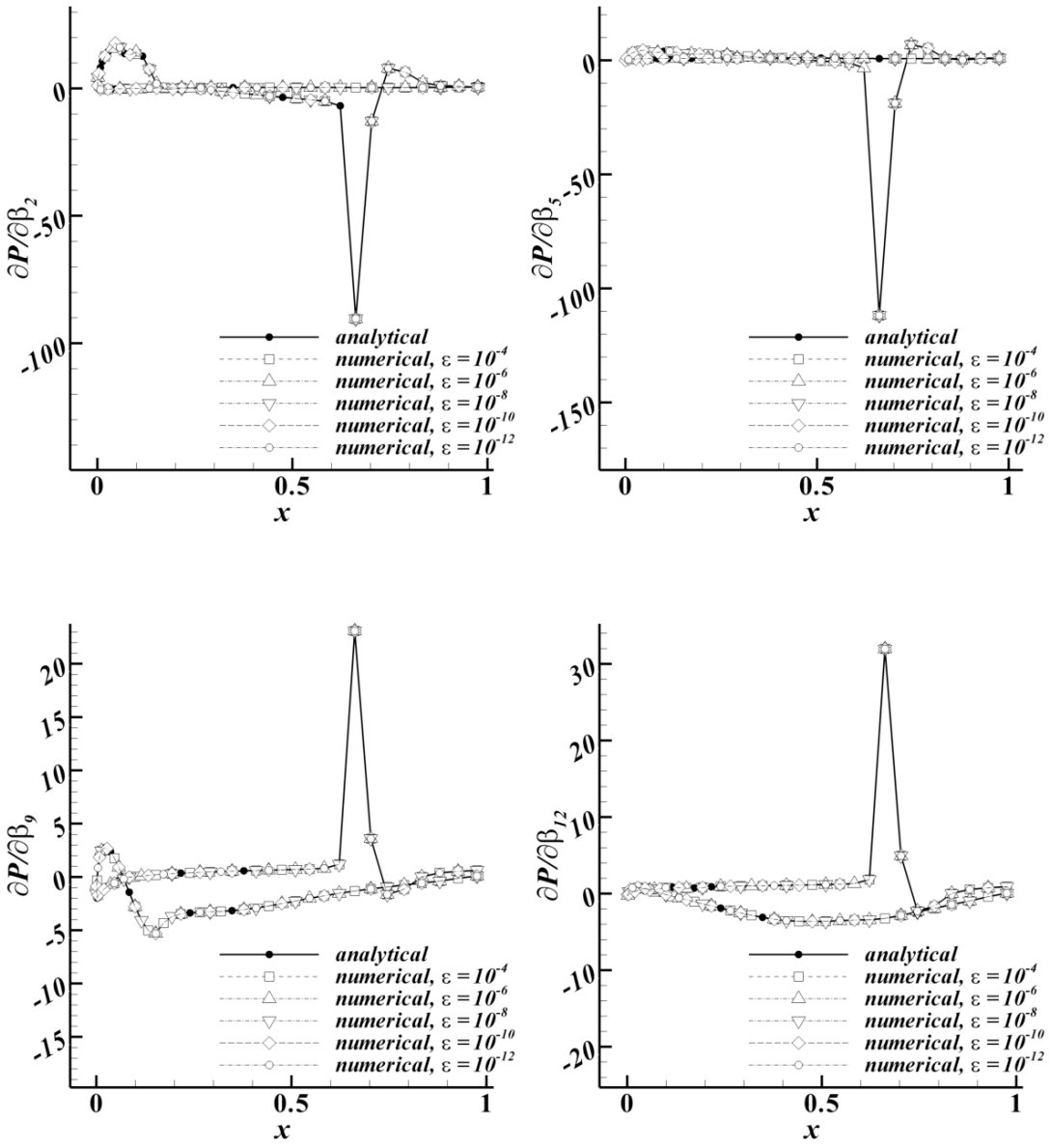

Figure 6.6 *Comparison of pressure sensitivities evaluated by analytical and numerical Jacobian matrices (Patched polynomials)*

