Flowfield Dependent Variation Method Applied to Compressible Euler Flow Equations

By

Bassem Raafat Rateb Girgis

B.Sc. in Aerospace Engineering, 2004

A Thesis Submitted to the
Faculty of Engineering at Cairo University
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science
in
Engineering Mechanics

Faculty of Engineering, Cairo University

Giza, Egypt

July 2007

Flowfield Dependent Variation Method Applied to Compressible Euler Flow Equations

By

Bassem Raafat Rateb Girgis

B.Sc. in Aerospace Engineering, 2004

A Thesis Submitted to the

Faculty of Engineering at Cairo University

in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

in

Engineering Mechanics

Under the Supervision of

Prof. Adel A. Megahed

Engineering Mathematics and Physics Dept.

Faculty of Engineering

Cairo University

Faculty of Engineering, Cairo University

Giza, Egypt

July 2007

Flowfield Dependent Variation Method Applied to Compressible Euler Flow Equations

By

Bassem Raafat Rateb Girgis

B.Sc. in Aerospace Engineering, 2004

A Thesis Submitted to the
Faculty of Engineering at Cairo University
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science
in
Engineering Mechanics

Approved by the Examining Committee:

Prof. Adel A. Megahed, Thesis Main Advisor

Prof. Adel A. Mohsen, Member

Prof. Talaat Fawzy Refai, Member

Faculty of Engineering, Cairo University
Giza, Egypt
July 2007

تطبيق طريقة التغيرالمعتمد على مجال السريان على معادلات أويلر الأنضغاطيه

إعداد

باسم رأفت راتب جرجس

بكالوريوس في هندسة الطيران و علوم الفضاء-2004

رسالة مقدمة إلى كلية الهندسة - جامعة القاهرة

كجزء من متطلبات الحصول على درجة الماجستير

في

الميكانيكا الهندسية

كلية الهندسة - جامعة القاهرة

الجيزة - جمهورية مصر العربية

يوليو 2007

تطبيق طريقة التغيرالمعتمد على مجال السريان على معادلات أويلر الأنضغاطيه

إعداد

باسم رأفت راتب جرجس

بكالوريوس في هندسة الطيران و علوم الفضاء-2004

رسالة مقدمة إلى كلية الهندسة - جامعة القاهرة

كجزء من متطلبات الحصول على درجة الماجستير

في

الميكانيكا الهندسية

تحت إشراف

أ.د. عادل عبد الرحمن مجاهد

أستاذ بقسم الرياضيات و الفيزيقا الهندسية

كلية الهندسة – جامعة القاهرة

كلية الهندسة - جامعة القاهرة

الجيزة - جمهورية مصر العربية

يوليو 2007

تطبيق طريقة التغيرالمعتمد على مجال السريان على معادلات أويلر الأنضغاطيه

إعداد

باسم رأفت راتب جرجس

بكالوريوس في هندسة الطيران و علوم الفضاء-2004

رسالة مقدمة إلى كلية الهندسة - جامعة القاهرة

كجزء من متطلبات الحصول على درجة الماجستير

في

الميكانيكا الهندسية

يعتمد من لجنة الممتحنين:

الأستاذ الدكتور/ عادل عبد الرحمن مجاهد، المشرف الرئيسي

الأستاذ الدكتور/ عادل عبد القادر محسن، عضو

الأستاذ الدكتور/ طلعت فوزى رفاعى، عضو

كلية الهندسة - جامعة القاهرة

الجيزة - جمهورية مصر العربية

يوليو 2007

To my family

ACKNOWLEDGMENT

All gratitude and thanks is due to the creator of all.

The author would like to acknowledge Prof. Adel Megahed for both his scientific and fatherly guidance since the beginning of this work to its very end.

Special thanks, gratitude, and deep appreciation is due to the late Dr Mourad El-Mallah, the godfather of this work.

The author would like also to thank Prof. T. J. Chung, University of Alabama in Huntsville, for his time and sincere replies regarding both his book and FDV theory.

Also the author would like to thank all the staff of the Engineering Mathematics and Physics Dept. especially the staff of Engineering Mechanics for their continuous help and support.

Finally, all author's love and appreciation is due to his parents and his family.

ABSTRACT

Many practical CFD problems in industry and research attain crucial physical situations. In recent decades all CFD research efforts were directed towards the development of numerical techniques that are able to solve these situations accurately. To accurately predict flow properties the solution technique should be able to handle the interactions between high/low speeds, compressible/incompressible, subsonic/transonic/supersonic flows.

The present thesis aims at developing a time accurate flow solver for Euler equations. This solver should be able to operate on different flow regimes seamlessly. The capabilities of the flowfield-dependent mixed explicit/implicit scheme, also known as flowfield dependent variation (FDV) method, have been investigated. Finite element techniques have been used to discretize the flow domain via standard Galerkin method. Several benchmarks have been tested. These test cases have been selected to clarify the ability of the FDV method to resemble complex flow situations and to cover a wide range of flow regimes. Good agreement with published literature has been obtained in all cases.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

Symbols

$\mathbf{a}_i$	$= \dfrac{\partial \mathbf{F}_i}{\partial \mathbf{U}}$ convection Jacobian tensor
a	scalar Jacobian
$\mathbf{A}$	left hand side domain integral
$\mathbf{B}$	left hand side contour integral
c_v	specific heat at constant volume
c_p	specific heat at constant pressure
$d(\ )$	Infinitesimal change of
$D, \mathbf{D}$	1-D and 2-D first intermediate FDV parameter
$E, \mathbf{E}$	1-D and 2-D second intermediate FDV parameter
e	static internal energy
e_t	total internal energy
$\mathbf{F}_i$	Euler flux vector (convection term)
F	scalar flux function
$\mathbf{H}$	right hand side domain integral
i, j	co-ordinate dimension counters = 1,2
k	thermal conductivity
L^*	element characteristic length
M	Mach number
n_i	boundary-normal unit vector
$\mathbf{N}$	right hand side contour integral
$O(\ )$	order of error
P	pressure
$Q, \mathbf{Q}$	1-D and 2-D third intermediate FDV parameter
R	gas constant
r	intermediate constant

s_1, s_2	first and second order convection FDV parameters
T	temperature
t	time
Δt	time step
$\mathbf{U}$	conservative variables vector
$\mathbf{u}$	velocity vector
u_i	velocity components in 2-D space
x_i	space co-ordinates

Greek symbols

α, β	nodal element counters	1,2 for 1-D element
		1,..,4 for quadrilateral 2-D elements
ε	error from exact solution	
$\Phi, \overset{*}{\Phi}$	domain and boundary shape functions	
γ	specific heat ratio	
$\underline{\nabla}(\ )$	gradient of a scalar function	
ρ	density	
Ω, Γ	domain and contour boundaries	
ξ_i	natural coordinates	

Scripts

$(\ )_{\min}$	minimal value
$(\ )_{\max}$	maximal value
$\overset{*}{(\ )}$	non-dimensional value
$\hat{(\ )}$	exact solution value
$(\ )_{\infty}$	free stream value

Abbreviations

ADI	Alternating Direct Implicit
CFD	Computational Fluid Dynamics
CFL	Courant-Frederich-Levy Number
DCF	Discontinuity Capturing Factor
DCO	Discontinuity Capturing Operator
ENO	Essentially Non-Oscillatory
FDM	Finite Difference Method
FDV	Flowfield-Dependent Variation
FEM	Finite Element Method
FVM	Finite Volume Method
JIM	Jacobi Iteration Method
LGSIM	Line Gauss-Seidel Iteration Method
LSOR	Line Successive over Relaxation
ODE	Ordinary Differential Equation
PDE	Partial Differential Equation
PGSIM	Point Gauss-Seidel Iteration Method
PSOR	Point Successive Over Relaxation
RKDG	Runge-Kutta Discontinuous Galerkin
SUPG	Streamline Upwind Petrov-Galerkin
TDS	Tri-Diagonal Solver
TVB	Total Variation Bounded
TVD	Total Variation Diminishing

Chapter One
Introduction

1.1 Background

Fluid dynamics applications span a wide range of human activities and interests on planet earth. It ranges from water supplies to space explorations. Think of any application in the daily life and certainly it will have something to do with fluid dynamics. The analysis and design of fluid dynamics application systems can be divided into three main approaches. A quick summery for each approach can be found in the next subsections. For complete discussions on this topic, the reader is referred to any fluid dynamics historical textbook or the introductory sections for fluid dynamics books such as [1]-[10].

1.1.1 The Experimental Approach

This approach states that: if a designer wants to know how a fluid will react towards an obstacle or how the fluid affects this obstacle, he/she has to build the complete system or an experimental scale model. This model facilitates the designer's ability to see exactly what the flow properties will be (velocity vector, pressure, temperature, etc.) by means of experimental setups and flow measurements hardware.

The experimental approach is a good compromise for small applications design processes. Hence, it dominated the fluid dynamics field in the seventeenth century because of the simplicity of the engineering applications at these days. Also it looks more engineering than scientific, despite the fact that to have good results and accurate measurements the experiment should hold some scientific concepts such as; flow similarity conditions, experimental setup validity to actual size prototype, measurements accuracy, and small disturbances from the measurement hardware.

The advantages of this approach come from the certain and assumption-less results, especially if accurate experimental setup could be built. Whereas, disadvantages are mainly due to the cost of building/rebuilding of such accurate setups during the design refinement process. The measurements hardware can be very expensive in some cases, and it may be impossible in other cases (it is common to have a combustion temperature about 3000 K in rockets, and there is no regular temperature sensor that can operate at this temperature). Also the interferences and disturbances from the measuring sensors can't be avoided in most of the applications. The disadvantages of this approach lead engineers and scientists to the theoretical approach.

1.1.2 The Theoretical Approach

It is the approach of pure mathematical solutions for the governing equations of fluid flows. In this approach the governing equations are to be solved analytically (exactly) to provide the flow properties everywhere in the solution domain. The complete governing equations for fluid dynamics are second order non-linear system of partial differential equations (PDEs). These equations are agreed to be called the Navier-Stokes equations in viscous flows and Euler equations in inviscid flows. The exact solution of these systems is not known yet in general, and it looks like such a general solution will not be found in the near future.

In the late seventeenth and nearly all the eighteenth century, the scientists and designers had to assume strong assumptions, such as inviscid and incompressible flows. They even had to assume the flow to be potential in some cases to solve the governing PDEs analytically. These assumptions limited their ability from solving stiff fluid dynamics problems. It is worth mentioning to say that even after assuming potential flow, a pure analytical solution could not be found for all geometries and later a numerical technique was implemented to account for this problem, see [11], named the singularity distribution method.

At one hand, the human applications are growing and the need for efficient and economical design is due. At the other hand, the theoretical approach suffers from its limited ability to account for actual engineering applications and the cost of the experimental approach can't be afforded for the preliminary design stages. This forced the fluid dynamics community to the computational approach.

1.1.3 The Computational Approach

In this approach the Navier-Stokes and Euler equations are discretized using any discretization technique (to be discussed in section 1.2). Then the flow properties are "computed" at some points in the flow domain. Some numerical considerations should be accounted for in order to guarantee the accuracy and validity of these discrete values in visualizing the actual flowfield. The origins of this approach are well established from long time ago. But, due to the limited computing abilities, actual implementation could not be carried out till the late fifties of the last century when the digital computers revolution started.

This approach has the power of solving the complete system of governing equations for fluid dynamics. Virtually all flow phenomena can be simulated such as turbulence, chemical reactions, magnetic fields interactions, and even relativistic chemistry. This approach in general is referred to as Computational Fluid Dynamics (CFD). A brief summary of this issue is presented in the next section. The reader is referred to more specialist texts such as [1]-[3] and [5] to find more information.

1.2 Computational Fluid Dynamics

One of the best descriptions of the role of CFD in the fluid dynamics research can be found in [5]. It is clear from the previous discussion on both the experimental and theoretical approaches that they constitute two distinctive dimensions in the development of fluid dynamics; at one hand, pure experimental work and at the other hand, pure mathematical solutions. CFD offers a third dimension by its unique ability to render the actual flow phenomena using numerical methods and computations without the need for experimental setups (for preliminary design stages at least) and without the limitations of theoretical analysis. Many scientists refer to CFD as the science of numerical experimentation.

It has been stated earlier that the origins of CFD, or to be precise, the origins of numerical solutions of the PDEs were sited long time ago. However, actual implementation for such techniques in solving engineering problems has been delayed because of the computational limitations without computers. After the revolution of digital computers, especially that of the last three decades, CFD became an affordable design tool in virtually all human applications related to fluid dynamics.

By the years, many numerical methods have been developed and verified for accurately simulating the physical flow situations. The most famous and important methods will be summarized in the following subsections and the reader is referred to more specialized textbooks and review articles on this topic. [1] is especially recommended for further readings. The introduction of the finite element method is delayed to section 1.3 for more detailed analysis.

1.2.1 Finite Difference Methods

In the Finite Difference Methods (FDMs), the governing PDEs are replaced by finite difference equations. Analytical derivatives of the flow properties are approximated to their numerical counterparts. This transforms the differential equations into a numerical scheme that can be solved using any numerical method. [6] and [7] are recommended for further readings on the origins of the FDMs. The most famous numerical methods that FDMs are based on in the explicit formulation are Jacobi Iteration Method (JIM), Point

Gauss-Seidel Iteration Method (PGSIM), and Point Successive over Relaxation (PSOR). While for implicit formulations, the Tri-Diagonal Solver (TDS) usually employed with the Line Gauss-Seidel Iteration Method (LGSIM), Line Successive over Relaxation (LSOR), and Alternating Direct Implicit (ADI) method.

Many numerical schemes have been developed for the solution of the Navier-Stokes and Euler equations using FDMs. Some of these methods are based on central differences such as; Lax-Wendroff and explicit McCormack. Other methods are based on the concept of up-winding. The first order upwind schemes include Flux-Vector Splitting and Godunov method. The second order upwind schemes include Low resolution (MUSCL), high resolution (TVD). Also there are some schemes based on the concept of Essentially Non-Oscillatory (ENO) upwind scheme. All the FDMs are called point based schemes, because they constitute only the approximation of the solution at individual points in the solution domain called "Nodes".

Earlier applications of FDM in CFD include Courant, Friedrichs, and Lew (1928), Evans and Harlow (1957), Godunov (1959), Lax and Wendroff (1960), McCormack (1969), Briley and McDonald (1973), van Leer (1974), Beam and Warming (1978), Harten (1978, 1983), Roe (1981, 1984), Jameson (1982), among many others. The literature on FDM in CFD is adequately documented in many text books such as Roache (1972, 1999), Patankar (1980), Peyret and Taylor (1983), Anderson, Tannehill, and Pletcher (1984, 1997), Hoffman (1989), Hirsch (1988, 1990), Fletcher (1988), Anderson (1995), and Ferziger and Peric (1999), among others.

The main disadvantage in FDMs is its inability to handle complex domain boundaries that are likely to rise in actual engineering problems. Even when numerical transformations for some complex boundaries can be found, huge transformation matrices are required to be stored in order to calculate the derivatives of the flow properties in the computational domain. Structured grids are essential in FDMs, while most of the engineering problems naturally call for unstructured grids. Due to the FDMs shortages, the development of a domain based solutions rather than point based solutions is a logical step. And this is what called for finite volume and finite element methods.

1.2.2 Finite Volume Methods

Finite volume methods (FVMs), also called control volume methods, can be regarded as a logical extension to the FDM literature. All FDM practitioners can easily extend their knowledge and already-working schemes to FVM, with major modifications of course. Because of the simple data structure of FVMs, they become increasingly popular in recent years. Their formulations can be regarded as a domain based counterpart of FDM formulation.

FVMs are formulated from the inner product of the governing PDEs with a unit weighting function. This process results in the spatial integration of the governing equations. Unlike the point based methods, the FVM requires the satisfaction of the governing equations in the integral form rather than the differential form. This gives the method the advantage of working on unstructured grids and, hence, the ability to simulate complex computational domains. The integrated terms are approximated by either finite differences or finite elements, discretely summed over the entire domain.

Traditional curvilinear coordinate transformation required for FDM is no longer needed. Designation of the components of a vector normal to boundary surfaces in FVM accommodates the unstructured grid configuration with each boundary surface integral constructed between nodal points. Many domain-approximation-techniques have been adapted and implemented in the FVM codes. From these techniques; the node-centered control volume, cell-centered control volume, and cell-centered average scheme concepts. [1] is especially recommended for more details about the relation between FVM and other discretization methods.

1.2.3 Other Discretization Techniques

There are many other numerical discretization methods that have been developed to be used in the solution of PDEs which raise from engineering applications like fluid dynamics. From these methods we mention:

- Spectral Element Methods.
- Boundary Element Methods.
- Conservation-Element/Solution-Element Method.
- Mont-Carlo Method.
- Finite Point Method.

Despite the fact that these methods share the same heart in their origin, which is the numerical solution of PDEs, they are different in terms of the amount of analytical work and computations on digital computers. Also the simplicity of the mathematical concepts varies by a significant ratio from one method to another. Some of these methods provide the solution at the boundaries and/or the solution in the domain itself; hence a specific application may naturally call for its suitable numerical method. The designer/engineer, who has a certain engineering problem at hand, should have the ability to choose from these methods the one that best suite his/her interests.

1.3 Finite Element Methods

In finite element methods (FEMs), the solution domain is discretized with smaller domains that are called finite elements. A certain distribution for the flow properties over each element is assumed via shape functions. All the derivatives can be computed with the aid of these shape functions. The governing equations in the differential forms are multiplied by a trial function (usually the same shape function and/or a perturbation from it) and integrated over the element. This technique is called the weighted residual approach. After that an assembling technique is employed to get the global contribution matrix. FEMs have the advantage of giving the Neumann boundary conditions explicitly using the contour integrals that results from the integration by parts in the element equations. Through all the numerical methods that have been developed since ever, FEM is considered to be unique in this advantage.

FEMs were originally introduced by civil engineers. The reader is referred to any finite element textbook's introduction for a summary of the method's history such as [1] and [12]. Every writer has his/her own theory about the origins of the FEM. The essence of the method was certainly available for a long time, but the actual implementation delayed to till the mid of last century.

Earlier applications of FEM in CFD include Zienkiewiez and Cheung (1965), Oden (1972, 1988), Chung (1978), Hughes et al. (1982), Baker (1983), Zienkiewiez and Taylor (1991), Carey and Oden (1986), Pironneau (1989), Pepper and Heinrich (1992). Other contributions of FEM in CFD for the past two decades include generalized Petrov-Galerkin methods (Heinrich et al., 1977; Hughes, Franca, and Mallett, 1986: Johnson, 1987), Taylor-Galerkin methods (Donea, 1984; Lohner, Morgan, and Zienkiewiez, 1985), adaptive methods (Oder et al., 1989), characteristic Galerkin methods (Zienkiewiez et al., 1995), discontinuous Galerkin methods (Oden, Babuska, and Baumann, 1998), and incompressible flows (Gresho and Sani, 1999), among others.

To illustrate how the researchers in the CFD field think of both FDM and FEM, a quoted paragraph from [1] is reported. T. J. Chung wrote; "Historically, FDMs have dominated the CFD community. Simplicity in formulations and computations contributed to this trend. FEMs, on the other hand, are known to be more complicated in formulations and more time-consuming in computations. However, this is no longer the case in many of the recent developments in FEM applications. Many examples of superior performance of FEM have been demonstrated."

In this thesis, FEM is employed in the development of the required numerical solutions for many reasons. From these reasons we mention:

- The method has sound mathematical foundations. In some situations, the solution corresponds to a variational principle for which an upper/lower bound can be determined.
- The FEM approximates complicated geometrical boundaries easily. Almost invariably the FDM starts by regularizing the domain, i.e. mapping complex regions into regular (rectangular) regions, and there is no need to say that this is not possible in all engineering applications.
- The FEM accounts for boundary conditions in an easy and straight forward manner. Especially the Neumann boundary conditions. This will be demonstrated for no penetration and pressure boundary conditions in Chapter Two and Chapter Three.
- It is also a more implicit procedure than FDM, since more grid points are connected together. This facilitates the solution with high Courant-Frederich-Levy (CFL) numbers as will be demonstrated in Chapter Four.
- The FEM is modular; with a "library" of elements built in a code, one can shift to successively more sophisticated elements to gain more accuracy without changing program structure. On the same grid one can also use different levels of interpolation for the different dependent and independent variables, while a staggered grid may be needed in a FDM approach.

In the following subsections each approach of the main finite element sub-methods will be introduced in brief.

1.3.1 Direct Approach

This approach is applicable only to simple problems governed by an algebraic relationship or a simple first order ordinary differential equation (ODE) for 1-D problems. It can be used to simulate simple engineering situations where the governing equations are linear. The most important use for this approach is that; it illustrates all the mechanics of the FEM in terms of discretization, obtaining the local influence matrix, assembling it in a global matrix, and solving it for easy problems. This can be considered as a good starting point for any novice in FEM.

1.3.2 Variational Approach

This approach is best suited for physical problems that are governed by an extremization law, i.e. a minimum or a maximum. Such equations rise in solid mechanics more than fluid mechanics. In fluid mechanics and especially fluid dynamics, non-linearity is essential and, hence, not many variational principles exist. The most important application of the variational approach in fluid dynamics is in the solution of the inviscid incompressible and subsonic/ transonic flows in 1-D applications via the method of artificial compressibility.

1.3.3 Weighted Residual Approach

This approach is considered to be the most general finite element approach. It can be used for any type of PDEs and especially those of CFD. It suits the problems where a variational principle does not exist, for nonlinear problems, and for unsteady problems. The residual of the PDE(s) is minimized by being weighted with a certain weighting function, integrated over each element domain, and equated to zero. The weighting functions depend on the particular residual approach that will be used. In this class of methods one can mention the following methods:

- Standard Galerkin method.
- Taylor-Galerkin Method.

- Streamline Upwind Petrov-Galerkin Method (SUPG).
- The sub-domain method (also known as FVM).
- The collocation method.
- The least-squares method.

It is also important to mention that; whenever a variational principle exists, the variational FEM and the standard Galerkin FEM yield identical equations.

1.4 Literature Review

In this section a quick review of the available literature about the numerical methods in the solution of Navier-Stokes and Euler equations is presented. In this review, interest has been given to the most successful and globally respected researchers' work. Many other methods exist but they are either un-mature or have no active research nowadays.

1.4.1 Taylor-Galerkin Method

In 1984 Lohner et al. [13] used a method based on writing the non-symmetric first-order differential operators along characteristics to achieve a self-adjoint form, and then applied the Galerkin method for the solution of a system of hyperbolic equations. Approximately at the same time, Donea [14] developed the Taylor-Galerkin algorithm in which the weak statement (resulting from the integration by parts) was formed on a Taylor series expansion of the unsteady equation, with higher-order derivatives re-expressed in terms of derivatives of the flux vector of the hyperbolic conservation laws.

In 1987, Baker and Kim [15] generalized these concepts and proposed a Galerkin weak-statement formulation which encompasses over a dozen independently derived finite difference and finite element dissipative algorithms. In 1986, Oden et al. [16] used a semi-explicit two-step algorithm for the analysis of unsteady inviscid compressible flow in arbitrary two-dimensional domains. Later on, In 1987 Oden et al. [17] used the Taylor-Galerkin method for the solution of the Euler equations in the supersonic regime, along with the flux-corrected-transport approach to avoid non-physical oscillations in the solution.

However, in many cases, Gibbs-type oscillations of the solutions can still be observed owing to the presence of discontinuities, which are the main difficulty in the numerical solution of first-order hyperbolic conservation laws. An artificial viscosity or a limiter function is needed to control such oscillatory behavior.

1.4.2 Streamline Upwind Petrov-Galerkin Method (SUPG)

In 1983, Harten [18] developed the concept of TVD (total variation diminishing) and constructed second-order shock-capturing schemes using finite difference methods which have proved to be very successful in solving the compressible Euler equations for high-speed flows ([19] and [23]). Many desirable properties of TVD schemes, such as stability and robustness in solving the hyperbolic conservation laws with strong shocks, have been demonstrated. One characteristic of TVD schemes is that they are at most first-order accurate at non-sonic critical points. This restricts the accuracy of TVD schemes to be at most first-order in the L_∞-norm and at most second-order in the L_1-norm for general problems.

To overcome this difficulty, in 1987 Harten and Osher [24] and Harten et al. [25] constructed ENO (essentially non-oscillatory) schemes which use a local adaptive stencil to obtain information automatically from regions of smoothness when the solution develops discontinuities. As a result, approximations using these methods can achieve uniformly high-order accuracy right up to discontinuities, while keeping a sharp, essentially non-oscillatory shock transition. However, a convergence theory for ENO schemes is still not available at the present time.

Numerical experiments on ENO schemes for the scalar conservation law in two dimensions and the Euler equation in one dimension have been reported in the work of Harten et al. [25]. Also, results for two-dimensional gas-dynamic problems involving multiple-shock (Euler equations) interactions have been given in J. Yang and C. Lombard [26].

C. W. Shu in [27] offered a class of TVB (total variation bounded) uniformly high-order schemes has been proposed for the hyperbolic conservation laws, which, they claim, share most of the advantages and may remove local degeneracy at the critical points of TVD schemes. The TVD, TVB and ENO concepts and the resulting so-called high-resolution schemes are mostly developed in the finite difference or finite volume setting.

Using the idea of upwind schemes in the finite difference method, in 1982 Brooks and Hughes [28] introduced the Streamline Upwind Petrov-Galerkin (SUPG) method, in which the weight function is modified by adding a perturbation to the standard Galerkin test function. The added perturbation creates an upwind effect by weighting more heavily the upstream nodes within the elements than the downstream nodes. In 1984, Tezduyar and Hughes [29] generalized the SUPG method to first-order multidimensional hyperbolic systems. Later on, the shock capturing ability of the method was improved by adding non-linear operators to the perturbation to account for the compressible Navier-Stokes and Euler equations ([30]-[34]).

Among all the other CFD methods, the SUPG is considered to be the most solid method. This is due to the fact that Shakib et al. in [34] presented a mathematical poof for the stability and convergence of the method. This can be considered as a unique accomplishment to those authors.

1.4.3 Least-Squares Residual Method

Another method is the least-squares weighted residual method. In this method the weighting function is replaced by the derivative of the residual itself. The method has good stability properties due to its minimization nature, and has been applied for the solution of a variety of problems.

As one of the earliest efforts in this field one can mention the technique presented by Polk and Lynn [35] in 1978 for the solution of unsteady gas dynamic equations, with elements that are constructed in both space and time. Another space-time finite element scheme was presented by Nguyen and Reynen [36] in 1984 and was applied to the solution of convection-dominated problems in one and two-dimensions.

In 1979 Fletcher [37] used the least-squares method to solve the Euler equations for sub-critical compressible flows. The special feature of his method was to represent groups of variables rather than single variables. In 1989, Bruneau et al. [38] used a similar method to study the vortical phenomena created by the subsonic and supersonic flow over a flat plate at different angles of attack. Actual application for the least-squares finite element method has been introduced to fluid dynamics by Jiang and Povinelli [39] in 1990 and Lefobvre and Paraire [40] in 1993 when they introduced linear and quadratic approximations for the solution of the Euler equations.

Application of the least-squares method to a governing equation of the general form: $L(\phi)=f$ leads to the favorable result of a symmetric and positive-definite coefficient matrix, if L is a first-order differential operator. When L is a higher-order operator, however, this property is completely lost during the integration-by-parts and moreover elements with higher-order continuity.

Despite the facts that the implementation of least squares is easy and straight forward, this method in general suffers from the excessive, inherited, and uncontrollable dissipation which can in some cases cause inaccuracy of the results. Many researchers who developed this technique had to attach any adaptation technique to get satisfactory performance from the least squares formulations.

1.4.4 Discontinuous Galerkin Method

The discontinuous Galerkin method was introduced in 1973 by Reed and Hill [41] and successively analyzed in 1974-1975 by Lesaint and Raviart [42]-[43] for the linear advection equation. More recently, in 1989-1991 Cockburn and Shu [44]-[46], and Chen et al. [47] devised a high order accurate (both in space and time) total variation bounded (TVB) "Runge-Kutta discontinuous Galerkin" (RKDG) method for the solution of nonlinear systems of conservation laws. The TVB property of the RKDG method is enforced by means of a "slope limiting" procedure designed to control the deviation from the element mean of the numerical solution.

Finally the method was mature enough to the solution of the compressible Navier-Stokes and Euler equations in 1993-1995 by F. Bassi and S. Rebay [48]-[50]. They extended a high-order discontinuous finite element method for the numerical solution of the Euler equations, which has proven to be very effective. Later on, in 1996 they extended the method in [51] to the case of the compressible Navier-Stokes.

1.4.5 Flowfield Dependent Variation Method (FDV)

The flowfield dependent variation (FDV) method is considered to be a general approach which leads to most of the currently available computational schemes as special cases. The original idea of FDV began in 1999 from the need to address the physics involved in shockwave turbulent boundary layer interactions, Chung [52] and Schunk et al. [53].

In turbulent shockwave boundary layers the transitions, interactions, and interferences between various flow phenomena dominate the flowfield. In such flows, strong coupling exists between sub-flowfields that are inviscid/ viscous, compressible/incompressible, or laminar/turbulent in nature. The solution constitutes not only the physical complexities but also computational difficulties. This is where the very low velocity in the vicinity of the wall ($M = 0, \mathrm{Re} = 0$) and very high velocity far away from the wall ($M = 3, \mathrm{Re} = 10^6$) coexist within a domain of study. Transitions from one type of flow to another and interactions between two distinctly different flows have been studied for many years, both experimentally and numerically.

Incompressible flows were analyzed using the pressure-based formulation with the primitive variables for the implicit solution of the Navier-Stokes equations. The precondition process for the time-dependent term intended for all speed flows was also discussed. Whereas compressible flows were analyzed using the density-based formulation with the conservation variables for the solution of the Navier-Stokes equations. However, in dealing with the domain which contains all speed flows with various physical properties where the equations of state for compressible and incompressible flows are different, and where the transitions between laminar and turbulent flows are involved in dilatational dissipation due to compressibility, the numerical method must provide very special and powerful treatments when applied to the governing PDEs. The FDV method has been devised toward resolving these issues.

Originally T. J. Chung et al. [54]-[55] in 1996 introduced 3-D mixed explicit-implicit generalized Galerkin method using spectral elements. This method had the ability to analyze the high speed turbulent compressible flows. Later on, the same author [55] in 1998 evolved the method to its current form. Many successful implementations have been demonstrated.

1.5 Aim of the Thesis

Practical CFD problems in industry and research attain crucial physical situations. In the recent decades, all CFD research efforts were directed towards the development of numerical techniques that are able to solve these situations accurately. To accurately predict flow properties, the solution technique should be able to handle the inherited strong interactions between high/low speeds, compressible/incompressible, subsonic/transonic/supersonic flows.

This thesis aims at developing a time-accurate flow solver for Euler equations. Certain characteristics have been drawn about this solver which have to be attained in the developed technique, these characteristics are:

- This solver should be able to operate on different flow regimes seamlessly. It should give satisfactory performance in the solution of supersonic flows while maintaining its robustness and accuracy in the solution of subsonic/transonic flows as well.
- The solution should be time accurate.
- The solver should support all types of boundary conditions which may rise from most of the engineering applications in CFD.
- It should be able to handle structured as well as unstructured grids.
- The solution should not suffer from neither very high nor very low numerical dissipation. I.e. the amount of the used numerical diffusion should provide numerical stability and peak-less solutions without altering the accuracy.
- The computer program should be modularized as much as possible in order to be easily maintained and upgraded.
- No external codes and/or built in functions should be used for major computational steps. This is to gain the power of changing every detail in the program without restrictions.
- Preconditioned sparse GMRES matrix solver is to be used.

As a response to the requirements stated above, a solid numerical method has to be used. To this aim, the capabilities of the flowfield-dependent mixed explicit/implicit scheme, also known as flowfield dependent variation (FDV) method, have been investigated.

As stated earlier, the FDV theory was devised in response to the need to characterize the complex physics of shockwave turbulent boundary layers in which transitions between, and strong interactions of inviscid/viscous, incompressible/compressible, and laminar/turbulent flows constitute the most complex physical phenomena in fluid dynamics (Chung et al. [52]-[55]). The complexities of physics, in general, lead directly to computational difficulties.

It is clear from the original purpose of the FDV method that this method was introduced to solve very complicated and sophisticated flowfields. This method meets the demands in the Euler solver stated above. So, in this thesis a step by step analysis is introduced toward the development of the finite element flowfield dependent Euler equations solver.

1.6 Thesis Layout

This thesis is divided into four chapters and three appendices. These four chapters have been designed to give the best presentation for the research efforts that have been paid in this thesis development process. Numerical test cases and/or illustrating figures have been used whenever possible to clarify the used techniques for the reader.

The three appendices are introduced to gain the convenience of separating mathematical details of the Jacobians, iterative matrix solver, and error analysis from the heart of the theory. In the beginning of each chapter, a more comprehensive explanation of the layout and construction of this individual chapter are presented in the first paragraph.

In Chapter One (this chapter), an introduction and the necessary background have been introduced to gain the reader's attention for the value of the present research work. In section 1.1, a brief background about fluid dynamics' analysis approaches has been introduced to clarify the role of CFD in engineering applications. Section 1.2 represents an introduction to computational methods in CFD and a quick summary for most of them. Section 1.3 is more specialized since it introduces finite element method and its sub-methods. Section 1.4 provides a comprehensive review for the available literature in the numerical solution of Navier-Stokes and Euler equations using the finite element method. Section 1.5 states the aim of the thesis and the characteristic and guide lines used in the development of the computer code. Finally, section 1.6 (this section) presents the thesis layout and the contents of each chapter and its sections.

Chapter Two represents the heart of the flowfield dependent variation (FDV) method and verifies this technique though the application of FDV method to a sample 1-D equation. The first three sections (2.1, 2.2, and 2.3) are the milestones in FDV method derivation. Section 2.4 introduces the 1-D finite element implementation, and section 2.5 represents the numerical results.

In Chapter Three, the FDV method is applied to the compressible Euler equations. The discussion is limited to two-dimensional flows. Section 3.1 briefly summarizes the system of PDEs representing inviscid compressible flow (Euler equations) and section 3.2 introduces the FDV treatment of these equations. Section 3.3 discusses the implicitness parameters and the various methods to calculate them, while section 3.4 represents the finite element implementation technique. Section 3.5 introduces the most common compressible inviscid boundary conditions.

Chapter Four contains the numerical results which validate the developed FDV Euler equations solver and their discussions, and also introduces the thesis conclusions. In section 4.1 the discontinuity capturing factor (DCF) will be tuned to ensure ripple-free solution, while in section 4.2 the ability of the FDV formulation to predict the transient flows accurately will be verified by solving a Riemann problem. Sections 4.3 and 4.4 are devoted to supersonic and subsonic/transonic test cases, respectively. Finally, section 4.5 introduces the thesis conclusions and future research recommendations.

Chapter Two

Flowfield-Dependent Variation (FDV) Method

In this chapter, the basics of the FDV method will be presented. This will be done by selecting a model PDE in one-dimensional space and applying the FDV method to clarify its merits. This chapter contains five sections; the first three sections (2.1, 2.2, and 2.3) represent the milestones in FDV method formulation, the fourth section (2.4) introduces the FEM implementation for the resulting equations, and the fifth section (2.5) represents the solution of two applications using FDV with FEM in 1-D, namely the solution of the linear first order wave equation and the inviscid 1-D Burgers equation. Consider the one-dimensional scalar PDE given by:

$$\frac{\partial \psi}{\partial t}+\frac{\partial F}{\partial x}=0 \tag{2.1}$$

where; ψ is a space-time dependent scalar function (i.e. $\psi=\psi\left(x,t\right)$), $F=F\left(\psi\right)$ is a scalar function.

Equation (2.1) has been chosen because the system of compressible Euler flow equations can be written in the same form using the conservation variables (discussed in detail in Chapter Three). Using the chain rule, equation (2.1) can be rewritten as follows:

$$\frac{\partial \psi}{\partial t}+\frac{\partial F}{\partial \psi}\frac{\partial \psi}{\partial x}=0 \tag{2.2}$$

renaming the Jacobian of transformation to be:

$$a=\frac{\partial F}{\partial \psi} \tag{2.3}$$

substituting from (2.3) into (2.2), we get:

$$\frac{\partial \psi}{\partial t}+a\frac{\partial \psi}{\partial x}=0 \tag{2.4}$$

where; the Jacobian a is considered to be constant at each time step and will be updated after each time step.

2.1 Special Taylor Expansion

The first step in FDV method is to expand the unknown variable ψ in a special form of Taylor expansion. This will be done by expanding ψ^{n+1} around ψ^{n} in both the explicit and implicit Taylor formulations up to and including the second order derivative as follows:

$$\psi^{n+1} = \psi^{n} + \Delta t \frac{\partial \psi^{n}}{\partial t} + \frac{(\Delta t)^2}{2} \frac{\partial^2 \psi^{n}}{\partial t^2} + O\left(\Delta t^3\right) \tag{2.5}$$

$$\psi^{n+1} = \psi^{n} + \Delta t \frac{\partial \psi^{n+1}}{\partial t} + \frac{(\Delta t)^2}{2} \frac{\partial^2 \psi^{n+1}}{\partial t^2} + O\left(\Delta t^3\right) \tag{2.6}$$

multiplying equation (2.5) by $(1-s)$ and equation (2.6) by s and adding we get:

$$\psi^{n+1} = \psi^{n} + \Delta t \left((1-s) \frac{\partial \psi^{n}}{\partial t} + s \frac{\partial \psi^{n+1}}{\partial t} \right) + \\ + \frac{(\Delta t)^2}{2} \left((1-s) \frac{\partial^2 \psi^{n}}{\partial t^2} + s \frac{\partial^2 \psi^{n+1}}{\partial t^2} \right) + O\left(\Delta t^3\right) \tag{2.7}$$

It is clear that equating s to 0.0 the explicit form of Taylor expansion is obtained, while equating to 1.0 the implicit form is recovered. Allowing the first and the second order derivatives to have different implicitness, equation (2.7) will have the following form:

$$\psi^{n+1} = \psi^{n} + \Delta t \frac{\partial \psi^{n+s_1}}{\partial t} + \frac{(\Delta t)^2}{2} \frac{\partial^2 \psi^{n+s_2}}{\partial t^2} + O\left(\Delta t^3\right) \tag{2.8}$$

where;

$$\frac{\partial \psi^{n+s_1}}{\partial t} = \frac{\partial \psi^{n}}{\partial t} + s_1 \frac{\partial \Delta \psi^{n+1}}{\partial t} \tag{2.9}$$

$$\frac{\partial^2 \psi^{n+s_2}}{\partial t^2} = \frac{\partial^2 \psi^{n}}{\partial t^2} + s_2 \frac{\partial^2 \Delta \psi^{n+1}}{\partial t^2} \tag{2.10}$$

$$\Delta \psi^{n+1} = \psi^{n+1} - \psi^{n} \tag{2.11}$$

s_1 and s_2 are called the implicitness parameters and they vary from 0.0 to 1.0. It is clear that at $s_1 = s_2 = 0.0$ (1.0) equation (2.5) returns to the explicit (implicit) form of the Taylor expansion. For mixed values between 0.0 and 1.0, equation (2.5) reads a mixed explicit/implicit form of the Taylor expansion. These parameters, s_1 and s_2, will be computed from the flowfield properties in the preceding time step. From these definitions, the FDV parameters can be summarized as follows:

- $s_1 \equiv F$ function's first order FDV parameter.
- $s_2 \equiv F$ function's second order FDV parameter.

2.2 Presentation of the FDV Implicitness Parameters

The second step in FDV formulation is to introduce the so called FDV implicitness parameters. The parameters s_1 and s_2 are the reason of the FDV method's name. These parameters may gain their physical meaning by calculating them from the flow variables' fluctuations (changes) in the derivative.

The first order FDV parameter s_1 is flowfield dependent, whereas the second order FDV parameters s_2 is assumed to be exponentially proportional to the first order parameter and mainly acts as artificial viscosity. The primary goal from s_1 is to provide the best possible solution accuracy at the regions of high rate of variations and discontinuities. While the goal from s_2 is to provide the solution with the required numerical diffusion without altering its accuracy. Following the same concept in [56], the proposed method to calculate s_1 from the current flowfield is given by:

$$s_1 = \min\left(r, 1\right) \tag{2.12}$$

$$s_2 = \frac{1}{2}\left(1 + s_1^{\eta}\right) \tag{2.13}$$

where; $r = L^* \left| \frac{\partial \psi}{\partial x} \right| \Big/ \psi_{\min}$, $0.05 < \eta < 0.2$.

The value $\psi_{\min}$ is defined by the minimum algebraic absolute between the neighboring nodes (i.e. between the element nodes). The main idea behind the special form of the Taylor expansion is that the regions of high numerical instability or high rate of variations in the solution domain will have high implicitness parameters, and vise versa. The computed values for s_1 and s_2 should attain two conditions. Theses conditions are:

- Good resolution for the discontinuities and high rate of variations in ψ.
- The minimum possible numerical diffusion that maintains stability and virtually has no effect on the solution accuracy.

Substituting from equations (2.9), (2.10), and (2.11) into equation (2.5) we get:

$$\Delta \psi^{n+1} = \Delta t \left(\frac{\partial \psi^n}{\partial t} + s_1 \frac{\partial \Delta \psi^{n+1}}{\partial t} \right) + \\ + \frac{(\Delta t)^2}{2} \left(\frac{\partial^2 \psi^n}{\partial t^2} + s_2 \frac{\partial^2 \Delta \psi^{n+1}}{\partial t^2} \right) + O\left(\Delta t^3\right) \tag{2.14}$$

Equation (2.1) can be rewritten as follows:

$$\frac{\partial \psi}{\partial t} = -\frac{\partial F}{\partial x} \tag{2.15}$$

also equation (2.4) can be rewritten as follows:

$$\frac{\partial \psi}{\partial t} = -a \frac{\partial \psi}{\partial x} \tag{2.16}$$

Differentiating equation (2.16) w.r.t. time we get:

$$\frac{\partial^2 \psi}{\partial t^2} = -a \frac{\partial}{\partial t} \left(\frac{\partial \psi}{\partial x} \right) \tag{2.17}$$

and interchanging the order of the time and spatial derivatives we have:

$$\frac{\partial^2\psi}{\partial t^2} = -a\frac{\partial}{\partial x}\left(\frac{\partial\psi}{\partial t}\right) \tag{2.18}$$

substituting from (2.15) into (2.18), we obtain:

$$\frac{\partial^2\psi}{\partial t^2} = a\frac{\partial^2 F}{\partial x^2} \tag{2.19}$$

substituting from (2.15) and (2.19) into (2.14) the result will take the following form:

$$\Delta\psi^{n+1} = \Delta t\left(-\frac{\partial F^n}{\partial x} - s_1\frac{\partial\Delta F^{n+1}}{\partial x}\right) + \\ +\frac{(\Delta t)^2}{2}\left(a\frac{\partial^2 F^n}{\partial x^2} + s_2 a\frac{\partial^2\Delta F^{n+1}}{\partial x^2}\right) + O\left(\Delta t^3\right) \tag{2.20}$$

Equation (2.20) is the Taylor expansion of the time term in (2.1). Again, at $s_1 = s_2 = 0.0$ (1.0) the explicit (implicit) expansion is obtained, and for values between 0.0, 1.0 a mixed explicit/implicit expansion results.

2.3 Residual Form of the FDV Method

The third and final step in the FDV method is to write the governing equation in a residual form in order to make it ready for any discretization technique such as FEM or FDM. Rewriting (2.20) in a residual form and expressing the terms associated with the FDV parameters in terms of their Jacobians we get:

$$\Delta\psi^{n+1} + \Delta t\, s_1\frac{\partial}{\partial x}\left(a\,\Delta\psi^{n+1}\right) - \frac{(\Delta t)^2}{2}as_2\frac{\partial^2}{\partial x^2}\left(a\,\Delta\psi^{n+1}\right) + \\ +\Delta t\frac{\partial F^n}{\partial x} - \frac{(\Delta t)^2}{2}a\frac{\partial^2 F^n}{\partial x^2} + O\left(\Delta t^3\right) = 0 \tag{2.21}$$

rearranging equation (2.21) we get:

$$\Delta\psi^{n+1} + D^n\frac{\partial}{\partial x}\left(\Delta\psi^{n+1}\right) + E^n\frac{\partial^2}{\partial x^2}\left(\Delta\psi^{n+1}\right) + Q^n + O\left(\Delta t^3\right) = 0 \tag{2.22}$$

where;

$$D^n = \Delta t s_1 a$$

$$E^n = -\frac{(\Delta t)^2}{2} s_2 a^2$$

$$Q^n = \Delta t \frac{\partial F^n}{\partial x} - \frac{(\Delta t)^2}{2} a \frac{\partial^2 F^n}{\partial x^2}$$

(2.23)

The superscript n is used to emphasize that these terms are calculated at the n^{th} time step. Equation (2.22) along with equation (2.23) represent the FDV method treatment for the original PDE given in (2.1). Equation (2.22) is ready to be solved for the time change of the unknown variable ($\Delta \psi^{n+1}$). It can be solved using any discretization technique such as FEM or FDM. Since (2.22) contains all the necessary numerical dissipation and stabilizing elements that are required for numerical stability and solution accuracy, no additional modifications are necessary from the discretization technique that will be used.

It is worth mentioning to say that, with the aid of the implicitness parameters (s_1 and s_2) any available FEM or FDM scheme may rise as a special case from FDV method when holding these parameters to certain values.

2.4 Finite Element Implementation via Standard Galerkin Method

Since all the numerical stability issues have been accounted for in the formulation of the FDV method, the application of the standard Galerkin method is the next and final step towards finite element implementation for equation (2.22). It worth motioning to say that the standard Galerkin method is the central difference schemes counterpart in the FEM. Standard Galerkin method is carried out by the inner product of the residual with the shape function as the weighting function:

$$\int_{x=x_1}^{x_2} \Phi_\alpha R(\psi, F) dx = 0 \tag{2.24}$$

where; $R(\psi, F)$ is the residual of (2.22) and Φ_α is the shape function for node α. In what follows the order of the solution error $O(\Delta t^3)$ will be omitted from (2.22) for convenience. Substituting from (2.22) into (2.24), we get:

$$\int_{x=x_1}^{x_2} \Phi_\alpha \left(\Delta\psi^{n+1} + D^n \frac{\partial}{\partial x}\left(\Delta\psi^{n+1}\right) + E^n \frac{\partial^2}{\partial x^2}\left(\Delta\psi^{n+1}\right) + Q^n \right) dx = 0 \quad (2.25)$$

The first step in calculating the integration given in equation (2.25); a certain shape function is to be assumed for the variation of the unknown $\Delta\psi^{n+1}$. The second step is to integrate by parts all the differential terms. The next two subsections summarize these steps for the 1-D element using linear one-dimensional shape function.

2.4.1 1-D Linear Shape Function

Consider the 1-D element shown in Figure 2.1. The solution can be either in terms of the global coordinate "x" or the natural coordinate "ξ". The later is used because it provides a simpler treatment in the finite element formulations. Both coordinate systems are shown in Figure 2.1.