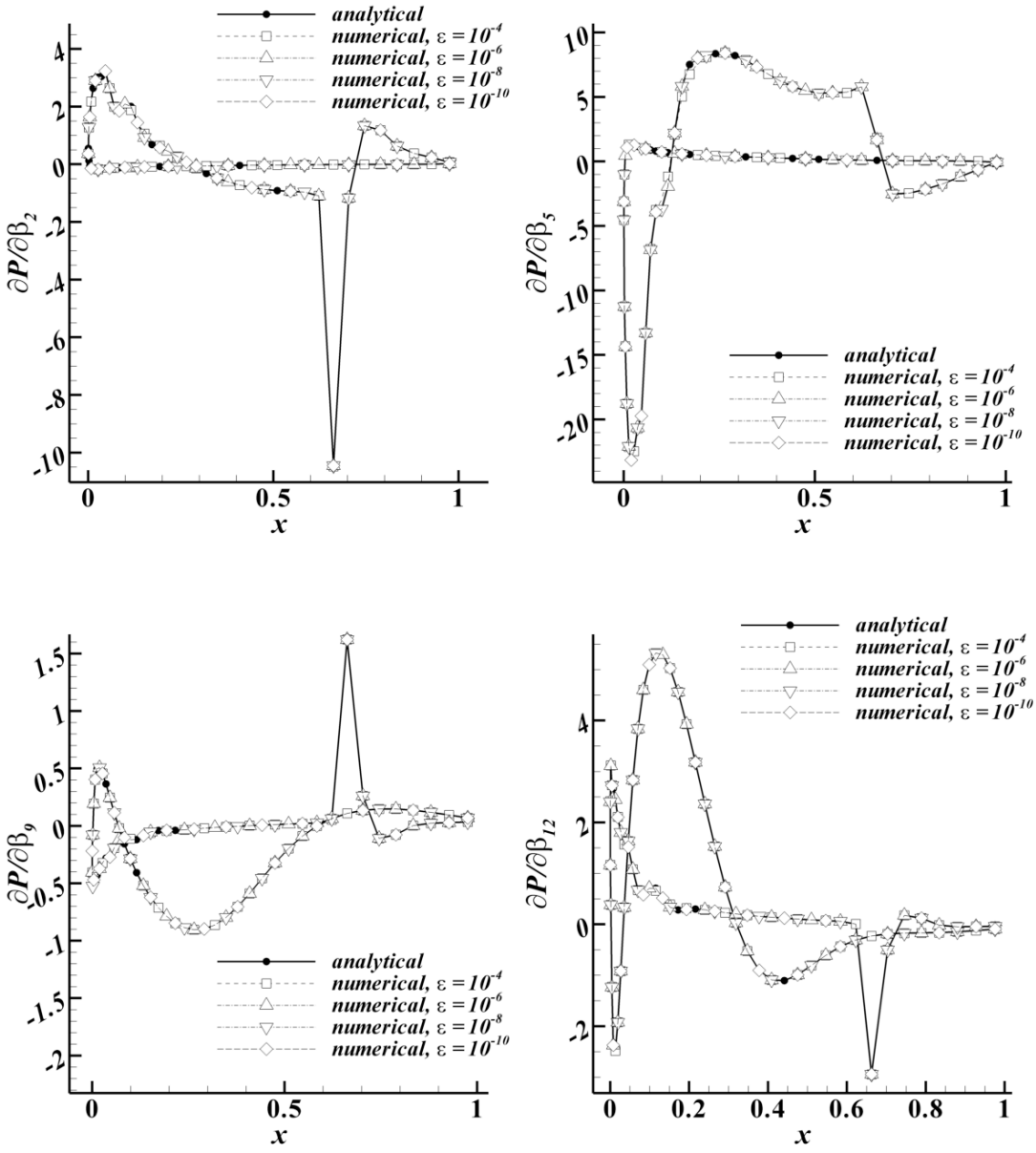

Figure 6.7 *Comparison of pressure sensitivities evaluated by analytical and numerical Jacobian matrices Legendre polynomials)*

Finally the effect of the finite difference perturbation magnitude on the efficiency of the design cycle is studied. The variations of the convergence parameter of the designs, which are performed with different numerical Jacobian matrices, are presented in Figure 6.8. The plots presented shows that, finite difference perturbation magnitude does not

affect the convergence and design objective is accomplished by the equal number of design cycles.

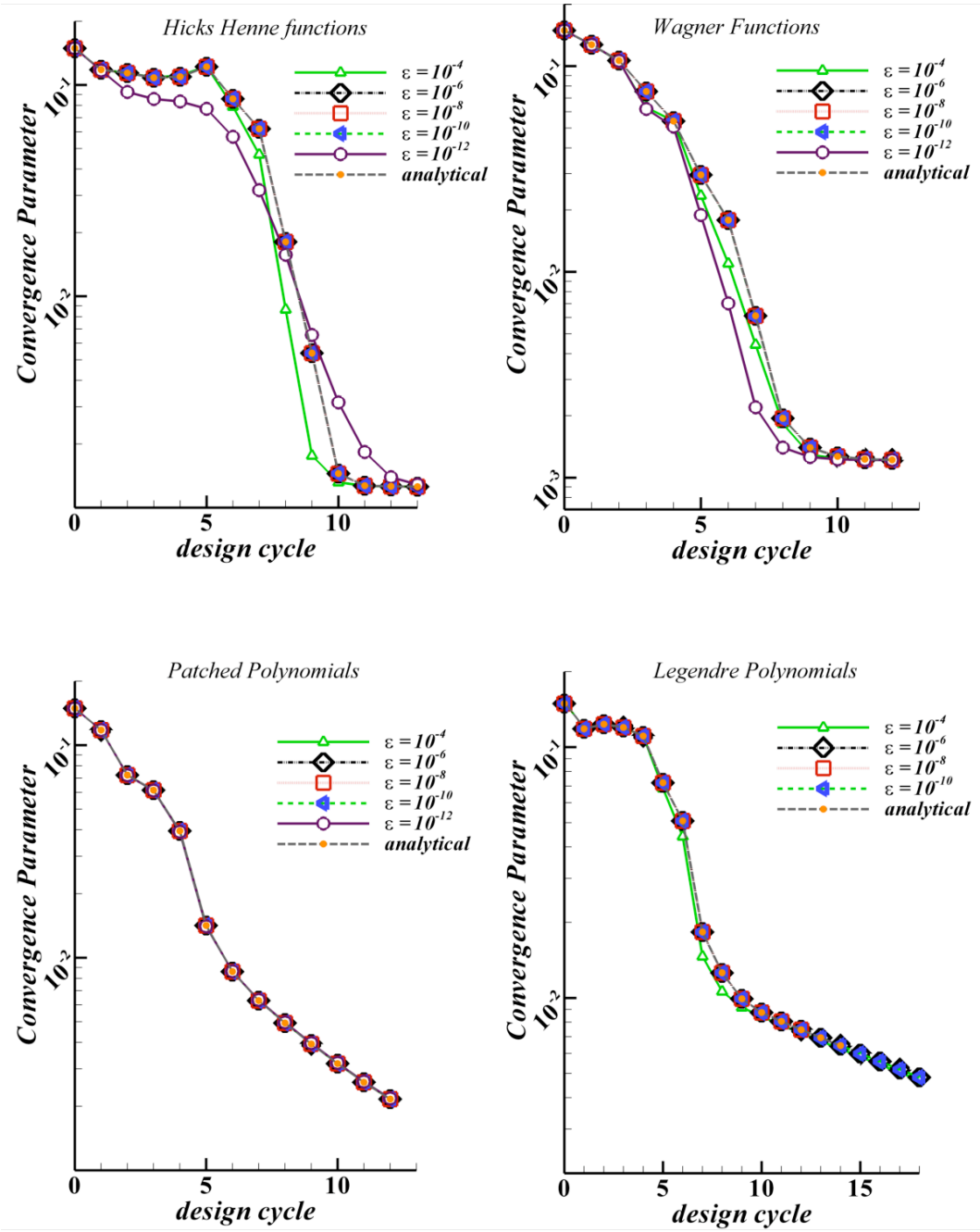

Figure 6.8 *History of convergence parameter of inverse design*

CHAPTER 7

CONCLUSION

In this thesis study, the Newton's method was successfully applied for the solution of the two dimensional Euler equations. The results are generated by the code developed throughout the study. The finite volume method is used in the discretization of the governing equations. Fluxes are evaluated by the upwinding methods and high order accuracy in the spatial discretization is attained by MUSCL interpolation. The steady flow solutions were performed for the varying flow regimes. The Jacobian matrix required by the Newton's method was evaluated both analytically and numerically. The UMFPACK, unsymmetrical multifrontal sparse matrix solver is utilized to factorize the Jacobian matrix, which is highly sparse. The gradient based optimization method is applied for the inverse aerodynamic design. The gradients are evaluated analytically by the sensitivity analysis using the direct differentiation method.

The one of the main objective of this thesis study was to analyze the accuracy of the numerically evaluated flux Jacobian. The difference between the numerical and the analytical Jacobian matrices is defined as error in the study. Detailed error analysis is performed by considering the effects of the finite difference perturbation magnitude. The variation of the error is examined for variety of flow solution cases by changing the discretization technique, grid resolution, geometry and the flow regime. Simple formula is derived for the approximation of the optimum perturbation magnitude which minimizes the error in the numerical Jacobian. For all cases the derived formula approximated the optimum finite difference perturbation magnitude successfully. The effect of the limiters, which are used in the high order discretization, on the accuracy of the Jacobian matrix is examined. Results showed that to deactivate the limiter in the smooth flow regions, proper tuning on the limiter function is required. Significant enlargement occurred in the magnitude of the second derivatives of the flux vector, in the cases where the limiter was not tuned properly. The enlargement of the second derivatives caused significant shift in the optimum perturbation magnitude.

The convergence of the flow solution by the Newton's method is analyzed. The Newton's method provides quadratic convergence as long as a good initial guess is provided. The presented results show that, the number of iterations required for the convergence will be tremendous or solution may even diverge if a poor initial guess is used. To strengthen the initial guess, time like terms are added to the diagonal of the Jacobian matrix in initial iterations. The initial and the withdrawal values of those added diagonal terms are found to be most critical factors on the convergence of the solution. Another factor which directly affects the convergence directly is the consistency between the schemes used in the Jacobian and the residual evaluations. It is found that, if the linearization of the Jacobian is not performed exactly the quadratic rate can not be achieved anymore and the iterations required for the converged solution will drastically increase. The effect of the accuracy of the Jacobian matrix on the convergence is also analyzed. Compared to effects of initial conditioning and the exactness of the linearization, the effect of numerical Jacobian accuracy on the convergence is found to be negligible. To sum up the studies conducted on the convergence, it can be said that, the convergence of the Newton's method is insensitive to the errors in the numerical Jacobian and quadratic convergence can be achieved independently from the Jacobian evaluation technique as long as the initial conditioning and the linearization of the matrix is applied properly.

The effects of the accuracy of the numerical Jacobian matrix on the accuracy of the flow variable sensitivities are analyzed. The sensitivities evaluated by analytical Jacobian is compared with one ones evaluated by numerical Jacobian matrices to define the error. The upper bound of the relative error is estimated by condition number. The actual relative error was found to be significantly smaller than that approximated bound. Although the accuracy of the sensitivities varied by the finite differencing perturbation magnitude used in Jacobian evaluation, the relative error was found to be very small even for large and very small perturbation magnitudes.

The effects of the sensitivity accuracy on the efficiency of the inverse design optimization were analyzed. Due to very small relative sensitivity error, identical

designs and design convergence were obtained from the usage of numerical and analytical Jacobian.

To conclude, the numerical evaluation of the Jacobian matrix can be very advantageous since the occurring relative error is very small and do not introduces any practical penalty. Moreover using the numerical evaluation, very complex schemes can be linearized effortlessly compared to the analytical evaluation. The only penalty of the numerical Jacobian usage is the increase in the CPU time spent.

REFERENCES

[1] Wigton, L B., "Application of MACSYMA and Sparse Matrix Technology to Multi-element Airfoil Calculations", AIAA Paper 87-1142, 1987.

[2] Bender, E.E and Kosla, P.K., "Application of Sparse Matrix Solvers and Newton's Method to Fluid Flow Problems", AIAA Paper 88-3700, 1988.

[3] Venkatakrishnan, V., "Newton Solution of Inviscid and Viscous Problems", *AIAA Journal*, Vol. 27, July 1989, pp. 885-891.

[4] Van Dam, C. P., M. Hafez and J. Ahmad, "Calculations of viscous flow with separation using Newton's method and direct solver", *AIAA Journal*, Vol. 28, No. 5, 1990, pp. 937-939.

[5] Orkwis, P. D., *A Newton's method Solver for the Two-Dimensional and Axisymmetric Navier-Stokes Equations*, Ph.D. Dissertation, North Carolina State University, Raleigh, NC, 1990.

[6] Orkwis, P. D., and McRae, D. S., "Newton's method Solver for High Speed Viscous Separated Flowfields", *AIAA Journal*, Vol. 30, January 1992, pp. 78-85.

[7] Orkwis, P. D., and Mc Rae, D. S., "Newton's Method Solver for High Speed Viscous Separated Flowfields", *AIA Journal*, Vol. 30, January 1992, pp. 1507-1514.

[8] Orkwis, P. D., "Comparison of Newton's and Quasi-Newton's Method Solvers for the Navier-Stokes Equations", *AIAA Journal*, Vol. 31, May 1993, pp. 832-836.

[9] Kim, D. B., and Orkwis, P. D., "Jacobian Update Strategies for Quadratic and Near-Quadratic Convergence of Newton and Newton-Like Implicit Schemes", AIAA Paper 93-0878, Proceedings of the AIAA 31st Aerospace Sciences and Meeting&Exhibit, Reno, Nevada, January 1993.

[10] Whitfield, D. L., and Taylor, L. K., “Discretized Newton-Relaxation Solution of High Resolution Flux-Difference Split Schemes”, AIAA Paper 91-1539, 1991.

[11] Vanden, K. J., *Direct and Iterative Algorithms for the Three-Dimensional Euler Equations,* Ph.D. Dissertation, Mississippi State University, Mississippi, 1992.

[12] Vanden, K. J., and Whitfield, D. L, “Direct and Iterative Algorithms for the Three-Dimensional Euler Equations”, AIAA Paper 93-3378, 1993.

[13] Orkwis,P.,D., and Venden, K.,J., ”On the Accuracy of Numerical Versus Analytical Jacobians”, AIAA Paper 94-0176, Proceedings of the AIAA 32nd Aerospace Science Meeting, Reno, Neveda1994, January 1994.

[14] Saad, Y., and Schultz, M.H., “GMRES: A Generalized Minimal Residual Algorithm For Solving Non-Symmetric Linear Systems”, *SIAM Journal on Scientific and Statistical Computing*, Vol. 7, July 1986.

[15] Venkatakrishnan, V., and Mavripilis, D.J., “Implicit Solvers for Unstructured Meshes”, AIAA-91-1537-CP, 1991.