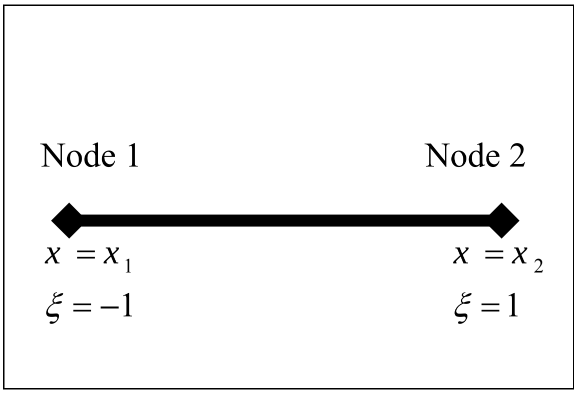

Figure 2.1 1-D element in global and natural coordinates

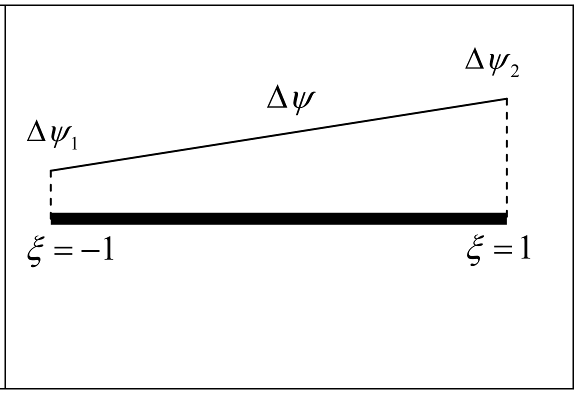

Figure 2.2 Linear 1-D scalar function variation

Assuming linearly varying scalar function $\Delta\psi$ as shown in Figure 2.2, the linear distribution of the scalar function in terms of the natural coordinate ξ can be written as follows:

$$\Delta\psi = \beta_1 + \beta_2 \xi \quad (2.26)$$

or in matrix form:

$$\Delta\psi = \begin{bmatrix}1 & \xi\end{bmatrix}\begin{Bmatrix}\beta_1 \\ \beta_2\end{Bmatrix} \tag{2.27}$$

Where β_1 and β_2 are unknown constants. Using the values at the element nodes summarized in Table 1 to determine these constants:

Table 1 Linear 1-D element nodal coordinates in the natural coordinate

Node	ξ	$\Delta\psi$
1	-1.0	$\Delta\psi_1$
2	1.0	$\Delta\psi_2$

$$\Delta\psi_1 = \beta_1 - \beta_2$$
$$\Delta\psi_2 = \beta_1 + \beta_2$$

or in a matrix form:

$$\begin{Bmatrix}\Delta\psi_1 \\ \Delta\psi_2\end{Bmatrix} = \begin{bmatrix}1 & -1 \\ 1 & 1\end{bmatrix}\begin{Bmatrix}\beta_1 \\ \beta_2\end{Bmatrix}$$

Using matrix inversion to obtain the unknown constants β_1 and β_2, we get:

$$\begin{Bmatrix}\beta_1 \\ \beta_2\end{Bmatrix} = \frac{1}{2}\begin{bmatrix}1 & 1 \\ -1 & 1\end{bmatrix}\begin{Bmatrix}\Delta\psi_1 \\ \Delta\psi_2\end{Bmatrix} \tag{2.28}$$

substituting from (2.28) into (2.27), we obtain:

$$\Delta\psi = \frac{1}{2}\begin{bmatrix}1 & \xi\end{bmatrix}\begin{bmatrix}1 & 1 \\ -1 & 1\end{bmatrix}\begin{Bmatrix}\Delta\psi_1 \\ \Delta\psi_2\end{Bmatrix}$$
$$\Delta\psi = \frac{1}{2}\begin{bmatrix}1-\xi & 1+\xi\end{bmatrix}\begin{Bmatrix}\Delta\psi_1 \\ \Delta\psi_2\end{Bmatrix}$$

letting:

$$\Phi_1 = \frac{1}{2}(1-\xi), \qquad \Phi_2 = \frac{1}{2}(1+\xi) \tag{2.29}$$

where; Φ_1 and Φ_2 are the first and second nodes' shape functions, shown in Figure 2.3. The reader is referred to any finite element textbook like [1] or [12].

The shape functions' derivatives w.r.t the natural co-ordinate is given by:

$$\Phi_{1,\xi} = \frac{-1}{2}, \qquad \Phi_{2,\xi} = \frac{1}{2} \tag{2.30}$$

Using the shape function, any scalar function and its derivative can be written as:

$$\Delta\psi = \begin{bmatrix} \Phi_1 & \Phi_2 \end{bmatrix} \begin{Bmatrix} \Delta\psi_1 \\ \Delta\psi_2 \end{Bmatrix} \tag{2.31}$$

$$\Delta\psi_{\xi} = \begin{bmatrix} \Phi_{1,\xi} & \Phi_{2,\xi} \end{bmatrix} \begin{Bmatrix} \Delta\psi_1 \\ \Delta\psi_2 \end{Bmatrix} \tag{2.32}$$

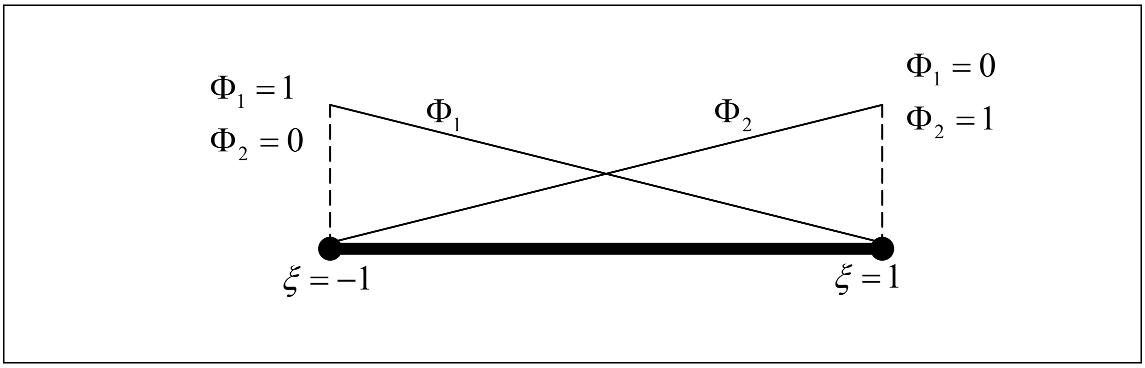

Figure 2.3 1-D linear shape functions in natural coordinates

From the definition of the shape function we have:

$$x = \Phi_1 x_1 + \Phi_2 x_2 \tag{2.33}$$

This leads to the following relations:

$$\xi = \frac{2x - (x_1 + x_2)}{L^*}, \quad \frac{\partial \xi}{\partial x} = \frac{2}{L^*} \tag{2.34}$$

$$dx = \frac{L^*}{2} d\xi \tag{2.35}$$

where; $L^* = x_2 - x_1$ is the element characteristic length.

2.4.2 Integration by Parts

The integration will be evaluated in terms of ξ, so rewriting the derivatives in equation (2.25) in terms of "ξ" rather than "x" using the linear 1-D shape function presented in the previous section:

$$\int_{\xi=-1.0}^{1.0} \Phi_\alpha \begin{pmatrix} \Delta\psi^{n+1} + D^n \dfrac{\partial}{\partial\xi}\left(\Delta\psi^{n+1}\right)\dfrac{\partial\xi}{\partial x} + E^n \dfrac{\partial}{\partial\xi}\left(\dfrac{\partial}{\partial\xi}\left(\Delta\psi^{n+1}\right)\dfrac{\partial\xi}{\partial x}\right)\dfrac{\partial\xi}{\partial x} + \\ +\Delta t\, \dfrac{\partial F^n}{\partial\xi}\dfrac{\partial\xi}{\partial x} - \dfrac{\left(\Delta t\right)^2}{2} a \dfrac{\partial}{\partial\xi}\left(\dfrac{\partial F^n}{\partial\xi}\dfrac{\partial\xi}{\partial x}\right)\dfrac{\partial\xi}{\partial x} \end{pmatrix} \frac{L^*}{2} d\xi = 0 \tag{2.36}$$

substituting from (2.34) and (2.35) into (2.36) we get:

$$\int_{-1.0}^{1.0} \Phi_\alpha \begin{pmatrix} \Delta\psi^{n+1}\dfrac{L^*}{2} + D^n \dfrac{\partial}{\partial\xi}\left(\Delta\psi^{n+1}\right) + \dfrac{2E^n}{L^*}\dfrac{\partial^2}{\partial\xi^2}\left(\Delta\psi^{n+1}\right) + \\ +\Delta t\, \dfrac{\partial F^n}{\partial\xi} - \left(\Delta t\right)^2 \dfrac{a}{L^*}\dfrac{\partial^2 F^n}{\partial\xi^2} \end{pmatrix} d\xi = 0 \tag{2.37}$$

expressing $\Delta\psi^{n+1}$ as a linear combination of the shape functions Φ_α we have:

$$\Delta\psi^{n+1}\left(\xi,t\right) = \Phi_\alpha\left(\xi\right)\Delta\psi_\alpha^{n+1}\left(t\right) \tag{2.38}$$

substituting from (2.38) into (2.37) and performing integration by parts we get:

$$\begin{aligned} &\int_{-1.0}^{1.0} \Phi_\alpha\Phi_\beta\Delta\psi_\beta^{n+1}\frac{L^*}{2}d\xi + D^n\,\overset{*}{\Phi}_\alpha\,\Delta\psi^{n+1}\Big|_{-1.0}^{1.0} - \int_{-1.0}^{1.0} D^n\,\Phi_{\alpha,\xi}\Phi_\beta\,\Delta\psi_\beta^{n+1}d\xi + \\ &+\frac{2E^n}{L^*}\overset{*}{\Phi}_\alpha\frac{\partial\left(\Delta\psi^{n+1}\right)}{\partial\xi}\Bigg|_{-1.0}^{1.0} - \int_{-1.0}^{1.0}\frac{2E^n}{L^*}\Phi_{\alpha,\xi}\Phi_{\beta,\xi}\Delta\psi_\beta^{n+1}d\xi + \\ &+\Delta t\,\overset{*}{\Phi}_\alpha\,\overset{*}{F}{}^n\Big|_{-1.0}^{1.0} - \Delta t\int_{-1.0}^{1.0}\Phi_{\alpha,\xi}F^n d\xi + \\ &-\left(\Delta t\right)^2\frac{a}{L^*}\overset{*}{\Phi}_\alpha\frac{\partial\overset{*}{F}{}^n}{\partial\xi}\Bigg|_{-1.0}^{1.0} + \left(\Delta t\right)^2\frac{a}{L^*}\int_{-1.0}^{1.0}\Phi_{\alpha,\xi}\frac{\partial F^n}{\partial\xi}d\xi = 0 \end{aligned} \tag{2.39}$$

2.4.3 Finite Element Equations

In order to obtain the finite element equations, the terms in equation (2.39) are rearranged after substituting from (2.23) in the following form:

$$\left(A_{\alpha\beta}^n + B_{\alpha\beta}^n\right)\Delta\psi_\beta^{n+1} = H_\alpha^n + N_\alpha^n \tag{2.40}$$

where;

$$A^n_{\alpha\beta} = \int_{-1.0}^{1.0} \left(\Phi_\alpha \Phi_\beta \frac{L^*}{2} - \Delta t s_1 a \Phi_{\alpha,\xi} \Phi_\beta + (\Delta t)^2 \frac{s_2 a^2}{L^*} \Phi_{\alpha,\zeta} \Phi_{\beta,\xi} \right) d\xi$$

$$B^n_{\alpha\beta} = \left(\Delta t s_1 a \overset{*}{\Phi}_\alpha \overset{*}{\Phi}_\beta - (\Delta t)^2 \frac{s_2 a^2}{L^*} \overset{*}{\Phi}_\alpha \overset{*}{\Phi}_{\beta,\xi} \right) \Bigg|_{-1.0}^{1.0}$$

$$H^n_\alpha = \int_{-1.0}^{1.0} \left(\Delta t \Phi_{\alpha,\xi} F^n - (\Delta t)^2 \frac{a}{L^*} \Phi_{\alpha,\xi} \frac{\partial F^n}{\partial \xi} \right) d\xi$$

$$N^n_\alpha = \left(-\Delta t \overset{*}{\Phi}_\alpha \overset{*}{F}^n + (\Delta t)^2 \frac{a}{L^*} \overset{*}{\Phi}_\alpha \frac{\partial \overset{*}{F}^n}{\partial \xi} \right) \Bigg|_{-1.0}^{1.0}$$

(2.41)

It is clear that in the inter-elements the values of $B^n_{\alpha\beta}$ and N^n_α cancel each other. They will only be available at the first element's first node and the last element's second node. At the first element's first node we have:

$$B^n_{11} = -\Delta t s_1 a - \frac{(\Delta t)^2}{2} \frac{s_2 a^2}{L^*}$$
$$N^n_1 = \Delta t \overset{*}{F}^n + \frac{(\Delta t)^2}{2} \frac{a}{L^*} \overset{*}{F}^n \tag{2.42}$$

and at the last element's second node we have:

$$B^n_{\mathrm{Imax,Imax}} = \Delta t s_1 a - \frac{(\Delta t)^2}{2} \frac{s_2 a^2}{L^*}$$
$$N^n_{\mathrm{Imax}} = -\Delta t \overset{*}{F}^n + \frac{(\Delta t)^2}{2} \frac{a}{L^*} \overset{*}{F}^n \tag{2.43}$$

It is evident from (2.42) and (2.43) that if there is any Neumann boundary condition in the solution, it can be easily implemented by simply setting it either at the first or the last node.

2.5 FDV Method Applied to 1-D Equations

To verify the applicability of the FDV method, two simple applications are presented in this section. These applications are the linear first order wave equation in 1-D and the inviscid 1-D Burgers equation. The analytical solutions of those problems provide good chance to test the capabilities of the FDV method and also to verify the role of implicitness parameters.

2.5.1 Linear First Order Wave Equation

The general form of the linear first order wave equation is given by:

$$\frac{\partial \psi}{\partial t} + \frac{\partial}{\partial x}(a\psi) = 0 \tag{2.44}$$

where; a is a constant greater than zero.

For this simple equation, the characteristic lines are straight lines given by the equation $x - at = const$. Here, the value of ψ is convected along these lines with constant velocity a. The general solution of (2.44) is given by:

$$\psi = \psi(x - at) \tag{2.45}$$

For instant, when the initial condition is given by:

$$\psi(x,0) = \begin{cases} 0 & 0.0 \le x \le 0.2 \\ \sin(kx) & 0.2 < x \le 0.3 \\ 0 & 0.3 < x \le 1.0 \end{cases} \tag{2.46}$$

where; $k = 20\pi$.

Figure 2.4 plots the initial condition given in (2.46). Using the general solution given in (2.45) and the initial condition in (2.46), the solution of (2.44) will take the form.

$$\psi(x,0) = \begin{cases} 0 & x \le x_{\min} \\ \sin(k(x - at)) & x_{\min} < x \le x_{\max} \\ 0 & x_{\max} < x \end{cases} \tag{2.47}$$

where; $x_{\min} = 0.2 + at$ and $x_{\max} = 0.3 + at$.

Letting $a = 20$ and discretizing the domain from 0.0 to 1.0 with 500 nodes and using $\Delta t = 10^{-5}$, the following results have been obtained. Figure 2.5 plots the numerical and exact solutions at different time steps. It is clear that the FDV method has succeeded in resolving the solution without major changes in the profile of the convected solution. A closer comparison with FDV and most of the other CFD methods like those of the FDM will reveal the superiority of the FDV-FEM formulation.

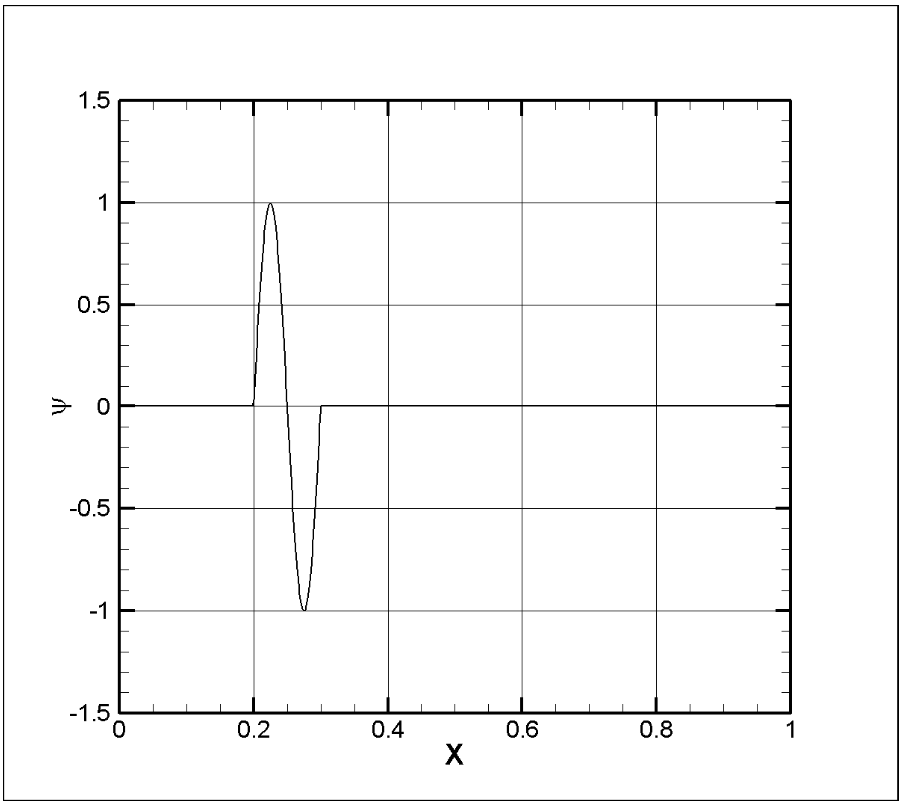

Figure 2.4 $\psi\left(x,0\right)$, linear wave equation initial condition

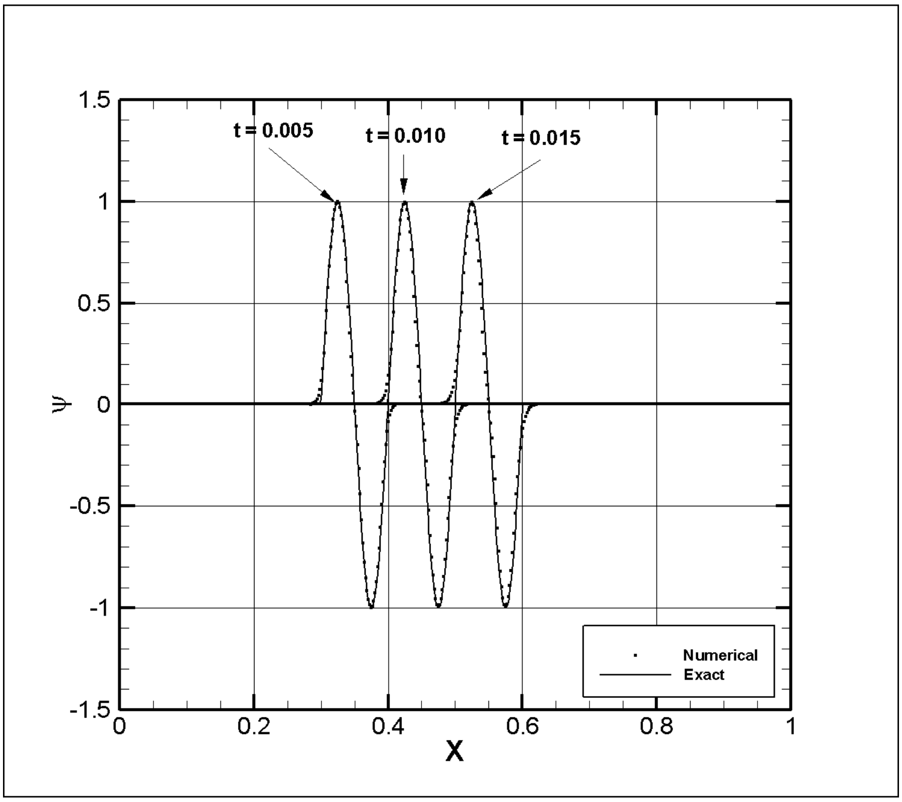

Figure 2.5 $\psi\left(x,t\right)$, linear wave equation solution

2.5.2 Inviscid Burgers Equation

The general form of the inviscid Burgers equation is given by:

$$\frac{\partial \psi}{\partial t}+\frac{\partial}{\partial x}\left(\frac{\psi^2}{2}\right)=0 \tag{2.48}$$

Equation (2.48) has a non-linear flux term proportional to the square of the variable ψ, which is very similar to the convection term in Euler equations. In this case, the flux function and its Jacobian are:

$$F=\frac{\psi^2}{2} \tag{2.49}$$

$$a=\psi \tag{2.50}$$

Assuming an initial shock discontinuity of the form:

$$\psi(x,0)=\begin{cases}\psi_L & 0.0 \le x \le 0.2\\ \psi_R & 0.2 < x \le 1.0\end{cases} \tag{2.51}$$

Figure 2.6 plots the initial condition given in (2.51). The analytical solution for this case is a moving shockwave in the positive x direction with speed c given by:

$$c=\frac{(\psi_L+\psi_R)}{2} \tag{2.52}$$

Letting $\psi_L=1.0$, $\psi_R=0.5$ and discretizing the domain from 0.0 to 1.0 with 500 nodes and using $\Delta t=5\times10^{-3}$, the following results have been obtained using FDV method. Figure 2.7 plots the numerical and exact solutions at different time steps. Again, FDV method has succeeded in resolving the solution without major changes in the profile of the convected non-linear discontinuity.

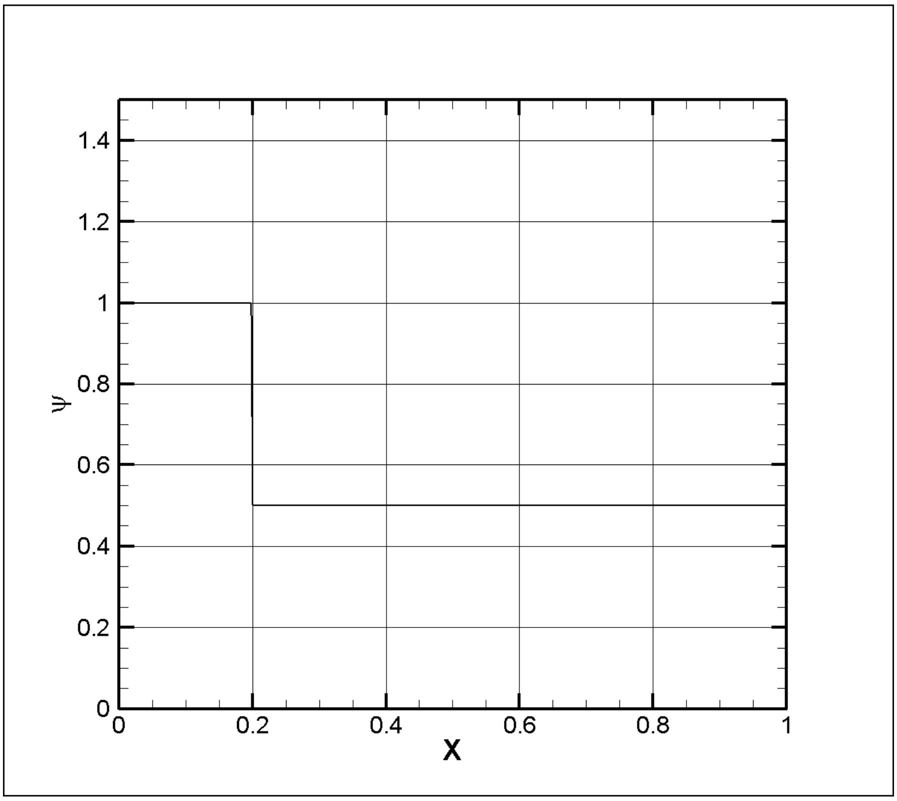

Figure 2.6 $\psi(x,0)$, inviscid Burgers equation initial condition

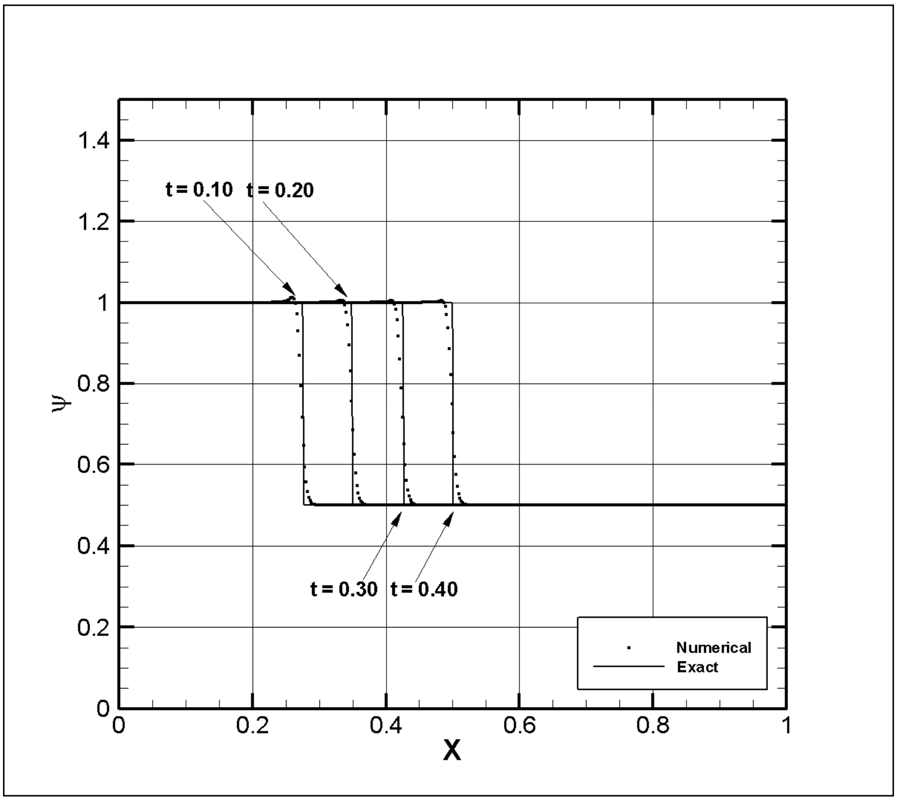

Figure 2.7 $\psi(x,t)$, inviscid Burgers equation solution

Chapter Three

FDV-FEM Compressible Euler Flow Solver

In this chapter the flowfield dependent variation method is applied to the two-dimensional compressible Euler flow equations. All theoretical issues are to be addressed in this chapter. This chapter is divided into five sections. A brief summary of Euler equations is given in section 3.1 with the non-dimensionalization procedure, followed by FDV method implementation in section 3.2. Section 3.3 is completely devoted to the analysis of the implicitness parameters, while section 3.4 addresses the finite element implementation for the resulting equations. Finally, section 3.5 introduces a brief analysis about the boundary conditions based on the theory of characteristics.

3.1 Euler Equations in 2-D Space

The system of continuity equation, x_i momentum equations, and energy equation in the case of inviscid compressible flow are called the Euler flow equations. These equations and their derivation from basic principles can be found in any fluid dynamics textbook such as [9]. The reader is especially referred to [3] for a sound mathematical analysis about the properties of this system of equations. The following is a listing of the system of Euler equations in the conservative form using the conserved fluxes:

$$\frac{\partial \rho}{\partial t}+\frac{\partial\left(\rho u_1\right)}{\partial x_1}+\frac{\partial\left(\rho u_2\right)}{\partial x_2}=0 \tag{3.1}$$

$$\frac{\partial\left(\rho u_1\right)}{\partial t}+\frac{\partial\left(\rho u_1^2+P\right)}{\partial x_1}+\frac{\partial\left(\rho u_1 u_2\right)}{\partial x_2}=0 \tag{3.2}$$

$$\frac{\partial\left(\rho u_2\right)}{\partial t}+\frac{\partial\left(\rho u_1 u_2\right)}{\partial x_1}+\frac{\partial\left(\rho u_2^2+P\right)}{\partial x_2}=0 \tag{3.3}$$

$$\frac{\partial\left(\rho e_t\right)}{\partial t}+\frac{\partial\left(\rho u_1 e_t+P u_1\right)}{\partial x_1}+\frac{\partial\left(\rho u_2 e_t+P u_2\right)}{\partial x_2}=0 \tag{3.4}$$

where; the total energy, e_t, is given in terms of the internal energy, e, and velocity components, u_i, by:

$$e_t = e + \frac{u_i u_i}{2} \tag{3.5}$$

and for a thermally perfect gas the internal energy can be written in terms of the temperature as follows:

$$e = c_v T = \frac{1}{(\gamma - 1)} RT \tag{3.6}$$

also from the equation of state for perfect gas we have:

$$P = \rho RT \tag{3.7}$$

substituting from equation (3.7) into equation (3.6), we get:

$$e = \frac{1}{(\gamma - 1)} \frac{P}{\rho} \tag{3.8}$$

substituting from equation (3.8) into equation (3.5) and solving for the pressure we get:

$$P = (\gamma - 1)\rho\left(e_t - \frac{u_1^2 + u_2^2}{2}\right) \tag{3.9}$$

It is clear that Euler equations are a system of first order nonlinear partial differential equations. This system is classified as hyperbolic, which means that characteristic lines and the propagation of waves is the most crucial issue in its solution. A special case from this system is the incompressible flow situation; this case rises naturally at low Mach numbers and the system changes to elliptic system. At sonic speed, the system changes to be parabolic. For a typical transonic case, where all subsonic, transonic, and supersonic flows exist, the system is of a mixed type.

It is important to remember that; despite the fact that a thorough understanding of the mathematics involved in the solution of such systems is important, the physical meaning of the various terms in the equations is an essential requirement for practical implementation. A correct mathematical solution may be obtained for such systems, but this solution may present a hypothetical case which has no interest in any application. So, engineers and academic researchers only have to be concerned with the solution of PDEs like Euler equations.

The numerical solution of the Euler system of equations in dimensional form typically involves operations between terms that vary by several orders of magnitude. This leads to a situation in which the numerical solution fails or become unstable as the computer floating point limits are exceeded. So, the governing equations are usually written in a non-dimensional form. When dimensionless forms are used the computations are maintained between 0.0 and 1.0. Also writing the system of Euler equations in non-dimensional form facilitates the generalization to embody large range of problems. Toward this end, the following non-dimensional variables are introduced:

$$\overset{*}{x}_i = \frac{x_i}{L}, \qquad \overset{*}{t} = \frac{t}{L/V_\infty}, \qquad \overset{*}{\rho} = \frac{\rho}{\rho_\infty}$$
$$\overset{*}{u}_i = \frac{u_i}{V_\infty}, \qquad \overset{*}{P} = \frac{P}{\rho_\infty V_\infty^2}, \qquad \overset{*}{T} = \frac{T}{T_\infty}$$
$$\overset{*}{e}_t = \frac{e_t}{V_\infty^2}, \qquad \overset{*}{R} = \frac{1}{\gamma M_\infty^2}$$

(3.10)

where; $L, V_\infty, \rho_\infty, T_\infty, M_\infty$ are the characteristic length, free stream velocity, free stream density, free stream temperature, and free stream Mach number, respectively.

Multiplying equation (3.1) by $L/V_\infty \rho_\infty$, equations (3.2) and (3.3) by $L/V_\infty^2 \rho_\infty$, and equation (3.4) by $L/V_\infty^3 \rho_\infty$ we get:

$$\frac{\partial \overset{*}{\rho}}{\partial \overset{*}{t}} + \frac{\partial \left(\overset{*}{\rho} \overset{*}{u}_1 \right)}{\partial \overset{*}{x}_1} + \frac{\partial \left(\overset{*}{\rho} \overset{*}{u}_2 \right)}{\partial \overset{*}{x}_2} = 0 \qquad (3.11)$$

$$\frac{\partial\left(\overset{*}{\rho}\overset{*}{u}_1\right)}{\partial \overset{*}{t}}+\frac{\partial\left(\overset{*}{\rho}\overset{*}{u}_1^2+\overset{*}{P}\right)}{\partial \overset{*}{x}_1}+\frac{\partial\left(\overset{*}{\rho}\overset{*}{u}_1\overset{*}{u}_2\right)}{\partial \overset{*}{x}_2}=0 \tag{3.12}$$

$$\frac{\partial\left(\overset{*}{\rho}\overset{*}{u}_2\right)}{\partial \overset{*}{t}}+\frac{\partial\left(\overset{*}{\rho}\overset{*}{u}_1\overset{*}{u}_2\right)}{\partial \overset{*}{x}_1}+\frac{\partial\left(\overset{*}{\rho}\overset{*}{u}_2^2+\overset{*}{P}\right)}{\partial \overset{*}{x}_2}=0 \tag{3.13}$$

$$\frac{\partial\left(\overset{*}{\rho}\overset{*}{e}_t\right)}{\partial \overset{*}{t}}+\frac{\partial\left(\overset{*}{\rho}\overset{*}{u}_1\overset{*}{e}_t+\overset{*}{P}\overset{*}{u}_1\right)}{\partial \overset{*}{x}_1}+\frac{\partial\left(\overset{*}{\rho}\overset{*}{u}_2\overset{*}{e}_t+\overset{*}{P}\overset{*}{u}_2\right)}{\partial \overset{*}{x}_2}+=0 \tag{3.14}$$

Closer look to equations (3.1)-(3.4) and equations (3.11)-(3.14) will reveal the fact that both the dimensional and non-dimensional forms are identical when replacing each dimensional variable with its non-dimensional counterpart.

So, from now on the superscript ($\overset{*}{}$) will be omitted from the analysis taking into consideration that proper non-dimensionalization procedures should be followed to gain the convenience of dealing with non-dimensional variables. Introducing the conservation variables vector $\mathbf{U}$, convective flux vector $\mathbf{F}_i$ and using the indicial notation, the compressible Euler flow equations can be written as follows:

$$\frac{\partial \mathbf{U}}{\partial t}+\frac{\partial \mathbf{F}_i}{\partial x_i}=0 \tag{3.15}$$

where; in two-dimensional space $\left(X_1-X_2\right)$ the conservation variable vector is given by:

$$\mathbf{U}=\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix}=\begin{bmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho e_t \end{bmatrix} \tag{3.16}$$

and the convection flux vector is given by:

$$\mathbf{F}_i = \begin{Bmatrix} \rho u_i \\ \rho u_i u_1 + P\delta_{1i} \\ \rho u_i u_2 + P\delta_{2i} \\ \rho u_i e_t + Pu_i \end{Bmatrix} = \begin{bmatrix} \rho u_1 & \rho u_2 \\ \rho u_1^2 + P & \rho u_2 u_1 \\ \rho u_1 u_2 & \rho u_2^2 + P \\ \rho u_1 e_t + Pu_1 & \rho u_2 e_t + Pu_2 \end{bmatrix} \tag{3.17}$$

In order to make the system of Euler equations ready for the FDV implementation, the whole system has to be written in terms of the conservation variables. So, rewriting equation (3.15) in a quasi-linear form we get:

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{a}_i \frac{\partial \mathbf{U}}{\partial x_i} = 0 \tag{3.18}$$

where; $\mathbf{a}_i$ is the convection Jacobian tensor. The reader is referred to Appendix A for more details about the derivation of this tensor and the various forms to present it.

3.2 FDV Formulation for the 2-D Euler Equations

Following the same three steps presented in Chapter Two to get the residual form of the compressible Euler equations given in equation (3.15) will result in the FDV formulation for the system of Euler equations. In 3.2.1 a special form of Taylor expansion is introduced for the vector of the conservation variables. While in 3.2.2 the time and spatial derivatives are interchanged. Finally, in 3.2.3 the system is written in the residual form.

3.2.1 Special Taylor Expansion

Expanding $\mathbf{U}^{n+1}$ around $\mathbf{U}^n$ in a special form of Taylor expansion using both the explicit and implicit formulations up to and including the second order derivative as mentioned in 2.1 we obtain:

$$\mathbf{U}^{n+1} = \mathbf{U}^n + \Delta t \frac{\partial \mathbf{U}^{n+s_1}}{\partial t} + \frac{(\Delta t)^2}{2} \frac{\partial^2 \mathbf{U}^{n+s_2}}{\partial t^2} + O\left((\Delta t)^3\right) \tag{3.19}$$

where;

$$\frac{\partial \mathbf{U}^{n+s_1}}{\partial t} = \frac{\partial \mathbf{U}^n}{\partial t} + s_1 \frac{\partial \Delta \mathbf{U}^{n+1}}{\partial t} \qquad 0 \le s_1 \le 1$$

$$\frac{\partial^2 \mathbf{U}^{n+s_2}}{\partial t^2} = \frac{\partial^2 \mathbf{U}^n}{\partial t^2} + s_2 \frac{\partial^2 \Delta \mathbf{U}^{n+1}}{\partial t^2} \qquad 0 \le s_2 \le 1$$

$$\Delta \mathbf{U}^{n+1} = \mathbf{U}^{n+1} - \mathbf{U}^n$$

The implicitness parameters, s_1 and s_2, are flowfield dependent. They may gain their physical meaning by being calculated from the flow variables' fluctuations. The various proposed methods to calculate these parameters and their meaning are delayed to section 3.3.

3.2.2 Interchanging Time and Spatial Derivatives

Substituting from equation (3.15) into the Taylor expansion by interchanging the time derivatives with the spatial derivatives we get:

$$\begin{aligned} \Delta \mathbf{U}^{n+1} = \Delta t & \left(-\frac{\partial \mathbf{F}_i^n}{\partial x_i} - s_1 \frac{\partial \Delta \mathbf{F}_i^{n+1}}{\partial x_i} \right) + \\ & + \frac{(\Delta t)^2}{2} \mathbf{a}_i \frac{\partial}{\partial x_i} \left(\frac{\partial \mathbf{F}_j^n}{\partial x_j} \right) + \\ & + \frac{(\Delta t)^2}{2} s_2 \mathbf{a}_i \frac{\partial}{\partial x_i} \left(\frac{\partial \Delta \mathbf{F}_j^{n+1}}{\partial x_j} \right) + O\left((\Delta t)^3 \right) \end{aligned} \tag{3.20}$$

It is evident that the same technique that has been adapted in Chapter Two is also used here, so many minor details have been omitted to avoid the repetition of redundant mathematical steps. The major difference is that; here the formulation is applied to a system of equations rather than to a single scalar function. However, the use of the indicial notation is very useful to retain the same mathematical steps.

3.2.3 Residual Form

Rewriting equation (3.20) in the residual form and substituting all the Δ terms with their Jacobian equivalents we obtain:

$$\Delta\mathbf{U}^{n+1} + \Delta t\, s_1 \mathbf{a}_i \frac{\partial}{\partial x_i}\left(\Delta\mathbf{U}^{n+1}\right) - \frac{(\Delta t)^2}{2} s_2 \mathbf{a}_i \mathbf{a}_j \frac{\partial^2}{\partial x_i \partial x_j}\left(\Delta\mathbf{U}^{n+1}\right) +$$
$$+\Delta t \frac{\partial \mathbf{F}_i^n}{\partial x_i} - \frac{(\Delta t)^2}{2} \mathbf{a}_i \frac{\partial}{\partial x_i}\left(\frac{\partial \mathbf{F}_j^n}{\partial x_j}\right) + O\left((\Delta t)^3\right) = 0$$

(3.21)

rearranging equation (3.21) we get:

$$\Delta\mathbf{U}^{n+1} + \mathbf{D}_i^n \frac{\partial}{\partial x_i}\left(\Delta\mathbf{U}^{n+1}\right) + \mathbf{E}_{ij}^n \frac{\partial^2}{\partial x_i \partial x_j}\left(\Delta\mathbf{U}^{n+1}\right) + \mathbf{Q}^n + O\left(\Delta t^3\right) = 0 \quad (3.22)$$

where;

$$\mathbf{D}_i^n = \Delta t\, s_1 \mathbf{a}_i$$
$$\mathbf{E}_{ij}^n = -\frac{(\Delta t)^2}{2} s_2 \mathbf{a}_i \mathbf{a}_j$$
$$\mathbf{Q}^n = \Delta t \frac{\partial}{\partial x_i}\left(\mathbf{F}_i^n\right) - \frac{(\Delta t)^2}{2} \mathbf{a}_i \frac{\partial^2}{\partial x_i \partial x_j}\left(\mathbf{F}_j^n\right)$$

(3.23)

The superscript n is used to emphasize that these terms are calculated at the n^{th} time step. Equation (3.22) along with equation (3.23) represents the FDV method treatment for the system of Euler equations given in (3.15). Equation (3.22) is ready to be solved for the time change of the conservation variables $\Delta\mathbf{U}^{n+1}$. It can be solved using any discretization technique in both FEM and FDM. Since all necessary numerical dissipation and stabilizing elements that are required for numerical stability and solution accuracy is already impeded, no additional modifications are necessary from the discretization technique that will be used.

As mentioned in 2.4, any available FEM or FDM scheme may rise as a special case from FDV method when fixing the FDV parameters to certain values. For instant, setting $s_1 = 0.0, s_2 = 1.0$ the so called Generalized Taylor-Galerkin method is recovered, and setting $s_1 = s_2 = 0.0(1.0)$ the explicit (implicit) Euler scheme is obtained. Also most of the FDM schemes can be obtained as special cases from FDV method. Setting $s_1 = s_2 = 0.0$ and letting:

$$\mathbf{a}_{j+\frac{1}{2}} = \mathbf{a}_{j-\frac{1}{2}} = \mathbf{a}$$

the Lax-Wendroff scheme without artificial viscosity is recovered as a special case. For a comprehensive comparison between FDV method and various methods in both the FEM and FDM, the reader is referred to [55]. As summary, the FDV method is proved to reduce as a special case to some of the FDM schemes:

- Lax-Wendroff scheme with and without viscosity.
- Explicit and implicit McCormack schemes.
- First-order upwind scheme.
- PISO and SIMPLE.

Also some special cases of the FDV method are famous FEM methods. For instant:

- Generalized Taylor-Galerkin (GTG).
- Generalized Petrov-Galerkin (GPG).
- Characteristic-based Zienkiewicz-Codina.

The reader is especially referred to [54] for mathematical proofs that verifies analogies and relations between these various methods and the FDV formulation.

3.3 FDV Implicitness Parameters

The FDV implicitness parameters are the heart of the FDV method. Under or over estimation of these parameters may result in an unsatisfactory performance of the solver, or even numerical failure for the entire solution. These parameters allow the FDV method to adapt itself to each flow situation and to supply each element with its unique and distinctive numerical scheme that guarantees both accurate solution and sufficient numerical dissipation for stability.

The first order FDV parameter s_1 controls all the high-gradient phenomena such as shockwaves. This parameter is to be calculated from the changes in the flowfield properties that affect the convection phenomena. While, the second order FDV parameter s_2 controls the numerical dissipation that is required for numerical stability. This is evident from its association with the second order time derivative in Taylor expansion.

When the value of s_1 reaches the zero limit, this means that convection effect is small or nearly negligible and the FDV method alters itself to take this effect by zeroing all the s_1 terms which represents convection. On the other hand, s_2 is believed to be exponentially proportional to the value of s_1. This is due to the fact that flowfield regions those suffer from high gradients (high s_1) adequately require high numerical dissipation (high s_2) and vise versa. The various methods available to calculate this parameter from the flowfield are presented in 3.3.1 and 3.3.2.

3.3.1 Original FDV Theory

The original FDV theory that has been proposed by T. J. Chung in [1] for calculating s_1 and s_2 from the current flowfield is given by:

$$s_1 = \begin{cases} \min(r,1) & r > \varepsilon \\ 0 & r < \varepsilon, M_{\min} \neq 0 \\ 1 & M_{\min} = 0 \end{cases} \tag{3.24}$$

$$s_2 = \frac{1}{2}\left(1 + s_1^{\eta}\right), \qquad 0.05 < \eta < 0.2 \tag{3.25}$$

where; $r = \sqrt{M_{\max}^2 - M_{\min}^2} \Big/ M_{\min}$, ε is a small number, and $M_{\min}, M_{\max}$ are defined by the minimum and maximum values of the Mach number in the neighboring nodes (i.e. between the element nodes).

Relation (3.25) is plotted in Figure 3.1 for wide span for η. It is evident from Figure 3.1 that as η gets smaller, the value of s_2 reaches the unity faster, which means higher artificial viscosity. Usually the value of 0.10 is typical for most applications.

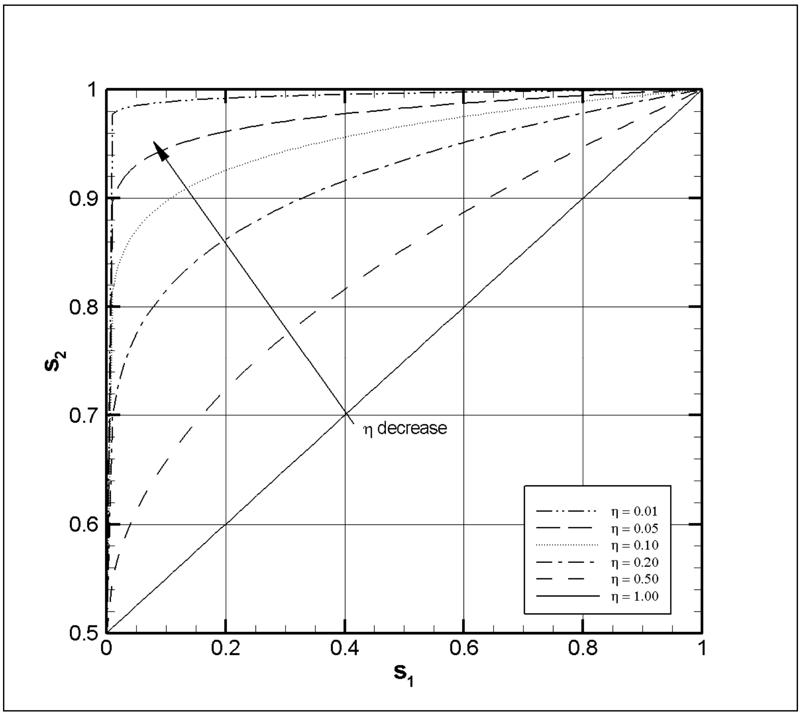

Figure 3.1 The relation between s_1 and s_2

It is clear from the definition of s_1 that it is associated with the changes of Mach number values in the flowfield. Mach number is a major factor in the analysis of convection flows and it distinguishes between various flow-situations which can be encountered in the same domain of study. For instant when there is no change in the Mach number, equation (3.24) reads $s_1 = 0.0$. Automatically the FDV formulation alters itself to take this into account and switches off all the s_1 terms. When the minimum Mach reaches the incompressible limit, the formulation generate terms that is equivalent to the pressure correction schemes like Poisson equation.

As s_1 gets higher the need for numerical dissipation increases and hence high s_2 values are logical. So from the definition given in (3.25), s_2 is exponentially proportional to s_1 and starting from the value of 0.5 to ensure numerical dissipation in all situations.

3.3.2 Modified FDV Theory

In [56] a modification to equation (3.24) was proposed. This modification was named as the Modified FDV (MFDV) method. In MFDV method, the relation used to recover the first order implicitness parameter from the current flowfield data was modified to give this parameter a deeper physical meaning. This can be accomplished by setting s_1 to be proportional to the first order derivative of the Mach number This is quite equivalent to the definition of s_1 given in [52]-[55]. Also to retain the dimensional similarity the result is multiplied by the elemental characteristic length and scaled by the minimal Mach number between the element nodes. The proposed modification is given by:

$$s_1 = \min(r, 1) \tag{3.26}$$

where; $r = L^* \left|\underline{\nabla} M\right| / M_{\min}$ and the other implicitness parameter (s_2) is calculated as in Eqn. (3.25).

As mentioned in [56], the original formula proposed by T. J. Chung and his co-workers is admitted to applicable and effective and this modification has been considered as a step towards deeper understanding of the role of the implicitness parameters. From the definition given in (3.26), the regions with small variations in the Mach number the derivative will reach the zero value. This gives s_1 the signal to switch off all its terms in the regions with smooth or no variations. As the derivatives gets steeper s_1 gets higher. In the limiting case when the minimum Mach number is zero, r will be infinity and the limiting minimum function sets s_1 to its higher value which is unity.

All the trends that can be obtained from (3.24) can be observed in (3.26). So, it is believed that FDV and MFDV methods will behave the same. But crucial benefit is evident from the MFDV. This benefit is that; s_1 in MFDV has a deeper physical meaning. This facilitates the ability to predict the response of the method to certain flow domain and helps in preparing the problem at hand for the solver.