[16] Venkatakrishnan, V., “Implicit Schemes and Parallel Computing in Unstructured Grid CFD”, ICASE report 95-28 CR-195071, NASA, 1995

[17] Rogers, S.E., “A Comparison of Implicit Schemes for the Incompressible Navier Stokes Equations with Artificial Compressibility”, AIAA 95-0567, January 1995

[18] Forsyth, P.A., and Jiang, H., “Iterative Methods for Full Newton Solution of the Euler Equations”, Sixth International Symposium on Computational Fluid Dynamics, pp. 318-323 , Lake Tahoe, Nevada, September 1995

[19] Brown, P., and Saad, Y., “Convergence Theory of Nonlinear Newton-Krylov Algorithms”, *SIAM J. Optimization*, Vol. 4., pp. 297-330

[20] Nielsen, E. J., Anderson, W.K., Walters, R.W., and Keyes, D.E., "Application of Newton-Krylov Methodology to Three-Dimensional Unstructured Euler Code", AIAA 95-1733-CP, June 1995

[21] Barth, T. J., and Linton, S.W., "An unstructured Mesh Newton Solver for Compressible Fluid Flow and Its Parallel Implementation", AIAA 95-0221, January 1995

[22] Davis, T. A., "UMFPACK Version 4.1 User Manual", University of Florida, Florida, 2003.

[23] Amestoy, P. R., Duff, I. S. and L'Excellent, J. -Y., "Multifrontal parallel Distributed Symmetric and Unsymmetric Solvers", *Computer Methods in Applied Mechanics and Engineering*, Vol. 184, Issues 2-4, 14 April 2000, pp. 501-520.

[24] Gupta, A., "Recent Advances in Direct Methods for Solving Unsymmetric Sparse Systems of Linear Equations", ACM Transactions on Mathematical Software, Vol. 28(3), 2002, pp.301–324.

[25] Schenk, O. and Gärtner, A., "Solving Unsymmetric Sparse Systems of Linear Equations with PARDISO", *Journal of Future Generation Computer Systems*, Vol.20, 2004, pp. 475-487.

[26] Onur, O. and Eyi, S., "Effects of the Jacobian Evaluation on Newton's Solution of the Euler Equations", *International Journal for Numerical Methods in Fluids*, Vol. 49, pp 211-231, 2005.

[27] Gelfgat, A. Y., "Stability of Convective Flows in Cavities: Solution of benchmark problems by a low-order finite volume method", *International Journal For Numerical Methods in Fluids*, 2007, Vol.53, pp. 485-506

[28] Raju,M.P, Tien, J.S. "Development of Direct Multifrontal Solvers for Combustion Problems", *An International Journal of Computation and Methodology*, 1521-0626, Vol. 53, Issue 3, 2008, pp. 189 – 205

[29] Sobieski, J. S., "Sensitivity of Complex, Internally Coupled Systems", *AIAA Journal*, Vol. 28, No. 1, January 1990, pp. 153-160.

[30] Haug, E.J., Choi, K.K. and Komkov, V., Design Sensitivity Analysis of Structural Systems, Academic Press, New York, 1986.

[31] Haftka, R. T. and Adelman H. M., "Recent Developments in Structural sensitivity Analysis", *Structural Optimization,* Vol. 1, 1989, pp. 137-151

[32] Haftka, R. T. and Grandhi, R. V., "Structural Shape Optimization-A Survey", *Computer Methods in Applied Mechanics*, Vol. 57, 1986, pp. 91-106.

[33] Frank, D. P., and Shubin, G. R., "A Comparison of Optimization-Based Approaches for a Model Computational Aerodynamic Design Problem", *Journal of Computational Physics*, Vol. 98, 1992, pp. 74-89

[34] Narducci, B., Grossmann, B. and Haftka, R. T., "Sensitivity Algorithms for Inverse Design Problem Involving a Shock Wave, "AIAA Paper 94-0096, January 1994.

[35] Rizk, M. H. "Optimization by Updating Design parameters as CFD Iterative Solutions Evolve", Symposium on Multidisciplinary Application of Computational Fluid Dynamics, [Ed: O. Baysal], FED-Vol. 129, ASME- Winter Annual Meeting, December 1991, pp. 51-62.

[36] Verhoff, A., Stookesberry, D. and Cain A., "An Efficient Approach to optimal Aerodynamic Design Part 1: Analytic Geometry Aerodynamic Sensitivity", AIAA Paper 93-0099, January 1993

[37] El-banna, H. and Carlson, L., "Determination of Aerodynamic Sensitivity Coefficients in the Transonic and Supersonic Regimes," AIAA Paper 89-0532, January 1989.

[38] Baysal, O., Eleshaky, M. E. and Burgereen, G. W., "Aerodynamic Shape Optimization Using Sensitivity Analysis on Third-Order Euler Equation", Journal of Aircraft, Vol. 30, No. 6, November-December 1993.

[39] Korivi, V. M., Taylor, A. C., III, Newman, P. A., Hou, G. H., and Jones, H. E., "An Approximately factored Incremental Strategy for Calculating Consistent Discrete Aerodynamic Sensitivity Derivatives", AIAA Paper 92-4746-CP, Proceedings, 4th AIAA/NASA/OAI Multidisciplinary Analysis and Optimization Conference. 1992, pp. 465-478

[40] Jameson, A., "Aerodynamic Design Via Control Theory", NASA-CR-181749, 1988.

[41] Newman III, J. C., Taylor III, A.C., Barnwell, R. W., Newman, P. A., Hou, G. J.-W., "Overview of Sensitivity Analysis and Shape Optimization for Complex Aerodynamic Configurations", *Journal of Aircraft*, 1999, 0021-8669, Vol.36, no.1, pp.87-96

[42] Volpe, G., "Inverse Design of Airfoil Contours: Constraints, Numerical Method, and Applications", Computational Methods for Aerodynamic Design and Optimization, AGARD Conference Proceedings No. 463, May, 1989.

[43] Sobieczky, H., "Progress in Inverse Design and Optimization in aerodynamics," Computational Methods for aerodynamic Design (Inverse) and Optimization. AGARD-CP-463, pp. 1-10, May, 1989.

[44] Giles, M. B. and Drela, M., "Two-Dimensional Transonic Aerodynamic Design Method," *AIAA Journal*, Vol. 25, No. 9, September 1987, pp. 1199-1205

[45] Malone, J. B., Narramore, J. C., and Sankar, L. N., "Airfoil Design Method Using the Navier-Stokes Equations," *Journal of Aircraft*, Vol. 28, No 3., March 1991, pp. 216-224.

[46] Birckelbaw, L., “Inverse Airfoil Design Using the Navier Stokes equations”, Proceedings AIAA 7th Applied Aerodynamic Conference, Seattle, WA, July 1989, pp. 346-353

[47] Nemec, M. and Zingg, D. W., “Newton-Krylov algorithm for aerodynamic design using the Navier-Stokes equations,” *AIAA Journal* , Vol. 40, No. 6, June 2002, pp. 1146–1154

[48] Nemec, M., Zingg, D. W., and Pulliam, T. H., “Multipoint and multi-objective aerodynamic shape optimization,” *AIAA Journal* , Vol. 42, No. 6, 2004, pp. 1057–1065

[49] Leung, T., and Zingg, D.W., "Single- and Multipoint Aerodynamic Shape Optimization Using a Parallel Newton-Krylov Approach", AIAA Paper, 2008-3803, 2009.

[50] Eyi, S. and Lee, K., “Aerodynamic Design Via Optimization”, *Journal of Aircfraft*, Vol. 29, No. 6, November-December 1992, pp. 1012-1029.

[51] Eyi, S. and Lee, K. “Effects of Sensitivty Analysis on Airfoil Design” *Proceedings AIAA 36th Aerospace Sciences Meeting and Exhibit,* Reno, NV, January, 1998

[52] Steger, J. L., and Warming, R. F., “Flux Vector Splitting of the Inviscid Gasdynamic Equations with Application to Finite-Difference Methods”, *Journal of Computational Physics*, Vol. 40, 1981, pp. 263-293.

[53] Van Leer, B., “Flux Vector Splitting for the Euler Equations”, ICASE Report 82-30, September 1982.

[54] Liou, M.-S. “A sequel to AUSM: AUSM+”, *Journal of Computational Physics*, Vol 129 (1996), pp. 364–382

[55] Roe, P. L., “Characteristics-Based Schemes for the Euler Equations”, Annual Review of Fluid Mechanics, Vol. 18, 1986, pp. 337-365.

[56] Van Leer, B., “Towards the Ultimate Conservative Difference Scheme, V. A Second Order Sequel to Godunov's Method”, *Journal of Computational Physics*, Vol. 32, 1979, pp. 101–136

[57] Van Albada, G.D., Van Leer, B., Roberts, W.W., “A Comparative Study of Computational Methods in Cosmic Gas Dynamics”, *Astronomy and Astrophysics*, Vol 108, 1982, pp. 76-84,

[58] Venkatakrishnan, V., “Preconditioned conjugate gradient methods for the compressible Navier- Stokes equations”, *AIAA Journal*, Vol. 29, June 1991, pp. 1092-1100.

[59] Blazek J., “Computational Fluid Dynamics: Principles and Applications”, Elsevier, 2005

[60] Hirsch, C., “Numerical Computation of Internal and External Flows”, Vol. I-II, John Wiley & Sons, Chichester, 1990.

[61] Dennis J.E., Schnabel R.B., “Numerical Methods for Unconstrained Optimization and Nonlinear Equations”, Prentice Hall, New Jersey, 1983.

[62] Pulliam, T. H., “Implicit methods in CFD”, Numerical methods for fluid dynamics III,1988

[63] Ni, R. H., “A Multiple-Grid Scheme for Solving the Euler Equations”, *AIAA Journal*, Vol.20, no.11, 1982, pp. 1565-1571.

[64] AGARD Subcommittee C., Test Cases for Inviscid Flow Field Methods, *AGARD Advisory Report 211*, 1986.

[65] Venkatakrishnan, V., “Convergence to Steady Sate Solutions of the Euler Equations on Unstructured Grid with limiters”, *Journal of Computational Physics*, Vol. 118, pp. 120-130.

[66] Gill, P.E., Murray W., and Wright M.H., Practical Optimization, Academic Press, London, 1992.

[67] Balasubramanian, R. and Newman III, J.C., "Discrete direct and adjoint sensitivity analysis for arbitrary Mach number flows" , *International Journal of Numerical Methods in Engineering*, 2006, Vol. 66, no. 2, pp. 297-318.

[68] Radespiel, R., and Kroll, N., "Accurate Flux Vector Splitting for Shocks and Shear Layers", *Journal of Computational Physics*, Vol. 121, 1995, pp 66-78.

[69] Godlewski, E., and Raviart, P.-A., Numerical Approximation of Hyperbolic Systems of Conservation Laws, Springer, New York,1996.

[70] Rizk, M. H., "The Use of Finite-Differenced Jacobians for Solving the Euler Equations for Evaluating Sensitivity Derivative", AIAA Paper 94-2213, 1994.

[71] Venkatakrishnan, V., "On the accuracy of limiters and convergence to steady state solutions", AIAA Paper, 93-0880, 1993.

[72] Kelley C.T., Iterative Methods for Linear and Nonlinear Equations, Society for Industrial and Applied Mathematics, Philadelphia, 1995.

[73] Korivi V.M., Taylor A.C., Newman P.A., Hou G.J.W. and Jones H.E.., "An Approximately Factored Incremental Strategy For Calculating Consistent Discrete Aerodynamic Sensitivity Derivatives", AIAA Paper 92-4746, 1992

[74] Lee, K. and Eyi, S., "Aerodynamic Design Via Optimization", *Journal of Aircraft,* Vol. 29, No.6 November-December 1992, pp 1012-1029.

[75] Taylor, A. C. III, Korivi, V. M., and Hou, G. W., "Sensitivity Analysis Applied to The Euler Equations : A Feasibility Study with Emphasis on Variation of Geometric Shape", AIAA Paper 91-0173 1991

[76] Taylor, A. C. III, , Hou, G. W., and Korivi, V. M., " A Methodology for Determining Aerodynamic Sensitivity Derivatives with Respect to Variation of Geometric Shape" AIAA Paper 91-1101 1991

[77] Hager, W., "Applied Numerical Linear Algebra", Prentice Hall, New Jersey, 1988

[78] Nadarajah, S. and Jameson, A., "Optimum Shape Design for Unsteady Flows with a Time Accurate Continuous and Discrete Adjoint Method", *AIAA Journal*, Vol. 45, No.7, pp 1478-1491, 2007

[79] Burgreen, G. W., and Baysal, O., "Three Dimensional Aerodynamic Shape Optimization Using Discrete Sensitivity Analysis", *AIAA Journal*, Vol. 34, No. 9, 1996, pp. 1761-1770

APPENDIX A

ANALYTICAL FLUX JACOBIANS FOR STEGER-WARMING FLUX VECTOR SPLITTING SCHEME

The Steger-Warming scheme defines the convective fluxes vectors as follows:

$$F=\frac{\rho}{2\gamma}\begin{bmatrix} 2(\gamma-1)\lambda_1+\lambda_3+\lambda_4 \\ \left(2(\gamma-1)\lambda_1+\lambda_3+\lambda_4\right)u+c\left(\lambda_3-\lambda_4\right)\eta_x \\ \left(2(\gamma-1)\lambda_1+\lambda_3+\lambda_4\right)v+c\left(\lambda_3-\lambda_4\right)\eta_y \\ \left(2(\gamma-1)\lambda_1+\lambda_3+\lambda_4\right)\frac{u^2+v^2}{2}+cU\left(\lambda_3-\lambda_4\right)+c^2\frac{\lambda_3+\lambda_4}{\gamma-1} \end{bmatrix} \tag{A.1}$$