Also, with the aid of the MFDV meaning the FDV method is no longer limited to solve flow problems. The MFDV method can be easily extended to any application that requires the solution of PDEs like the Euler equations. In this case the s_1 parameter will be associated with the derivatives of the solution field variables. This has been already made in section 2.5, where the FDV method was used to solve both the wave equation and inviscid Burgers equations.

3.4 Finite Element Implementation

Applying the standard Galerkin method, this is equivalent to the central difference schemes in FDMs, by integrating the inner product of the residual with the shape function as the weighting function. By applying this we get:

$$\int_{\Omega} \Phi_{\alpha} \mathbf{R}\left(\mathbf{U}, \mathbf{F}_i\right) d\Omega = 0 \tag{3.27}$$

where; $\mathbf{R}\left(\mathbf{U}, \mathbf{F}_i\right)$ is the residual of (3.22). In what follows the order of the solution error will be omitted for convenience. Substituting from (3.22) into (3.27), we obtain:

$$\int_{\Omega} \Phi_{\alpha} \left(\Delta\mathbf{U}^{n+1} + \mathbf{D}_i^n \frac{\partial}{\partial x_i}\left(\Delta\mathbf{U}^{n+1}\right) + \mathbf{E}_{ij}^n \frac{\partial^2}{\partial x_i \partial x_j}\left(\Delta\mathbf{U}^{n+1}\right) + \mathbf{Q}^n \right) d\Omega = 0$$

(3.28)

To calculate the integration given in equation (3.28); a certain shape function is to be assumed for the variation of the unknown $\Delta\mathbf{U}^{n+1}$. The second step is to integrate by parts all the differential terms. The next two subsections summarize these steps for the 2-D bilinear quadrilateral element shape function.

3.4.1 2-D Bilinear Quadrilateral Element Shape Function

Consider the 2-D element shown in Figure 3.2. The integrations encountered in finite element analysis can be either in terms of the global coordinates x_i or the natural coordinates ξ_i. The later is used because it provides a simpler treatment using Gauss quadrature.

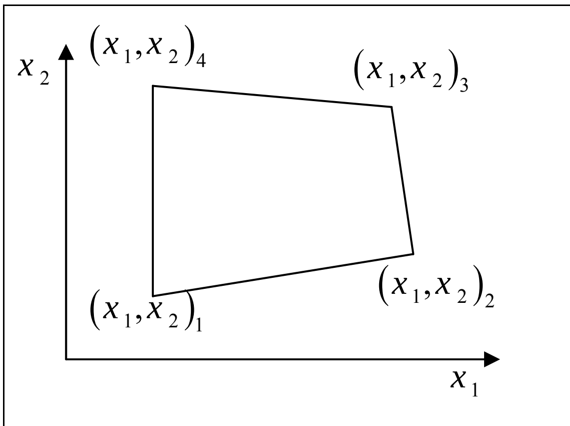

Figure 3.2 Actual 2-D quadrilateral element in global coordinates

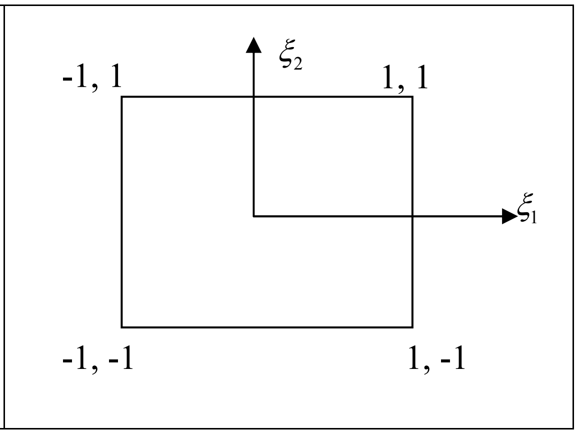

Figure 3.3 Transformed 2-D element in natural coordinates

Both coordinate systems are shown in Figure 3.2 and Figure 3.3, respectively. Any scalar function u can be written as follows:

$$u = \beta_1 + \beta_2 \xi_1 + \beta_3 \xi_2 + \beta_4 \xi_1 \xi_2 \tag{3.29}$$

or in a matrix form as follows:

$$u = \begin{bmatrix} 1 & \xi_1 & \xi_2 & \xi_1\xi_2 \end{bmatrix} \begin{bmatrix} \beta_1 & \beta_2 & \beta_3 & \beta_4 \end{bmatrix}^{\mathrm{T}} \tag{3.30}$$

where; β_1, β_2, β_3 and β_4 are unknown constants. Using the values of the element nodes' coordinates those summarized in Table 2 to determine these constants.

Table 2 Bilinear 2D element nodal coordinates in natural coordinates

Node	ξ_1	ξ_2	u
1	-1.0	-1.0	u_1
2	1.0	-1.0	u_2
3	1.0	1.0	u_3
4	-1.0	1.0	u_4

Substituting from Table 2 into (3.30), we get:

$$\begin{Bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{Bmatrix} = \begin{bmatrix} 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{Bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{Bmatrix} \tag{3.31}$$

solving for β_1 to β_4, we get:

$$\begin{Bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{Bmatrix} = \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ -1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{Bmatrix} \tag{3.32}$$

substituting from equation (3.32) into equation (3.30), we obtain:

$$u = \begin{bmatrix} 1 & \xi_1 & \xi_2 & \xi_1\xi_2 \end{bmatrix} \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ -1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{Bmatrix} \tag{3.33}$$

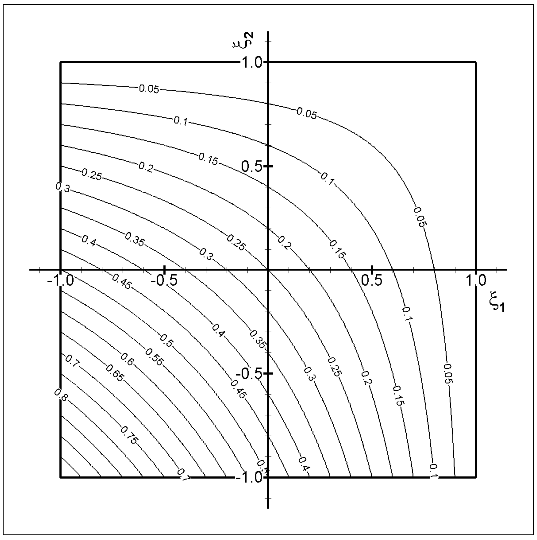

Figure 3.4 First shape function contours

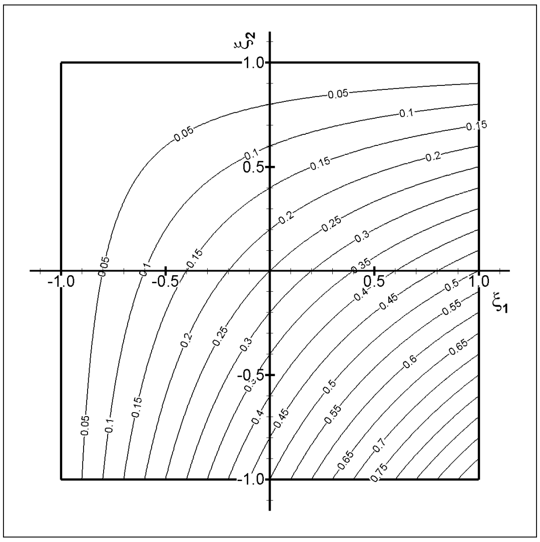

Figure 3.5 Second shape function contours

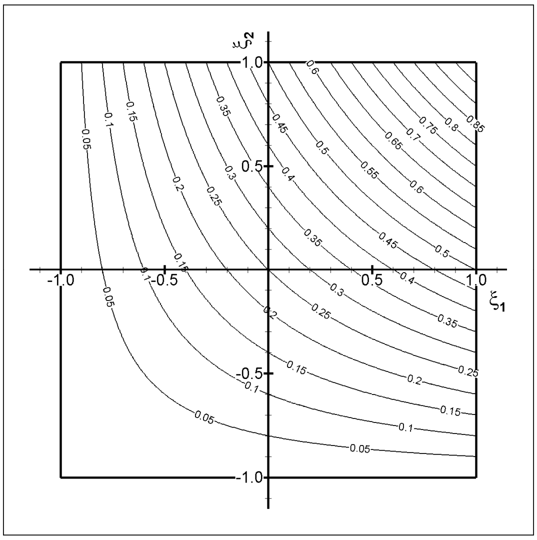

Figure 3.6 Third shape function contours

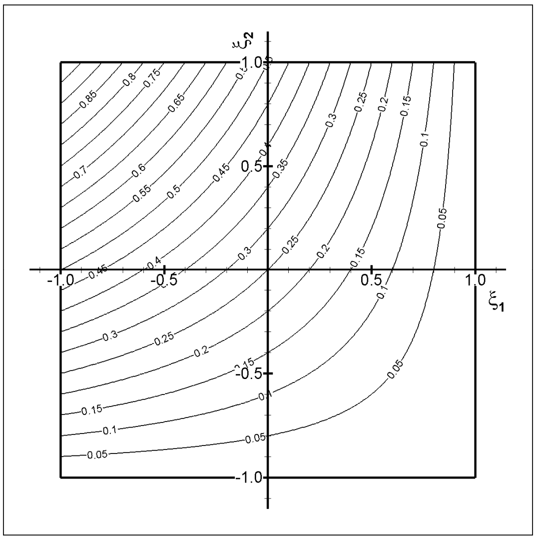

Figure 3.7 Fourth shape function contours

$$u = \frac{1}{4}\left[1-\xi_1-\xi_2+\xi_1\xi_2 \quad 1+\xi_1-\xi_2-\xi_1\xi_2 \quad 1+\xi_1+\xi_2+\xi_1\xi_2 \quad 1-\xi_1+\xi_2-\xi_1\xi_2\right]\begin{Bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{Bmatrix}$$

(3.34)

and renaming these terms to be as follows:

$$\begin{aligned} \Phi_1 &= \frac{1}{4}(1-\xi_1)(1-\xi_2), \qquad & \Phi_2 &= \frac{1}{4}(1+\xi_1)(1-\xi_2) \\ \Phi_3 &= \frac{1}{4}(1+\xi_1)(1+\xi_2), \qquad & \Phi_4 &= \frac{1}{4}(1-\xi_1)(1+\xi_2) \end{aligned} \tag{3.35}$$

where; Φ_i is the i$^{\text{th}}$ shape function. Figure 3.4, Figure 3.5, Figure 3.6, and Figure 3.7 display contour plots for the 2-D linear shape functions in the natural coordinates.

Now, using the shape functions, any scalar variable can be written as follows:

$$u = \left[\Phi_1 \quad \Phi_2 \quad \Phi_3 \quad \Phi_4\right]\left[u_1 \quad u_2 \quad u_3 \quad u_4\right]^{\mathrm{T}} \tag{3.36}$$

Usually finite element formulations contain differentiations w.r.t. x_i, so it will be quite useful at this point to get these derivatives in order to be ready whenever it is needed. To get these derivatives, the derivatives w.r.t ξ_i will be introduced and the transformation will be done using the transformation Jacobian. The ξ_i derivatives are given by:

$$\begin{aligned} \Phi_{1\xi_1} &= \frac{-1}{4}(1-\xi_2), \qquad & \Phi_{2\xi_1} &= \frac{1}{4}(1-\xi_2) \\ \Phi_{3\xi_1} &= \frac{1}{4}(1+\xi_2), \qquad & \Phi_{4\xi_1} &= \frac{-1}{4}(1+\xi_2) \\ \Phi_{1\xi_2} &= \frac{-1}{4}(1-\xi_1), \qquad & \Phi_{2\xi_2} &= \frac{-1}{4}(1+\xi_1) \\ \Phi_{3\xi_2} &= \frac{1}{4}(1+\xi_1), \qquad & \Phi_{4\xi_2} &= \frac{1}{4}(1-\xi_1) \end{aligned}$$

(3.37)

this leads to the following equation:

$$u_{\xi_i} = \left[\Phi_{1,\xi_i} \quad \Phi_{2,\xi_i} \quad \Phi_{3,\xi_i} \quad \Phi_{4,\xi_i}\right]\left[u_1 \quad u_2 \quad u_3 \quad u_4\right]^{\mathrm{T}} \tag{3.38}$$

To get differentiations with respect to x_i there should be a relation between the two coordinate systems; this can be done using the isoparametric element condition. I.e. use the same interpolation function to describe x_i we get:

$$x_i = [\Phi_1 \quad \Phi_2 \quad \Phi_3 \quad \Phi_4]\left[(x_i)_1 \quad (x_i)_2 \quad (x_i)_3 \quad (x_i)_4\right]^{\mathrm{T}} \tag{3.39}$$

furthermore the following relations are obtained:

$$\begin{aligned}\frac{\partial \Phi_i}{\partial \xi_1} &= \frac{\partial \Phi_i}{\partial x_1}\frac{\partial x_1}{\partial \xi_1} + \frac{\partial \Phi_i}{\partial x_2}\frac{\partial x_2}{\partial \xi_1}\\ \frac{\partial \Phi_i}{\partial \xi_2} &= \frac{\partial \Phi_i}{\partial x_1}\frac{\partial x_1}{\partial \xi_2} + \frac{\partial \Phi_i}{\partial x_2}\frac{\partial x_2}{\partial \xi_2}\end{aligned} \tag{3.40}$$

or in a matrix form as follows:

$$\begin{Bmatrix}\dfrac{\partial \Phi_i}{\partial \xi_1}\\ \dfrac{\partial \Phi_i}{\partial \xi_2}\end{Bmatrix} = \begin{bmatrix}\dfrac{\partial x_1}{\partial \xi_1} & \dfrac{\partial x_2}{\partial \xi_1}\\ \dfrac{\partial x_1}{\partial \xi_2} & \dfrac{\partial x_2}{\partial \xi_2}\end{bmatrix}\begin{Bmatrix}\dfrac{\partial \Phi_i}{\partial x_1}\\ \dfrac{\partial \Phi_i}{\partial x_2}\end{Bmatrix} \tag{3.41}$$

from equation (3.41) the transformation Jacobian is given by:

$$Jac = \begin{bmatrix}\dfrac{\partial x_1}{\partial \xi_1} & \dfrac{\partial x_2}{\partial \xi_1}\\ \dfrac{\partial x_1}{\partial \xi_2} & \dfrac{\partial x_2}{\partial \xi_2}\end{bmatrix} \tag{3.42}$$

and using matrix inversion we get:

$$\begin{Bmatrix}\dfrac{\partial \Phi_i}{\partial x_1}\\ \dfrac{\partial \Phi_i}{\partial x_2}\end{Bmatrix} = \frac{1}{|Jac|}\begin{bmatrix}\dfrac{\partial x_2}{\partial \xi_2} & -\dfrac{\partial x_2}{\partial \xi_1}\\ -\dfrac{\partial x_1}{\partial \xi_2} & \dfrac{\partial x_1}{\partial \xi_1}\end{bmatrix}\begin{Bmatrix}\dfrac{\partial \Phi_i}{\partial \xi_1}\\ \dfrac{\partial \Phi_i}{\partial \xi_2}\end{Bmatrix} \tag{3.43}$$

this leads to the following relation:

$$u_{x_i} = \left[\Phi_{1,x_i} \quad \Phi_{2,x_i} \quad \Phi_{3,x_i} \quad \Phi_{4,x_i}\right][u_1 \quad u_2 \quad u_3 \quad u_4]^{\mathrm{T}} \tag{3.44}$$

Equation (3.44) is to be used whenever a derivative in x_i is required. In order to use it the determinate of the transformation Jacobian defined in (3.42) should be calculated and this calls for the calculation of the derivatives in ξ_i. Those derivatives can be computed from equation (3.38).

3.4.2 Integration by Parts and Element Equations

Expressing the conservation variables as a linear combination of the trial functions Φ_α we have:

$$\mathbf{U}(\mathbf{x},t) = \Phi_\alpha(\mathbf{x})\mathbf{U}_\alpha(t) \tag{3.45}$$

and substituting from equation (3.45) into equation (3.28) and performing integration by parts we get:

$$\begin{aligned}
&\int_\Omega \Phi_\alpha \Phi_\beta \Delta\mathbf{U}_\beta^{n+1} d\Omega + \int_\Gamma \mathbf{D}_i^n \overset{*}{\Phi}_\alpha \overset{*}{\Phi}_\beta \Delta\mathbf{U}_\beta^{n+1} n_i \, d\Gamma - \int_\Omega \mathbf{D}_i^n \Phi_{\alpha,i} \Phi_\beta \Delta\mathbf{U}_\beta^{n+1} d\Omega + \\
&+ \int_\Gamma \mathbf{E}_{ij}^n \overset{*}{\Phi}_\alpha \overset{*}{\Phi}_{\beta,j} \Delta\mathbf{U}_\beta^{n+1} n_i \, d\Gamma - \int_\Omega \mathbf{E}_{ij}^n \Phi_{\alpha,i} \Phi_{\beta,j} \Delta\mathbf{U}_\beta^{n+1} d\Omega + \\
&+ \Delta t \int_\Gamma \overset{*}{\Phi}_\alpha \overset{*}{\mathbf{F}}{}_i^n n_i \, d\Gamma - \Delta t \int_\Omega \Phi_{\alpha,i} \mathbf{F}_i^n d\Omega + \\
&- \frac{(\Delta t)^2}{2} \int_\Gamma \mathbf{a}_i \overset{*}{\Phi}_\alpha \frac{\partial}{\partial x_j}\left(\overset{*}{\mathbf{F}}{}_j^n\right) n_i \, d\Gamma + \frac{(\Delta t)^2}{2} \int_\Omega \mathbf{a}_i \Phi_{\alpha,i} \frac{\partial}{\partial x_j}\left(\mathbf{F}_j^n\right) d\Omega = 0
\end{aligned} \tag{3.46}$$

rearranging these terms and substituting from equation (3.23) we have:

$$\left(\mathbf{A}_{\alpha\beta}^n + \mathbf{B}_{\alpha\beta}^n\right)\Delta\mathbf{U}_\beta^{n+1} = \mathbf{H}_\alpha^n + \mathbf{N}_\alpha^n \tag{3.47}$$

where;

$$\begin{aligned}
\mathbf{A}_{\alpha\beta}^n &= \int_\Omega \left(\Phi_\alpha \Phi_\beta - \Delta t \, s_1 \mathbf{a}_i \Phi_{\alpha,i} \Phi_\beta + \frac{(\Delta t)^2}{2} s_2 \mathbf{a}_i \mathbf{a}_j \Phi_{\alpha,i} \Phi_{\beta,j} \right) d\Omega \\
\mathbf{B}_{\alpha\beta}^n &= \int_\Gamma \left(\Delta t \, s_1 \mathbf{a}_i \overset{*}{\Phi}_\alpha \overset{*}{\Phi}_\beta - \frac{(\Delta t)^2}{2} s_2 \mathbf{a}_i \mathbf{a}_j \overset{*}{\Phi}_\alpha \overset{*}{\Phi}_{\beta,j} \right) n_i \, d\Gamma \\
\mathbf{H}_\alpha^n &= \int_\Omega \left(\Delta t \Phi_{\alpha,i} \mathbf{F}_i^n - \frac{(\Delta t)^2}{2} \mathbf{a}_i \Phi_{\alpha,i} \frac{\partial \mathbf{F}_j^n}{\partial x_j} \right) d\Omega \\
\mathbf{N}_\alpha^n &= \int_\Gamma \left(-\Delta t \, \overset{*}{\Phi}_\alpha \overset{*}{\mathbf{F}}{}_i^n + \frac{(\Delta t)^2}{2} \mathbf{a}_i \overset{*}{\Phi}_\alpha \frac{\partial}{\partial x_j}\left(\overset{*}{\mathbf{F}}{}_j^n\right) \right) n_i \, d\Gamma
\end{aligned} \tag{3.48}$$

Equation (3.47) along with (3.48) are called the element equations. They represent the first building block in the Euler solver. All other modules that will be included in the solver are to serve these equations. It is clear that $\mathbf{B}^n_{\alpha\beta}$ and $\mathbf{N}^n_{\alpha}$ terms will cancel each other at the inter-elements boundaries. These terms will only have effect on elements at the boundaries.

After assembling all the element equations for all elements in the solution domain, a large sparse system will result. This system has to be solved to obtain the time change $\Delta\mathbf{U}^{n+1}_{\beta}$. The preconditioned GMRES sparse matrix solver discussed in Appendix B will be employed for this purpose.

3.4.3 Discontinuity-Capturing Operator

As will be discussed in section 4.1, the FDV/MFDV methods are capable of introducing a satisfactory solution without the use of any further treatment. That is why only standard Galerkin method has been chosen. However, for strong shocks the solution suffers from Gibbs type errors at small time steps (small CFL values) that spoil the flowfield near the shock. To overcome this situation a Discontinuity-Capturing Operator (DCO) may be used. Using the same DCO presented in [31]-[34], the FDV/MFDV methods presented a good performance. It will be proved by numerical experimentation that using this DCO will not affect the solution accuracy, it just guarantees the ripple-free solution. The form of this DCO is given by:

$$DCO = \int_{\Omega} \delta \frac{\partial \Phi_\alpha}{\partial x_i} \frac{\partial \Phi_\beta}{\partial x_i} \mathbf{U}_\beta d\Omega \tag{3.49}$$

where; δ is the Discontinuity-capturing Factor (DCF) given by:

$$\delta = \left(\frac{C^{-1}_{rs} a_{itr} a_{jus} U_{t,i} U_{u,j}}{C_{uv} g^{mn} U_{v,m} U_{w,n}} \right)^{1/2} \tag{3.50}$$

where; $\mathbf{C}$ is the entropy variables' Jacobian, $\mathbf{a}_i$ is the convection Jacobian, and g^{mn} is the contravariant metric tensor given by:

$$g^{mn} = \frac{\partial \xi_m}{\partial x_p} \frac{\partial \xi_n}{\partial x_p} \tag{3.51}$$

In equations (3.50) and (3.51), the indices i,j,k,m,n,p refer to the special coordinates (having the range 1,2 in 2-D case) and r,s,t,u,v,w refer to the equation number (having the range 1,2,3,4 in 2-D case). The reader is referred to Appendix A for more details about Jacobian tensors.

Adding the DCO given in equation (3.49) to the standard Galerkin integral given in equation (3.27), we get:

$$\int_{\Omega} \Phi_\alpha \mathbf{R}(\mathbf{U},\mathbf{F}_i) d\Omega + \int_{\Omega} \delta \frac{\partial \Phi_\alpha}{\partial x_i}\frac{\partial \Phi_\beta}{\partial x_i}\mathbf{U}_\beta d\Omega = 0 \tag{3.52}$$

Where; δ is the DCF given in equation (3.50).

The term $\mathbf{U}_\beta$ can be either in "n" or "$n+1$" time steps, which corresponds to either the explicit or implicit forms of the DCO, respectively. Since strong interactions are expected, the implicit form is adopted here. This modification will alter the element equations given in (3.47) and (3.48) to be as follows:

$$\left(\mathbf{A}^n_{\alpha\beta} + \mathbf{B}^n_{\alpha\beta}\right)\Delta\mathbf{U}^{n+1}_\beta = \mathbf{H}^n_\alpha + \mathbf{N}^n_\alpha \tag{3.53}$$

where;

$$\mathbf{A}^n_{\alpha\beta} = \int_{\Omega}\left(\Phi_\alpha\Phi_\beta - \Delta t\, s_1 \mathbf{a}_i\, \Phi_{\alpha,i}\Phi_\beta + \frac{(\Delta t)^2}{2} s_2 \mathbf{a}_i \mathbf{a}_j \Phi_{\alpha,i}\Phi_{\beta,j} + \delta \frac{\partial \Phi_\alpha}{\partial x_i}\frac{\partial \Phi_\beta}{\partial x_i}\right) d\Omega$$

$$\mathbf{B}^n_{\alpha\beta} = \int_{\Gamma}\left(\Delta t\, s_1 \mathbf{a}_i\, \overset{*}{\Phi}_\alpha \overset{*}{\Phi}_\beta - \frac{(\Delta t)^2}{2} s_2 \mathbf{a}_i \mathbf{a}_j\, \overset{*}{\Phi}_\alpha \overset{*}{\Phi}_{\beta,j}\right) n_i\, d\Gamma$$

$$\mathbf{H}^n_\alpha = \int_{\Omega}\left(\Delta t \Phi_{\alpha,i}\, \mathbf{F}^n_i - \frac{(\Delta t)^2}{2}\mathbf{a}_i \Phi_{\alpha,i}\frac{\partial \mathbf{F}^n_j}{\partial x_j} - \delta \frac{\partial \Phi_\alpha}{\partial x_i}\frac{\partial \mathbf{U}^n}{\partial x_i}\right) d\Omega$$

$$\mathbf{N}^n_\alpha = \int_{\Gamma}\left(-\Delta t\, \overset{*}{\Phi}_\alpha \overset{*}{\mathbf{F}}{}^n_i + \frac{(\Delta t)^2}{2}\mathbf{a}_i\, \overset{*}{\Phi}_\alpha \frac{\partial}{\partial x_j}\left(\overset{*}{\mathbf{F}}{}^n_j\right)\right) n_i\, d\Gamma$$

(3.54)

As will be demonstrated in 4.1, unlike the SUPG formulation the FDV/MFDV methods "optionally" need about $0.10 \sim 0.20$ of the DCF to give a shot free solution. It is clear from the definition of the DCO that it has virtually no effect on the solution domain where the gradients of the flow properties are smooth or of low gradients. So, only at the elements of high rate of variations the DCO is functioning. The discontinuity capturing factor δ (DCF) will be tuned in section 4.1.

3.5 Boundary Conditions

Back to the mathematical properties of the Euler system of equations, it is well known that an efficient and robust boundary conditions imposition technique is a crucial subject in all CFD problems. In this section the most important boundary conditions are discussed. These boundary conditions are used to simulate many practical flow situations that can be encountered in real flow problems.

Since this thesis deals with the heart of the method and the various aspects in it, no special attention has been paid for implementing specialized problems that requires sophisticated boundary conditions. Great attention has been given to prove the applicability of the FDV/MFDV methods to various flow regimes by solving many numerical test cases that span various regimes and have mixed subsonic/transonic/supersonic flows (as will be demonstrated in Chapter Four), which is the ultimate goal for this thesis.

According to the theory of characteristics, the number of physical and numerical boundary conditions varies for different Mach numbers at both inlet and exit flows. In what follows some of the most common boundaries that can be encountered in many problems are introduced. For further analysis of the boundary conditions the reader is especially referred to [3].

3.5.1 Inviscid Wall Boundary Condition (No Penetration)

There are many ways to implement the wall boundary conditions inside the Euler solver. One of the most successful methods is the method of coordinates rotation given in J. N. Reddy [12] and implemented successfully by many others including F. Moussaoui [57]-[58] , T. E. Tezduyar et al. [58]. In this method the velocity components at the solid wall are transformed to the normal and tangential components, as shown in Figure 3.8.

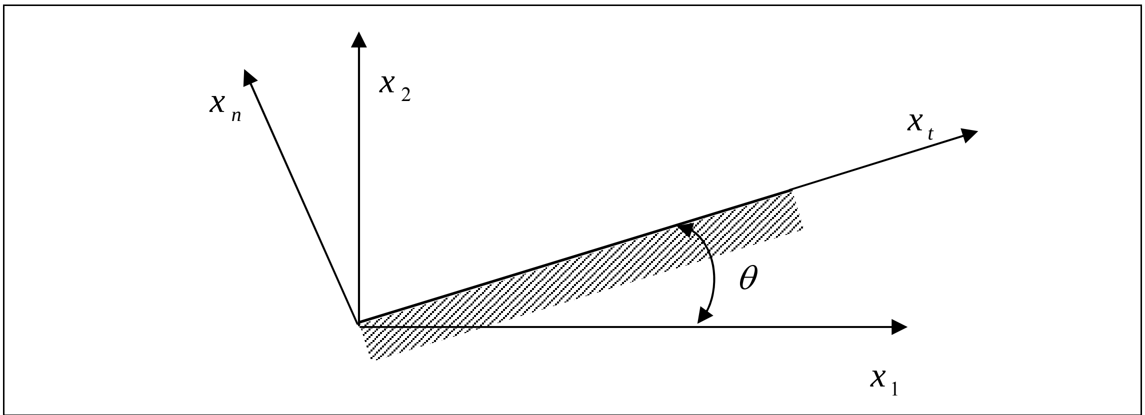

Figure 3.8 Co-ordinates rotation at a solid wall

These axes are used to facilitate zeroing the normal velocity ($u_n = 0.0$) and simulating the inviscid wall condition. Let the wall inclination angle be θ and the normal and tangential components are u_n, u_t, respectively. The coordinate transformation matrix will take the form:

$$\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} u_t \\ u_n \end{pmatrix} \tag{3.55}$$

For the nodes on the solid boundary a new vector of unknowns is to be introduced via the following rotation matrix:

$$\begin{pmatrix} \rho \\ \rho u_t \\ \rho u_n \\ \rho e_t \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho e_t \end{pmatrix} \tag{3.56}$$

Using the nodal transformation matrix given in (3.56) the implementation of the zero normal velocity becomes a straight forward process in the solver.

3.5.2 Supersonic Inlet Boundary Condition

For all the nodes on the supersonic inlet boundary, all the characteristic lines are entering the solution domain, hence the four flow properties ($\rho, \rho u_1, \rho u_2, \rho e_t$) should be specified exactly (physical conditions). The implementation of this boundary condition in the Euler solver is a straight forward step. This can be done by zeroing the change vector $\Delta \mathbf{U}^{n+1}$ at the supersonic inlet boundary nodes.

3.5.3 Supersonic Exit Boundary Condition

For all the nodes on the supersonic exit boundary, all the characteristic lines are leaving the solution domain, hence the four flow properties ($\rho, \rho u_1, \rho u_2, \rho e_t$) are left free. One of the finite element formulation benefits is that, all numerical boundary conditions (free variables) are specified from the shape function that was predefined in the Galerkin formulation. Unlike the FDMs which needs extra extrapolation techniques. So the simplest boundary is the supersonic exit where no changes are imposed on the element equations.

3.5.4 Subsonic Inlet Boundary Condition

In the subsonic inlet boundary condition, only three characteristic lines are entering the solution domain. Hence, three conditions have to be specified (physical conditions) at all nodes on this boundary and the other variable has to be left free (numerical boundary condition).

The most common boundary conditions for the subsonic inlet boundary are the specification of the stagnation pressure (P_0) and stagnation density (ρ_0) along with the flow direction angle. The flow direction angle can be imposed similar to the no penetration condition discussed in 3.5.1. While the stagnation conditions can be implemented in the Euler solver using the relations between the stagnation and static properties. These relations are given by:

$$\frac{P_0}{P} = \left(\frac{\rho_0}{\rho}\right)^{\gamma} \tag{3.57}$$

$$c_p T_0 = c_p T + \frac{(u_i u_i)^{1/2}}{2} \tag{3.58}$$

Rewriting equations (3.57) and (3.58) in terms of the conservation variables and using the Δ terms only the following relations results in the following relations:

$$\frac{\gamma P_0}{(\gamma-1)\rho_0}\left(2U_1 - \frac{\gamma+1}{\rho_0^{\gamma-1}} U_1^{\gamma}\right)\Delta U_1 - U_2 \Delta U_2 - U_3 \Delta U_3 = 0 \tag{3.59}$$

$$\left(\frac{\gamma P_0}{(\gamma-1)\rho_0^{\gamma}} U_1^{\gamma-1} - \frac{U_2^2}{2U_1^2}\right)\Delta U_1 + \frac{U_2}{U_1}\Delta U_2 + \frac{U_3}{U_1}\Delta U_3 - \Delta U_4 = 0 \tag{3.60}$$

Equations (3.59) and (3.60) are ready to be imposed on the subsonic inlet boundary nodes.

3.5.5 Subsonic Exit Boundary Condition

In the subsonic exit boundary condition, only one characteristic line is entering the solution. Hence, one condition (physical condition) has to be specified at all the nodes on this boundary and the other three variables have to be left free (numerical boundary conditions). The most common boundary condition for the subsonic exit boundary is the specification of the back pressure (P_b). From equation (3.9), pressure is given by:

$$P = (\gamma-1)\rho\left(e_t - \frac{u_1^2 + u_2^2}{2}\right) \tag{3.61}$$

or in terms of the conservation variable we have:

$$P = (\gamma-1)\left(U_4 - \frac{U_2^2 + U_3^2}{2U_1}\right) \tag{3.62}$$

using the Δ terms only to write the equation we get:

$$\frac{(U_2^2 + U_3^2)}{2U_1^2}\Delta U_1 - \frac{U_2}{U_1}\Delta U_2 - \frac{U_3}{U_1}\Delta U_3 + \Delta U_4 = 0 \tag{3.63}$$

Chapter Four
Numerical Test Cases and Conclusions

In this chapter, the FDV-FEM compressible Euler flow solver that has been theoretically developed in Chapter Three will be tested to verify the claim that FDV/MFDV formulations are capable of dealing with all flow regimes seamlessly. Great attention has been given in selecting the test cases to span wide ranges of flow regimes that vary between subsonic, transonic, and supersonic. No specific attention has been given to certain type of problems or applications. The MFDV relations given in section 3.3.2 are used in all the following outputs. The CFL number has been fixed to 1.0 in all cases otherwise will be specified.

The aim of the thesis, as mentioned in 1.5, is to develop a time-accurate flow solver for Euler equations and to verify the ability of the FDV formulation to deal with various flow situations. In section 4.1, the DCO that has been introduced in section 3.4.3 will be tuned. Unlike the SUPG formulation, FDV method only needs a fraction from the proposed DCF. This will be demonstrated by solving the famous shock reflection test case in section 4.1.1. Parametric analysis will be used to some extent to select the fraction of the DCF that virtually has no effect on the shock width and ensures ripple-free solution. To verify that the selection of the DCF fraction is not arbitrary, another problem which also has a known analytical solution (compression corner) will be solved in 4.1.2.

Section 4.2 verifies the time accuracy of the FDV formulation via solving the shock tube problem. Section 4.3 verifies the ability of the solver to handle supersonic flows. This will be done by solving three complex internal flow problems in 4.3.1, 4.3.2, and 4.3.3. Section 4.4 deals with both the completely subsonic and subsonic/transonic flows. Both of sections 4.4.1 and

4.4.2 are concerned with the solution of subsonic/transonic internal flow problems. Finally, section 4.5 introduces the conclusions that have been drawn from the thesis and the recommendations for future research to evolve the developed Euler equations solver to be a state of art.

4.1 Discontinuity Capturing Factor Tuning

As will be demonstrated from the results below, unlike the SUPG not all the DCF is needed in the FDV/MFDV formulations. A fraction of this factor is sufficient to guarantee ripple-free solution. This is due to the fact that FDV formulation already has a built in artificial viscosity term. In 4.1.1 some numerical experimentation are carried on the shock refection problem to select the fraction of the DCF that does not affect the shock thickness and ensures ripple-free solution. After that in 4.1.2, the supersonic compression corner problem is solved using the selected DCF fraction to ensure that this value is not arbitrary.

4.1.1 Shock Reflection

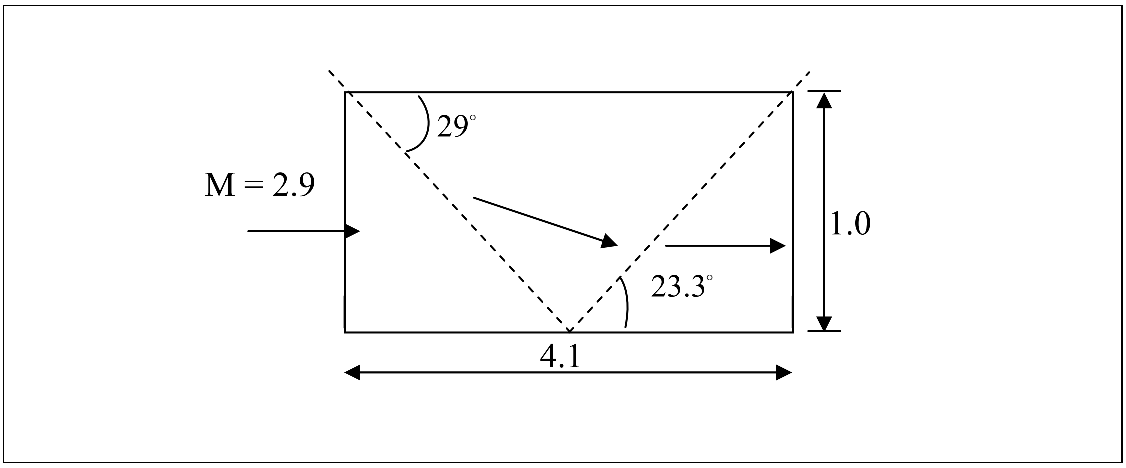

Figure 4.1 Shock reflection domain

The shock reflection problem has become an interesting test case for the validation and testing of any Euler solver in its preliminary stages. This test case with its known steady state analytical solution and relatively complex shock structure offers a very good opportunity to tune in any unsettled parameters in the solver. The solution domain of shock reflection from inviscid wall, as shown in Figure 4.1, is rectangular. At both inlet and upper boundaries, the supersonic inlet boundary condition is used with the following values:

Inlet:

$$\rho = 1.0, \qquad \rho u_1 = 2.9$$
$$\rho u_2 = 0.0, \qquad \rho e_t = 5.99075$$

Upper:

$$\rho = 1.7, \qquad \rho u_1 = 4.453$$
$$\rho u_2 = \text{-}0.86, \qquad \rho e_t = 9.87$$

On lower boundary, the inviscid wall (no-penetration) boundary condition is used and the exit boundary is left free being a supersonic exit. A uniform 120x60 grid is used to solve this test case. The inlet properties are used for the domain initial conditions. The discontinuity capturing factor is allowed to change in order to see the effect of its change on the shockwave structure.

4.1.1.1 DCF = 0.0

Figure 4.2 to Figure 4.9 plot the transient Mach number contours at various time steps. The solution reaches its steady state at t = 3.300 sec. The contours of the conservative variables, $\rho, \rho u_1, \rho u_2, \rho e_t$, at steady state are shown in Figure 4.10, Figure 4.11, Figure 4.12, and Figure 4.13, respectively. The Mach number steady state contours are shown in Figure 4.14, while s_1, s_2 steady state contours are shown in Figure 4.15 and Figure 4.16, respectively. As stated earlier, the solution is satisfactory for the supersonic problems without the discontinuity operator and these results match very well the results obtained in [59] and the analytical solution.

It is clear that s_1 parameter resolves the solution itself and this interesting property for s_1 may be utilized if an adaptation technique is to be used. Figure 4.17 shows a comparison between the steady state Mach number at $x_2 = 0.5$ for this gird and a courser one (60x30) with the exact analytical solution. Calculating both the pointwise and spacewise norm errors from equations (C.2) and (C.4) for the steady state solution for the two grid sets will give the results in Table 3.

Table 3 Space and point wise L_2 norm errors, shock reflection, DCF = 0.00

	60x30	120x60
$\lVert\varepsilon\rVert_{\text{Space}\,L_2}$	0.06169	0.04501
$\lVert\varepsilon\rVert_{\text{Point}\,L_2\,\%}$	0.01313	0.00837

It is clear from both Figure 4.17 and Table 3 that as the grid gets finer the exact solution is approached. Also a higher over/under-shots can be observed for finer grids. This is due to the lost solution modes resulting from the grid selectivity. It is expected that when the DCF increases these under/over shots will start to disappear.