Where η_x and η_y are the components of the face normal vector. The eigenvalues λ_1 , λ_2 , λ_3 and the speed of sound c is defined as:

$$\begin{aligned} c&=\sqrt{\gamma(\gamma-1)\left(e_t-\frac{1}{2}\left(u^2+v^2\right)\right)} \\ \lambda_1&=U\ ,\quad \lambda_2=U+c\ ,\quad \lambda_3=U-c \end{aligned} \tag{A.2}$$

Then taking the derivatives of each flux F_i with respect to the each flow variable $\boldsymbol{w}_j$ flux Jacobians F_{ij} are obtained as follows:

$$\begin{aligned} F_{11}&=\frac{\rho}{2\gamma}\left(2(\gamma-1)\lambda_{11}+\lambda_{31}+\lambda_{41}\right)+\frac{F_1}{\rho} \\ F_{12}&=\frac{\rho}{2\gamma}\left(2(\gamma-1)\lambda_{12}+\lambda_{32}+\lambda_{42}\right) \\ F_{13}&=\frac{\rho}{2\gamma}\left(2(\gamma-1)\lambda_{13}+\lambda_{33}+\lambda_{43}\right) \\ F_{14}&=\frac{\rho}{2\gamma}\left(2(\gamma-1)\lambda_{14}+\lambda_{34}+\lambda_{44}\right) \end{aligned} \tag{A.3}$$

$$
\begin{aligned}
F_{21} &= u\left(F_{11}-\frac{F_1}{\rho}\right)+\eta_x\frac{\rho}{2\gamma}\left(c_1(\lambda_3-\lambda_4)+c(\lambda_{31}-\lambda_{41})+\frac{c}{\rho}(\lambda_3-\lambda_4)\right)\\
F_{22} &= uF_{12}+\frac{F_1}{\rho}+\eta_x\frac{\rho}{2\gamma}\left(c_2(\lambda_3-\lambda_4)+c(\lambda_{32}-\lambda_{42})\right)\\
F_{23} &= uF_{13}+\eta_x\frac{\rho}{2\gamma}\left(c_3(\lambda_3-\lambda_4)+c(\lambda_{33}-\lambda_{43})\right)\\
F_{24} &= uF_{14}+\eta_x\frac{\rho}{2\gamma}\left(c_4(\lambda_3-\lambda_4)+c(\lambda_{34}-\lambda_{44})\right)
\end{aligned}
\tag{A.4}
$$

$$
\begin{aligned}
F_{31} &= v\left(F_{11}-\frac{F_1}{\rho}\right)+\eta_y\frac{\rho}{2\gamma}\left(c_1(\lambda_3-\lambda_4)+c(\lambda_{31}-\lambda_{41})+\frac{c}{\rho}(\lambda_3-\lambda_4)\right)\\
F_{22} &= vF_{12}+\eta_y\frac{\rho}{2\gamma}\left(c_2(\lambda_3-\lambda_4)+c(\lambda_{32}-\lambda_{42})\right)\\
F_{23} &= vF_{13}+\frac{F_1}{\rho}+\eta_y\frac{\rho}{2\gamma}\left(c_3(\lambda_3-\lambda_4)+c(\lambda_{33}-\lambda_{43})\right)\\
F_{34} &= vF_{14}+\eta_y\frac{\rho}{2\gamma}\left(c_4(\lambda_3-\lambda_4)+c(\lambda_{34}-\lambda_{44})\right)
\end{aligned}
\tag{A.5}
$$

$$
\begin{aligned}
F_{41} &= \frac{1}{2}\left(u^2+v^2\right)\left(F_{11}-\frac{2F_1}{\rho}\right)+\frac{\rho}{2\gamma}U\left(c_1(\lambda_3-\lambda_4)+c(\lambda_{31}-\lambda_{41})\right)\\
&\quad+\frac{\rho c}{\gamma(\gamma-1)}\left(c_1(\lambda_3+\lambda_4)+\frac{c}{2}(\lambda_{31}+\lambda_{41})+\frac{c}{2\rho}(\lambda_3+\lambda_4)\right)\\
F_{42} &= \frac{1}{2}\left(u^2+v^2\right)F_{12}+u\frac{F_1}{\rho}+\frac{\rho}{2\gamma}U\left(c_2(\lambda_3-\lambda_4)+c(\lambda_{32}-\lambda_{42})\right)\\
&\quad+\frac{\rho c}{\gamma(\gamma-1)}\left(c_2(\lambda_3+\lambda_4)+\frac{c}{2}(\lambda_{32}+\lambda_{42})+\eta_x\frac{c}{2\gamma}(\lambda_3-\lambda_4)\right)\\
F_{43} &= \frac{1}{2}\left(u^2+v^2\right)F_{13}+v\frac{F_1}{\rho}+\frac{\rho}{2\gamma}U\left(c_3(\lambda_3-\lambda_4)+c(\lambda_{33}-\lambda_{43})\right)\\
&\quad+\frac{\rho c}{\gamma(\gamma-1)}\left(c_3(\lambda_3+\lambda_4)+\frac{c}{2}(\lambda_{33}+\lambda_{43})+\eta_y\frac{c}{2\gamma}(\lambda_3-\lambda_4)\right)\\
F_{44} &= \frac{1}{2}\left(u^2+v^2\right)F_{14}+\frac{\rho}{2\gamma}U\left(c_4(\lambda_3-\lambda_4)+c(\lambda_{34}-\lambda_{44})\right)\\
&\quad+\frac{\rho c}{\gamma(\gamma-1)}\left(c_4(\lambda_3+\lambda_4)+\frac{c}{2}(\lambda_{34}+\lambda_{44})\right)
\end{aligned}
\tag{A.6}
$$