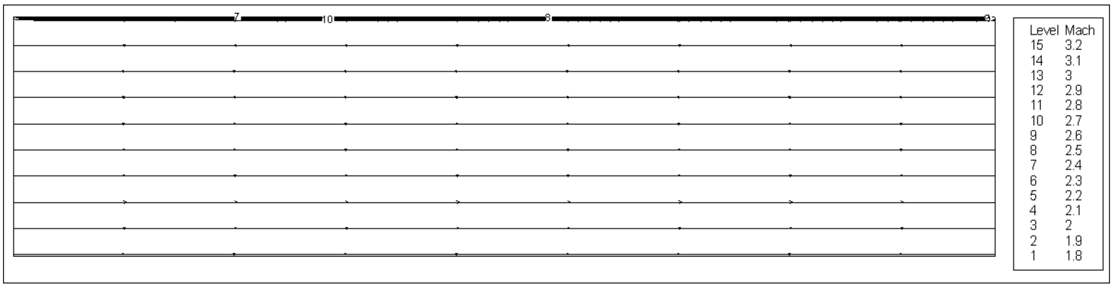

Figure 4.2 Mach No. contours at t = 0.000, DCF = 0.00, shock reflection

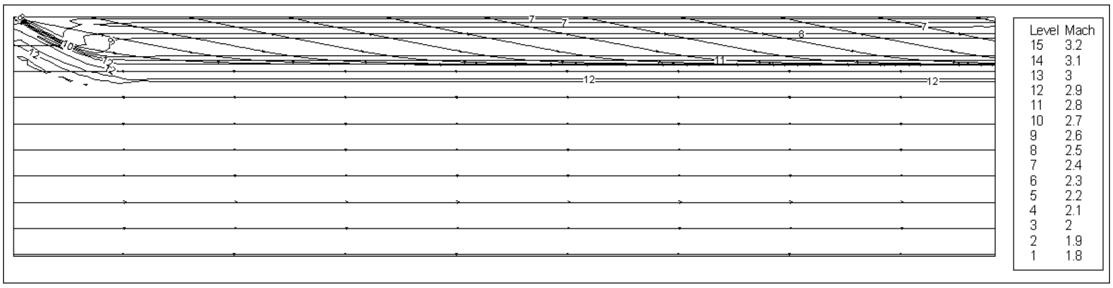

Figure 4.3 Mach No. contours at t = 0.118, DCF = 0.00, shock reflection

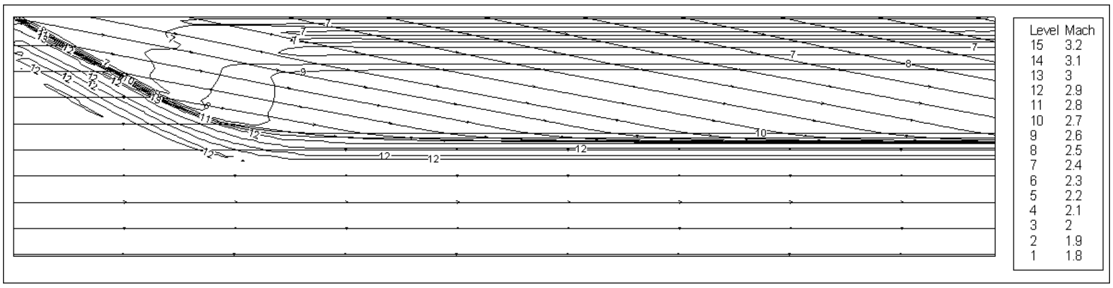

Figure 4.4 Mach No. contours at t = 0.352, DCF = 0.00, shock reflection

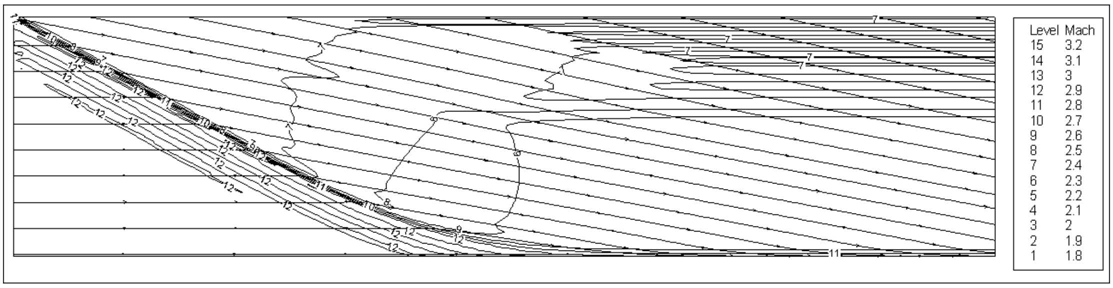

Figure 4.5 Mach No. contours at t = 0.704, DCF = 0.00, shock reflection

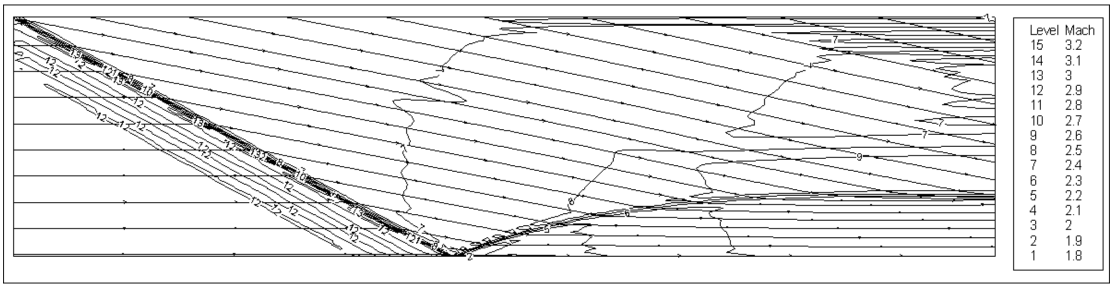

Figure 4.6 Mach No. contours at t = 1.000, DCF = 0.00, shock reflection

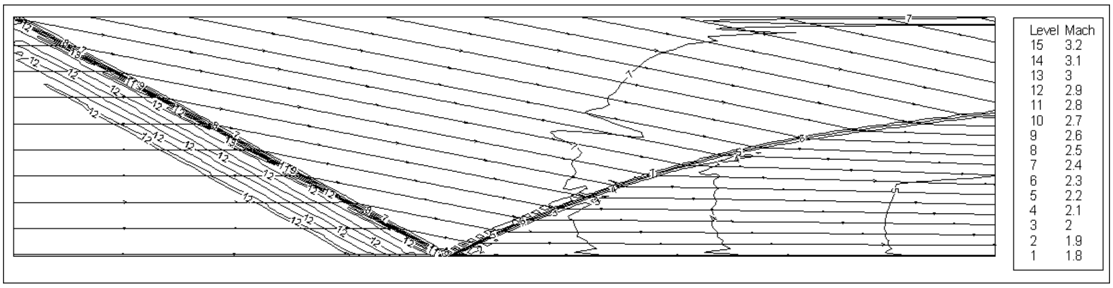

Figure 4.7 Mach No. contours at t = 1.404, DCF = 0.00, shock reflection

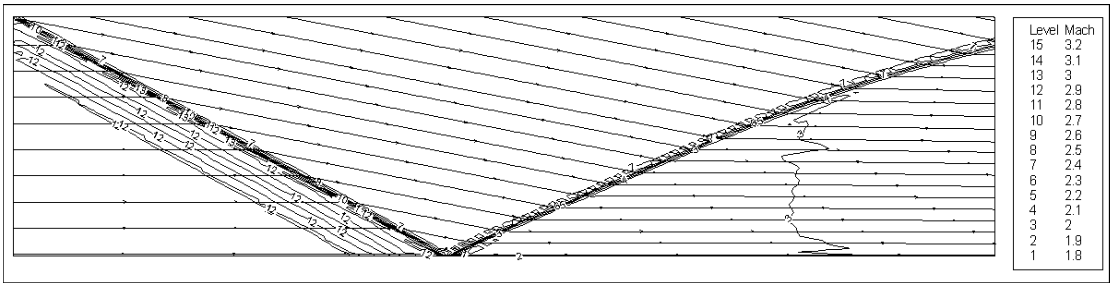

Figure 4.8 Mach No. contours at t = 2.000, DCF = 0.00, shock reflection

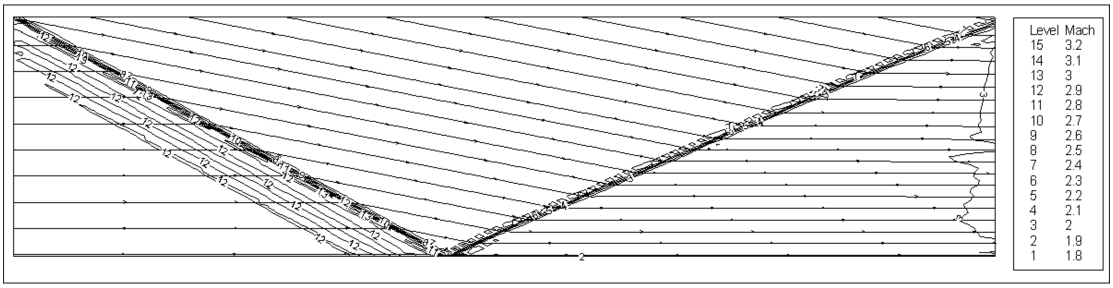

Figure 4.9 Mach No. contours at t = 2.500, DCF = 0.00, shock reflection

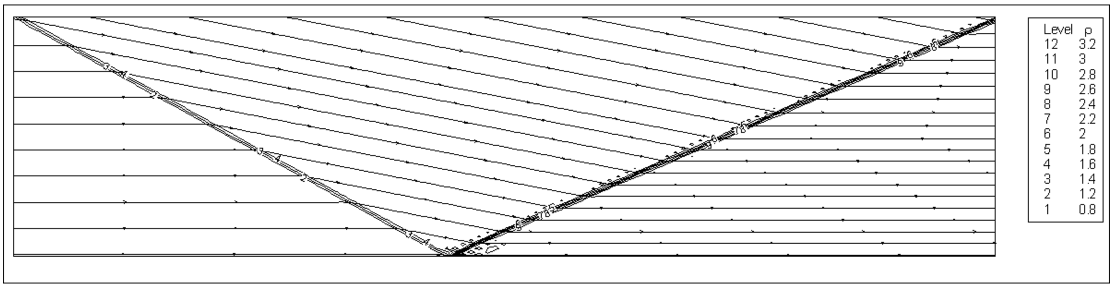

Figure 4.10 Steady state ρ contours, DCF = 0.00, shock reflection

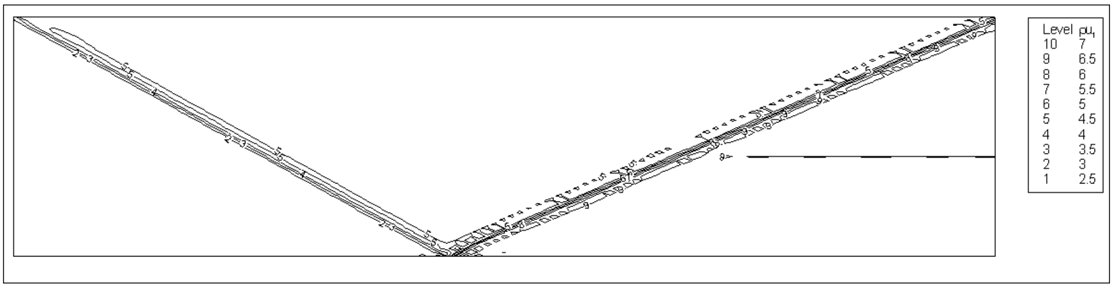

Figure 4.11 Steady state ρu_1 contours, DCF = 0.00, shock reflection

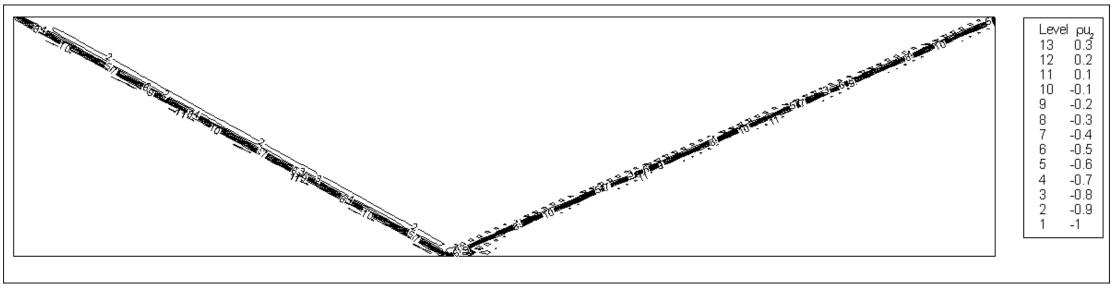

Figure 4.12 Steady state ρu_2 contours, DCF = 0.00, shock reflection

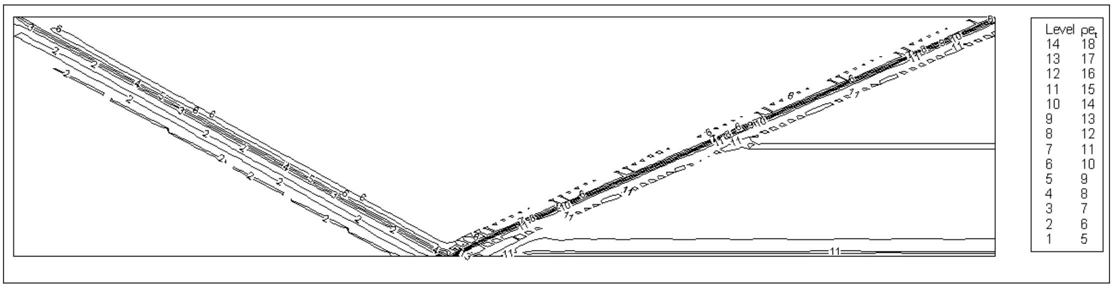

Figure 4.13 Steady state ρe_t contours, DCF = 0.00, shock reflection

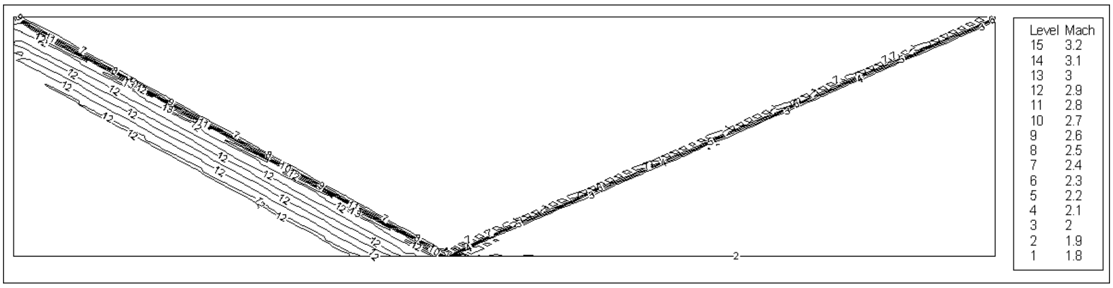

Figure 4.14 Steady state Mach No. contours, DCF = 0.0, shock reflection

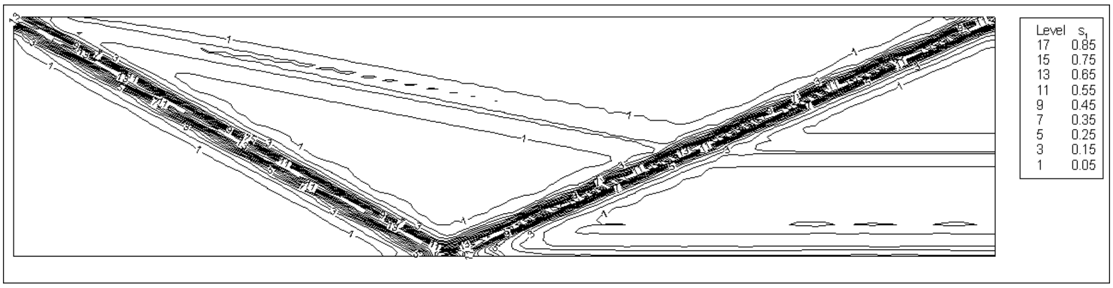

Figure 4.15 Steady state s_1 contours, DCF = 0.00, shock reflection

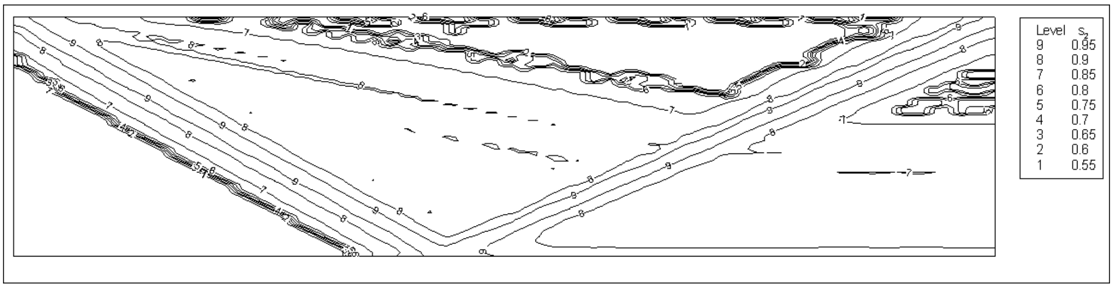

Figure 4.16 Steady state s_2 contours, DCF = 0.00, shock reflection

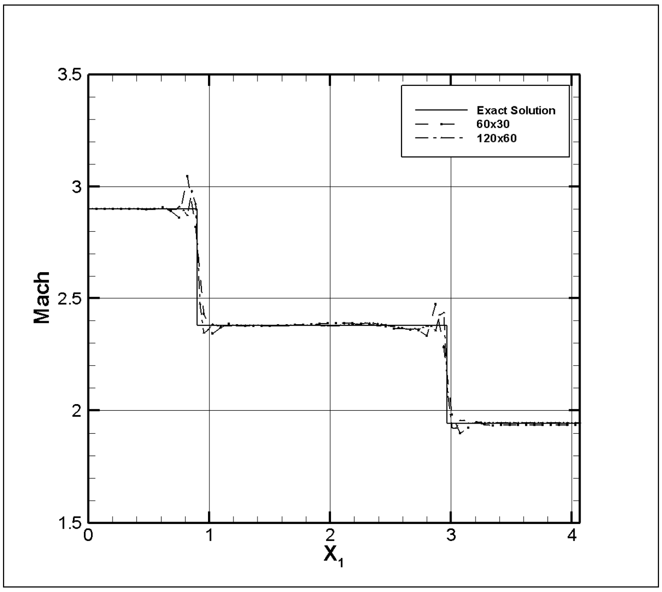

Figure 4.17 Analytical Vs numerical solution

at $x_2 = 0.5$, DCF = 0.00, shock reflection

4.1.1.2 Comparison between Various DCFs

In order to demonstrate the effect of the discontinuity capturing factor changing, a parametric study has been investigated to the same shock reflection problem stated earlier for the two grid sets.

The first grid is 60x30 and the other one is 120x60. Figure 4.18 shows the whole span steady state Mach number for the various values of the DCF at $x_2 = 0.5$ for the first grid set, while Figure 4.19 and Figure 4.20 show zoom in for the first and second shocks, respectively. It is evident that as the DCF increases the under/over shots system disappear and also the shockwave become smeared. A more precise look will reveal the fact that the values between 0.15 and 0.25 give a clear pattern without effective smearing in the shock structure.

The same data has been plotted for the second grid (120x60) in Figure 4.21, Figure 4.22, and Figure 4.23. The same trend that has been observed in the first set has been obtained here also. But the most interesting feature that can be seen in this data set is that; the smearing out doesn't depend on the spatial distribution of the grid nodes, it is dependent on the number of grid nodes. This note is vital if the grid adaptation is considered because as the grid gets adapted the smearing out will not be a problem.

By this parametric study, the DCF has settled itself to the values between 0.15 and 0.25. Higher values can be used as the grid gets finer. But for moderate grids these values are of good effect on the under/over shots with virtually no effect on the shockwave thickness. The next subsection contains the data for DCF = 0.20.

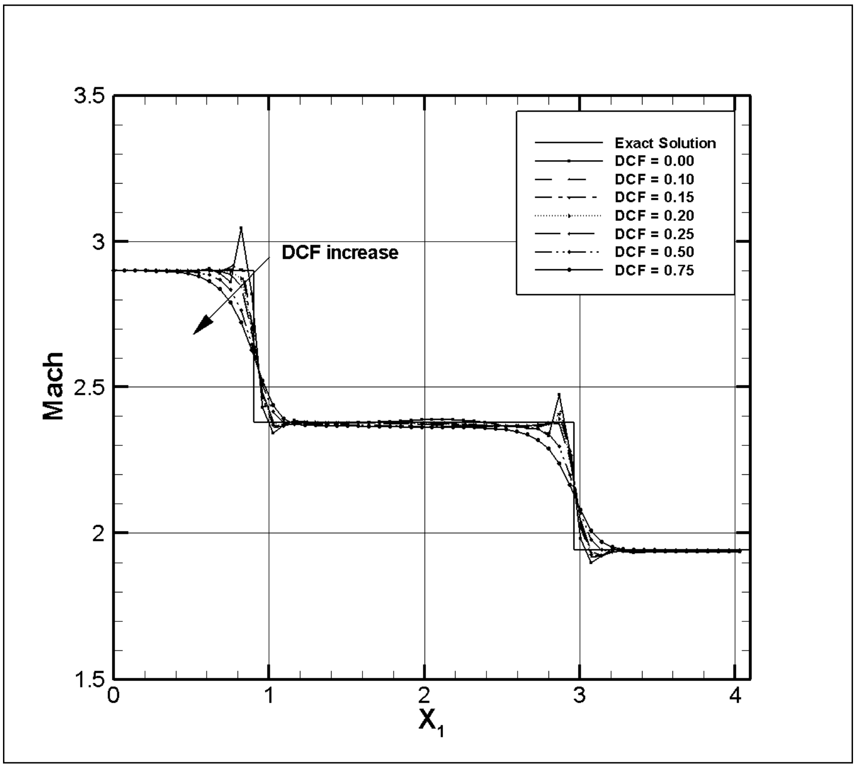

Figure 4.18 DCF effect at $x_2 = 0.5$,

shock reflection, 60x30

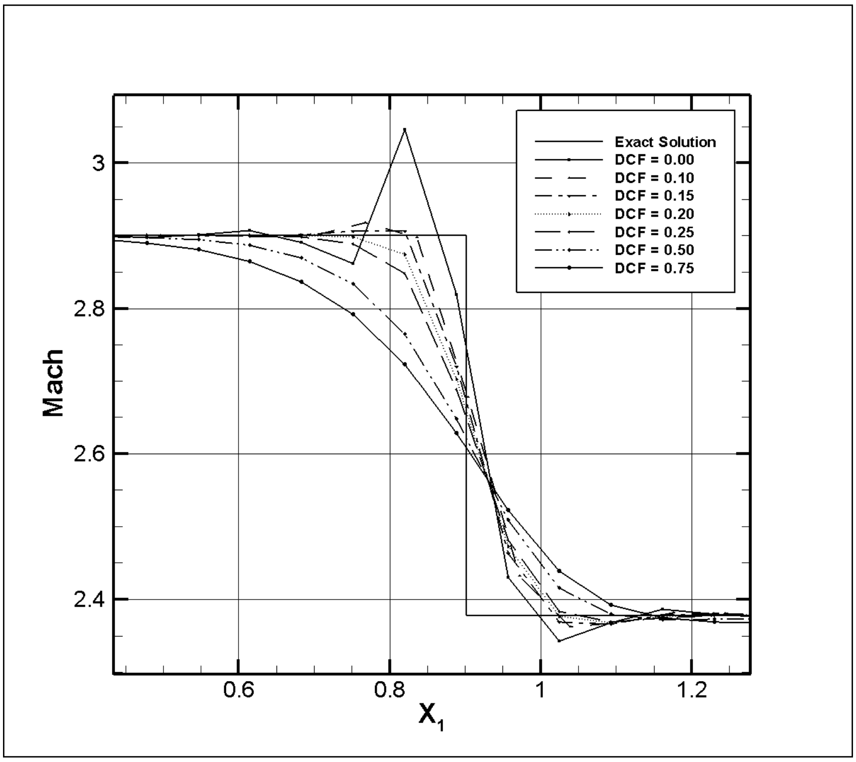

Figure 4.19 Zoom on the first shock at $x_2 = 0.5$,

shock reflection, 60x30

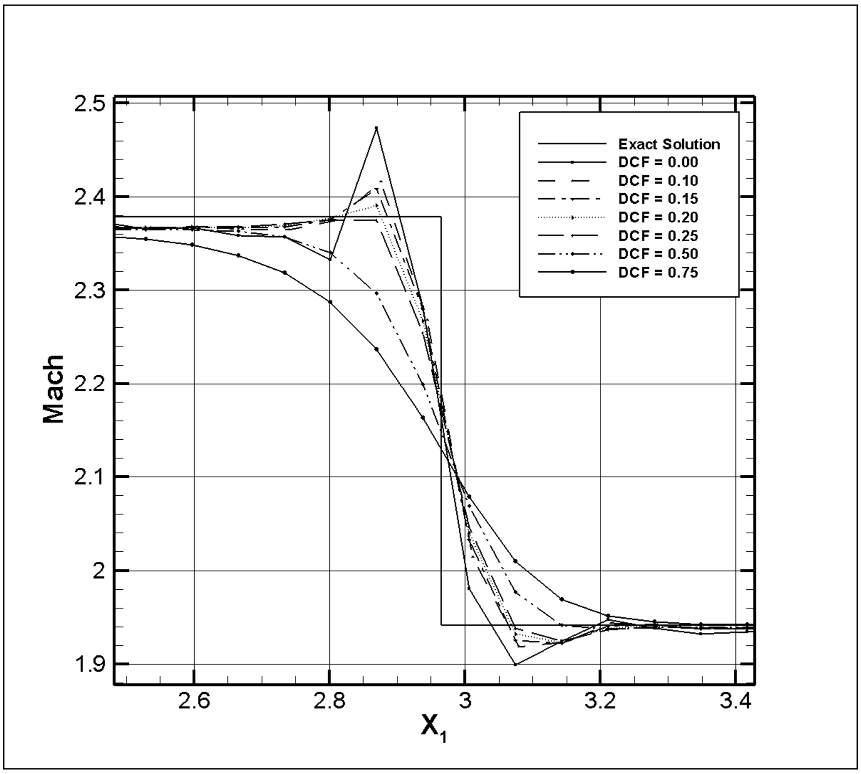

Figure 4.20 Zoom on the second shock at $x_2 = 0.5$,

shock reflection, 60x30

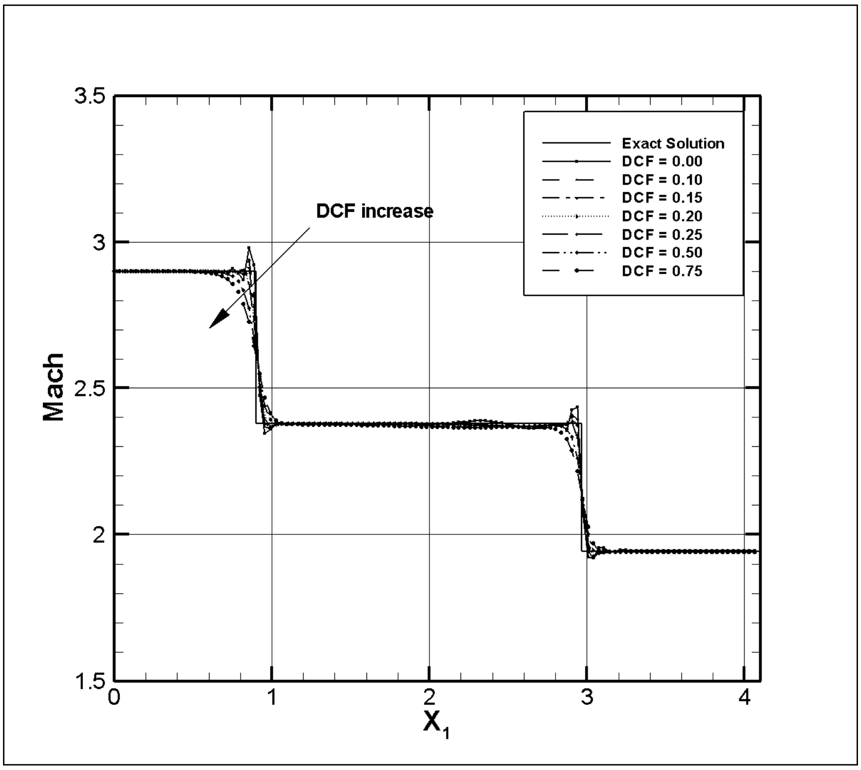

Figure 4.21 DCF effect at $x_2 = 0.5$,

shock reflection, 120x60

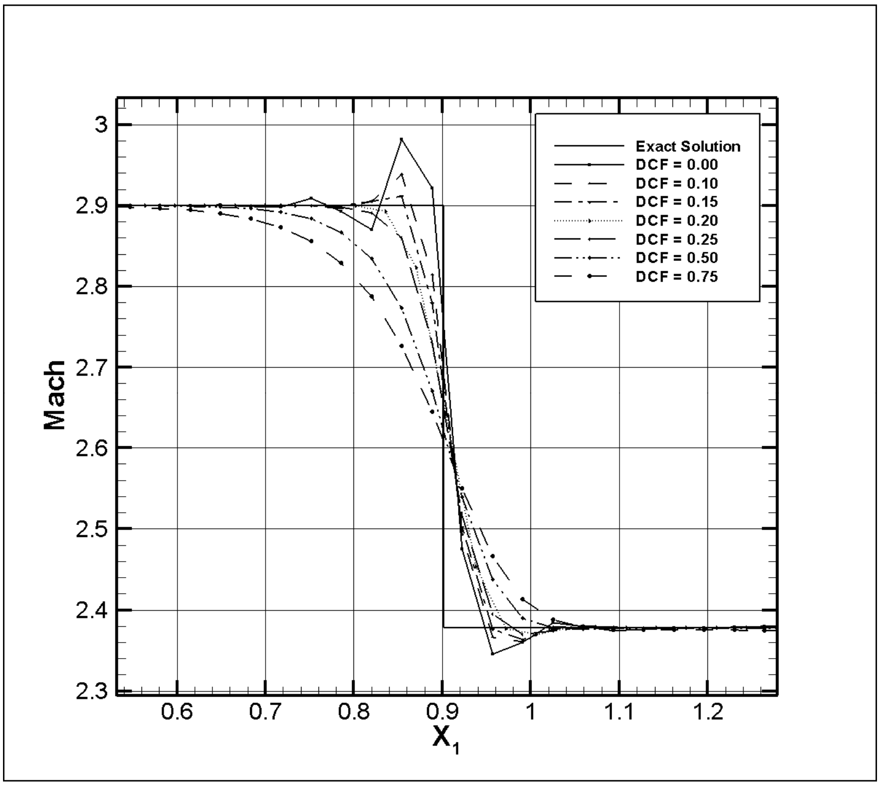

Figure 4.22 Zoom on the first shock at $x_2 = 0.5$,

shock reflection, 120x60

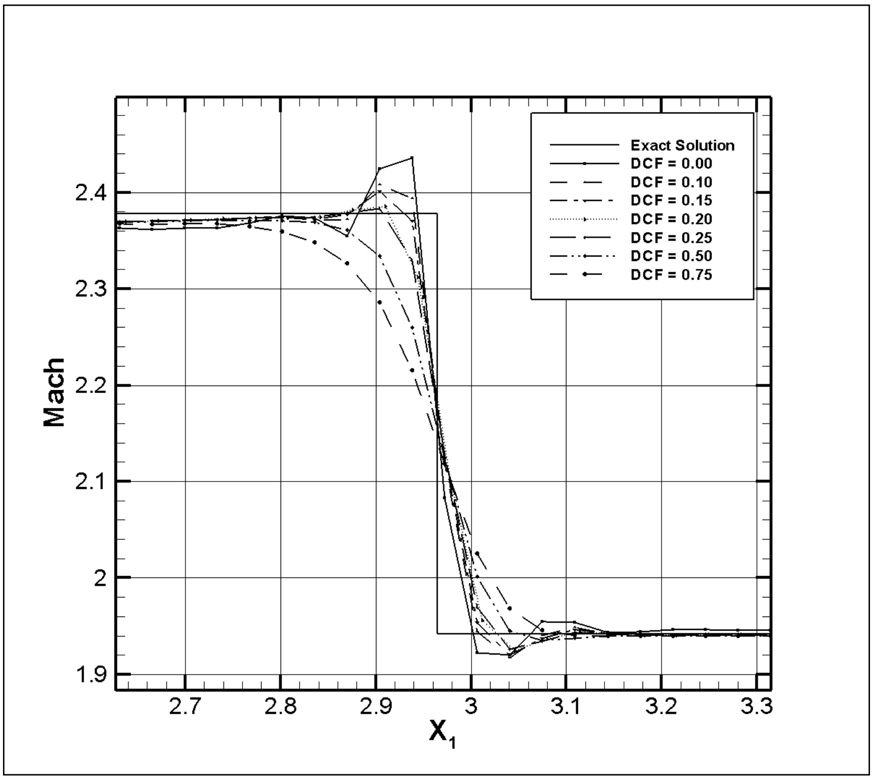

Figure 4.23 Zoom on the second shock at $x_2 = 0.5$,

shock reflection, 120x60

4.1.1.3 DCF = 0.20

Figure 4.24 to Figure 4.31 plot the transient Mach number contours at various time steps. Figure 4.32, Figure 4.33, Figure 4.34, and Figure 4.35 show the contour plot for the conservation variables $\rho, \rho u_1, \rho u_2, \rho e_t$, respectively at steady state.

The smoothing effect of the shock capturing operator can be clearly noticed. Figure 4.36, Figure 4.37, and Figure 4.38 show the steady state Mach number, s_1, and s_2 contours, respectively. Figure 4.39 plots the comparison between the two gird sets at the settled DCF for $x_2 = 0.5$.

Calculating both the pointwise and spacewise norm errors from equations (C.2) and (C.4) for the steady state solution for the two grid sets will give the results in Table 4. It is clear from both Figure 4.39 and Table 4 that as the grid gets finer the exact solution is approached. Also the smoothing effect of the DCF is evident from the peakless solution.

Table 4 Space and point wise L_2 norm errors, shock reflection, DCF = 0.20

	60x30	120x60
$\|\varepsilon\|_{\text{Space}\,L_2}$	0.07529	0.05833
$\|\varepsilon\|_{\text{Point}\,L_2\%}$	0.01447	0.01276

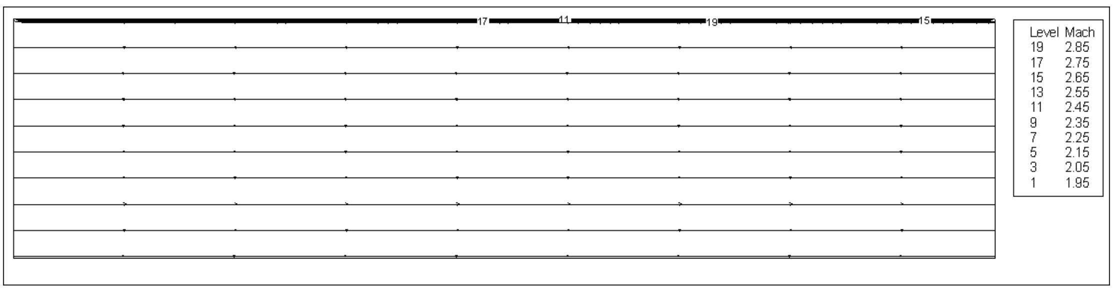

Figure 4.24 Mach No. contours at t = 0.000, DCF = 0.20, shock reflection

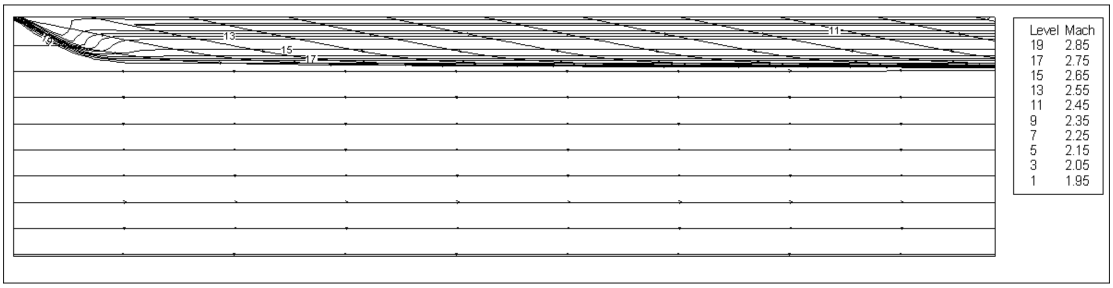

Figure 4.25 Mach No. contours at t = 0.118, DCF = 0.20, shock reflection

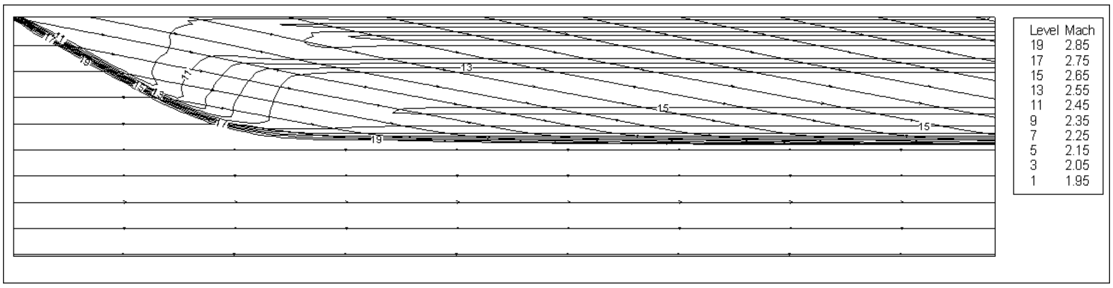

Figure 4.26 Mach No. contours at t = 0.354, DCF = 0.20, shock reflection

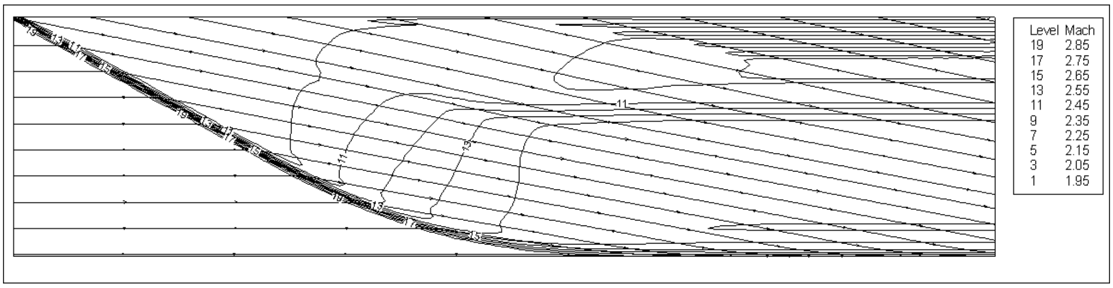

Figure 4.27 Mach No. contours at t = 0.705, DCF = 0.20, shock reflection

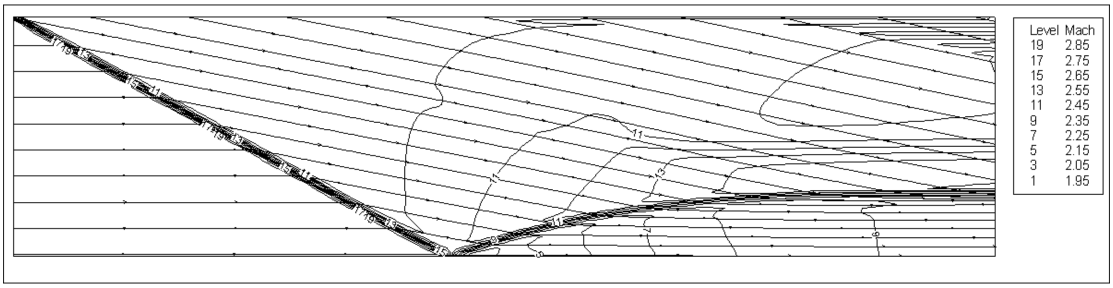

Figure 4.28 Mach No. contours at t = 1.000, DCF = 0.20, shock reflection

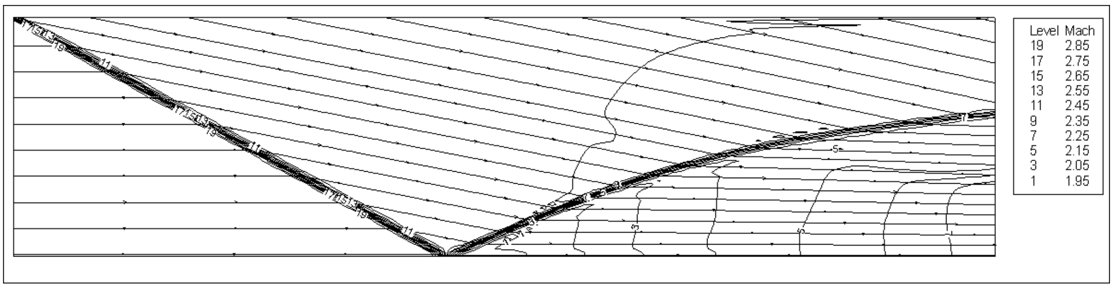

Figure 4.29 Mach No. contours at t = 1.405, DCF = 0.20, shock reflection

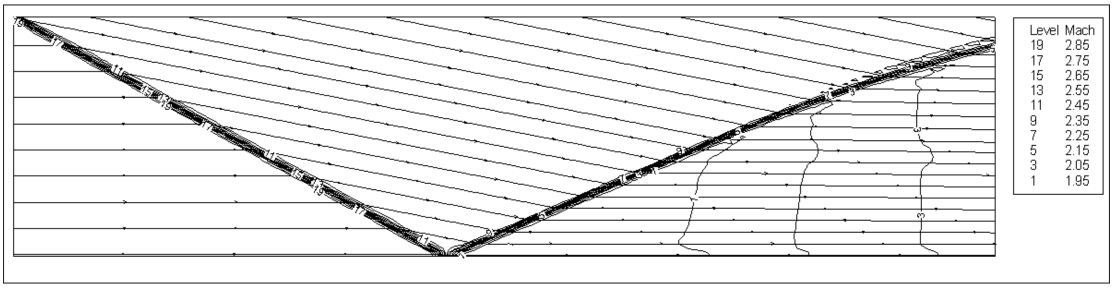

Figure 4.30 Mach No. contours at t = 2.015, DCF = 0.20, shock reflection

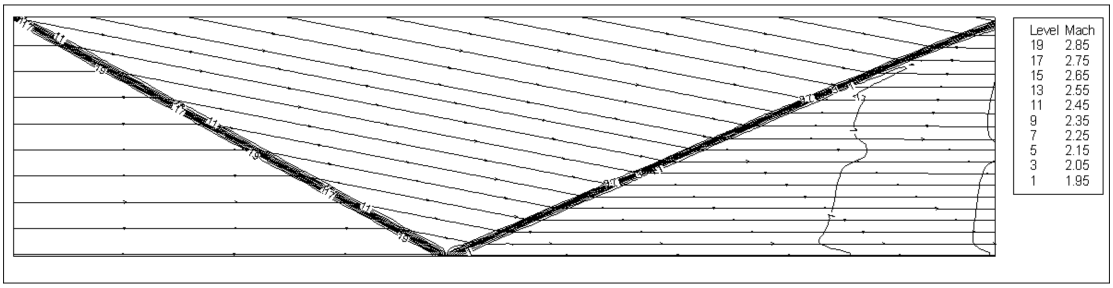

Figure 4.31 Mach No. contours at t = 2.500, DCF = 0.20, shock reflection

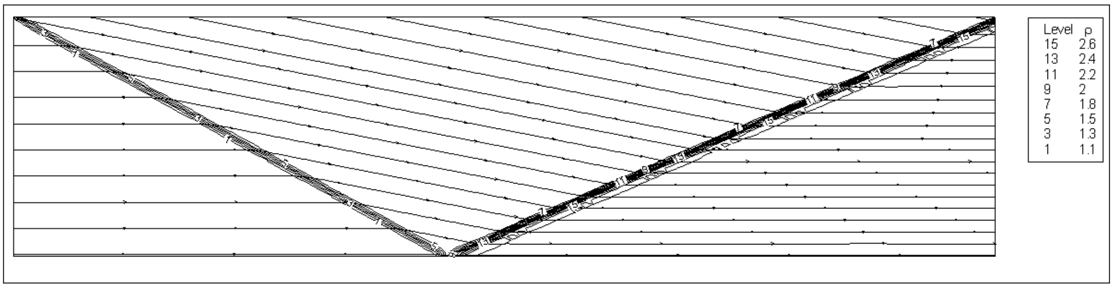

Figure 4.32 Steady state ρ contours, DCF = 0.20, shock reflection

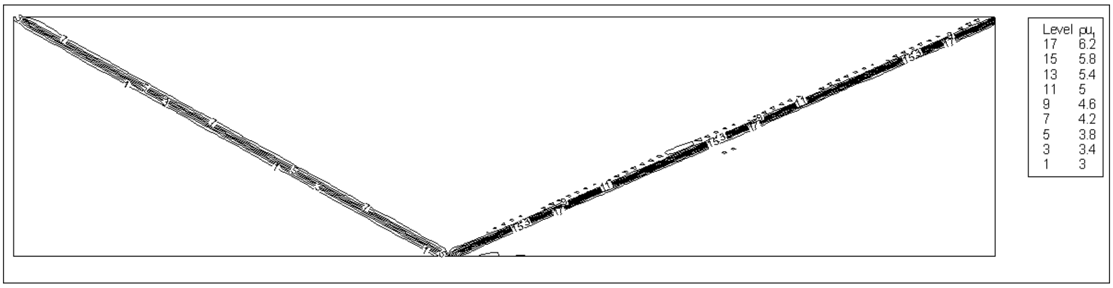

Figure 4.33 Steady state ρu_1 contours, DCF = 0.20, shock reflection

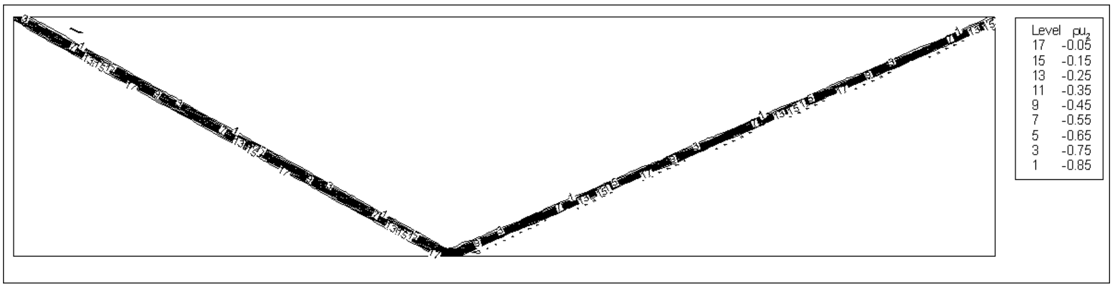

Figure 4.34 Steady state ρu_2 contours, DCF = 0.20, shock reflection

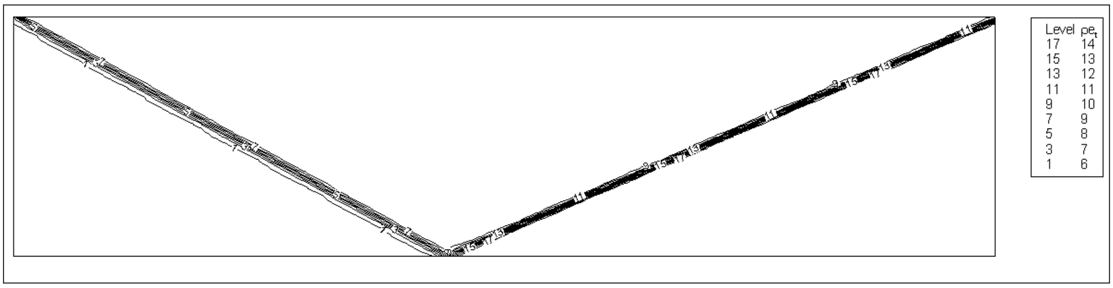

Figure 4.35 Steady state ρe_t contours, DCF = 0.20, shock reflection

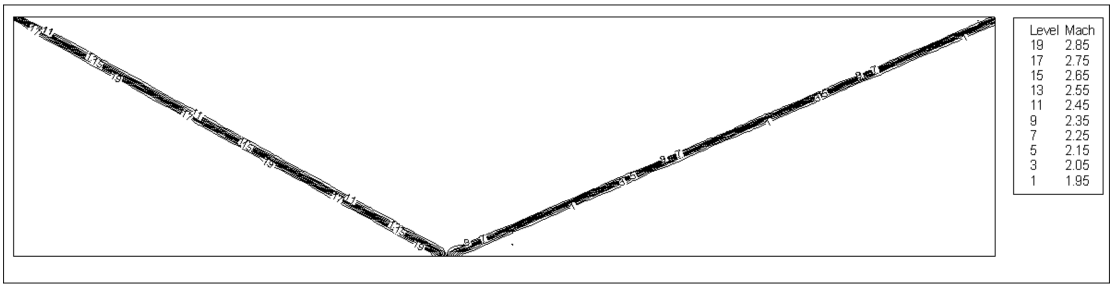

Figure 4.36 Steady state Mach No. contours, DCF = 0.20, shock reflection

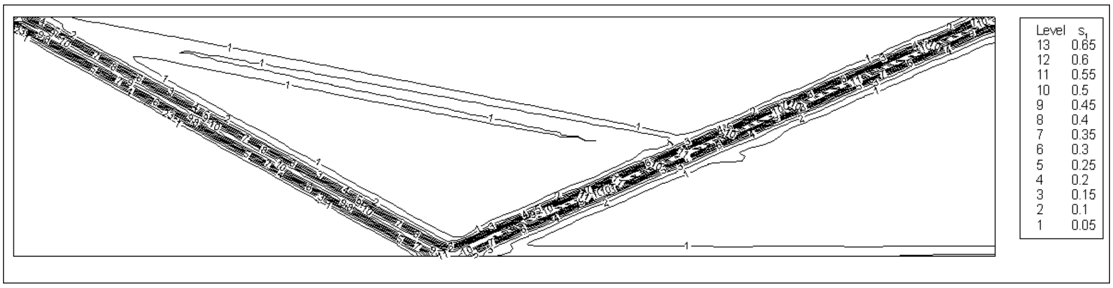

Figure 4.37 Steady state s_1 contours, DCF = 0.20, shock reflection

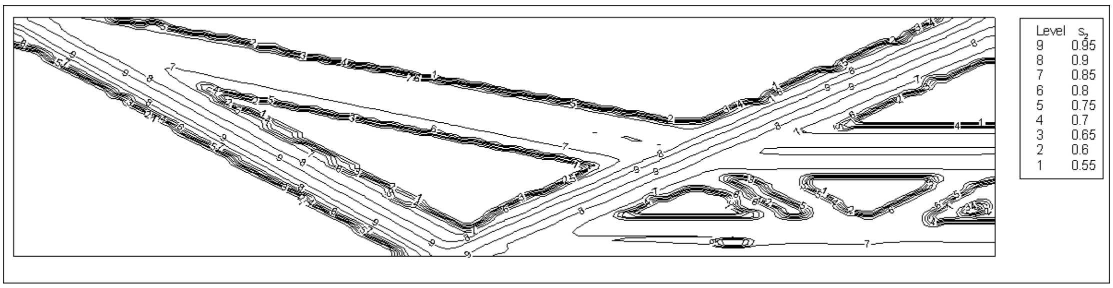

Figure 4.38 Steady state s_2 contours, DCF = 0.20, shock reflection

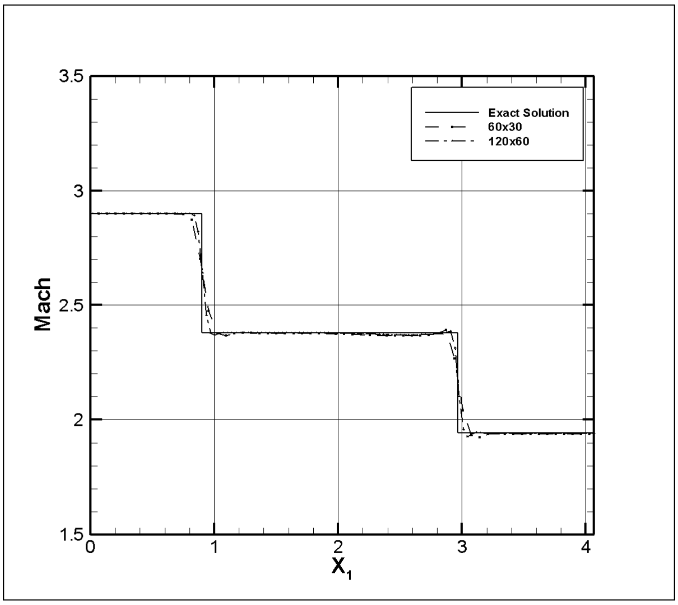

Figure 4.39 Analytical Vs numerical solution at $x_2 = 0.5$, DCF = 0.20, shock reflection

4.1.2 Compression Corner

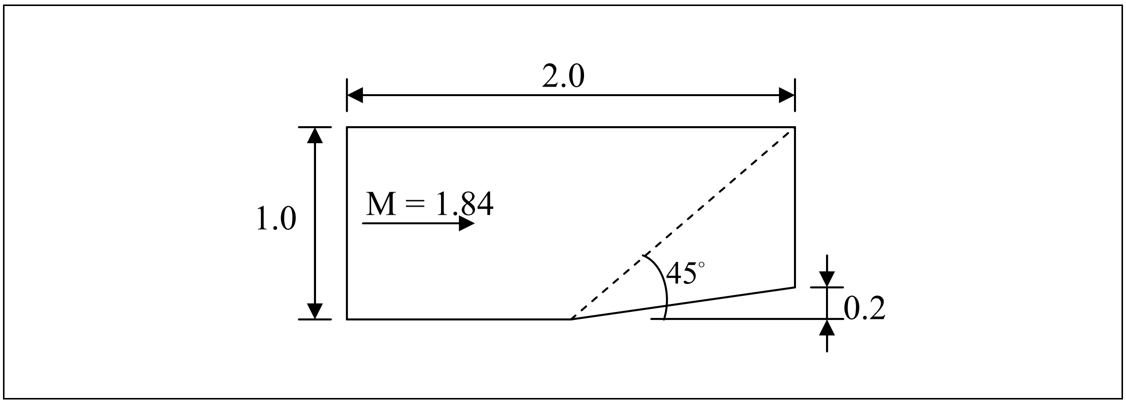

Figure 4.40 Compression corner domain

To make sure that the selection of DCF = 0.20 is not arbitrary for satisfactory performance of the DCO; another test case with known analytical steady state solution has been solved. The compression corner problem is shown in Figure 4.40. This test case constitutes to a typical oblique shockwave problem.

The solution domain is discretized using 120x80 bilinear quadrilateral elements. The no-penetration boundary condition is applied on both the upper and lower boundaries and the exit boundary is left free being supersonic. The inlet boundary is supersonic and the following values have been used:

$$\rho = 1.0, \qquad \rho u_1 = 1.84$$
$$\rho u_2 = 0.0, \qquad \rho e_t = 3.47855$$

Figure 4.41 to Figure 4.45 show the transient Mach number contours at various time steps. The steady state solution is obtained at t = 2.514 sec. Figure 4.46, Figure 4.47, Figure 4.48, and Figure 4.49 show the steady state contour plot for the conservation variables $\rho, \rho u_1, \rho u_2, \rho e_t$, respectively.

Again the shock capturing operator effect is evident in the outputs. Figure 4.50, Figure 4.51, and Figure 4.52 show the steady state Mach number, s_1, and s_2 contours, respectively. Figure 4.53 and Figure 4.54 show the effect of grid refinement on the shockwave at $x_2 = 0.5$.

Calculating both the pointwise and spacewise norm errors from equations (C.2) and (C.4) for the steady state solution for the two grid sets will give the results in Table 5. It is clear from Figure 4.54 and Table 5 that as the grid gets finer near the shock, the exact solution is approached without under/over shots.