Where the derivatives of speed of sound c and eigenvalues λ_i with respect to flow variables $\boldsymbol{w}_j$, being c_j and λ_{ij} respectively are defined as:

$$c_1 = \frac{\partial c}{\partial \rho} = \frac{\gamma(\gamma-1)}{2\rho c}\left(u^2+v^2+e_t\right) \qquad \lambda_{11} = \frac{\partial \lambda_1}{\partial \rho} = -\frac{\lambda_1}{\rho}$$

$$c_2 = \frac{\partial c}{\partial (\rho u)} = -\frac{\gamma(\gamma-1)u}{2\rho c} \qquad \lambda_{12} = \frac{\partial \lambda_1}{\partial (\rho u)} = \frac{\eta_x}{\rho}$$

$$c_3 = \frac{\partial c}{\partial (\rho v)} = -\frac{\gamma(\gamma-1)v}{2\rho c} \qquad \lambda_{13} = \frac{\partial \lambda_1}{\partial (\rho v)} = \frac{\eta_y}{\rho}$$

$$c_4 = \frac{\partial c}{\partial (\rho E)} = -\frac{\gamma(\gamma-1)}{2\rho c} \qquad \lambda_{14} = \frac{\partial \lambda_1}{\partial (\rho E)} = 0$$

$$\lambda_{31} = \frac{\partial \lambda_3}{\partial \rho} = -\frac{\lambda_1}{\rho} + c_1 \qquad \lambda_{41} = \frac{\partial \lambda_4}{\partial \rho} = -\frac{\lambda_1}{\rho} - c_1$$

$$\lambda_{32} = \frac{\partial \lambda_3}{\partial (\rho u)} = \frac{\eta_x}{\rho} + c_2 \qquad \lambda_{42} = \frac{\partial \lambda_4}{\partial (\rho u)} = \frac{\eta_x}{\rho} - c_2$$

$$\lambda_{33} = \frac{\partial \lambda_3}{\partial (\rho v)} = \frac{\eta_y}{\rho} + c_3 \qquad \lambda_{43} = \frac{\partial \lambda_4}{\partial (\rho v)} = \frac{\eta_y}{\rho} - c_3$$

$$\lambda_{34} = \frac{\partial \lambda_3}{\partial (\rho E)} = c_4 \qquad \lambda_{44} = \frac{\partial \lambda_4}{\partial (\rho E)} = -c_4$$

The positive and negative flux vectors and their Jacobian matrices are obtained from Equation (A.1) to Equation(A.6) by substituting λ's by λ^+'s and λ^-'s respectively. Here, $\lambda_i^{\mp}$ and its derivative with respect to the flow variables $\boldsymbol{w_j}$, $\lambda_{ij}^{\mp}$ are defined as:

$$\lambda_i^{\pm} = \frac{\lambda_i \pm |\lambda_i|}{2}$$

$$\lambda_{ij}^{\pm} = \lambda_{ij}\left(\frac{1 \pm sign(\lambda_i)}{2}\right)$$

APPENDIX B

DESIGN VARIABLES

Airfoil Shape Modification

The geometry was modified by adding smooth perturbations. The perturbation was defined as a linear combination of base functions f_k

$$\Delta y(x) = \sum_{k=1}^{K} \beta_k f_k(x) \tag{B.1}$$

Where x is the normalized chord wise position, β_k is design variable, f_k is the shape function, and K is the number of design variables to be used.

Hicks-Henne Functions:

The sinusoidal shape functions are frequently used in airfoil optimization.

$$\begin{aligned} f_k(x) &= sin^3(\pi x^{e(k)}) \\ e(k) &= \frac{log(0.5)}{log(x_k)} \\ x_k &= 0.05, 0.15, 0.3, 0.5, 0.65, 0.8, 0.95 \end{aligned} \tag{B.2}$$

Here x_k are the locations of maximum height of the corresponding shape funtions.

Wagner Functions:

Wagner functions provide large variations with high harmonics.

$$
\begin{aligned}
f_1(x) &= \frac{[\theta + sin(\theta)]}{\pi} - sin^2\left(\frac{\theta}{2}\right) \\
f_k(x) &= \frac{sin(k\theta)}{k\pi} + \frac{sin[(k-1)\theta]}{\pi} \quad for \quad k > 1 \\
\theta &= 2sin^{-1}\left(\sqrt{x}\right)
\end{aligned} \tag{B.3}
$$

Patched Polynomials:

A cubic on one side of x_k is patched with another cubic on the other side to produce a smooth curve of second order continuity. x_k is the location of maximum perturbation .

$$
\begin{aligned}
f_k(x) &= 1 - \left(\frac{x - x_k}{x_k}\right)^2 \left(1 + \frac{A}{(1-x_k)^2}\frac{x}{x_k}\right) \quad for \quad 0 \le x \le x_k \\
f_k(x) &= 1 - \left(\frac{x - x_k}{1 - x_k}\right)^2 \left(1 + \frac{B}{x_k^2}\frac{1-x}{1-x_k}\right) \quad for \quad x_k \le x \le 1 \\
A &= max(0, 1 - 2x_k) \\
B &= max(0, 2x_k - 1) \\
x_k &= 0.05, 0.15, 0.3, 0.5, 0.65, 0.8, 0.95
\end{aligned} \tag{B.4}
$$

Legendre Polynomials:

The orthogonal functions can also be used as shape functions. The first seven shape functions of the Legendre polynomials are given as follows

$$
\begin{aligned}
f_1(x) &= (1-x)^3\sqrt{x} \\
f_2(x) &= (1-x)^3 2x \\
f_3(x) &= (1-x)^3(6x^2 - 6x) \\
f_4(x) &= (1-x)^3(20x^3 - 30x^2 + 12x) \\
f_5(x) &= (1-x)^3(70x^4 - 140x^3 + 90x^2 - 20x) \\
f_6(x) &= (1-x)^3(252x^5 - 630x^4 + 560x^3 - 210x^2 + 30x) \\
f_7(x) &= (1-x)^3(924x^6 - 2772x^5 + 3150x^4 - 1680x^3 + 420x^2 - 42x)
\end{aligned} \tag{B.5}
$$