Table 5 Space and point wise L_2 norm errors, compression corner

	60x40	120x80
$\lVert\varepsilon\rVert_{\text{Space}\,L_2}$	0.041646	0.019929
$\lVert\varepsilon\rVert_{\text{Point}\,L_2\%}$	0.01477	0.009190

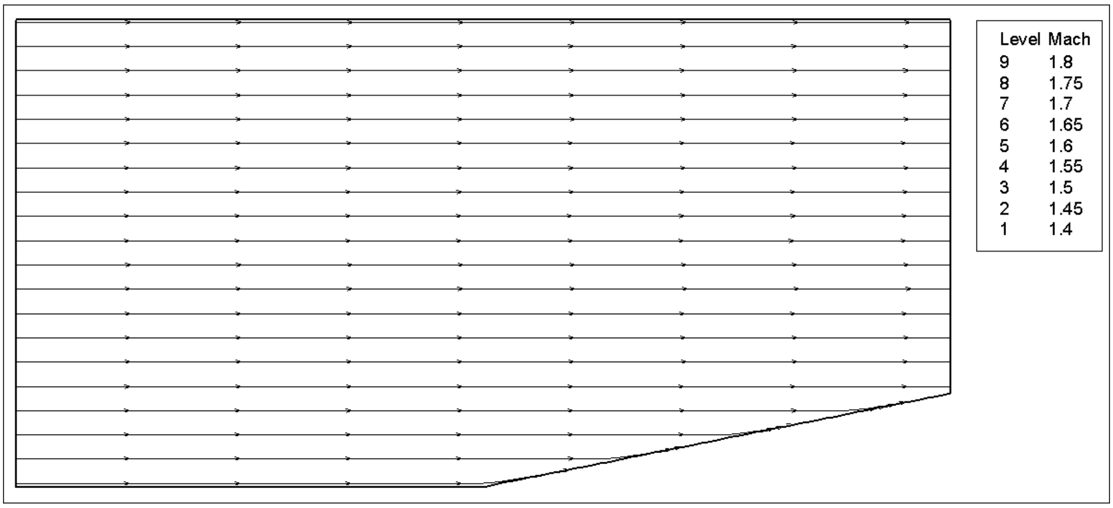

Figure 4.41 Mach No. contours at t = 0.000, DCF = 0.20, compression corner

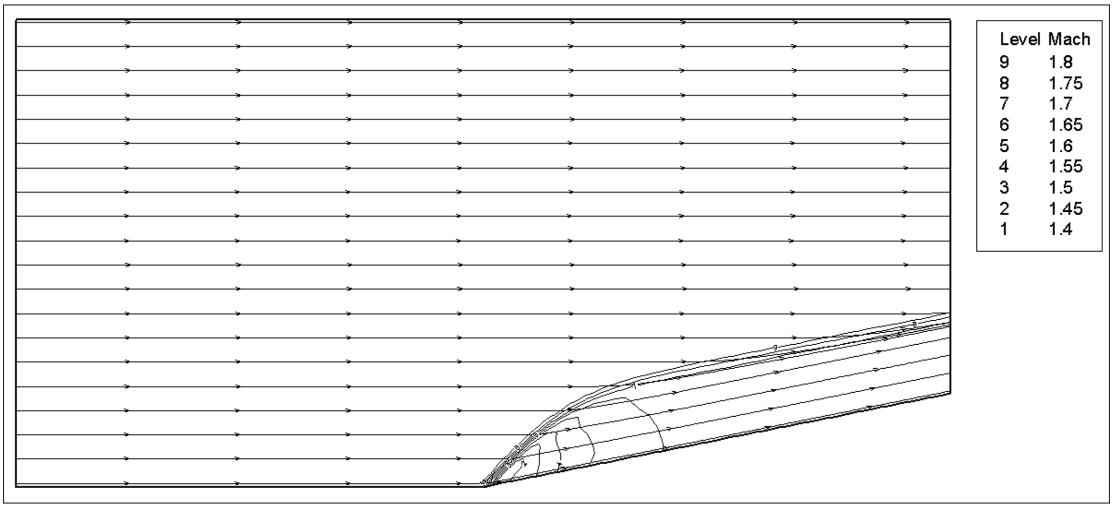

Figure 4.42 Mach No. contours at t = 0.179, DCF = 0.20, compression corner

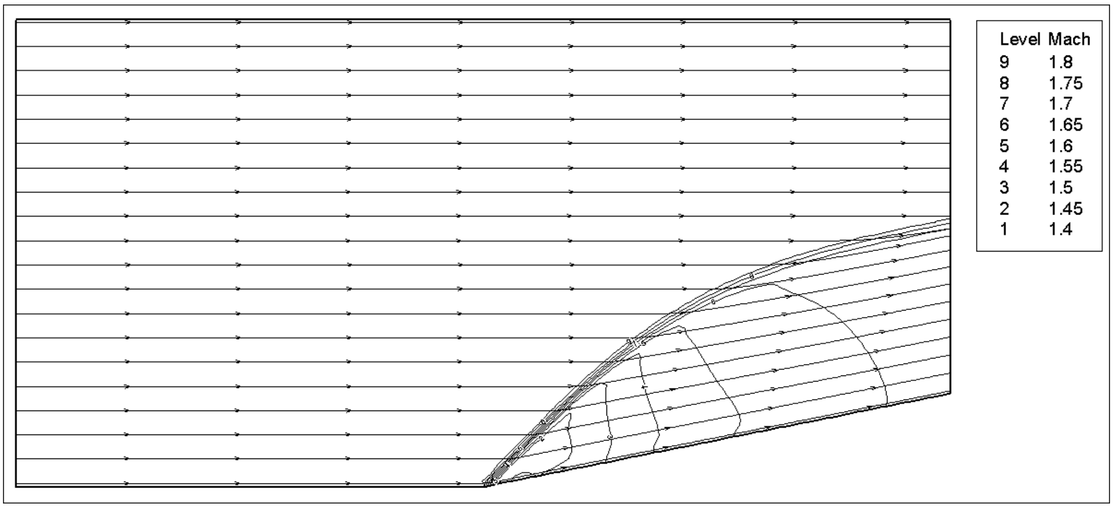

Figure 4.43 Mach No. contours at t = 0.403, DCF = 0.20, compression corner

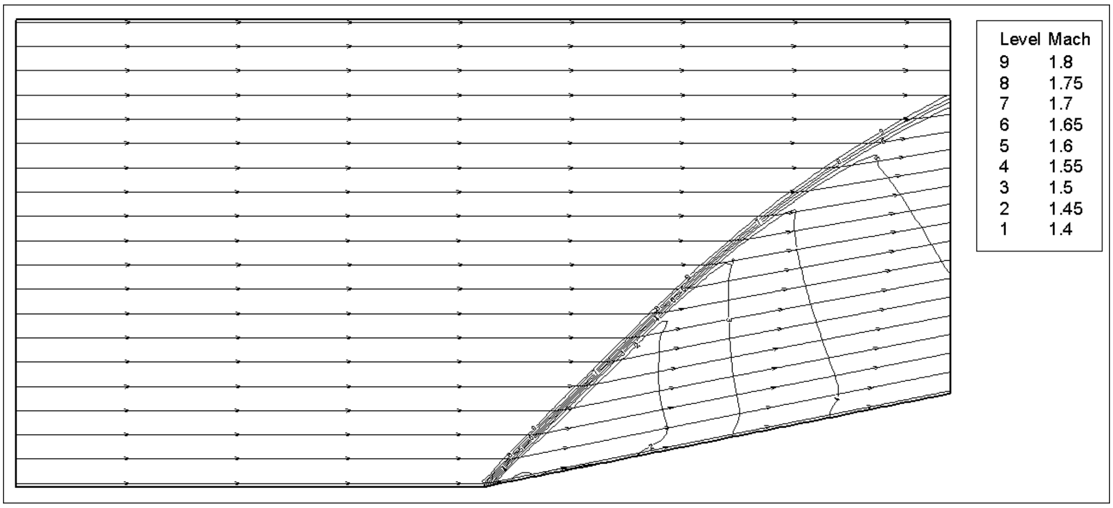

Figure 4.44 Mach No. contours at t = 0.812, DCF = 0.20, compression corner

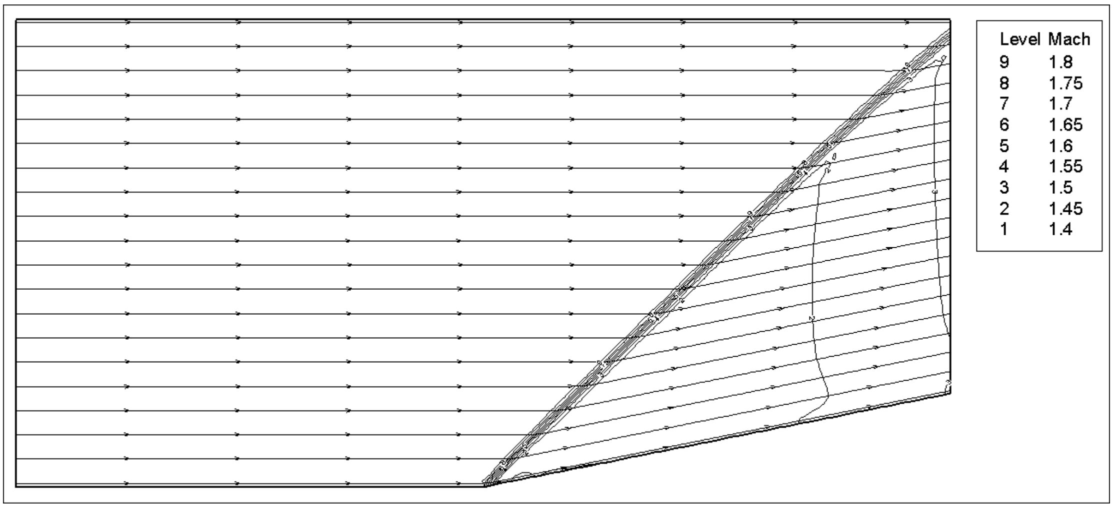

Figure 4.45 Mach No. contours at t = 1.494, DCF = 0.20, compression corner

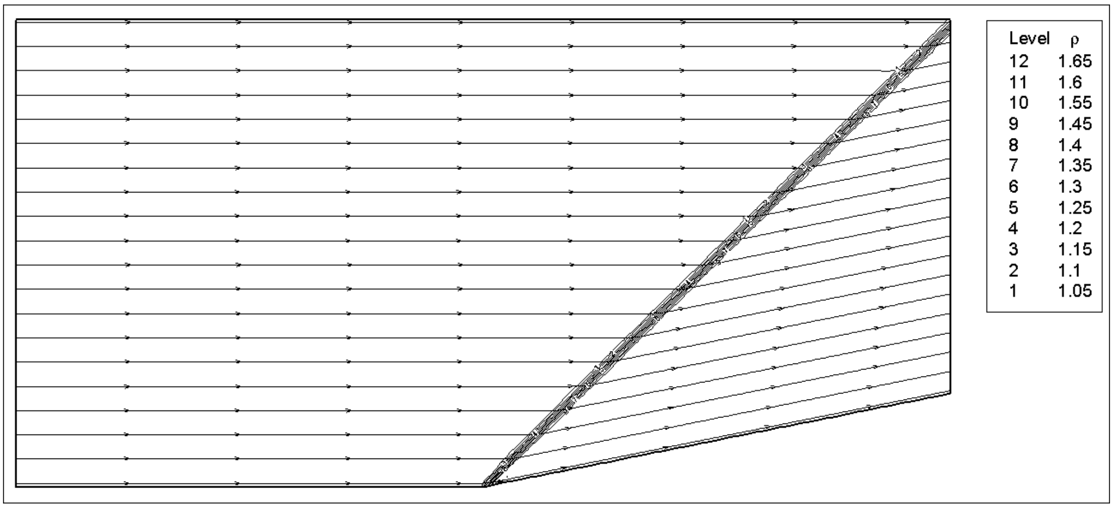

Figure 4.46 Steady state ρ contours, DCF = 0.20, compression corner

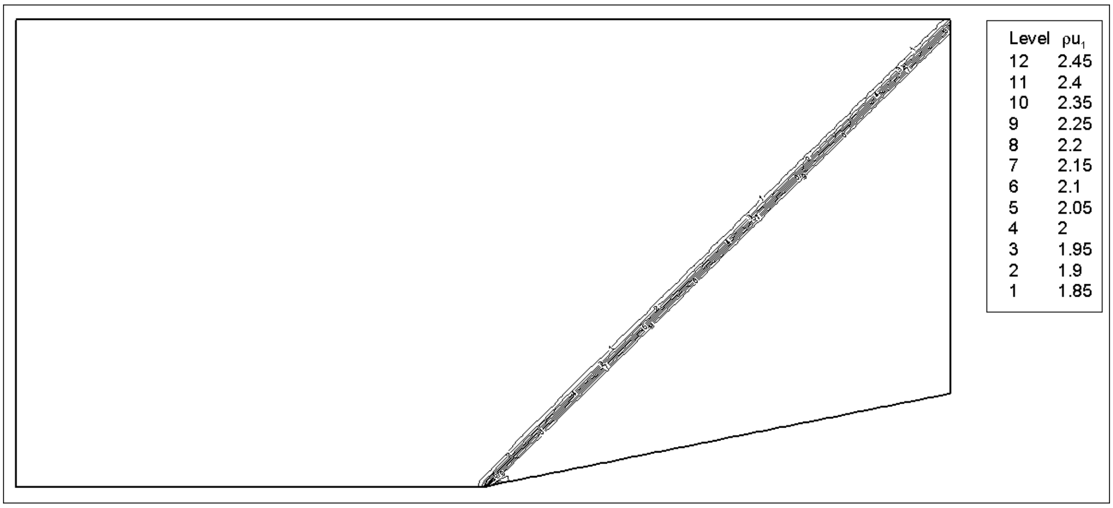

Figure 4.47 Steady state ρu_1 contours, DCF = 0.20, compression corner

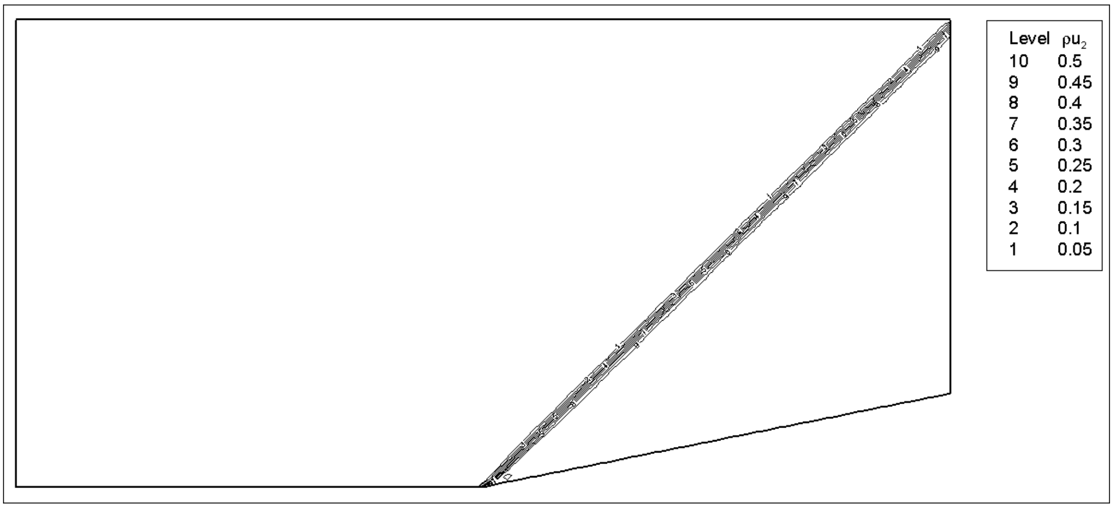

Figure 4.48 Steady state ρu_2 contours, DCF = 0.20, compression corner

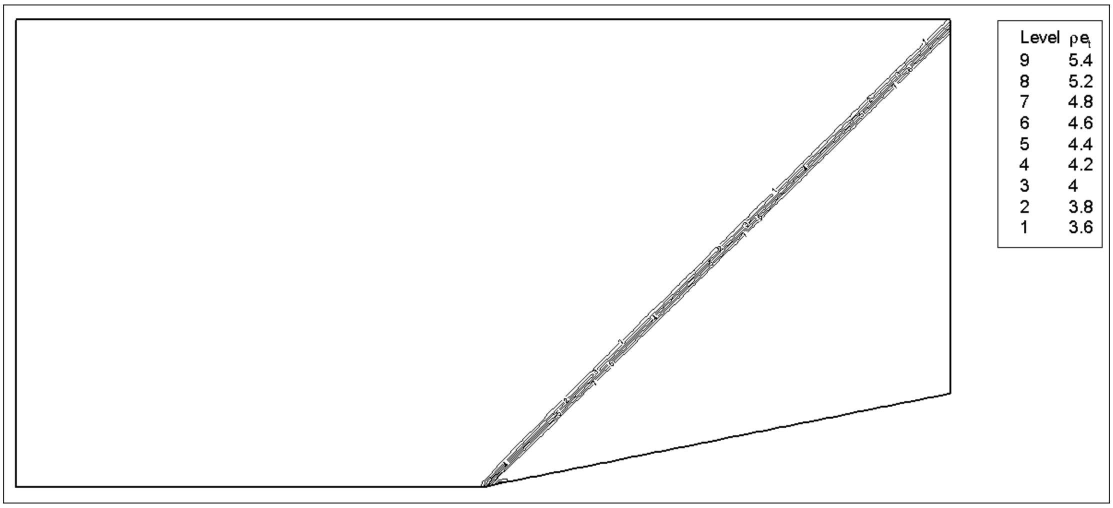

Figure 4.49 Steady state ρe_t contours, DCF = 0.20, compression corner

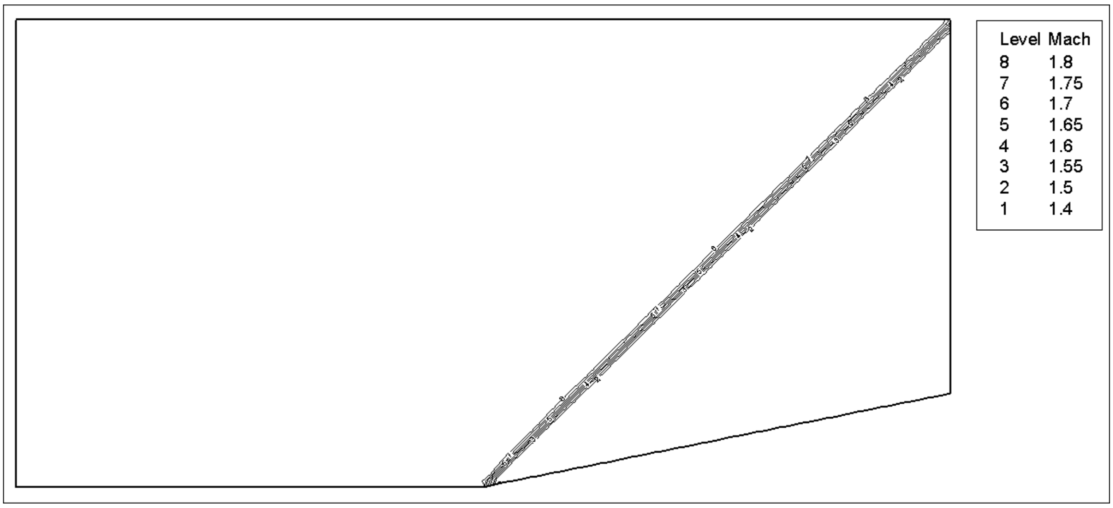

Figure 4.50 Steady state Mach No. contours, DCF = 0.20, compression corner

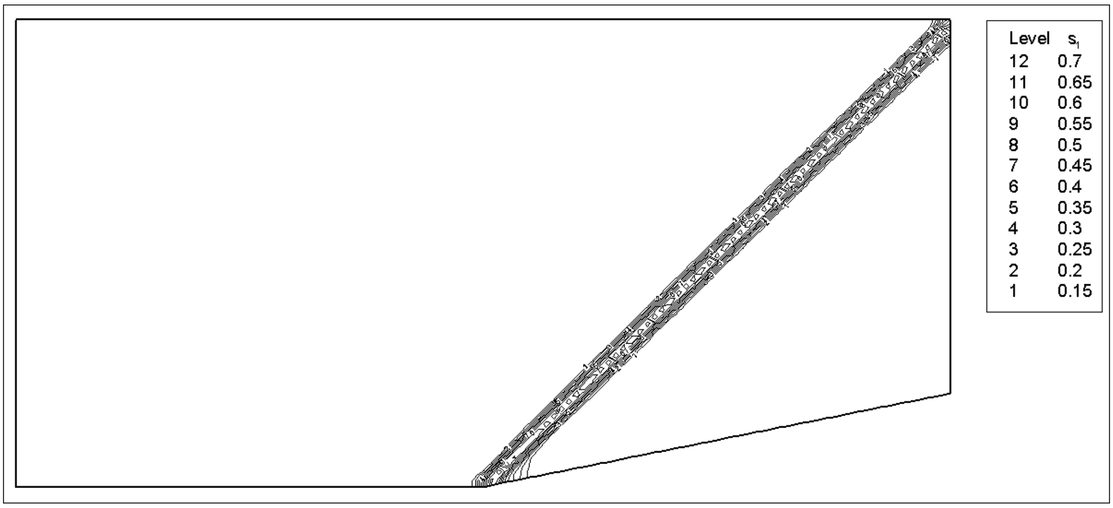

Figure 4.51 Steady state s_1 contours, DCF = 0.20, compression corner

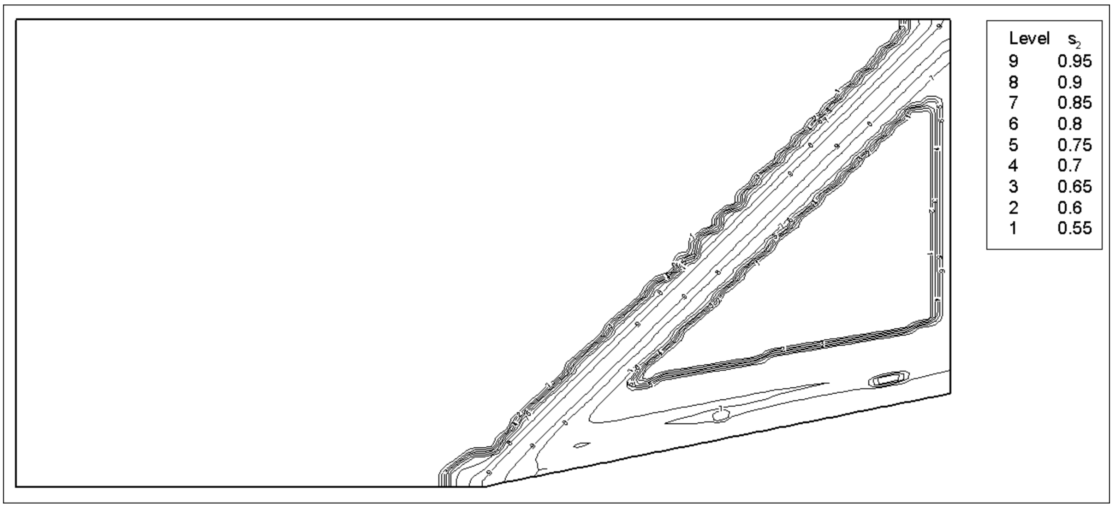

Figure 4.52 Steady state s_2 contours, DCF = 0.20, compression corner

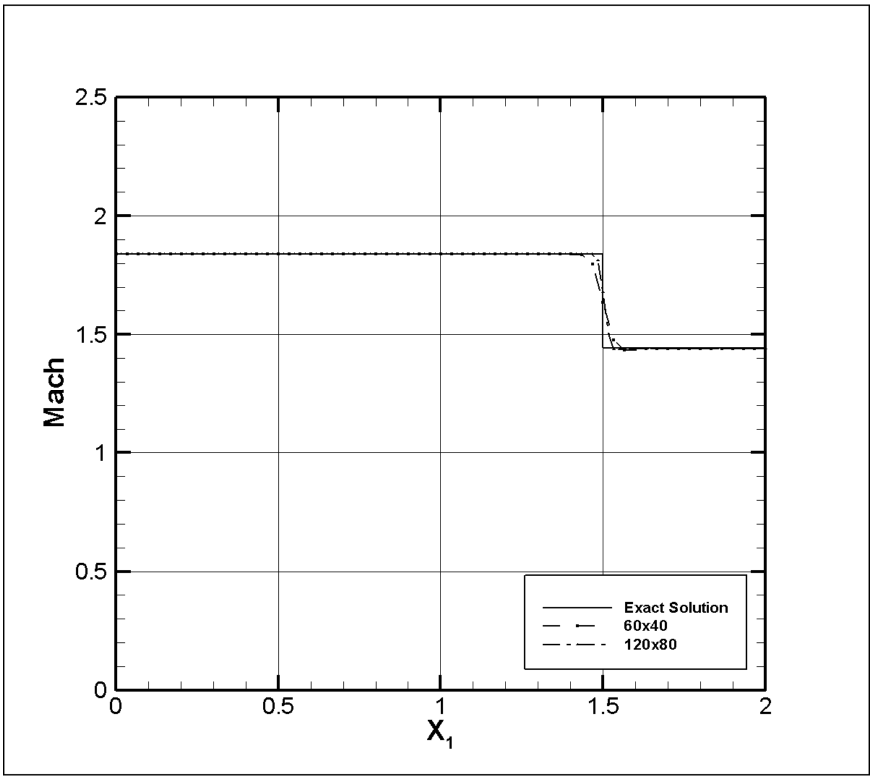

Figure 4.53 Analytical Vs numerical solution

at $x_2 = 0.5$, DCF = 0.20, compression corner

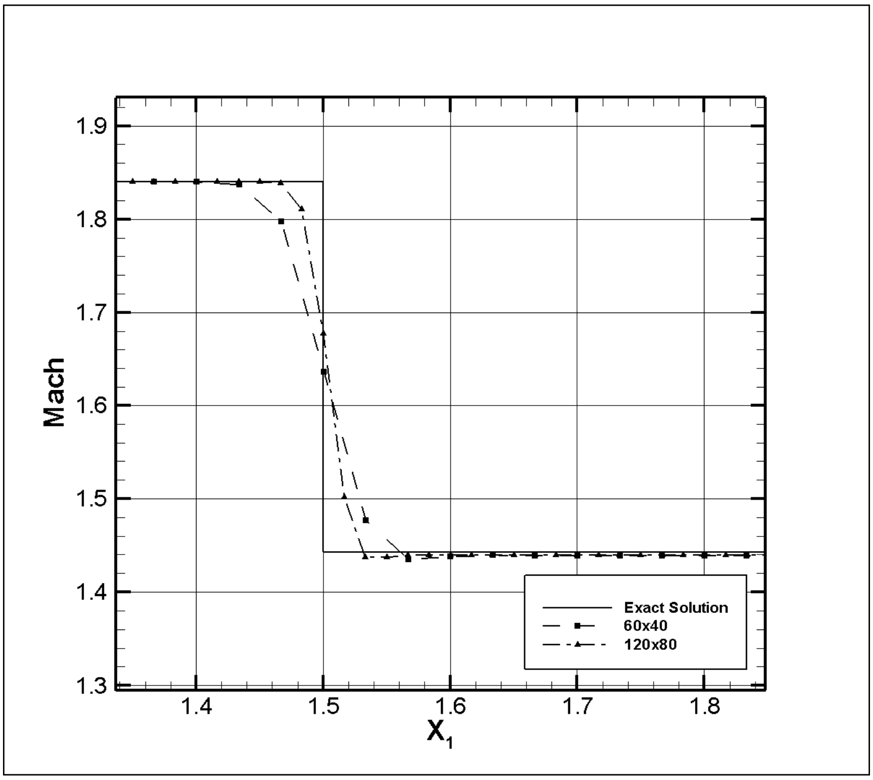

Figure 4.54 Zoom in the shockwave at $x_2 = 0.5$,

DCF = 0.20, compression corner

4.2 Time Dependent Test Case: Shock Tube Problem

The transient response accuracy of FDV formulation has been validated for the solution of the linear 1-D wave equation and the inviscid Burgers equation in section 2.5. To validate the ability of the developed FDV-FEM compressible Euler flow solver to accurately predict the transient behavior for inviscid flow problems, a famous test case with analytical solution has been solved. This test case is the shock tube problem (Riemann problem).

Initially, there is a diaphragm that separates two different gases at rest. Each has its pressure and density. Upstream, state L is defined by P_L, ρ_L, and downstream state R is defined by P_R, ρ_R. Suddenly this diaphragm is removed generating shockwave, contact discontinuity, and expansion fan as shown in Figure 4.55.

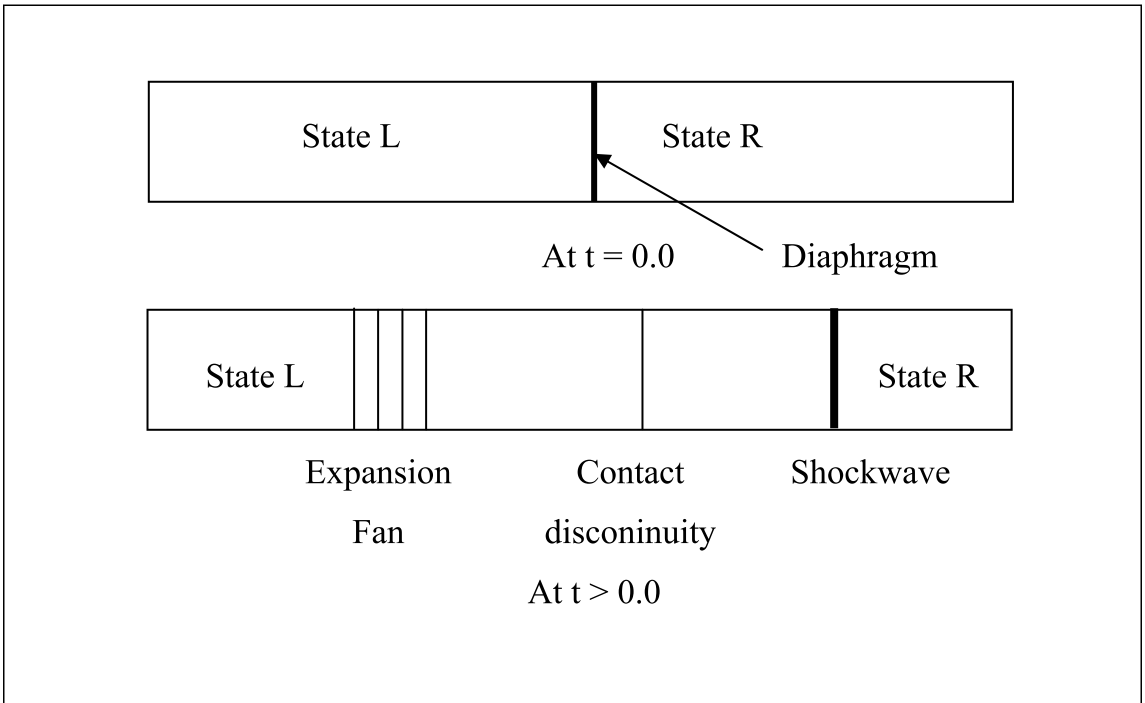

Figure 4.55 Shock tube domain

It is a simple concept problem, but a very difficult test case for any numerical scheme because it has very different flow regimes in the same domain. The analytical solution of this problem can be found in any gas dynamics or computational fluid dynamics textbook such as [1]-[4]. The problem is essentially one dimensional, so the domain may be discretized by very small number of element in the lateral direction without altering the solution accuracy. On the other hand the chord wise have significant variation and heavy grid is essential for good resolution. A grid of 1000x4 has been selected to solve this test case.

Letting $P_L = 10^5, \rho_L = 1.0, P_R = 10^4, \rho_R = 0.125$ and setting the CFL number to 0.10, the following results have been obtained. Figure 4.56 plots the variation of the density, while Figure 4.57 presents the variation of the normalized ρu_1 at different time instants. Figure 4.58 and Figure 4.59 shows the variation of the normalized ρe_t and the Mach number, respectively.

Figure 4.60 and Figure 4.61 plot the variation of the normalized pressure and relative change in entropy, respectively. Same as observed in section 2.5, the FDV formulation succeeded in accurate prediction for the transient response of a complicated convection problem. The method could recover the solution with time without major losses in the convected solution. Good resolution for the contact discontinuity as well as the shockwave has been obtained.

Figure 4.61 is of great importance because it illustrates the fact that the method provides numerical diffusion for the regions in the flowfield that needs it. For smooth or small variations the method virtually gives no dissipation. This is evident from the zero entropy change provided in all regions expect near the contact discontinuity and the shockwave.

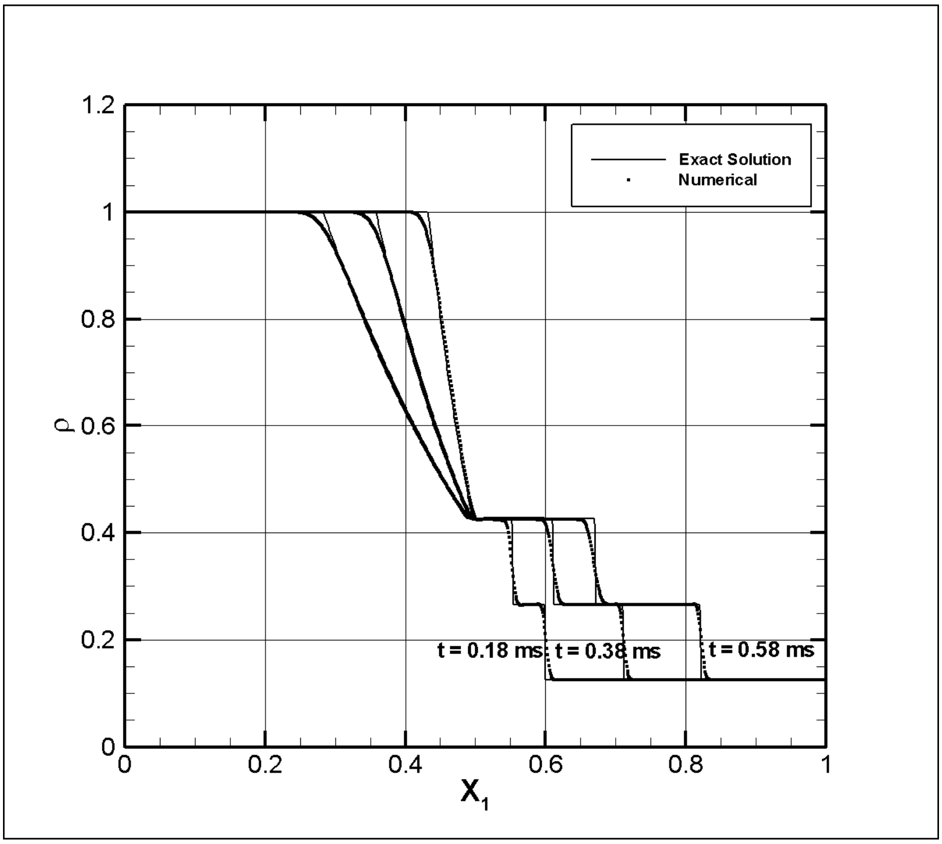

Figure 4.56 ρ vs x_1, shock tube

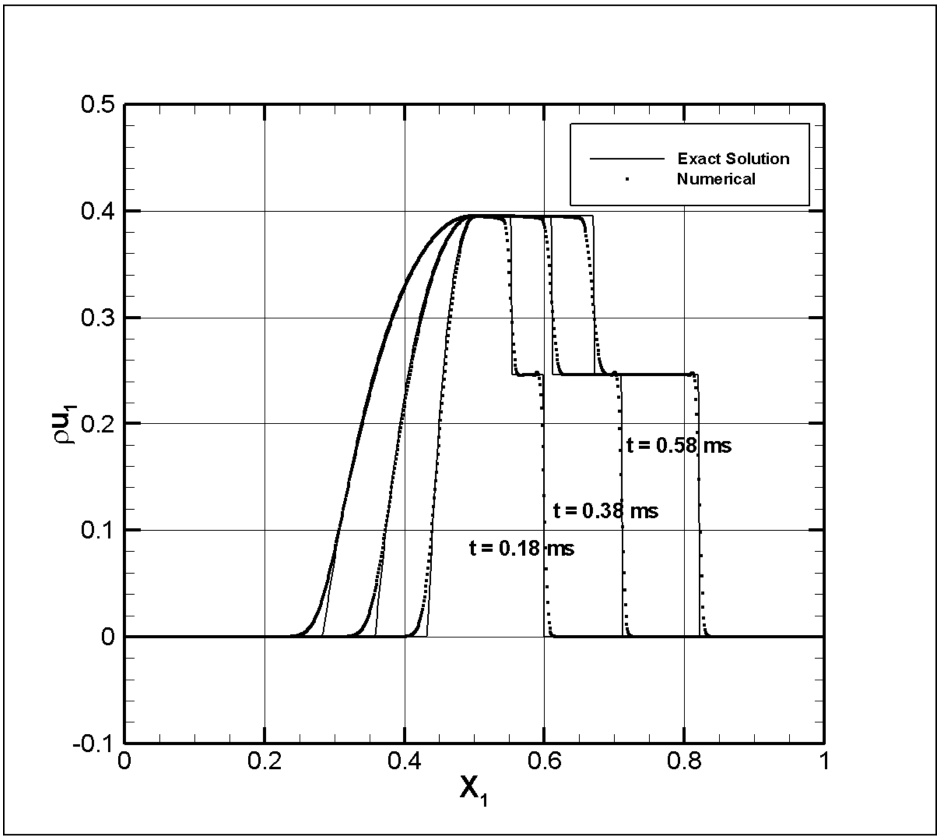

Figure 4.57 ρu_1 vs x_1, shock tube

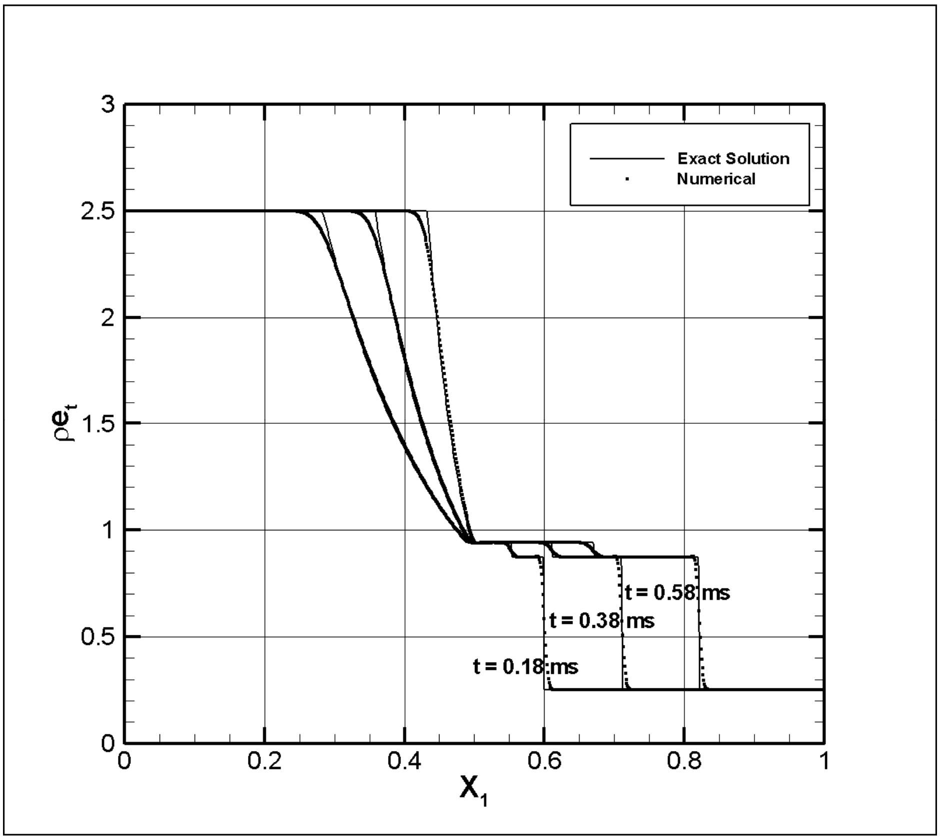

Figure 4.58 ρe_t vs x_1, shock tube

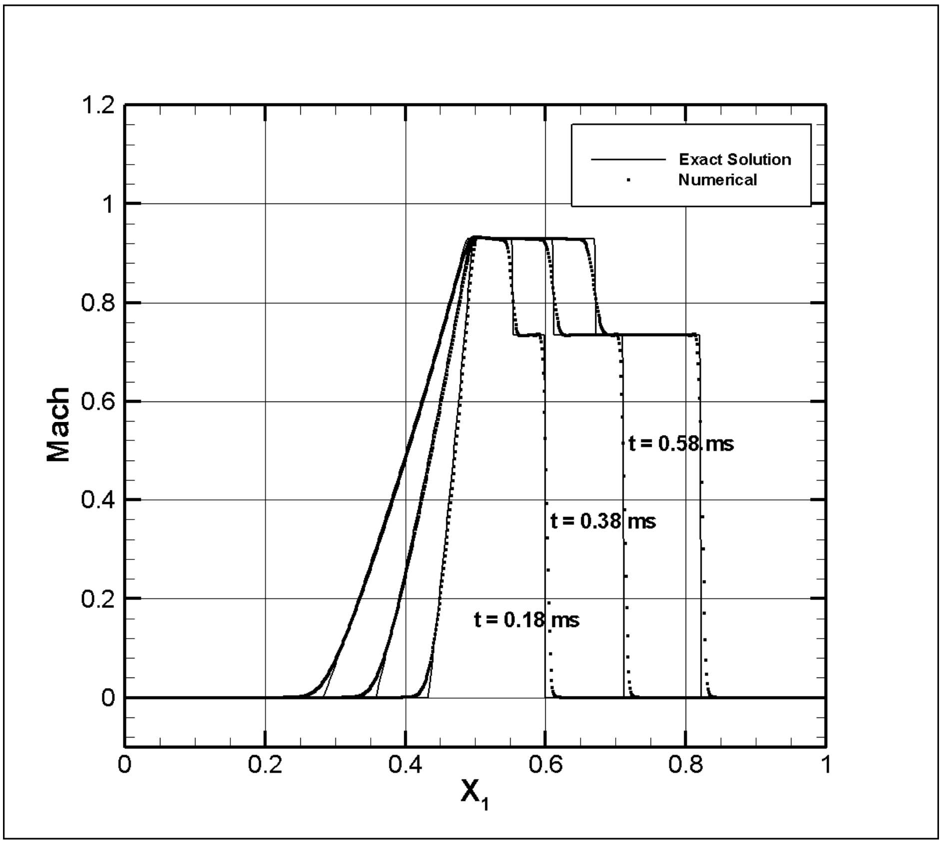

Figure 4.59 Mach No. vs x_1, shock tube

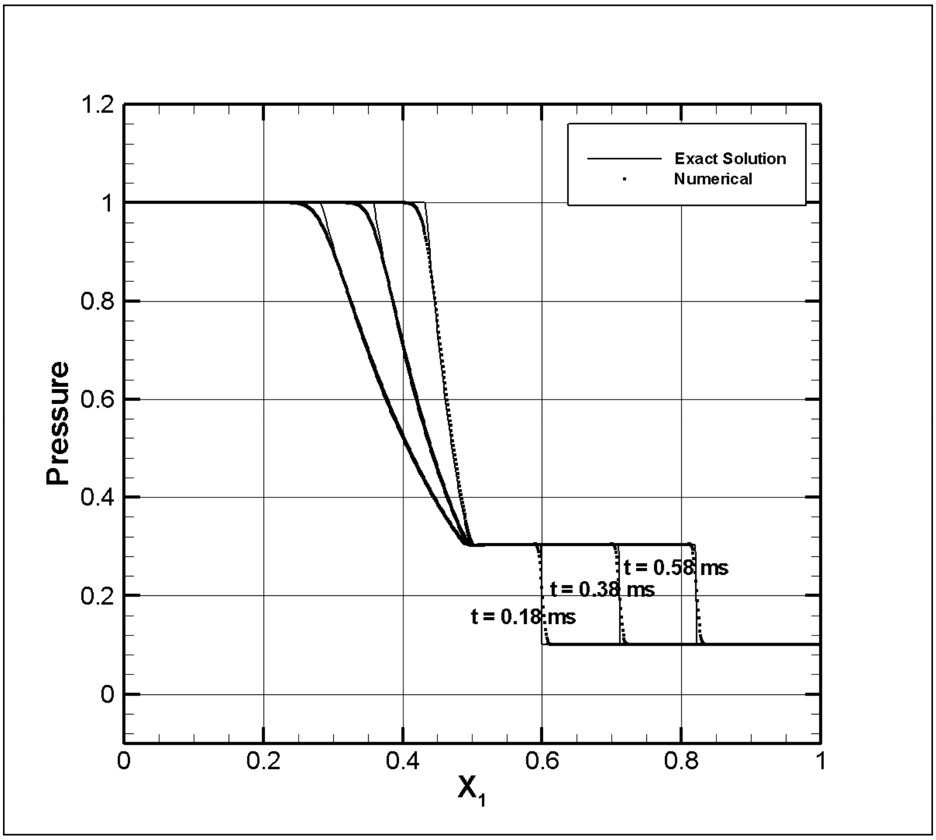

Figure 4.60 Pressure vs x_1, shock tube

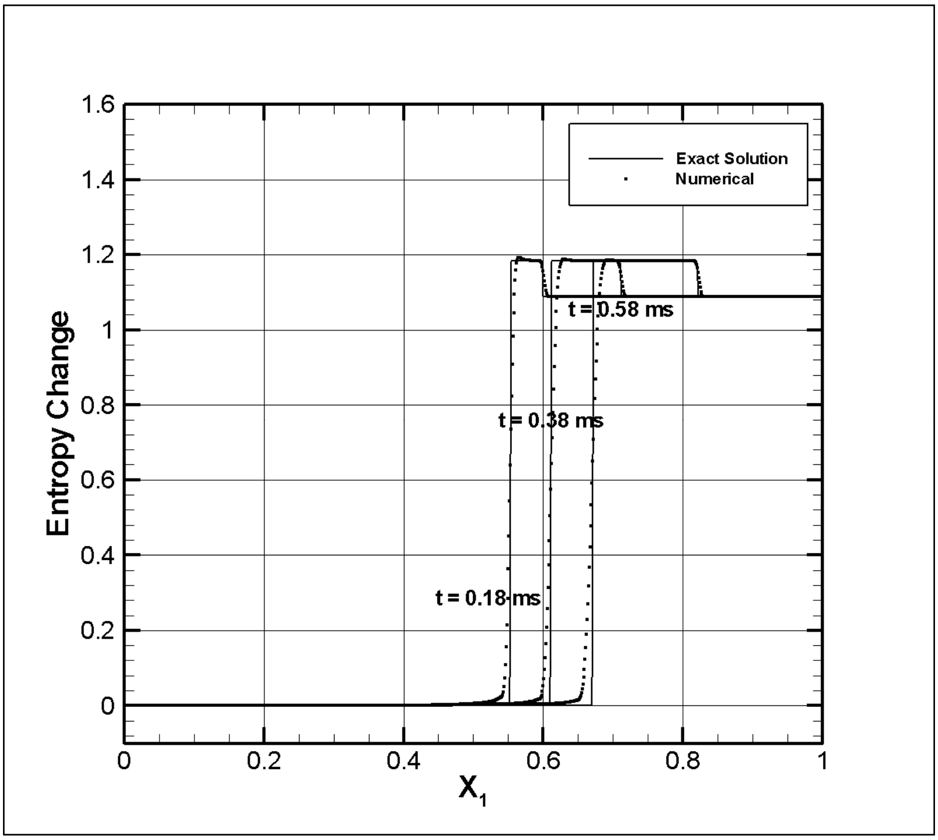

Figure 4.61 Entropy change vs x_1, shock tube

4.3 Supersonic Flows

This section is devoted to the validation of the FDV/MFDV formulations in handling the supersonic flows. The discussion is limited to internal flows only. [60] has been submitted for publication during this research work. This paper contains discussions and validations for the ability of the MFDV formulation to correctly analyze the supersonic internal flow problems. Some of [60] results are repeated here in this section for convenience.

In the next three subsections three problems will be investigated. The complexity of these problems varies significantly. The first problem is of relatively simple shock structure than the second one. The third problem is of special importance because of its bow shock structure encountered in its second case.

4.3.1 Extended Compression Corner

The extended compression corner problem is shown in Figure 4.62. The value of 0.2 has been chosen for the height of the ramp. The solution domain is discretized using 120x60 uniform bilinear quadrilateral elements. The value of 0.20 is used for the DCF.

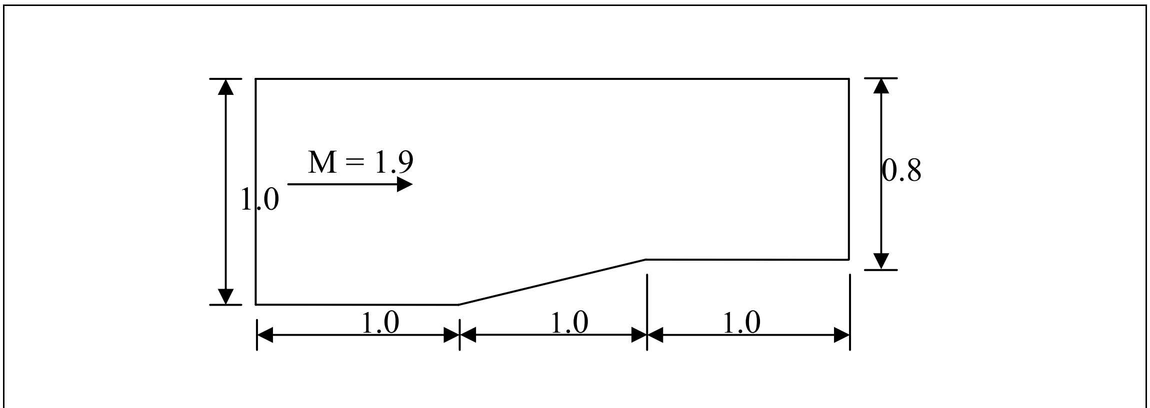

Figure 4.62 Extended compression corner domain

The inlet boundary is supersonic and the following values have been used:

$$\rho = 1.0, \qquad \rho u_1 = 1.90$$
$$\rho u_2 = 0.0, \qquad \rho e_t = 3.47855$$

Again the no-penetration boundary condition is applied on both the upper and lower boundaries and the exit boundary is left free being supersonic.

Figure 4.63 to Figure 4.70 show the transient Mach number contours at various time steps. The steady state solution is reached at t = 3.321 sec. Figure 4.71, Figure 4.72, Figure 4.73, and Figure 4.74 show the contour plot for the conservation variables $\rho, \rho u_1, \rho u_2, \rho e_t$, respectively, at steady state. Figure 4.75, Figure 4.76, and Figure 4.77 show the steady state Mach number, s_1, and s_2 contours, respectively.

Figure 4.78 shows the effect of grid refinement on the solution at $x_2 = 0.5$. It is clear that the FDV/MFDV formulation has succeeded in recovering the interference between the reflected shock and the expansion fan. Also the method is capable of approaching the exact solution by increasing the computational grid.

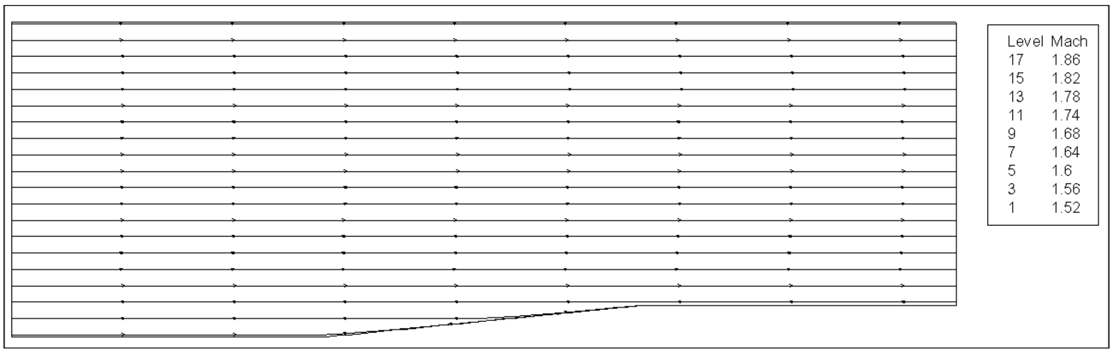

Figure 4.63 Mach No. contours at t = 0.000, extended compression corner

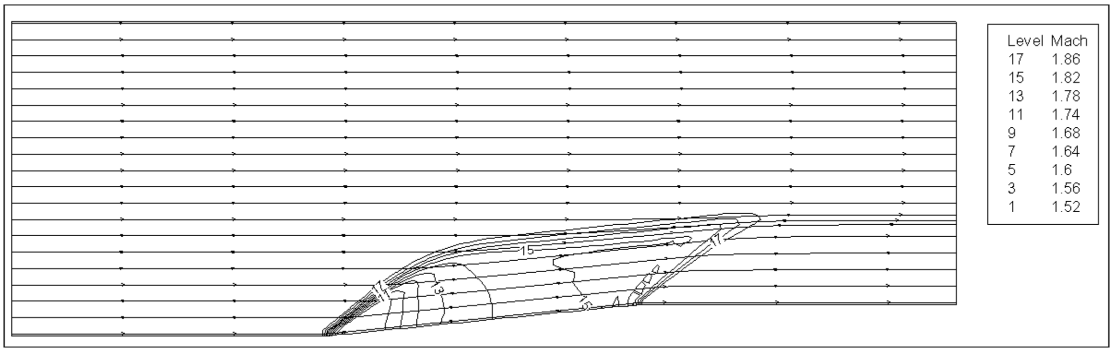

Figure 4.64 Mach No. contours at t = 0.259, extended compression corner

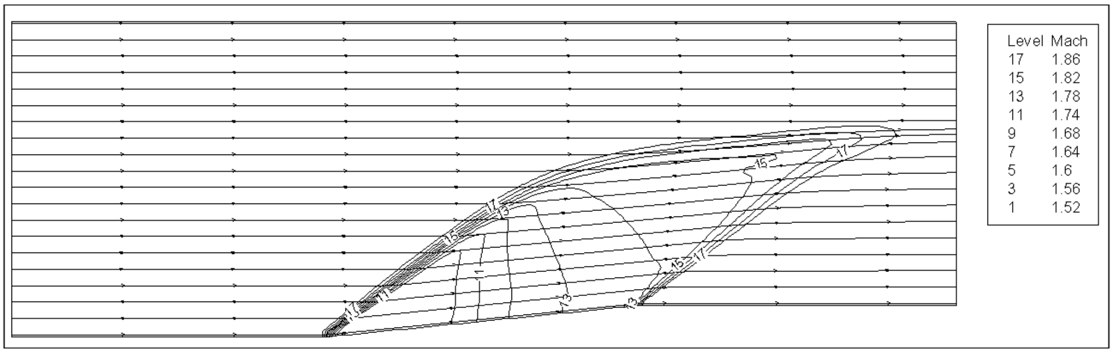

Figure 4.65 Mach No. contours at t = 0.517, extended compression corner

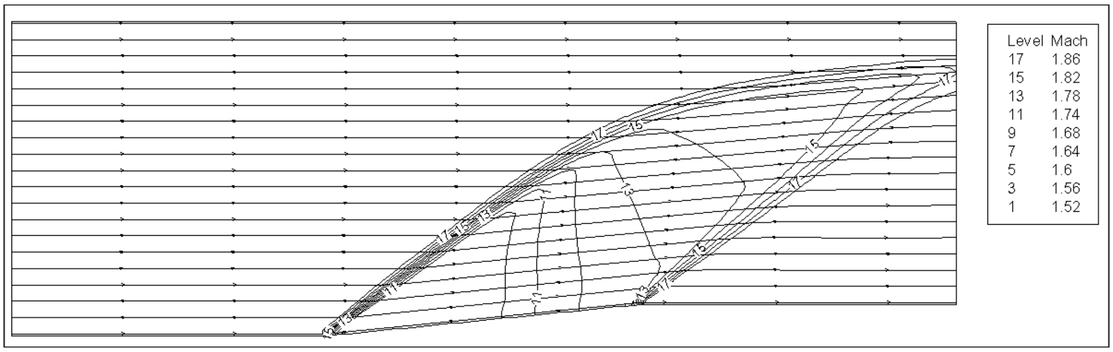

Figure 4.66 Mach No. contours at t = 0.712, extended compression corner

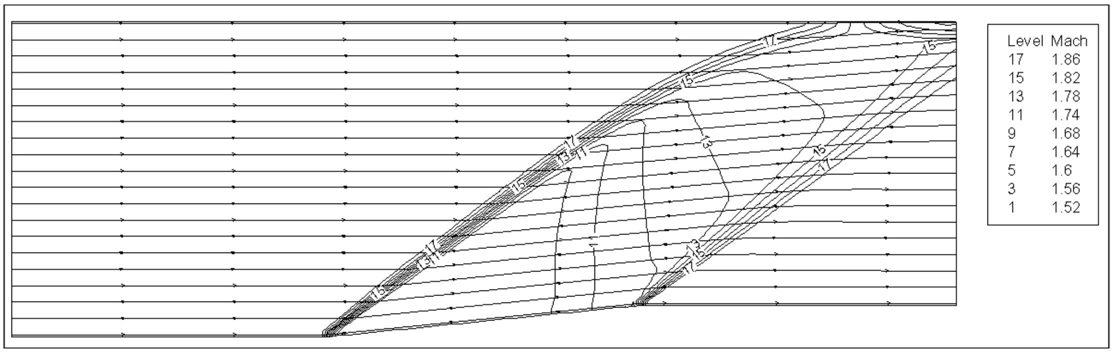

Figure 4.67 Mach No. contours at t = 0.907, extended compression corner

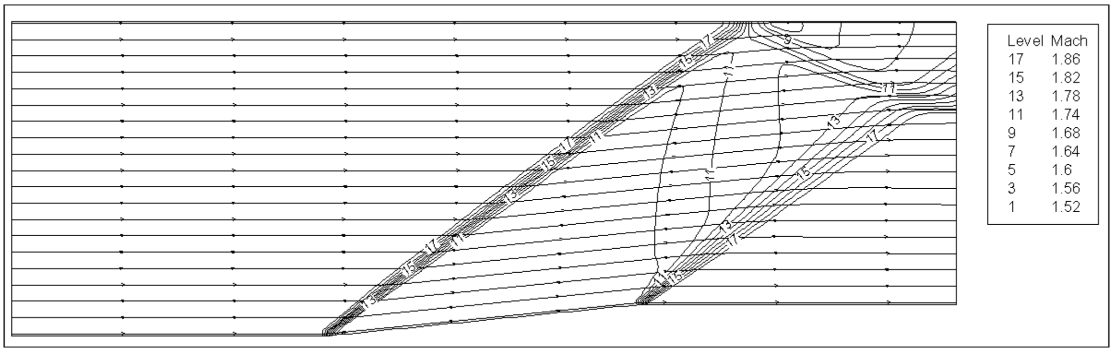

Figure 4.68 Mach No. contours at t = 1.300, extended compression corner

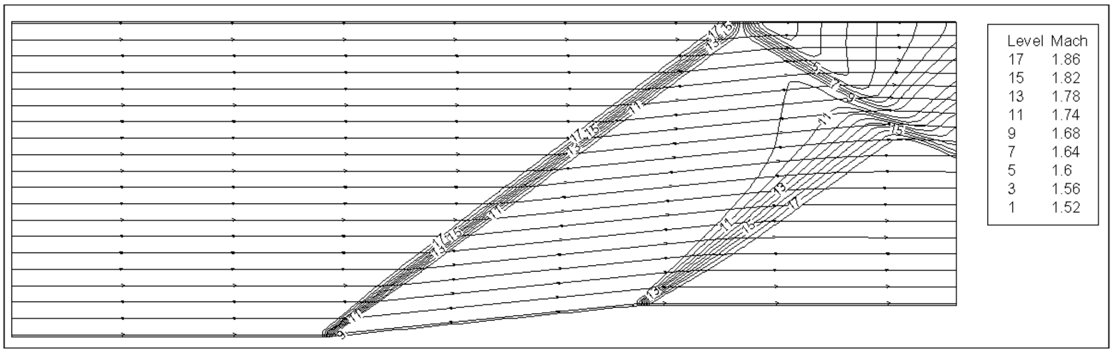

Figure 4.69 Mach No. contours at t = 1.760, extended compression corner

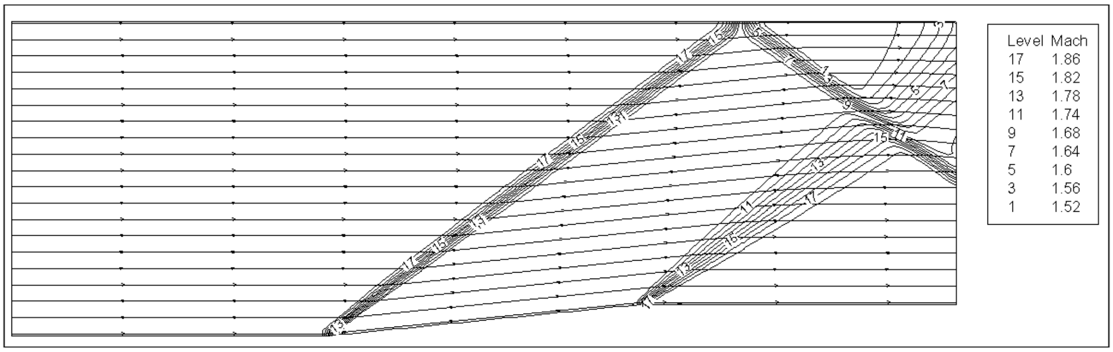

Figure 4.70 Mach No. contours at t = 2.490, extended compression corner

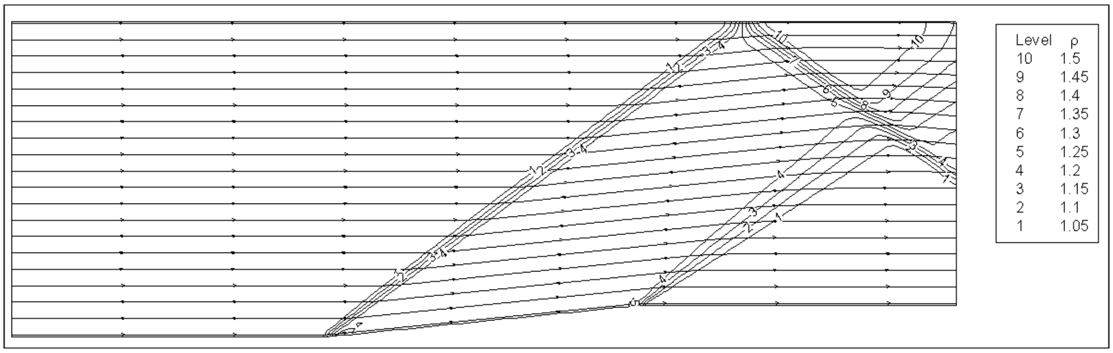

Figure 4.71 Steady State ρ contours, extended compression corner

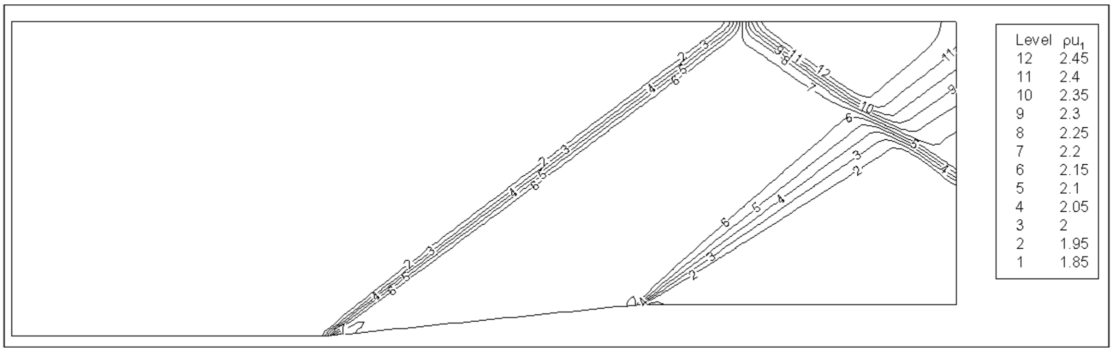

Figure 4.72 Steady State ρu_1 contours, extended compression corner

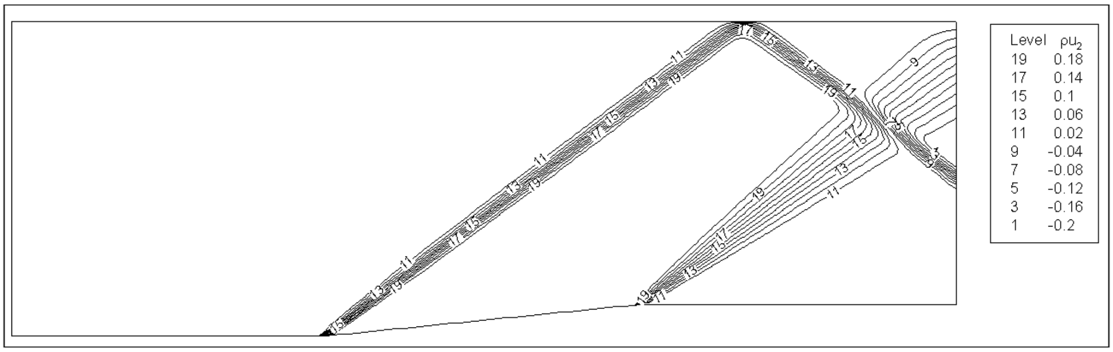

Figure 4.73 Steady State ρu_2 contours, extended compression corner

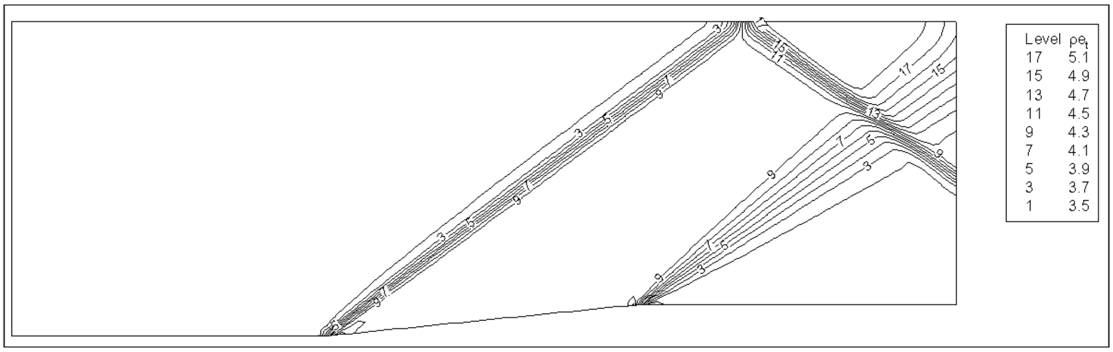

Figure 4.74 Steady State ρe_t contours, extended compression corner

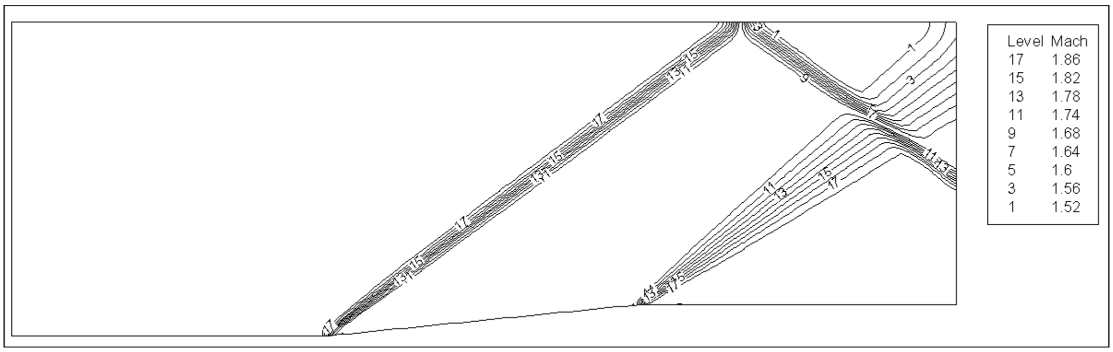

Figure 4.75 Steady State Mach No. contours, extended compression corner

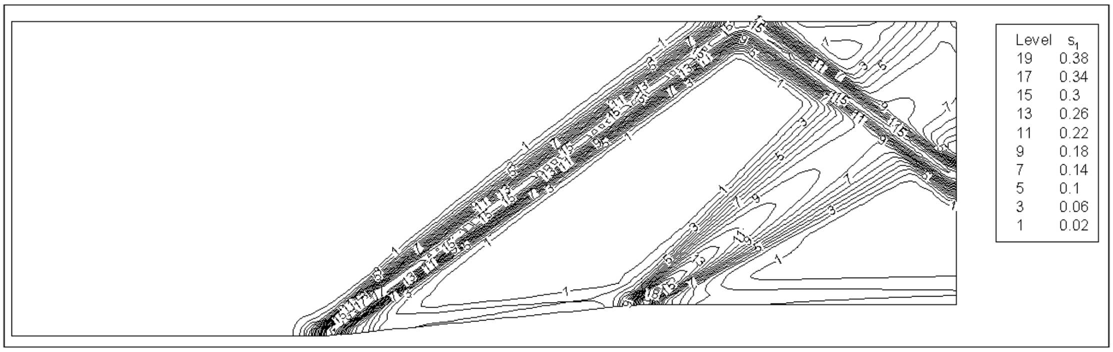

Figure 4.76 Steady State s_1 contours, extended compression corner

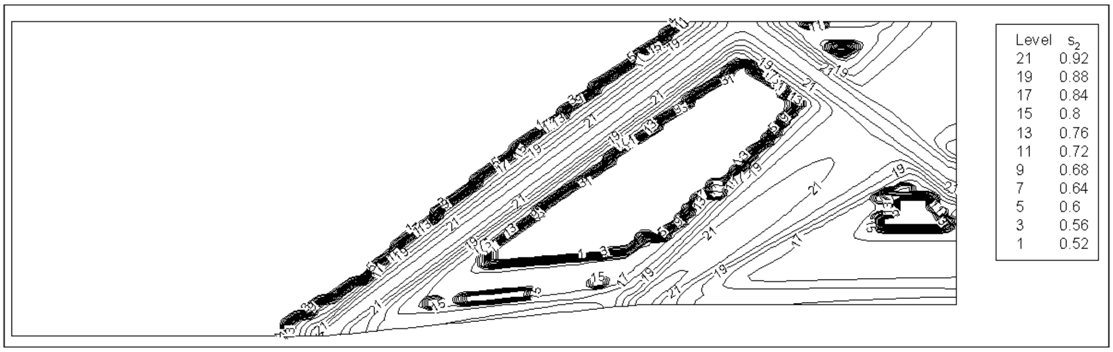

Figure 4.77 Steady State s_2 contours, extended compression corner

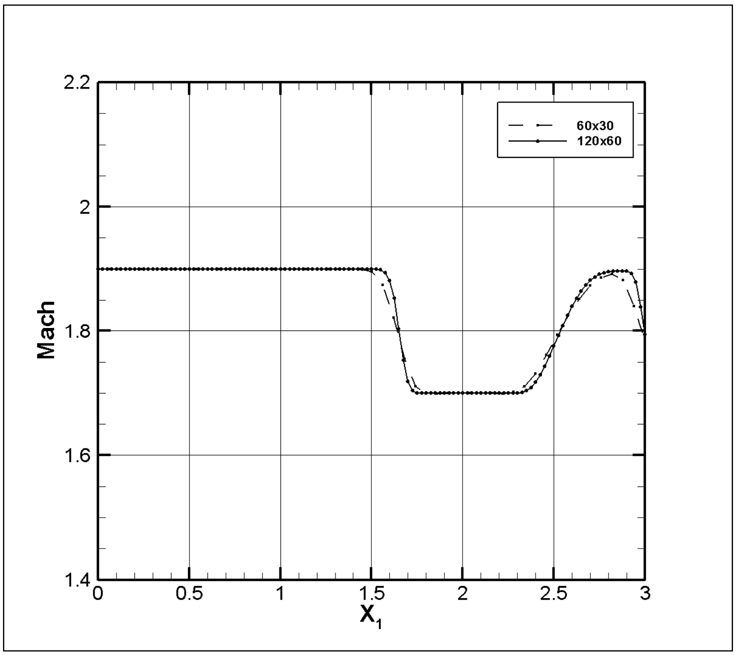

Figure 4.78 Effect of grid refinement at $x_2 = 0.5$,

extended compression corner

4.3.2 Half Wedge in a Supersonic Wind Tunnel

Figure 4.79 shows the configuration of a half wedge that is placed in a supersonic-wind-tunnel. The half wedge maximum height has been set to 0.04. The solution domain is discretized using 120x60 uniform bilinear quadrilateral elements. The value of 0.20 is used for the DCF. The inlet boundary is supersonic and the following values have been used:

$$\rho = 1.0, \qquad \rho u_1 = 1.40$$
$$\rho u_2 = 0.0, \qquad \rho e_t = 2.76575$$

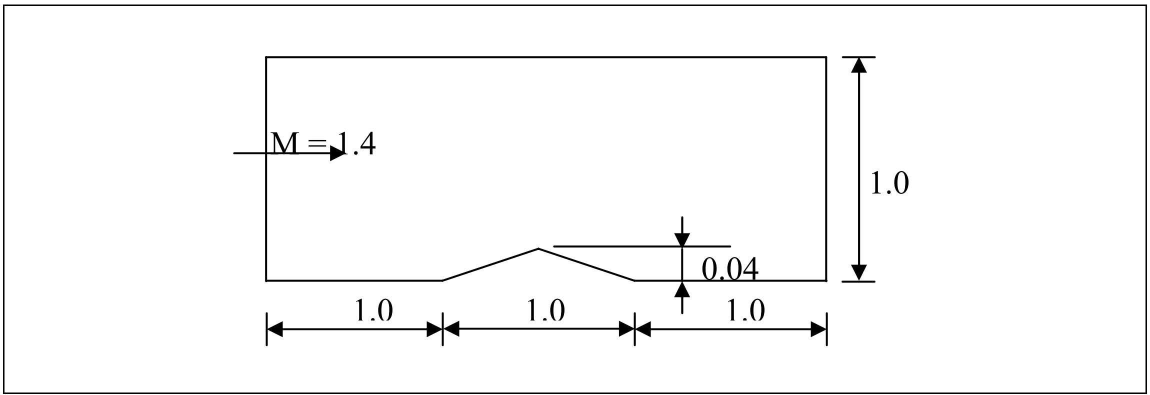

Figure 4.79 Half wedge domain

The inviscid wall boundary condition is applied on both the upper and lower boundaries and the exit boundary is left free being supersonic.

Figure 4.80 to Figure 4.88 show the transient Mach number contours at various time steps. The steady state solution is obtained at t = 9.597 sec. Figure 4.89, Figure 4.90, Figure 4.91, and Figure 4.92 plot the contours for the conservation variables $\rho, \rho u_1, \rho u_2, \rho e_t$ at the steady state, respectively.

Figure 4.93, Figure 4.94, and Figure 4.95 show the Mach number, s_1, and s_2 steady state contours, respectively. Figure 4.96 shows the effect of grid refinement on the solution at $x_2 = 0.5$.

It is clear that the FDV/MFDV formulation succeeded in recovering the interference between the reflected shock and the expansion fan as well as the interaction between the two intersecting shocks. Also as observed before, the method is capable of approaching the exact solution by increasing the computational grid.

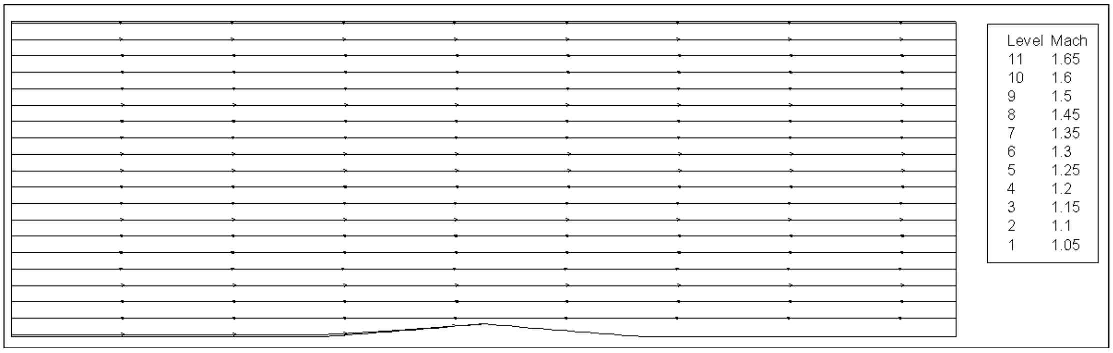

Figure 4.80 Mach No. contours at t = 0.000, half wedge

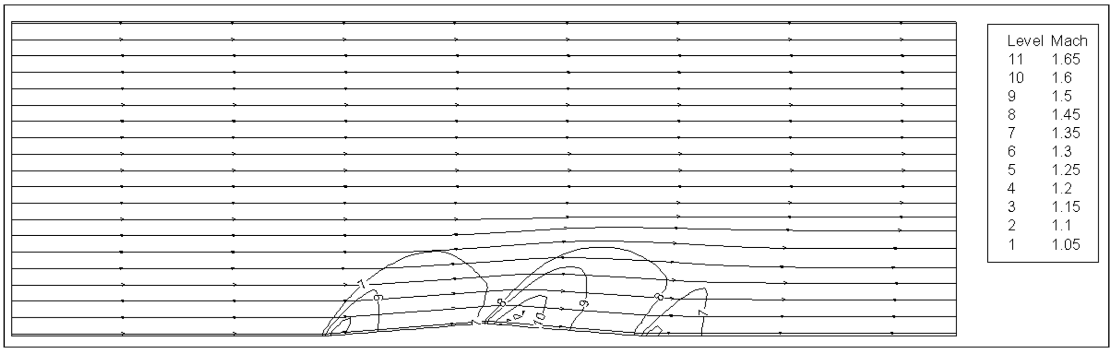

Figure 4.81 Mach No. contours at t = 0.317, half wedge

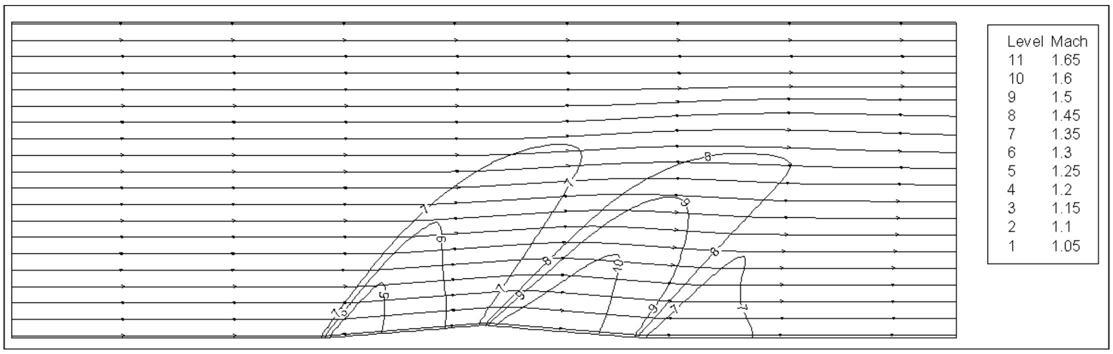

Figure 4.82 Mach No. contours at t = 0.714, half wedge

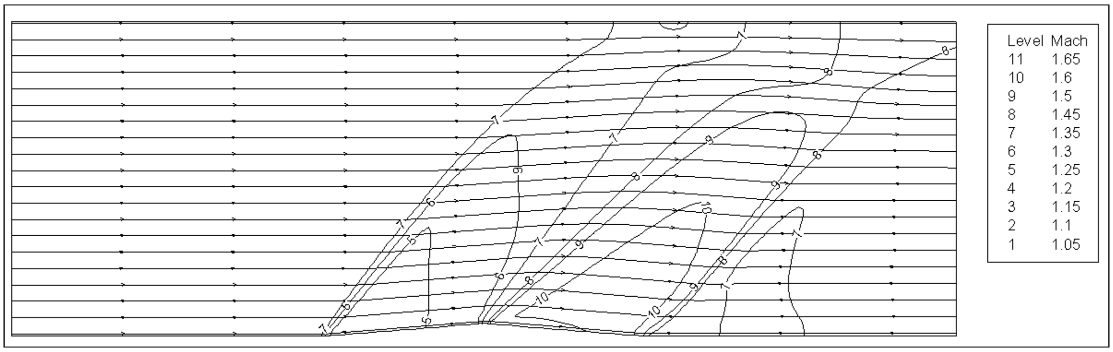

Figure 4.83 Mach No. contours at t = 1.194, half wedge

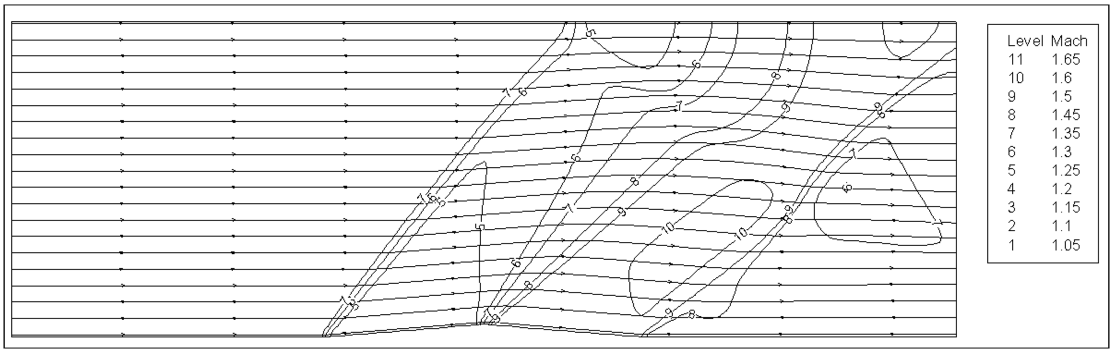

Figure 4.84 Mach No. contours at t = 1.757, half wedge

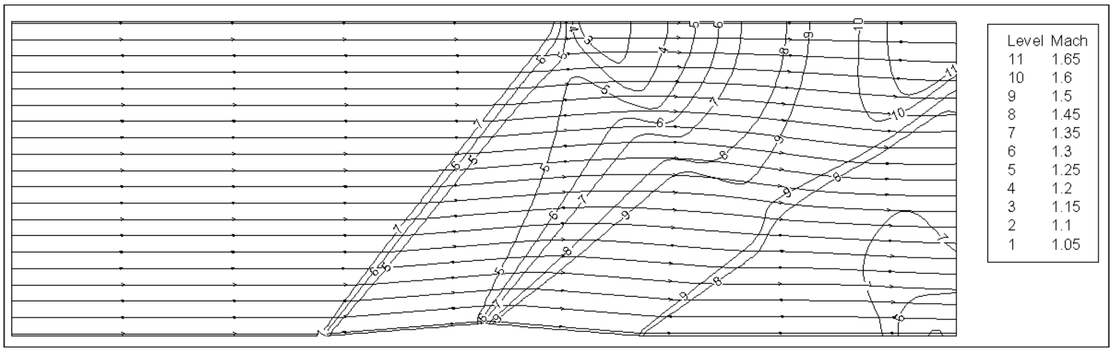

Figure 4.85 Mach No. contours at t = 2.559, half wedge

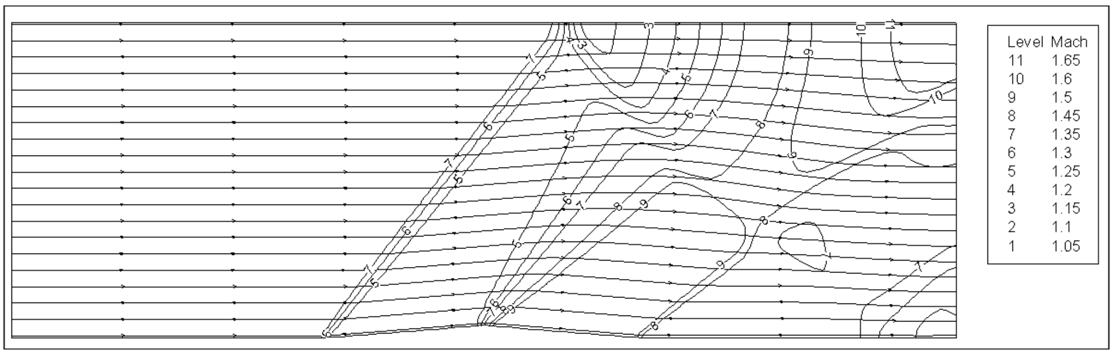

Figure 4.86 Mach No. contours at t = 3.038, half wedge

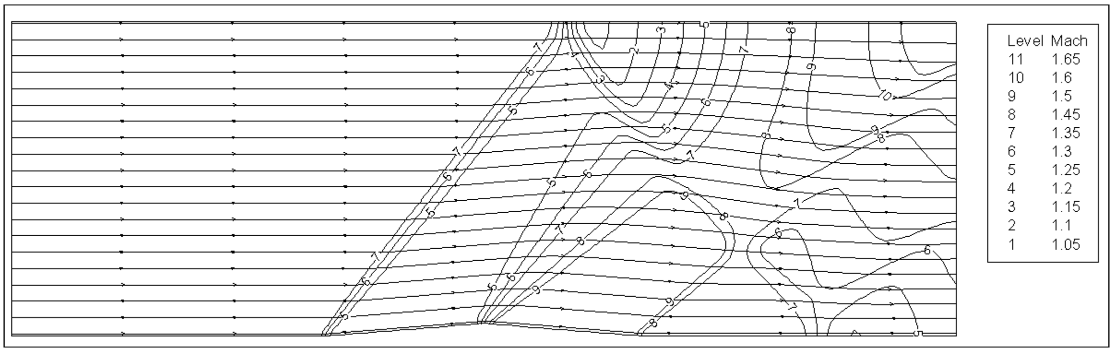

Figure 4.87 Mach No. contours at t = 4.079, half wedge

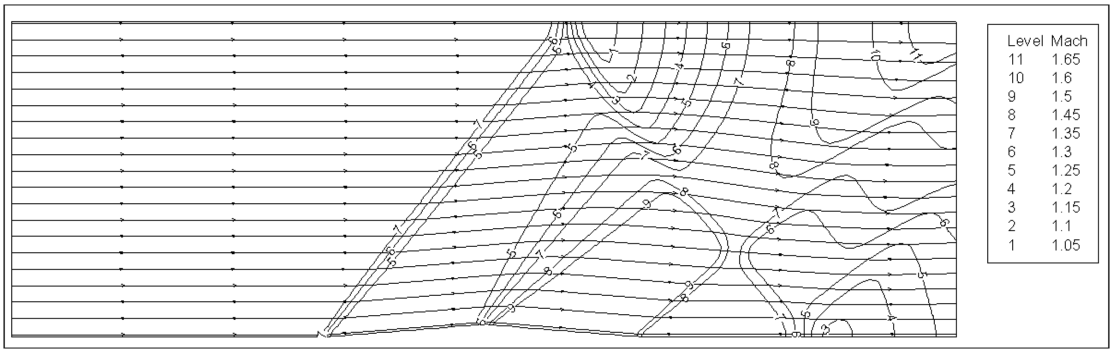

Figure 4.88 Mach No. contours at t = 5.044, half wedge

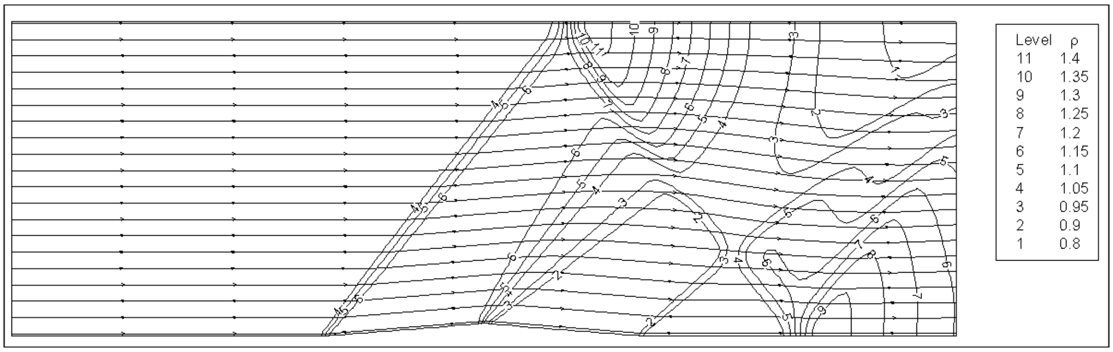

Figure 4.89 Steady State ρ contours, half wedge

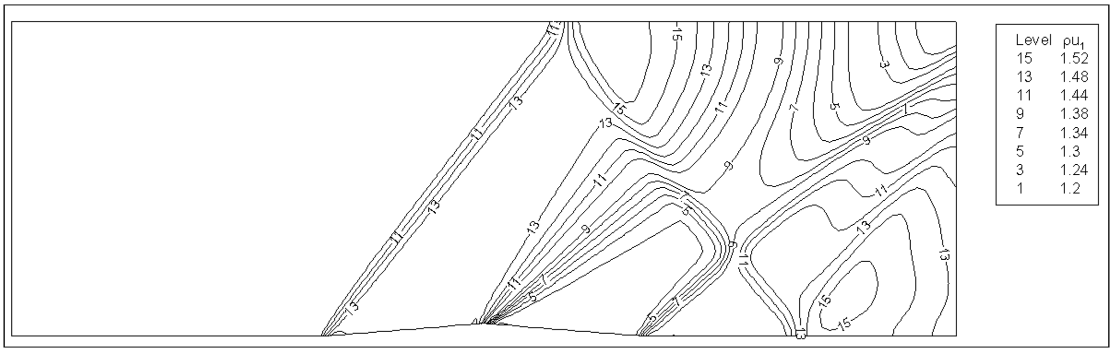

Figure 4.90 Steady State ρu_1 contours, half wedge

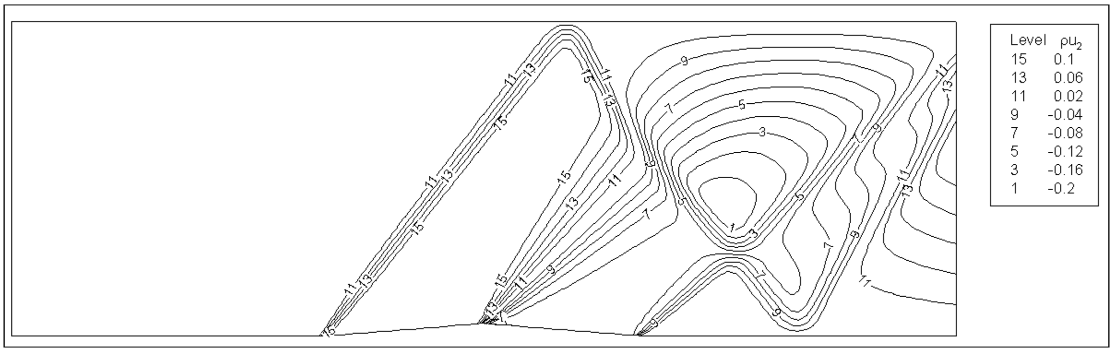

Figure 4.91 Steady State ρu_2 contours, half wedge

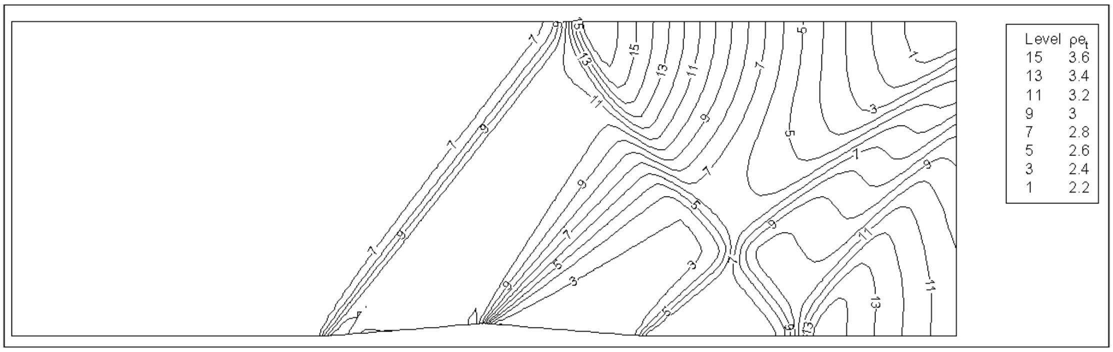

Figure 4.92 Steady State ρe_t contours, half wedge

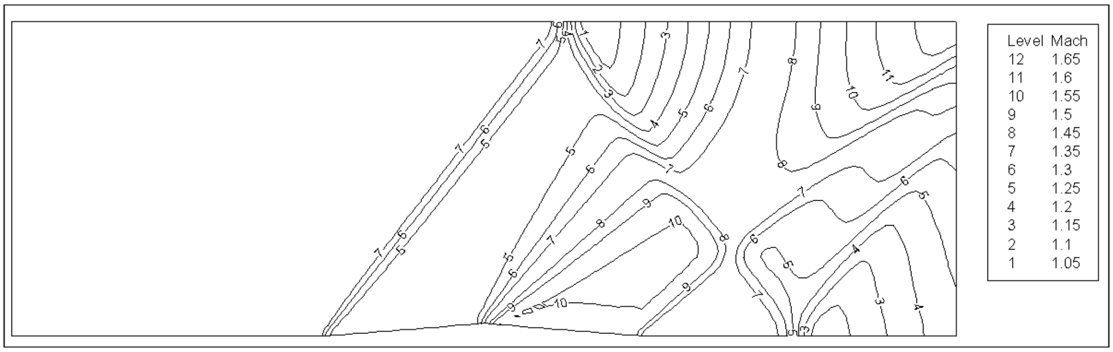

Figure 4.93 Steady State Mach No. contours, half wedge

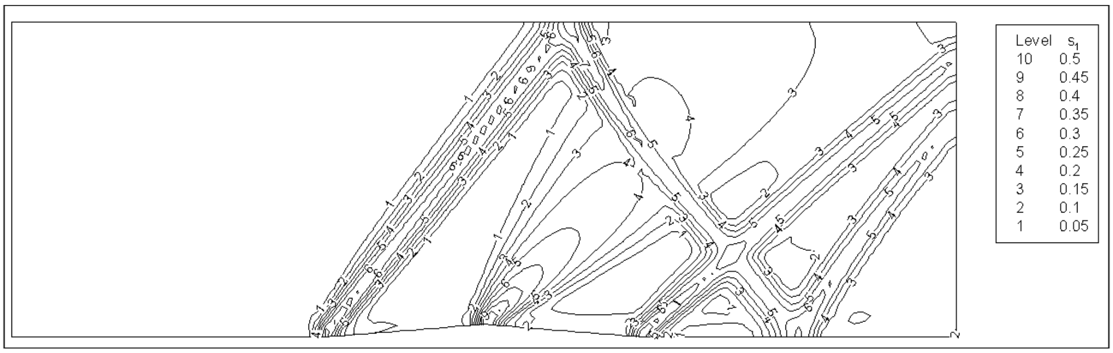

Figure 4.94 Steady State s_1 contours, half wedge

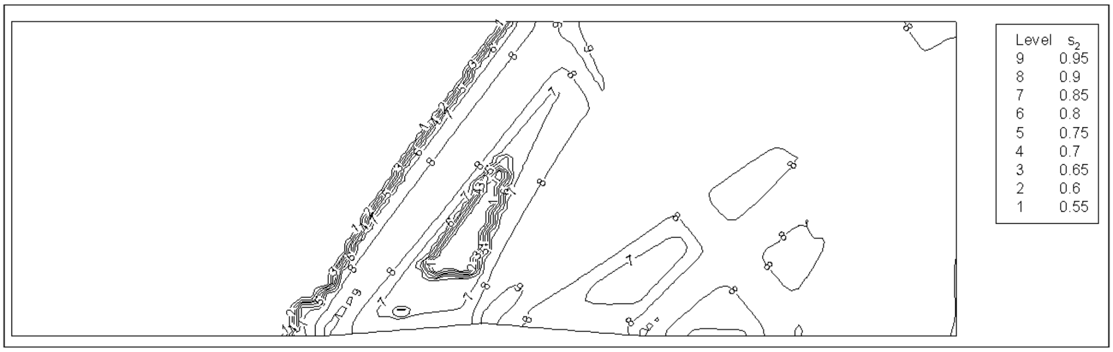

Figure 4.95 Steady State s_2 contours, half wedge

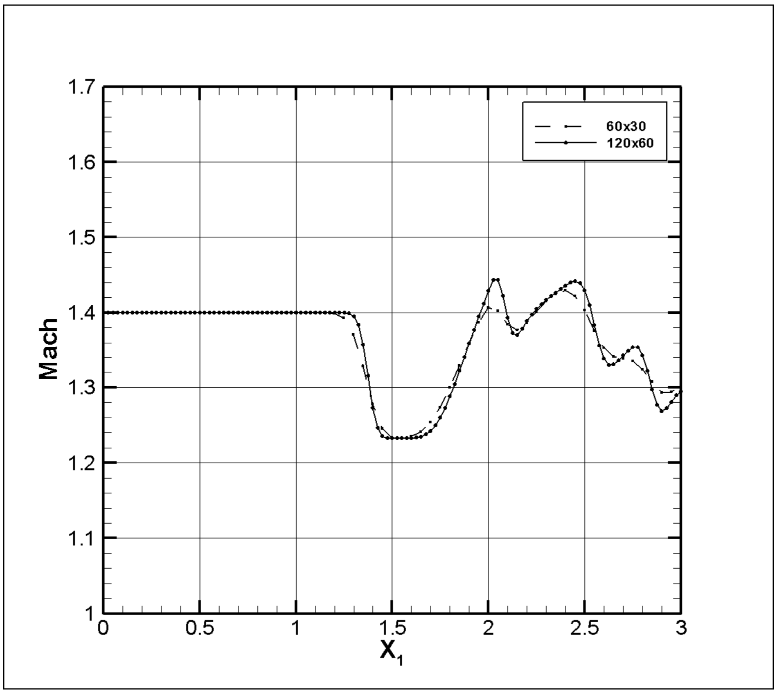

Figure 4.96 Effect of grid refinement at

$x_2 = 0.5$, half wedge

4.3.3 Circular Arc Bump in Supersonic Wind Tunnel

This problem is the third and final test case to verify the ability of the FDV/MFDV formulation in handling supersonic flows. In this test case a circular arc is placed on the lower wall of a supersonic wind tunnel. Two numerical experimentation setups have been investigated.