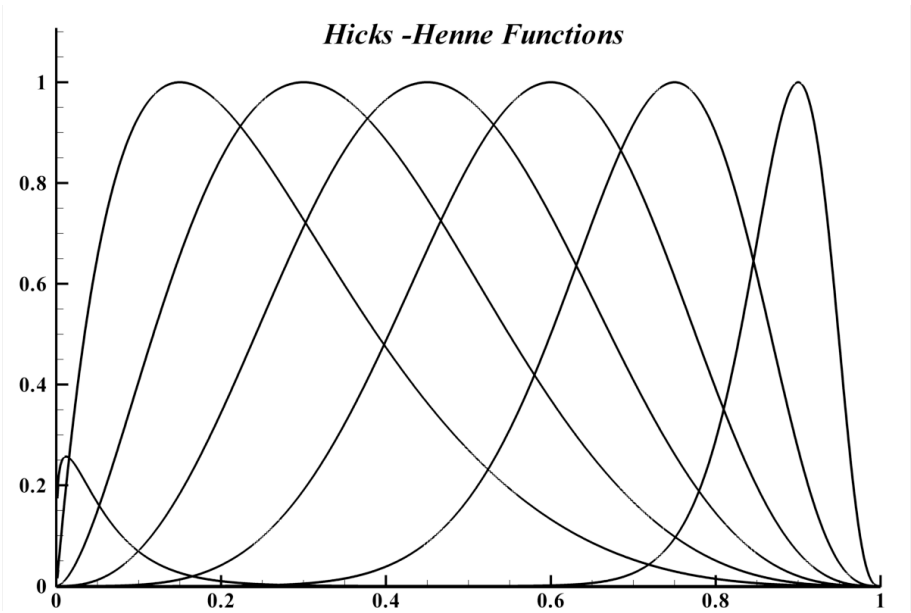

Figure B.1 *Hicks-Henne Functions*

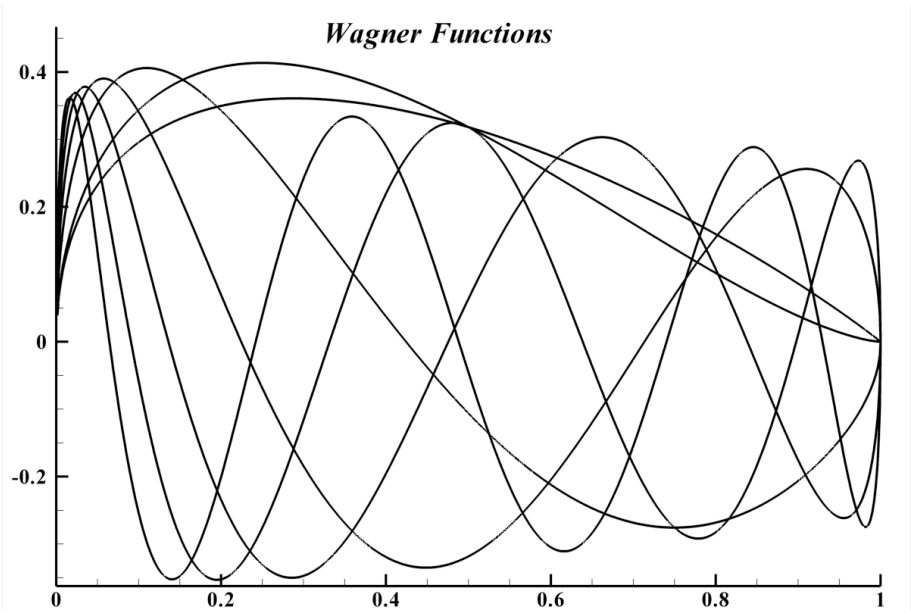

Figure B.2 *Wagner Functions*

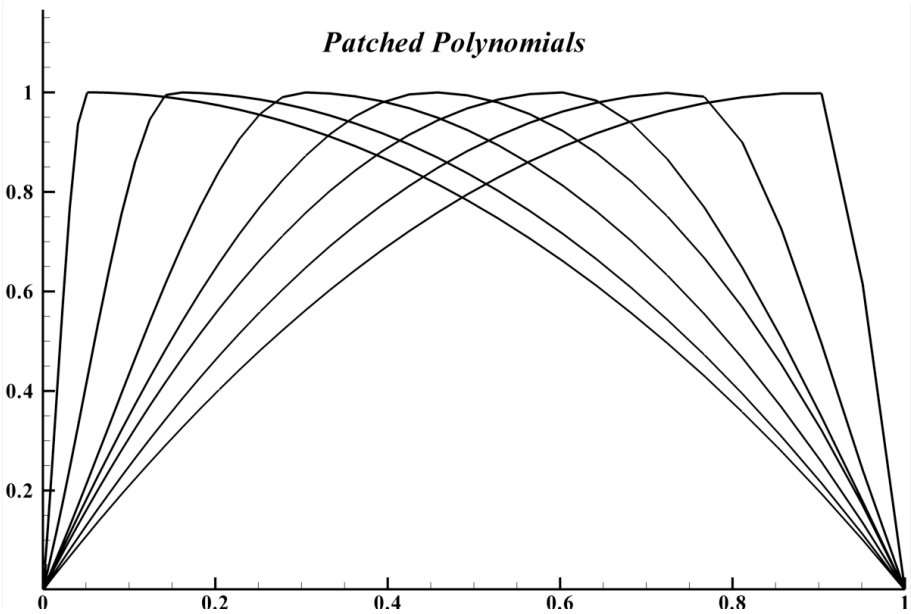

Figure B.3 *Patched Polynomials*

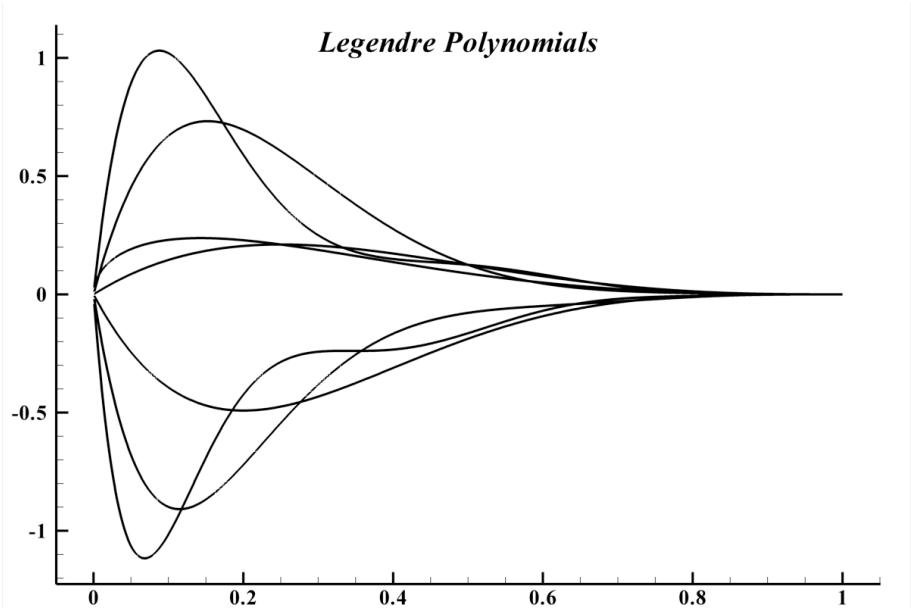

Figure B.4 *Legendre Polynomials*