The first is close to the previous test case. In which the height of the arc is relatively small. While the second one offer a relatively complex flow situation when the height of the circular arc increases. This is due to the bow shock that will be generated. In the second case the height of the wind tunnel section will be increased to ensure no interference from the upper wall as much as possible. While in the first case the upper wall was seated relatively close to the lower wall to see how the proposed theory will resolve the interferences from the shocks and the expansion over the circular arc.

4.3.3.1 0.04 Circular Arc

Figure 4.97 shows the configuration of a circular arc that is placed in a supersonic wind tunnel. The circular arc maximum height has been set to 0.04. The solution domain is discretized using 120x60 uniform bilinear quadrilateral elements. The value of 0.20 is used for the DCF.

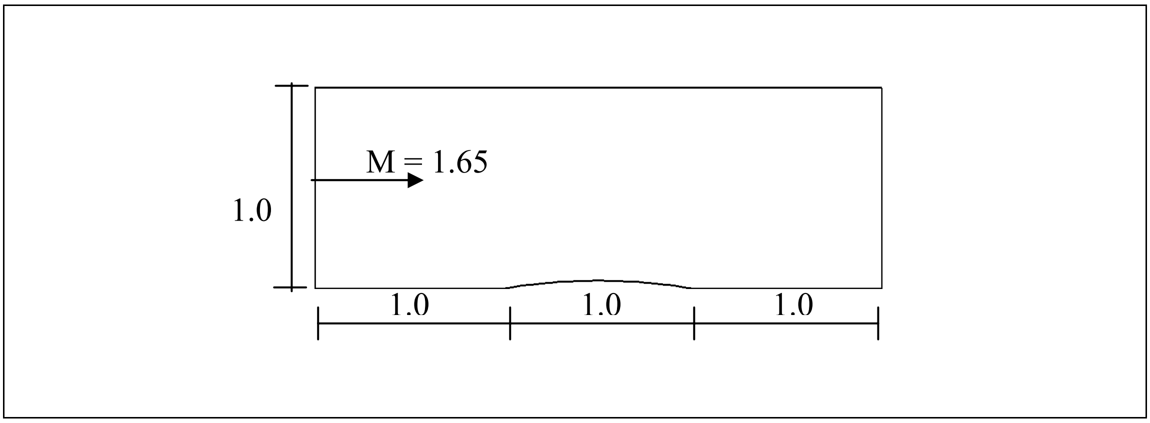

Figure 4.97 0.04 circular arc domain

The inlet boundary is supersonic and the following values have been used:

$$\rho = 1.0, \qquad \rho u_1 = 1.65$$
$$\rho u_2 = 0.0, \qquad \rho e_t = 3.147$$

The inviscid wall boundary condition is applied on both the upper and lower boundaries and the exit boundary is left free being supersonic. Figure 4.98 to Figure 4.105 show the transient Mach number contours at various time steps. The steady state solution is obtained at t = 4.328 sec. Figure 4.106, Figure 4.107, Figure 4.108, and Figure 4.109 show the steady state contours of the conservation variables $\rho, \rho u_1, \rho u_2, \rho e_t$, respectively. Figure 4.110, Figure

4.111, and Figure 4.112 show the Mach number, s_1, and s_2 steady state contours, respectively. Figure 4.113 shows the effect of grid refinement on the solution at $x_2 = 0.5$. Again, the FDV/MFDV formulation succeeded in recovering the interference between the reflected shock and the expansion over the circular arc as well as the intersection of the two interfering shocks. Also as observed before, the method is capable of approaching the exact solution by increasing the computational grid.

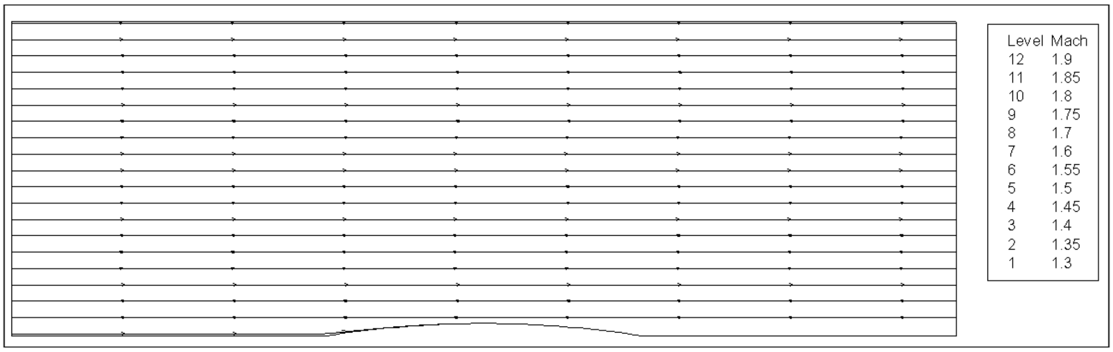

Figure 4.98 Mach No. contours at t = 0.000, 0.04 circular arc

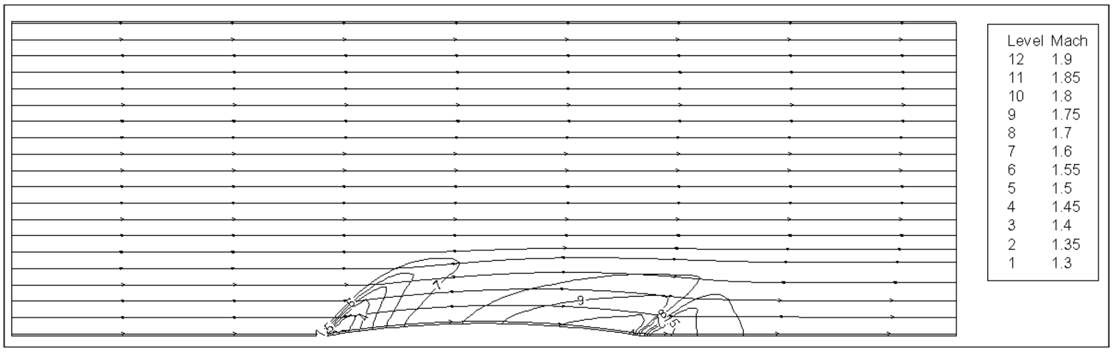

Figure 4.99 Mach No. contours at t = 0.222, 0.04 circular arc

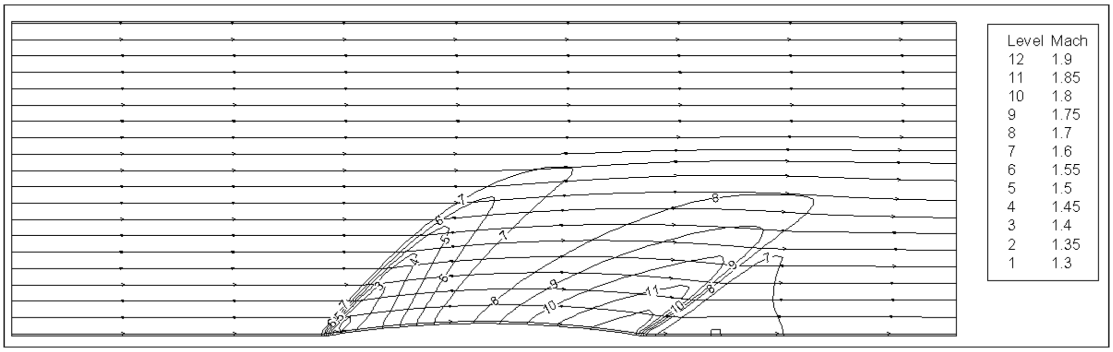

Figure 4.100 Mach No. contours at t = 0.515, 0.04 circular arc

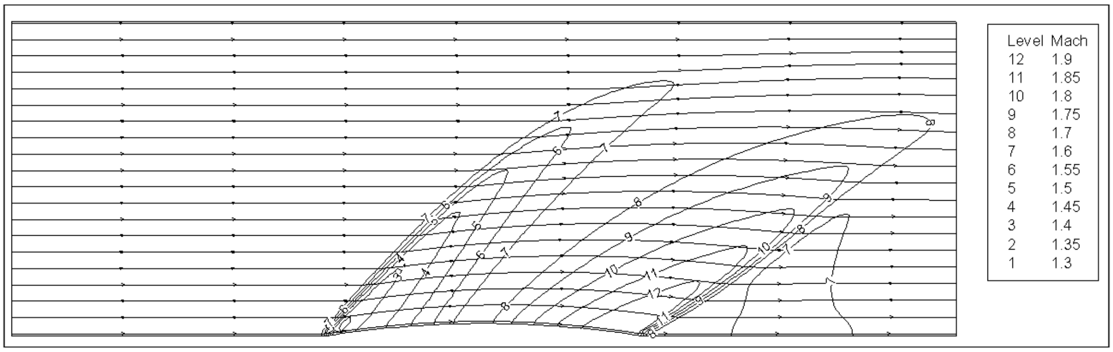

Figure 4.101 Mach No. contours at t = 0.807, 0.04 circular arc

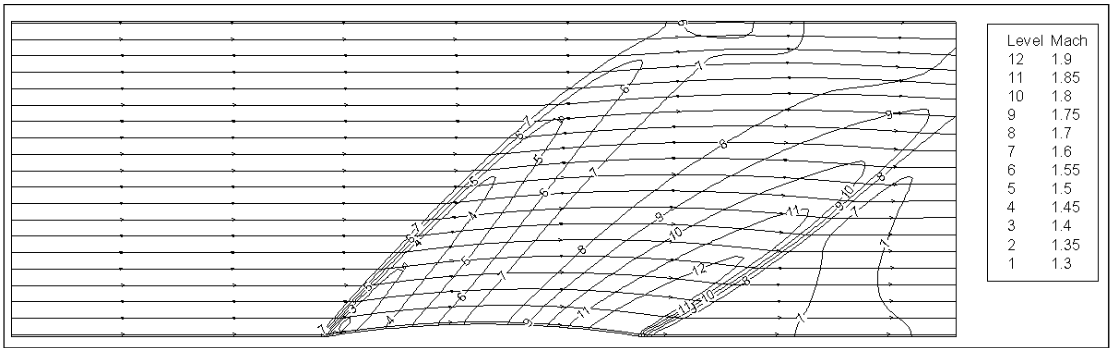

Figure 4.102 Mach No. contours at t = 1.100, 0.04 circular arc

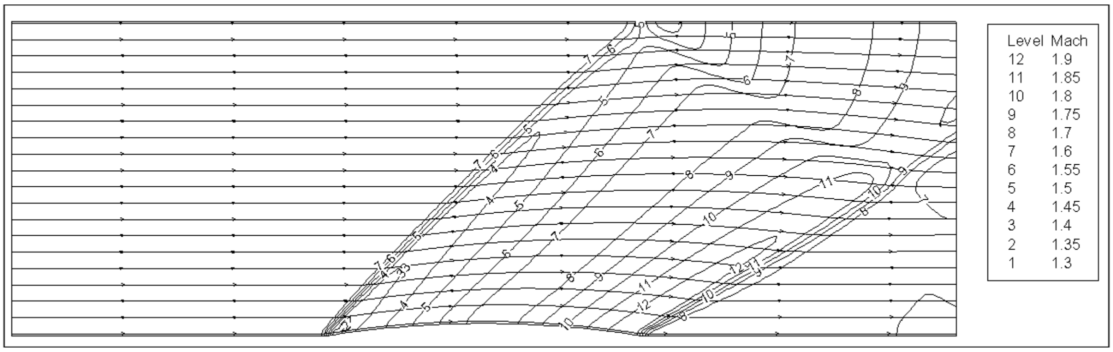

Figure 4.103 Mach No. contours at t = 1.539, 0.04 circular arc

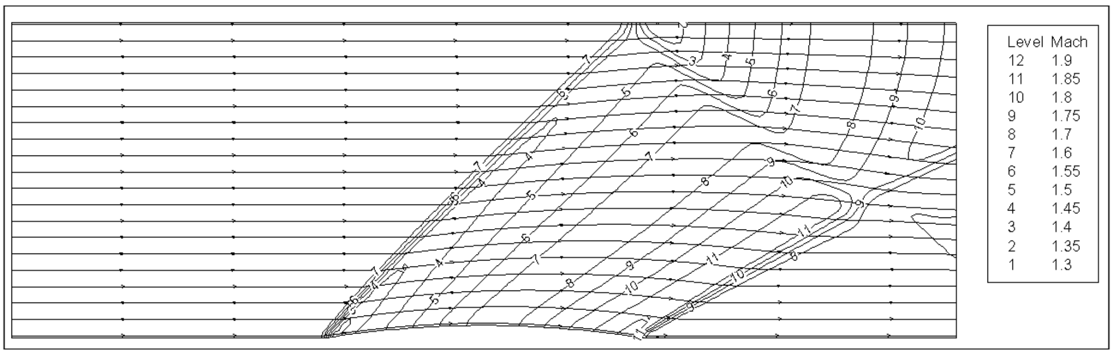

Figure 4.104 Mach No. contours at t = 2.056, 0.04 circular arc

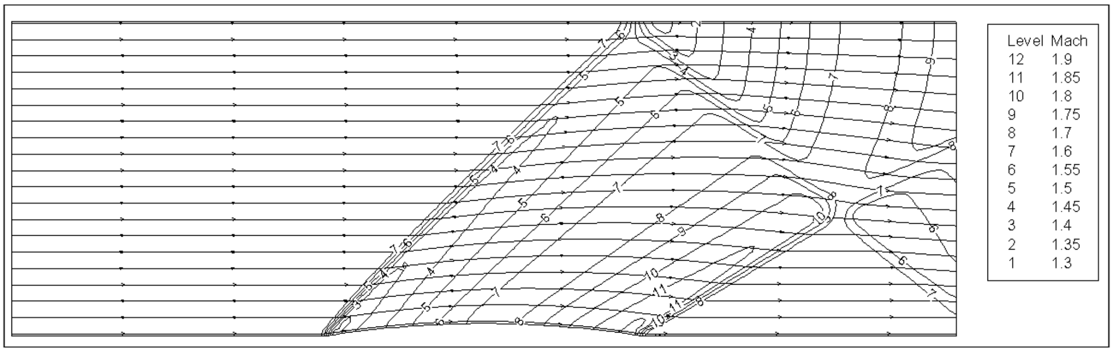

Figure 4.105 Mach No. contours at t = 3.092, 0.04 circular arc

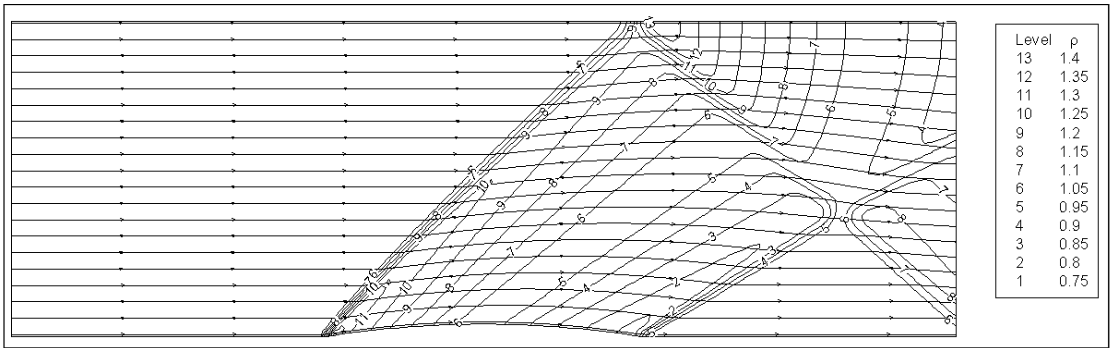

Figure 4.106 Steady State ρ contours, 0.04 circular arc

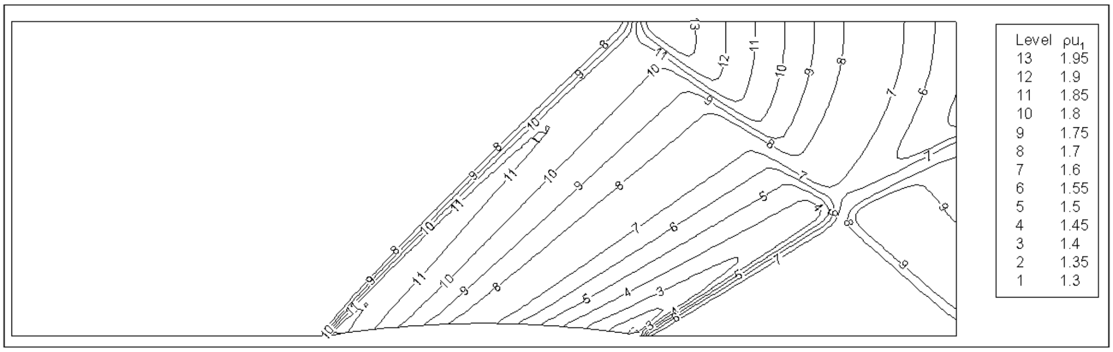

Figure 4.107 Steady State ρu_1 contours, 0.04 circular arc

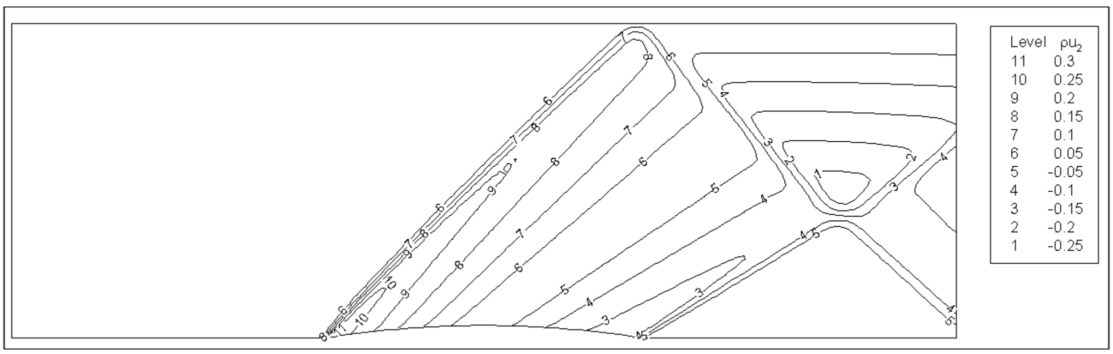

Figure 4.108 Steady State ρu_2 contours, 0.04 circular arc

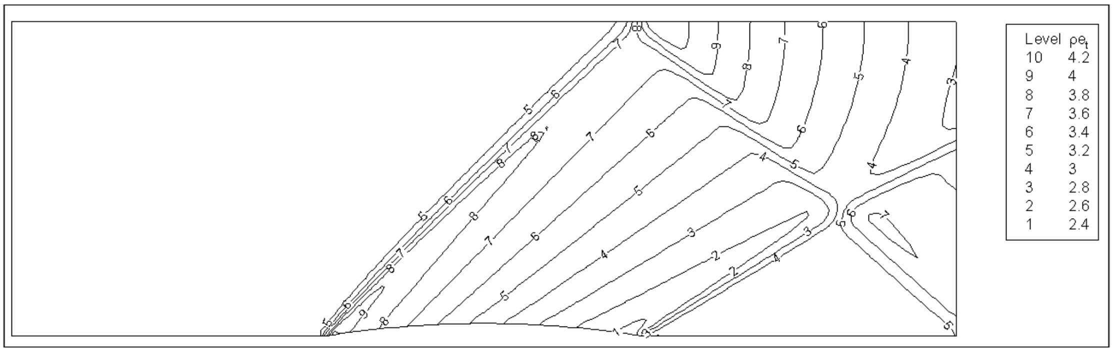

Figure 4.109 Steady State ρe_t contours, 0.04 circular arc

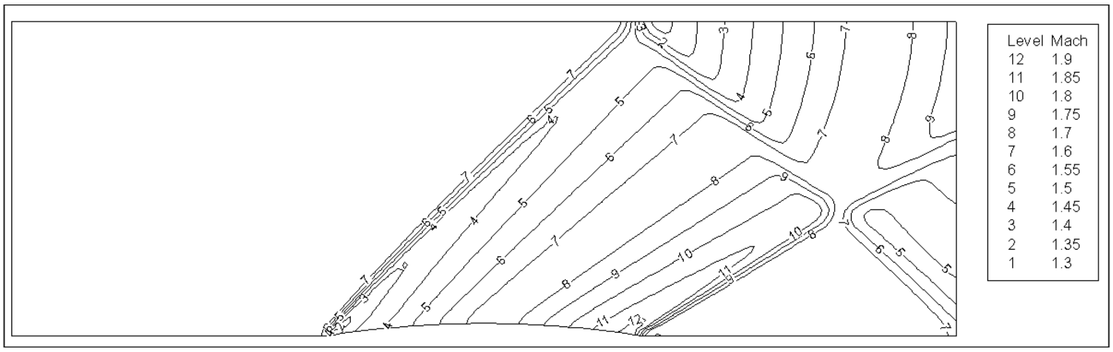

Figure 4.110 Steady State Mach No. contours, 0.04 circular arc

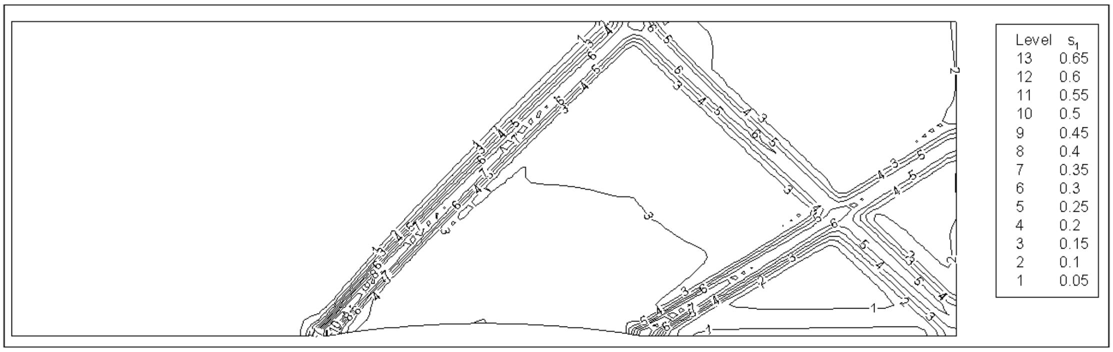

Figure 4.111 Steady State s_1 contours, 0.04 circular arc

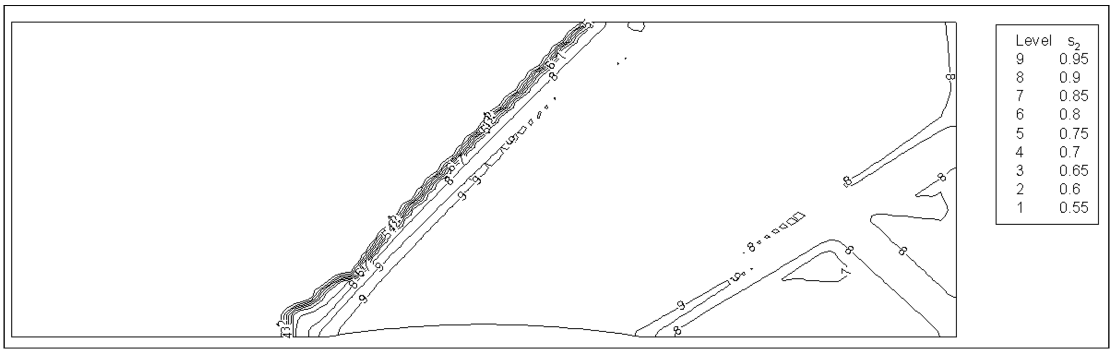

Figure 4.112 Steady State s_2 contours, 0.04 circular arc

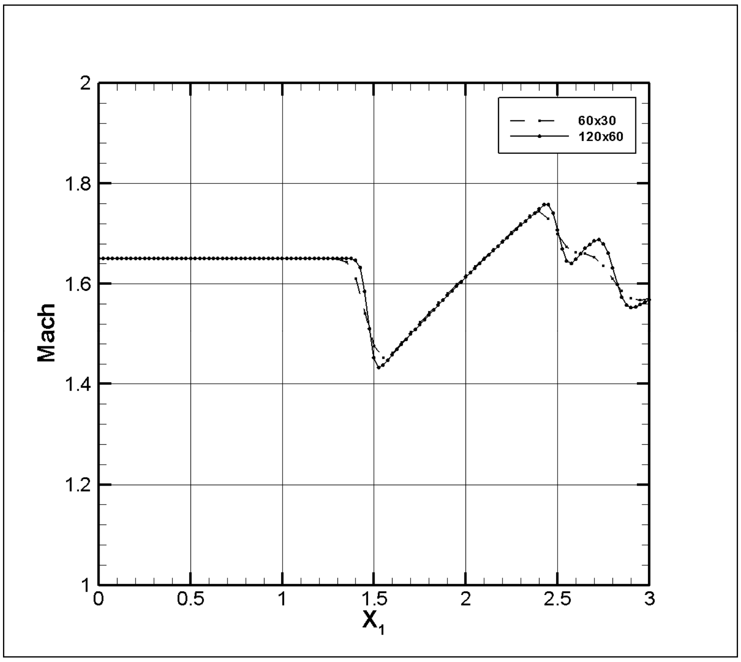

Figure 4.113 Effect of grid refinement at

$x_2 = 0.5$, 0.04 circular arc

4.3.3.2 0.20 Circular Arc

Figure 4.114 shows the configuration of a circular arc that is placed in a supersonic wind tunnel. The circular arc maximum height and the wind tunnel section height have been set to 0.20 and 10.0, respectively. It is believed that separating the upper wall from the lower wall with large span will contribute to the solution and minimize the interferences between the detached shock and the upper wall. The solution domain is discretized using 100x100 bilinear quadrilateral elements clustered in the lateral direction with expansion ratio 1.035. The value of 0.20 has been used for the DCF.

The inlet boundary is supersonic and the same values that have been used in the previous problem are used again:

$$\rho = 1.0, \qquad \rho u_1 = 1.65$$
$$\rho u_2 = 0.0, \qquad \rho e_t = 3.147$$

Same as before, the inviscid wall boundary condition is applied on both the upper and lower boundaries and the exit boundary is left free being supersonic.

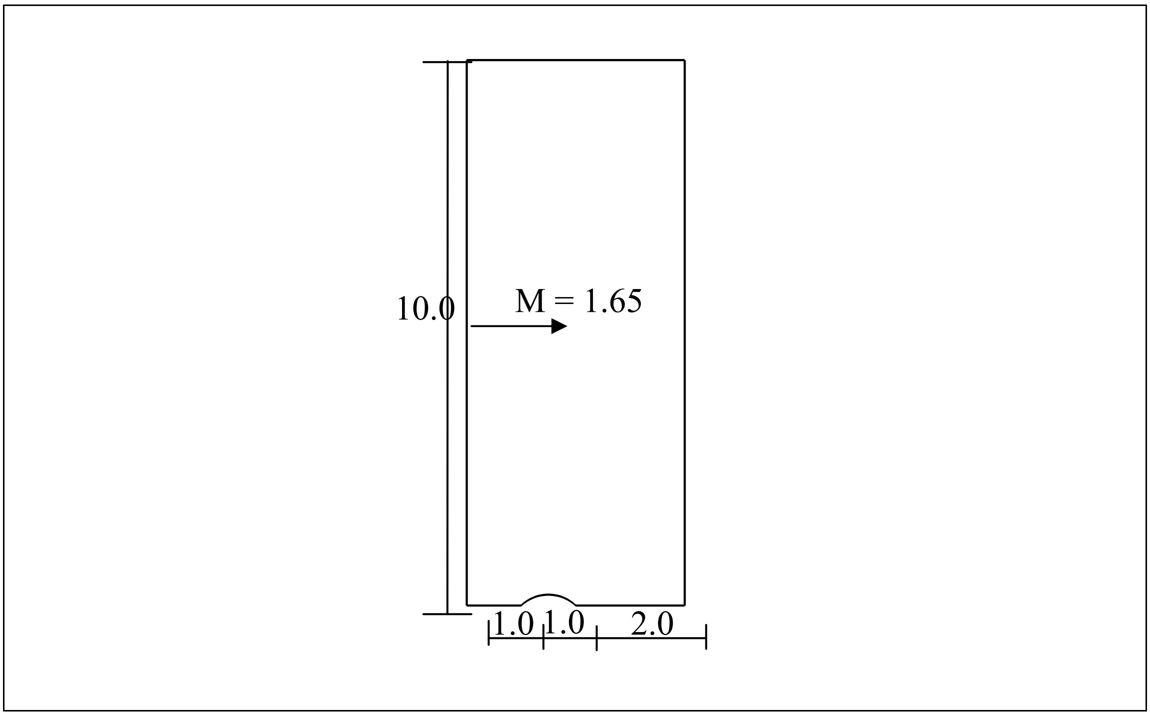

Figure 4.114 0.20 circular arc domain

Figure 4.115 to Figure 4.124 show the transient Mach number contours at various time steps. The steady state solution is obtained at t = 4.328 sec. Figure 4.125, Figure 4.126, Figure 4.127, and Figure 4.128 show the steady state contours of the conservation variables $\rho, \rho u_1, \rho u_2, \rho e_t$, respectively. Figure 4.129 and Figure 4.130 show the steady state Mach number and pressure contours, respectively, while Figure 4.131 and Figure 4.132 plot the s_1 and s_2 contours, respectively, at steady state. Figure 4.133 shows the effect of grid refinement on the solution at $x_2 = 0.6$.

Again, the FDV/MFDV formulation succeeded in recovering the detached shock generated from the relatively large deflection angle encountered at the first point of the circular arc. The maximum deflection angle for attached shockwave can be computed from the oblique shockwave theory. The results is; at Mach 1.65 the maximum angle is 15.8 degrees; while the deflection angle in this test case is 35.5 degrees and this generate the bow (detached) shockwave. In this case both supersonic and subsonic flows exist in the same domain of study and it is evident that FDV method could handle such a situation. Also as observed before, the method approaches the exact solution by increasing the computational grid.

4.4 Subsonic/Transonic Flows

This section is devoted to the validation of the FDV/MFDV formulations in handling the subsonic/transonic flows. The discussion is limited to internal flows only. (It is expected to publish soon more results to demonstrate the FDV/MFDV capabilities to correctly analyze the subsonic/transonic internal flow problems.)

In the next two subsections two problems will be investigated for completely subsonic and subsonic/transonic situations. The complexity of these problems comes from the physics involved in enforcing the boundary conditions.

The first problem is a convergent-divergent nozzle. The second problem is a circular arc that is similar to that of the supersonic flow but now it is solved for subsonic/transonic flow. In both test cases the inviscid wall boundary condition is applied on the upper and lower boundaries, while subsonic inlet and subsonic exit boundary conditions are imposed on the inlet and exit sections, respectively. To allow the inlet Mach number to change, the enforced back pressure should be changed. For each problem, two data sets will be presented each data set constitutes a different back pressure.

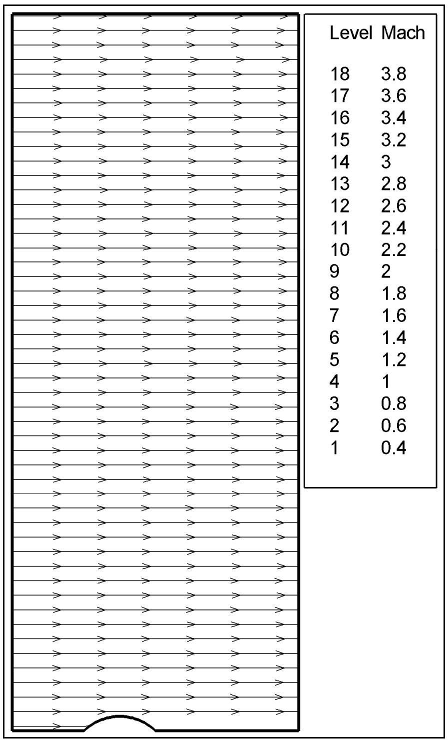

Figure 4.115 Mach No. contours at t = 0.000, 0.20 circular arc

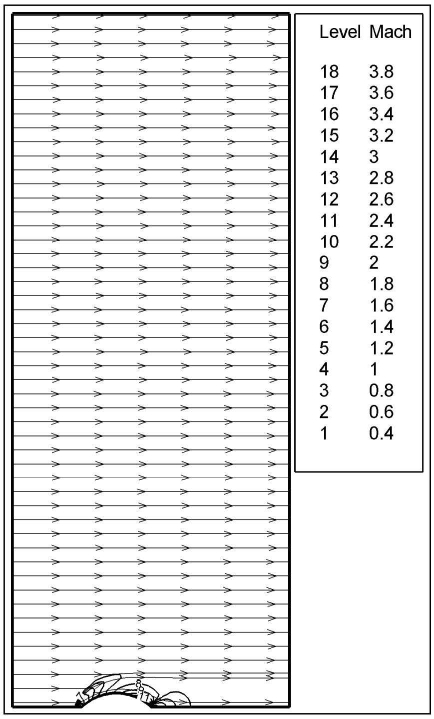

Figure 4.116 Mach No. contours at t = 0.221, 0.20 circular arc

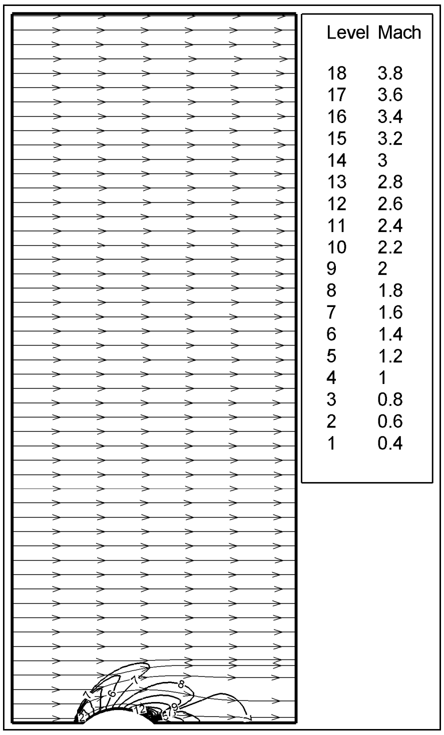

Figure 4.117 Mach No. contours at t = 0.533, 0.20 circular arc

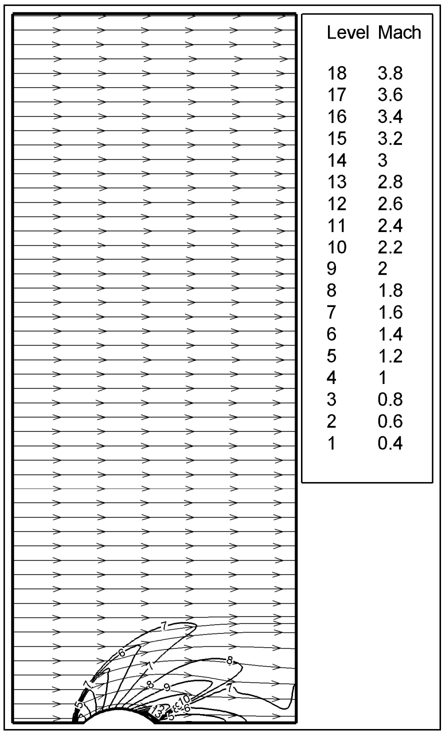

Figure 4.118 Mach No. contours at t = 1.043, 0.20 circular arc

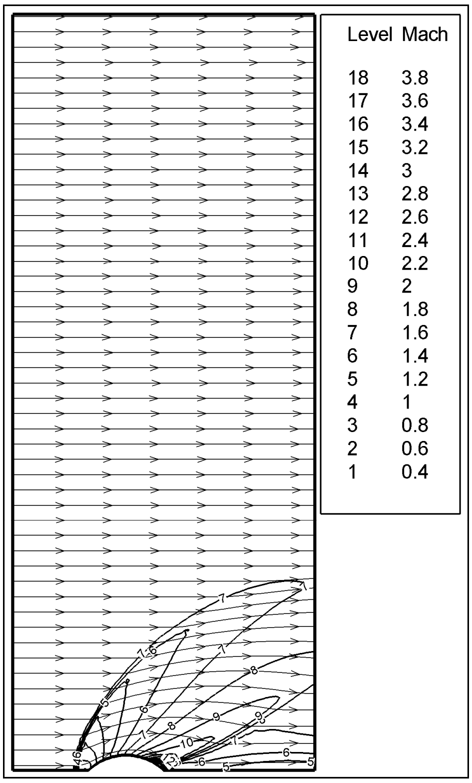

Figure 4.119 Mach No. contours at t = 2.081, 0.20 circular arc

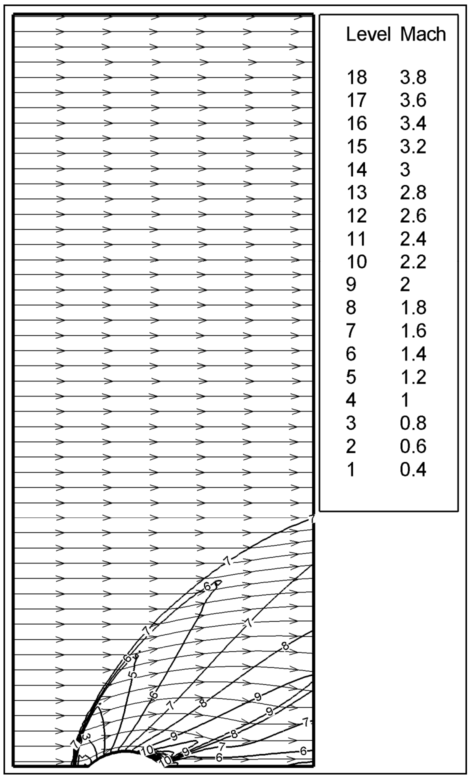

Figure 4.120 Mach No. contours at t = 3.025, 0.20 circular arc

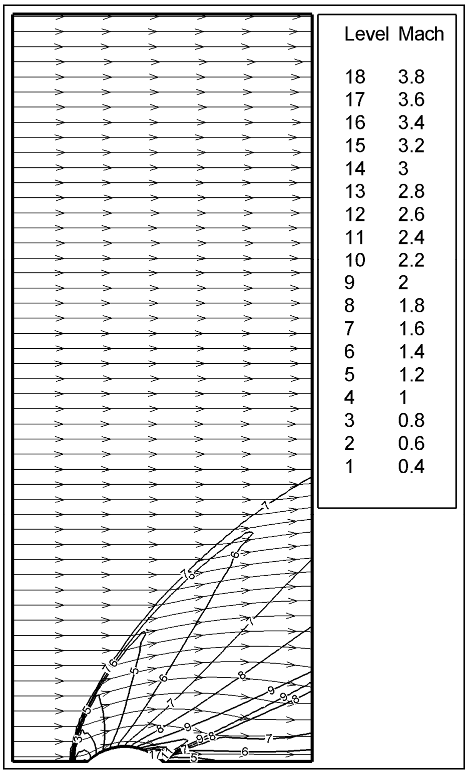

Figure 4.121 Mach No. contours at t = 4.078, 0.20 circular arc

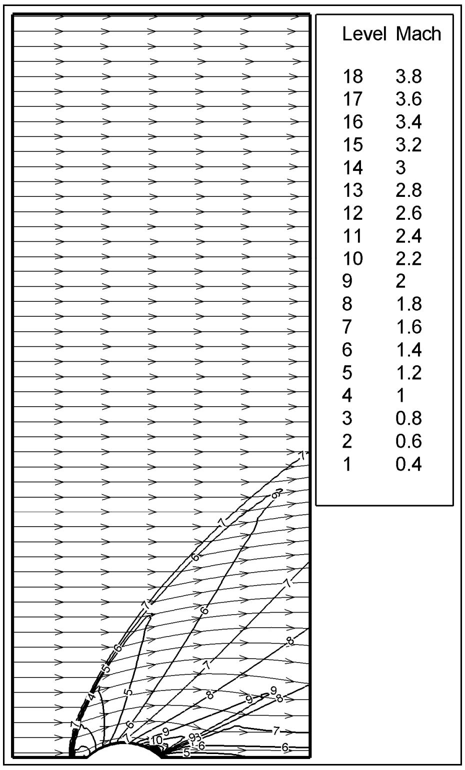

Figure 4.122 Mach No. contours at t = 5.029, 0.20 circular arc

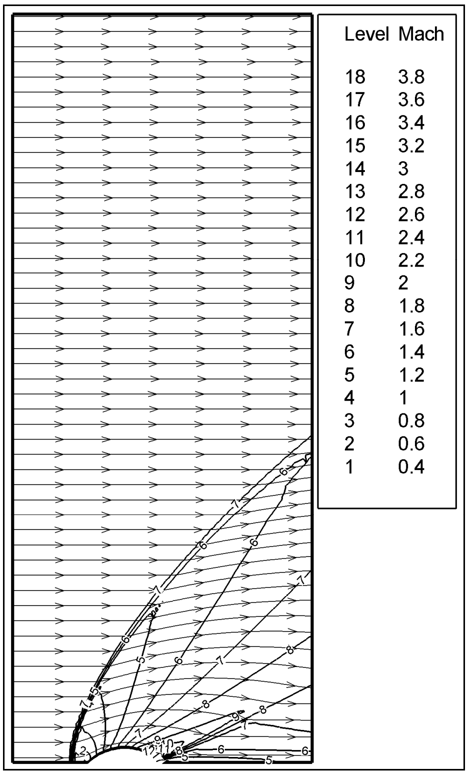

Figure 4.123 Mach No. contours at t = 6.086, 0.20 circular arc

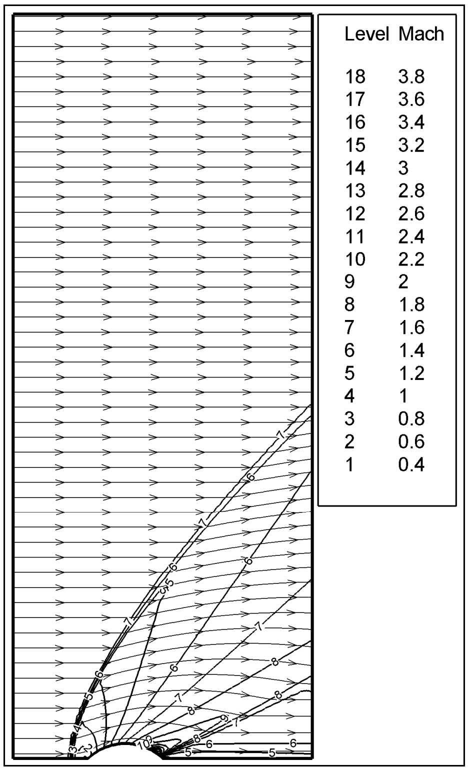

Figure 4.124 Mach No. contours at t = 10.003, 0.20 circular arc

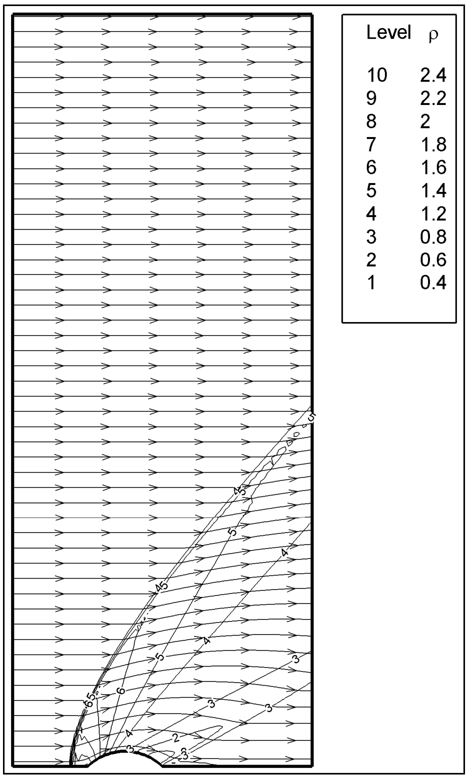

Figure 4.125 Steady State ρ contours, 0.20 circular arc

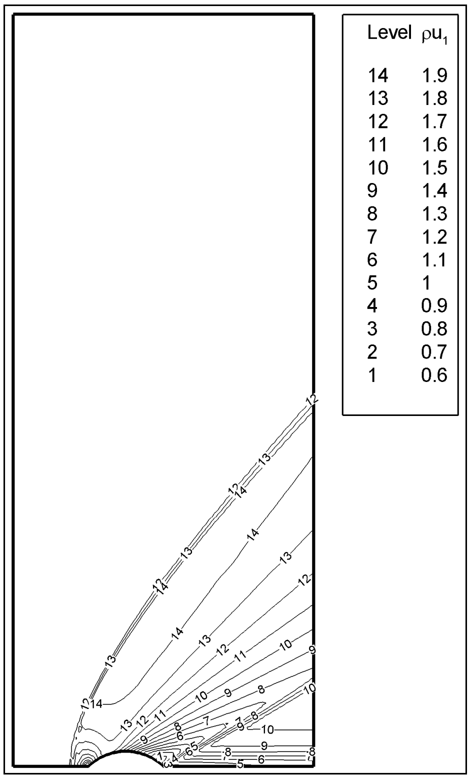

Figure 4.126 Steady State ρu_1 contours, 0.20 circular arc

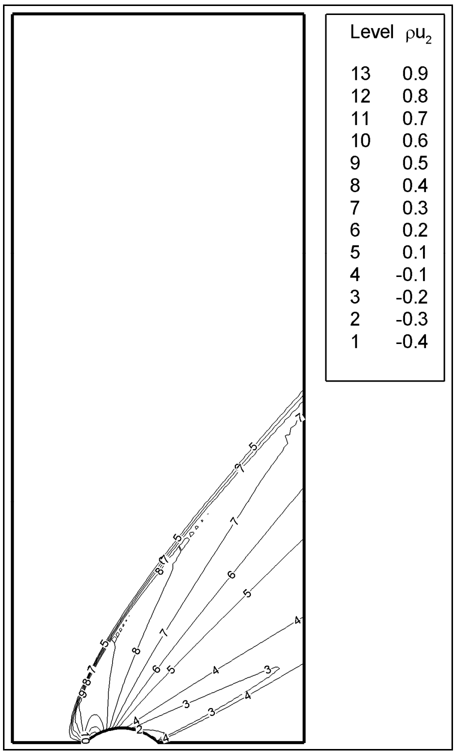

Figure 4.127 Steady State ρu_2 contours, 0.20 circular arc

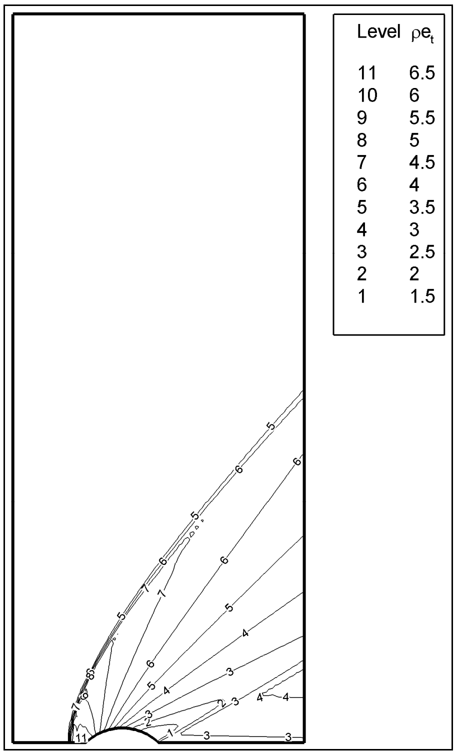

Figure 4.128 Steady State ρe_t contours, 0.20 circular arc

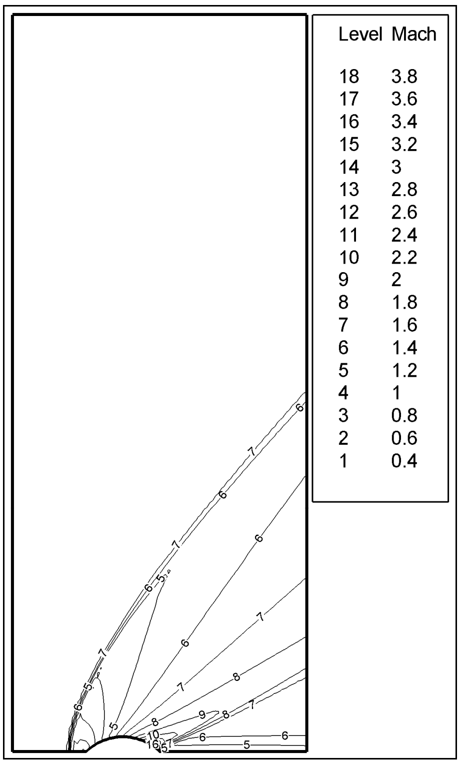

Figure 4.129 Steady State Mach No. contours, 0.20 circular arc

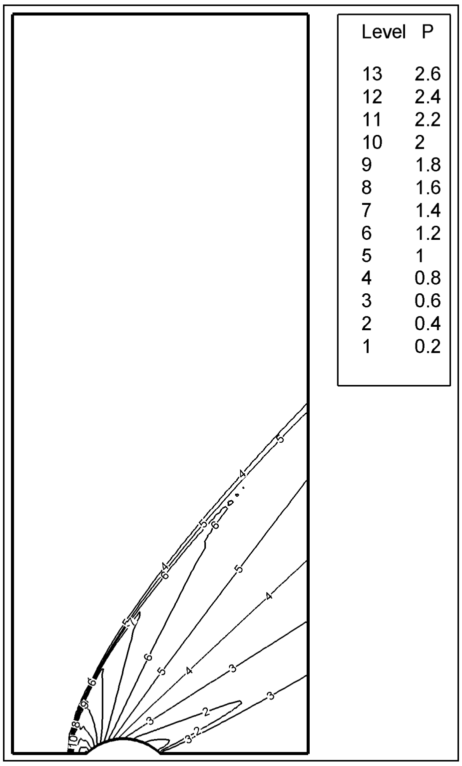

Figure 4.130 Steady State Pressure contours, 0.20 circular arc

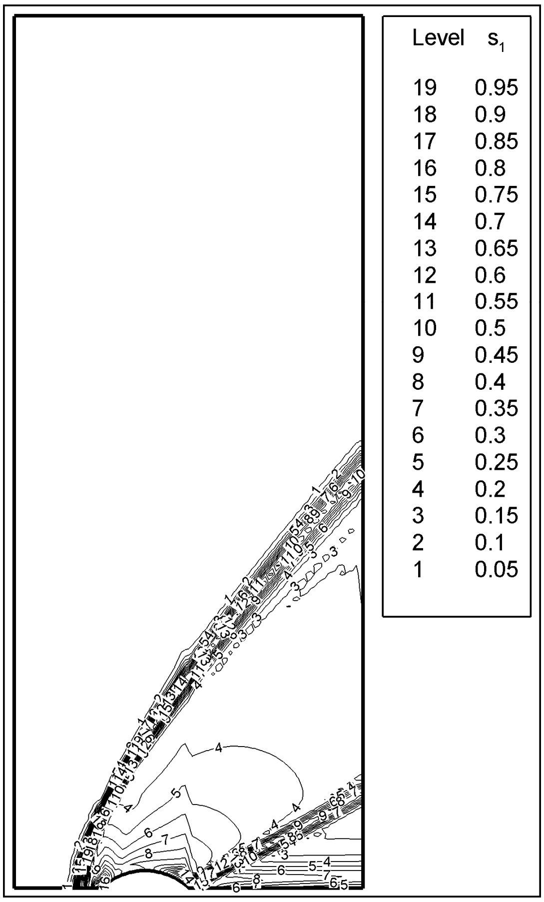

Figure 4.131 Steady State s_1 contours, 0.20 circular arc

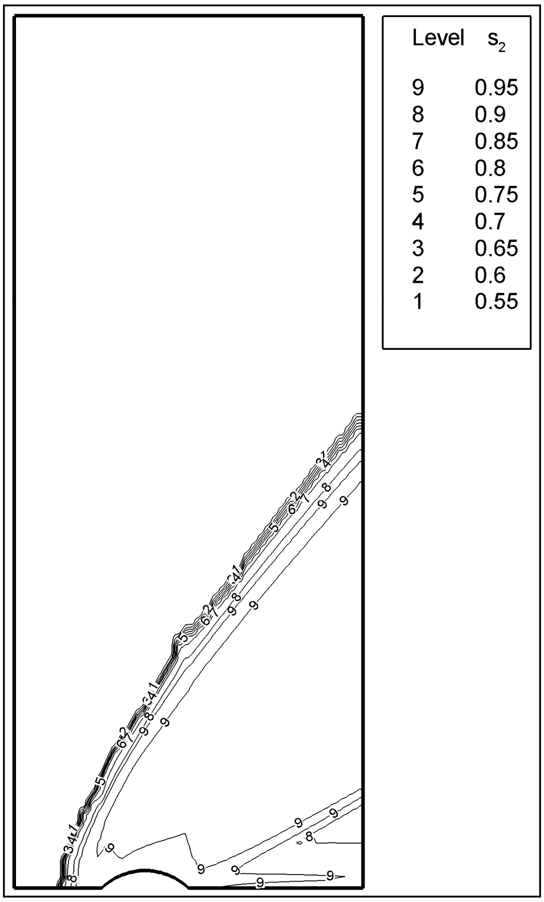

Figure 4.132 Steady State s_2 contours, 0.20 circular arc

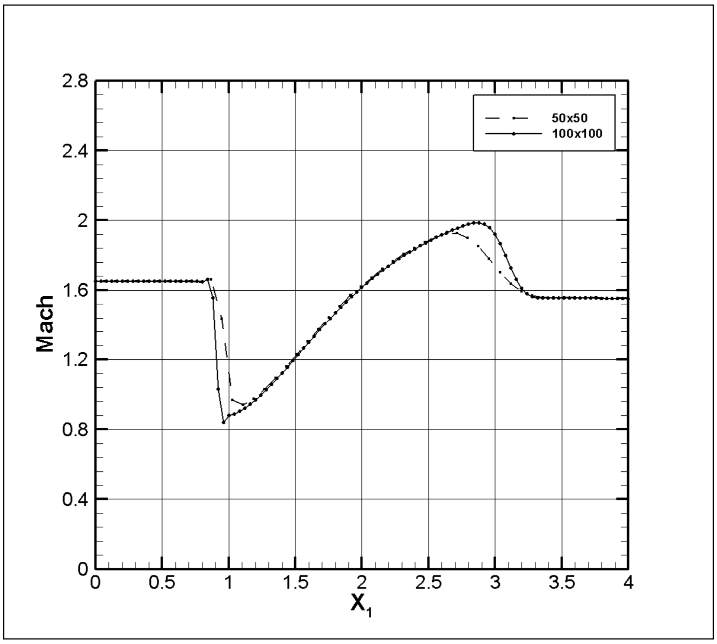

Figure 4.133 Effect of grid refinement at $x_2 = 0.6$, 0.20 circular arc

4.4.1 Convergent-Divergent Nozzle

Figure 4.134 shows the configuration of a convergent-divergent nozzle. The nozzle convergent and divergent parts shapes are assumed to vary cubically with x_1. Letting $h_1 = 0.20$, $h_2 = 0.10$, and $h_3 = 0.15$ the upper profile will have the equation:

$$x_2 = \begin{cases} 1.6x_1^3 - 1.2x_1^2 + 0.2 & 0.0 \le x_1 \le 0.5 \\ -0.8x_1^3 + 1.8x_1^2 - 1.2x_1 + 0.35 & 0.5 < x_1 \le 1.0 \end{cases}$$

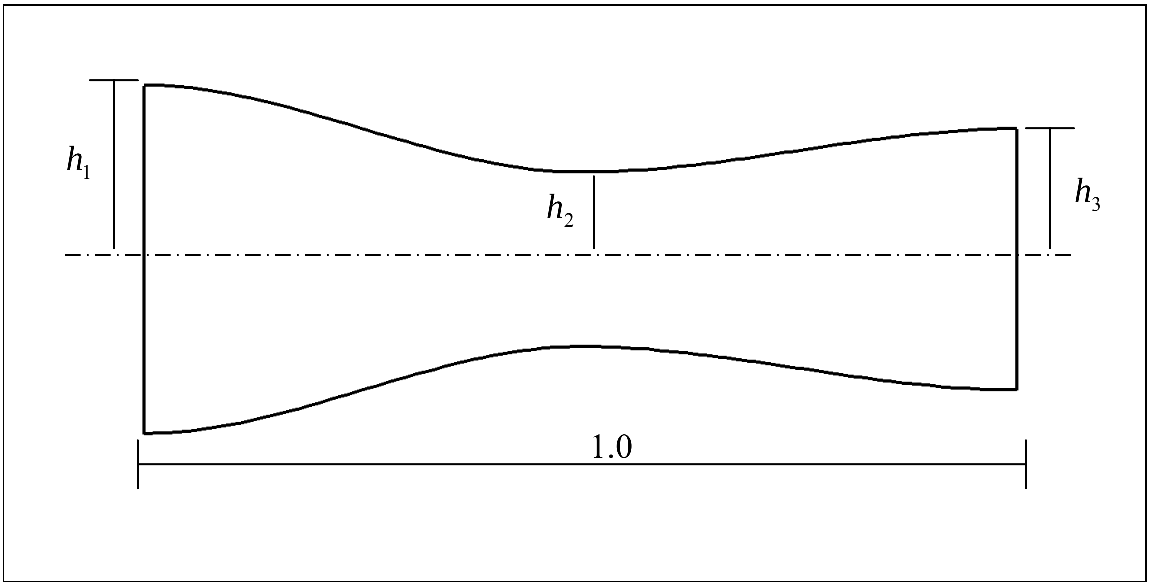

Figure 4.134 Convergent-divergent nozzle domain

The lower profile is assumed to be exactly the same as the upper but inversed in the x_2 direction (having negative sign). The stagnation pressure (P_0) and stagnation density (ρ_0) at the inlet are fixed to the values:

$$P_0 = 3.39$$
$$\rho_0 = 1.13$$

The solution domain is discretized using 90x30 uniform bilinear quadrilateral elements. The value of 0.20 is used for the DCF. Two data sets are obtained for two different back pressures. The first case is subsonic/transonic case where a shockwave is standing in the convergent part, while the second one constitutes to a completely subsonic flow.

4.4.1.1 Subsonic/Transonic Flow, $P_b = 2.48$

In order to encounter a shockwave in the convergent part the back pressure should be relatively low. So the value of 2.48 has been used. Figure 4.135, Figure 4.136, Figure 4.137, and Figure 4.138 show the contours of the conservation variables $\rho, \rho u_1, \rho u_2, \rho e_t$, respectively. Figure 4.139, Figure 4.140, and Figure 4.141 show the Mach number contours, pressure contours, and streamlines, respectively, while Figure 4.142 and Figure 4.143 plot the s_1 and s_2 contours, respectively. Figure 4.144 shows the comparison with the exact quasi one-dimensional solution.

It is evident that FDV/MFDV formulation succeeded in recovering the shock generated in the convergent part. In this case both subsonic and transonic flows exist in the same domain of study and it is clear that FDV method could handle such a situation. It is noticeable that the solution is symmetric about the center line. This is due to the symmetry of the nozzle shape. Figure 4.137 is of special importance because it indicates that the solution is symmetric with zero u_2 on the centerline. As shown in Figure 4.144, the solution is comparable to the quasi one dimensional solution. The deviation is due to the 2-D nature of the generated shockwave.

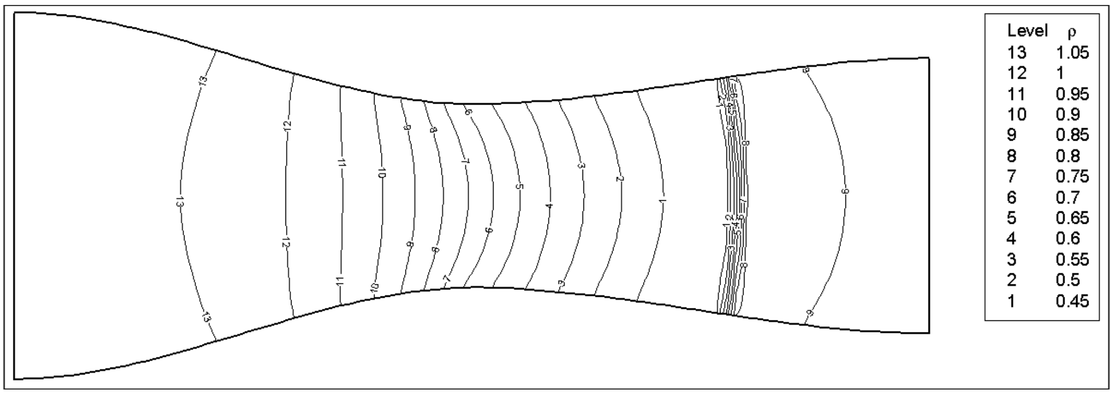

Figure 4.135 Steady state ρ contours, Nozzle, $P_b = 2.48$

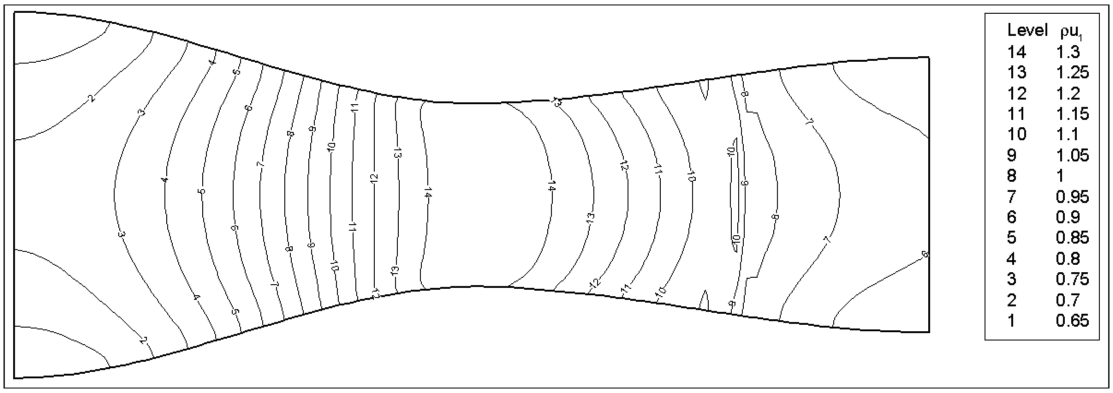

Figure 4.136 Steady state ρu_1 contours, Nozzle, $P_b = 2.48$

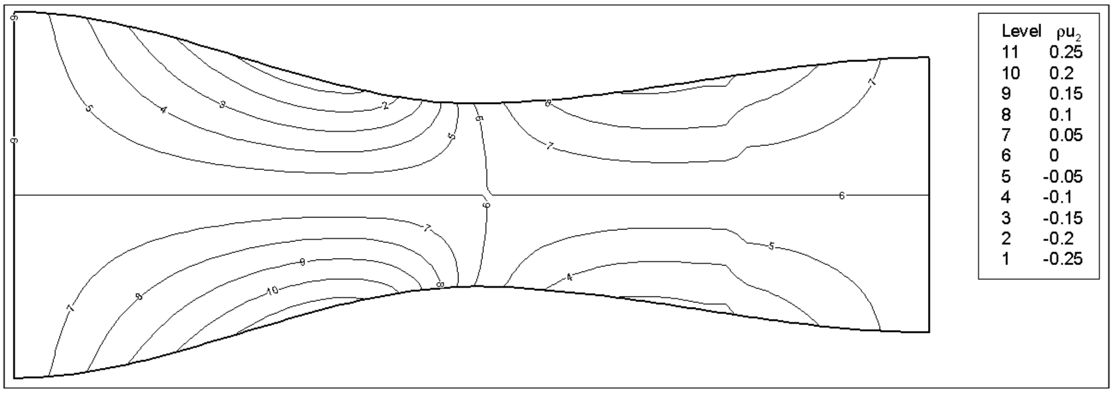

Figure 4.137 Steady state ρu_2 contours, Nozzle, $P_b = 2.48$

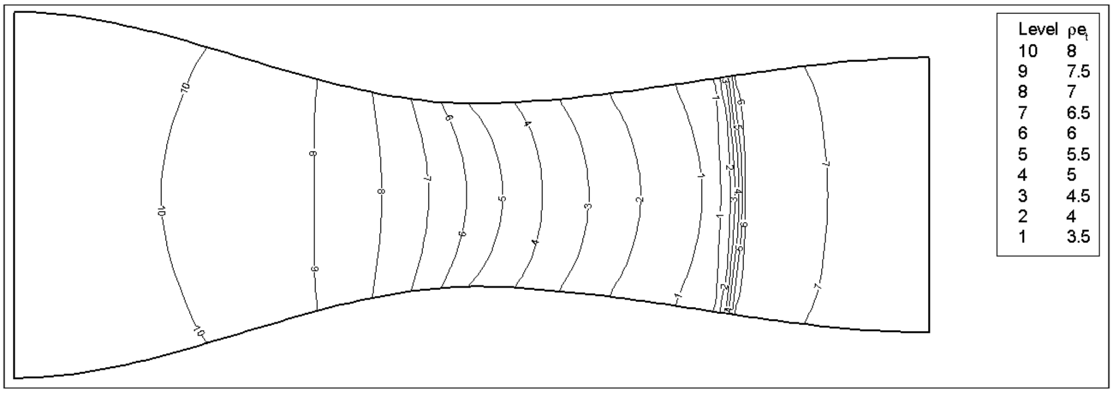

Figure 4.138 Steady state ρe_t contours, Nozzle, $P_b = 2.48$

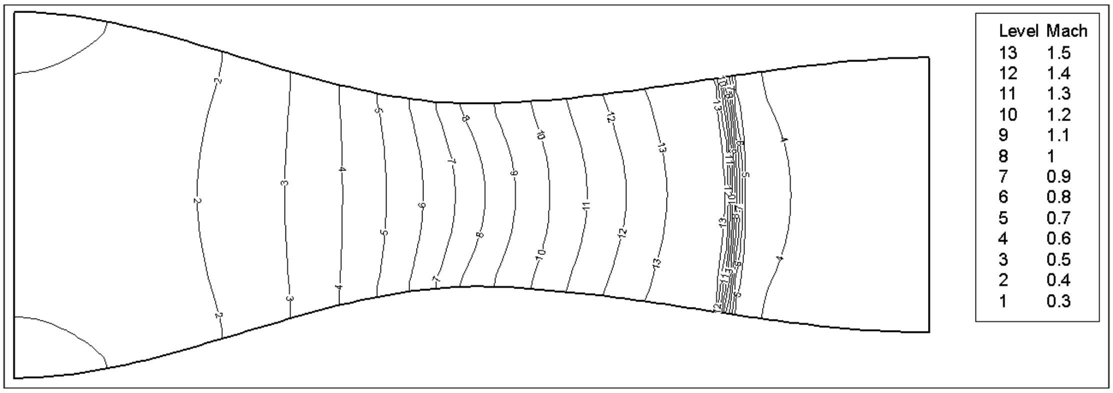

Figure 4.139 Steady state Mach No. contours, Nozzle, $P_b = 2.48$

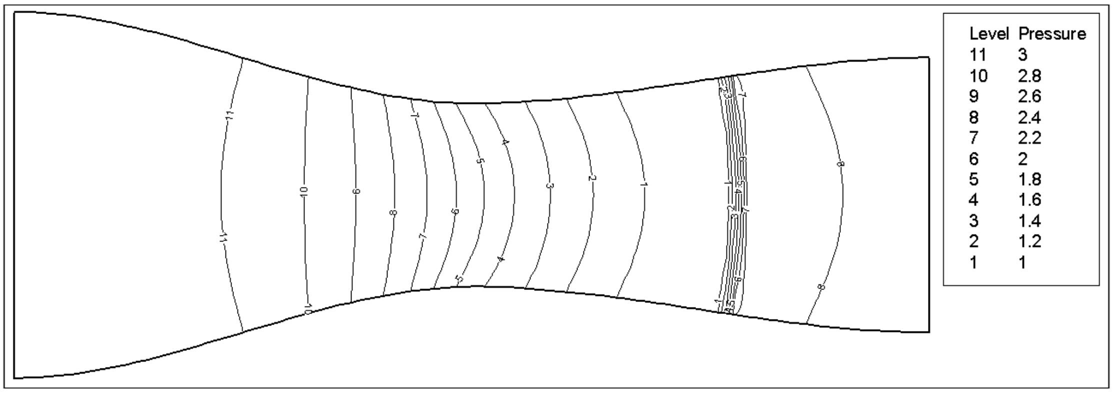

Figure 4.140 Steady state Pressure contours, Nozzle, $P_b = 2.48$

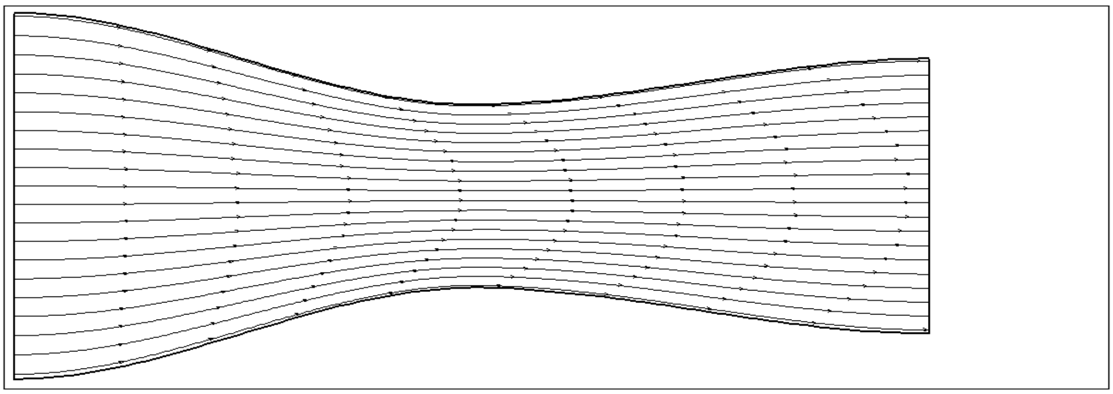

Figure 4.141 Steady state Streamlines, Nozzle, $P_b = 2.48$

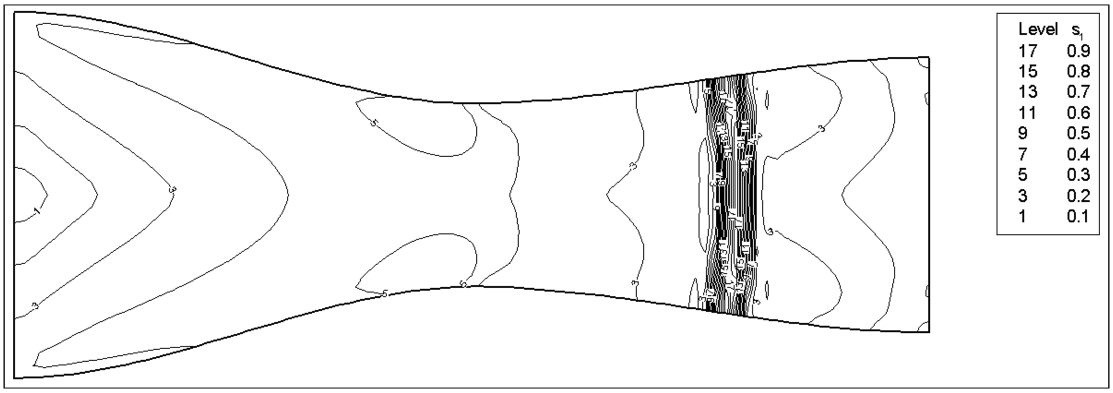

Figure 4.142 Steady state s_1 contours, Nozzle, $P_b = 2.48$

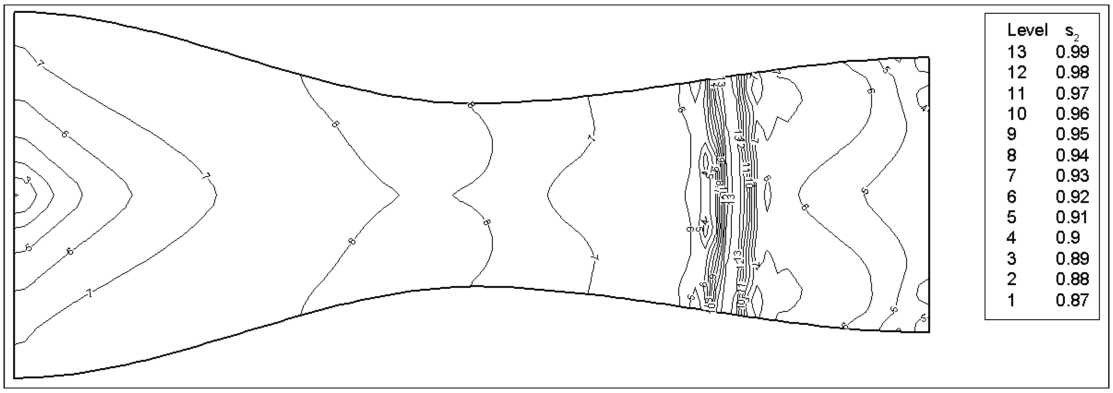

Figure 4.143 Steady state s_2 contours, Nozzle, $P_b = 2.48$

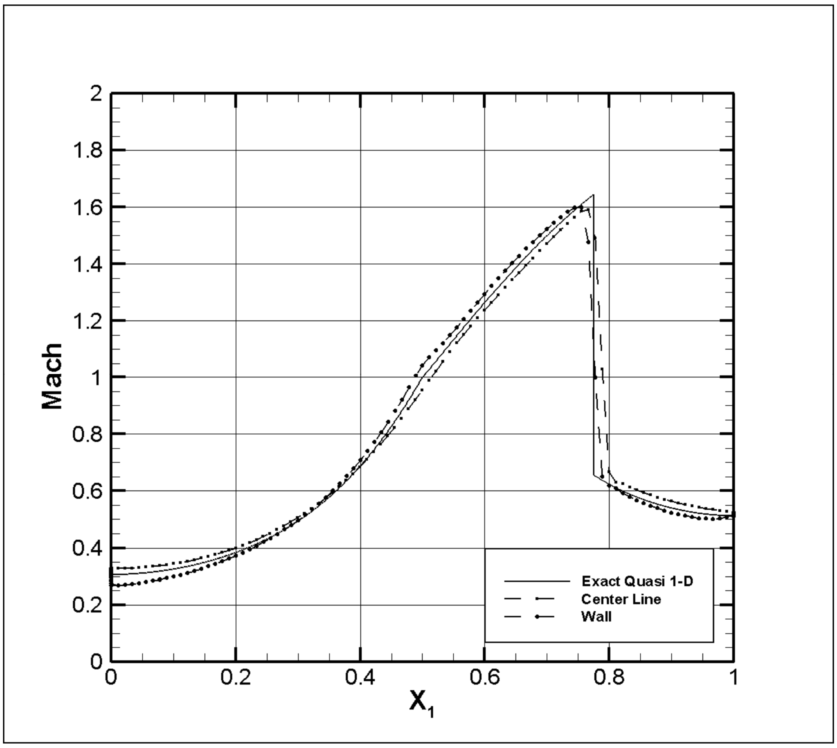

Figure 4.144 Comparison with exact quasi 1-D solution, Nozzle, $P_b = 2.48$

4.4.1.2 Completely Subsonic flow, $P_b = 3.27$

In order to have a completely subsonic flow thought out the nozzle, the back pressure should be relatively high. So the value of 3.27 has been used. Figure 4.145, Figure 4.146, Figure 4.147, and Figure 4.148 show the contours of the conservation variables $\rho, \rho u_1, \rho u_2, \rho e_t$, respectively. Figure 4.149, Figure 4.150, and Figure 4.151 show the Mach number contours, pressure contours, and streamlines, respectively, while Figure 4.152 and Figure 4.153 plot the s_1 and s_2 contours, respectively. Figure 4.154 shows the comparison with the exact quasi one-dimensional solution.

It is evident that FDV/MFDV formulation succeeded in handling the completely subsonic flow with relatively low Mach number. Again the solution symmetry is obtained. Figure 4.147 is of special importance because it indicates that the solution is symmetric with zero u_2 on the centerline. A shown in Figure 4.154, the solution is comparable to the quasi one dimensional solution. The deviation is due to the 2-D nature of the solution.

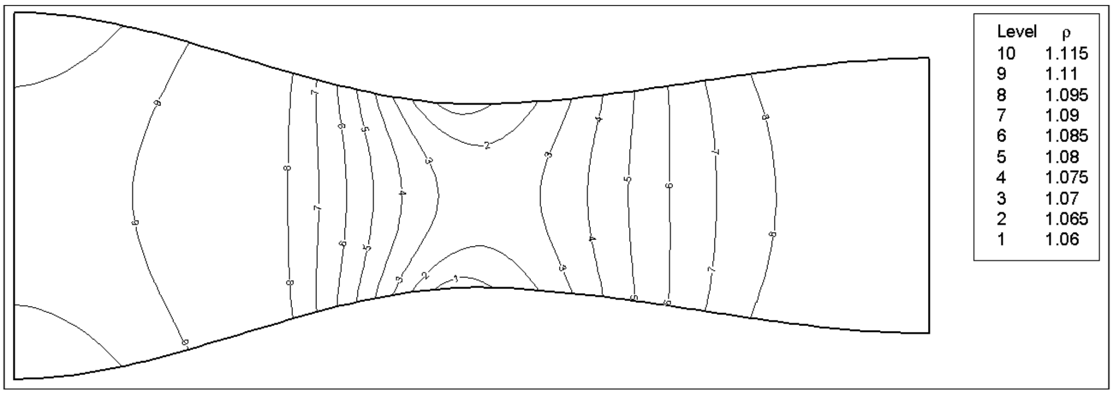

Figure 4.145 Steady state ρ contours, Nozzle, $P_b = 3.27$

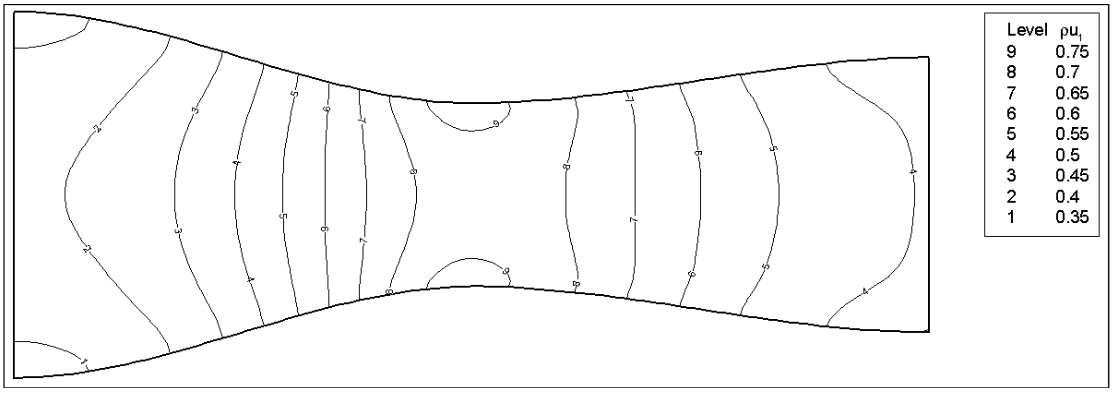

Figure 4.146 Steady state ρu_1 contours, Nozzle, $P_b = 3.27$

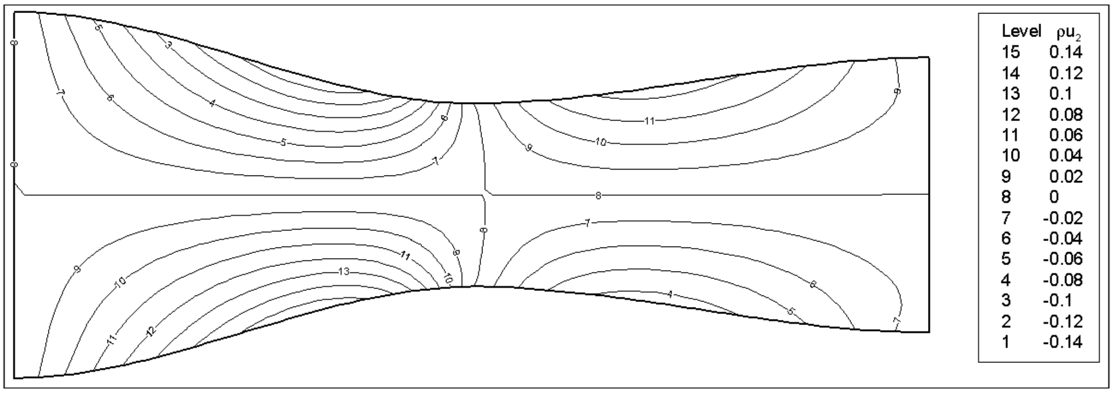

Figure 4.147 Steady state ρu_2 contours, Nozzle, $P_b = 3.27$

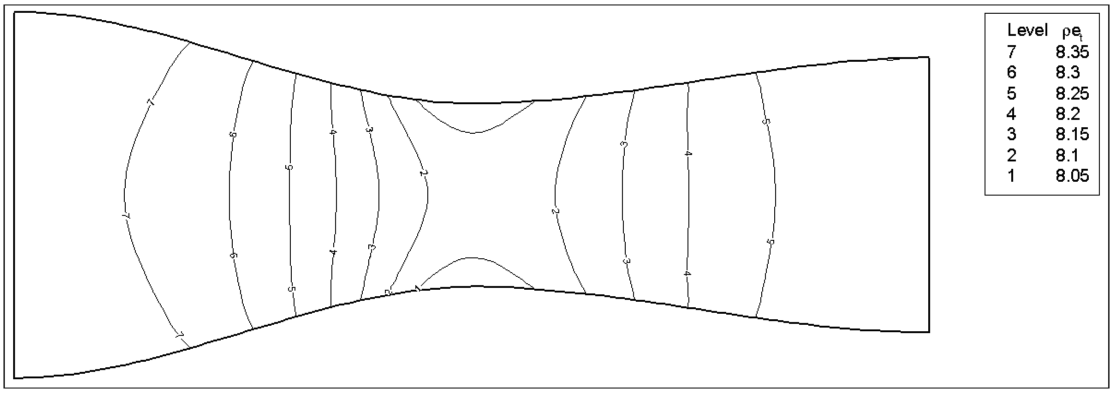

Figure 4.148 Steady state ρe_t contours, Nozzle, $P_b = 3.27$

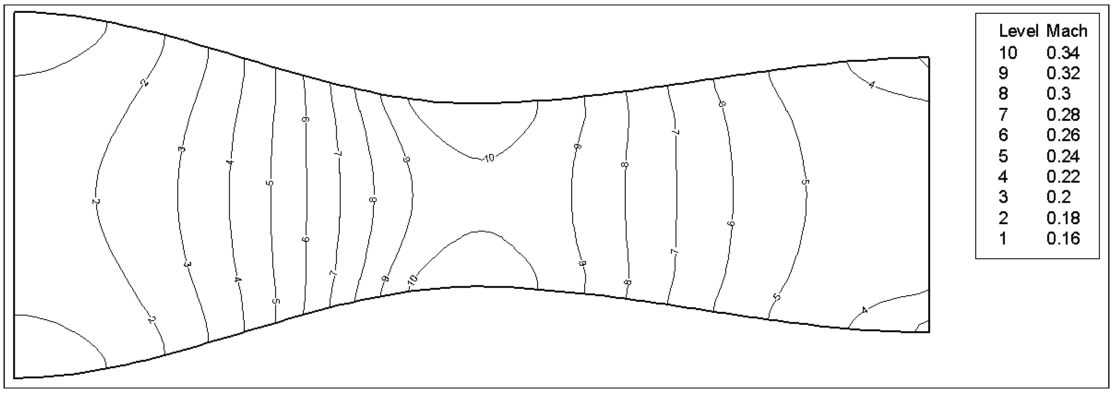

Figure 4.149 Steady state Mach No. contours, Nozzle, $P_b = 3.27$

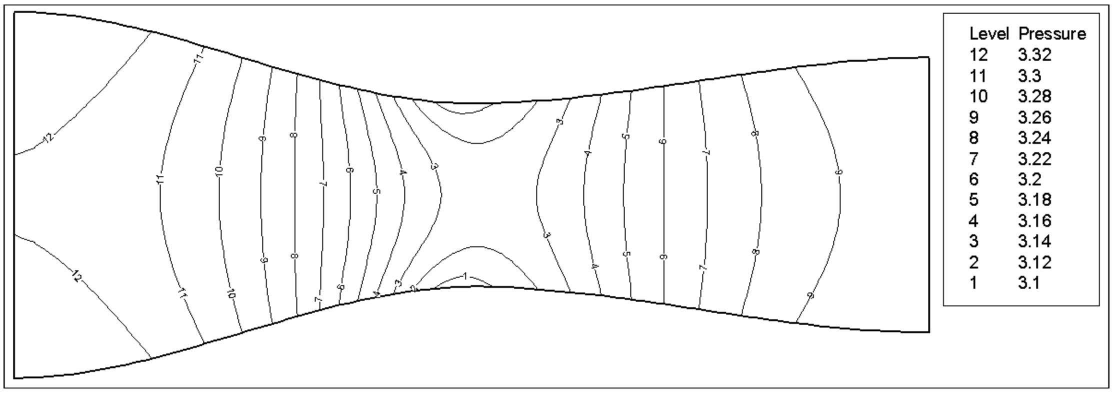

Figure 4.150 Steady state Pressure contours, Nozzle, $P_b = 3.27$

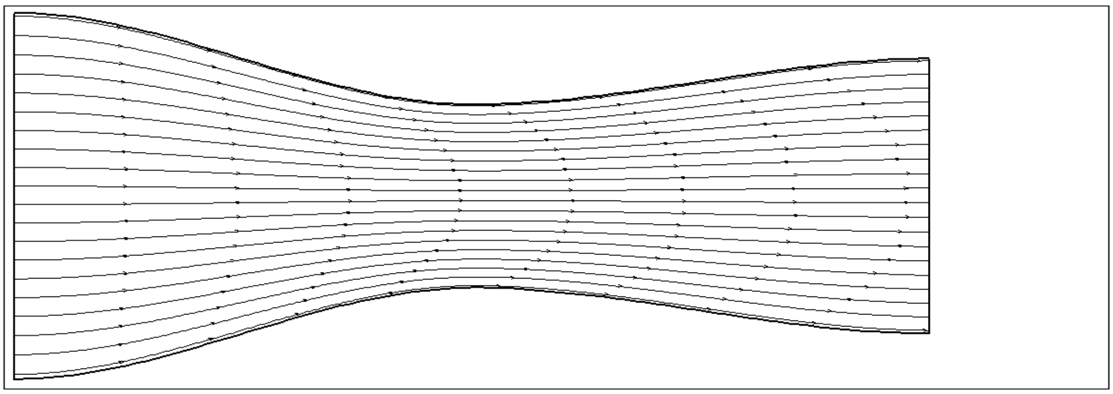

Figure 4.151 Steady state Streamlines, Nozzle, $P_b = 3.27$

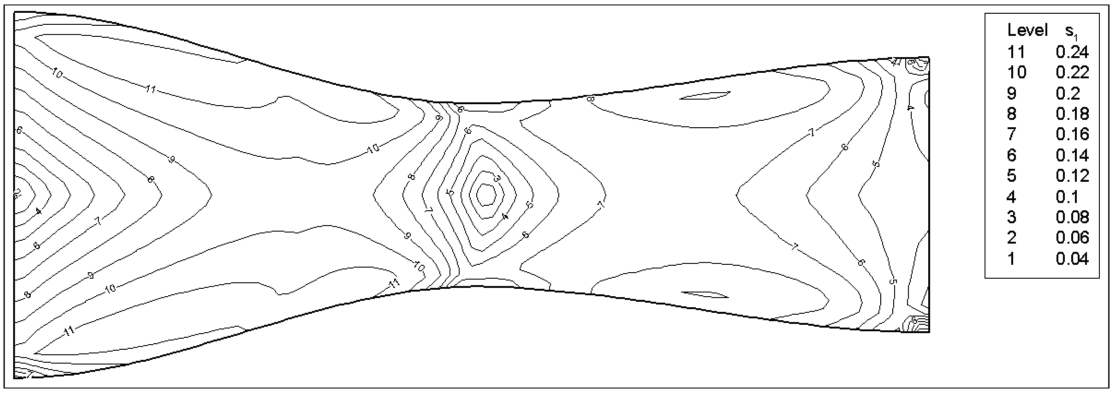

Figure 4.152 Steady state s_1 contours, Nozzle, $P_b = 3.27$

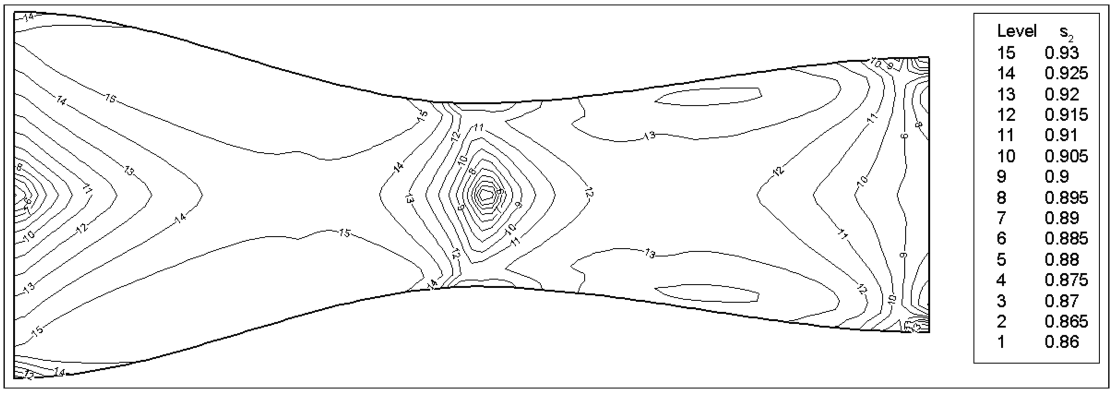

Figure 4.153 Steady state s_2 contours, Nozzle, $P_b = 3.27$

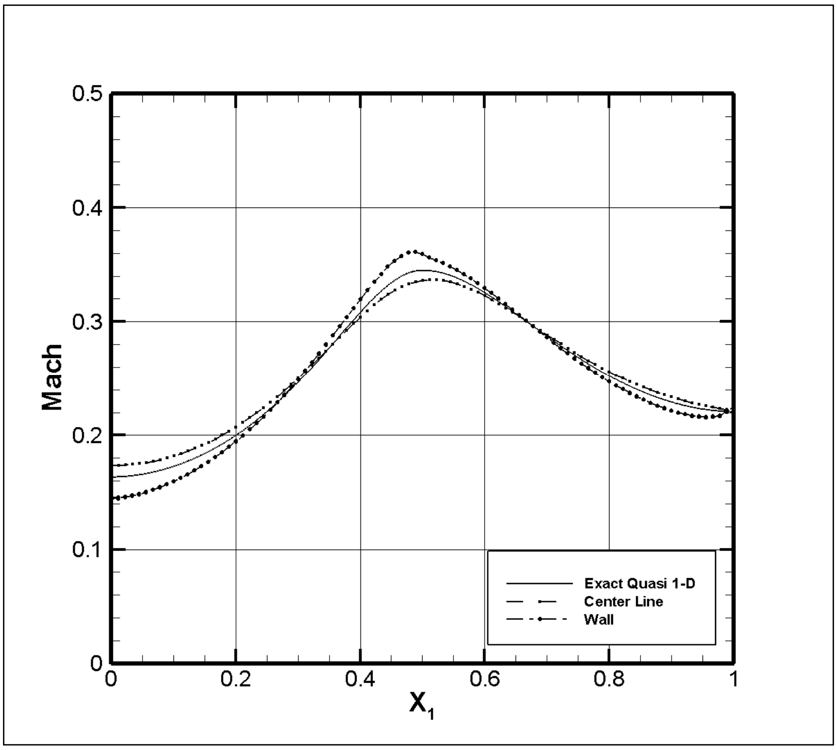

Figure 4.154 Comparison with exact quasi 1-D solution,

Nozzle, $P_b = 3.27$

4.4.2 Circular Arc Bump in Subsonic Wind Tunnel

Figure 4.155 shows the configuration of a circular arc placed on the lower wall of a subsonic wind tunnel. The bump maximum height has been set to 0.10. The stagnation pressure (P_0) and stagnation density (ρ_0) at the inlet are fixed to:

$$P_0 = 3.39$$
$$\rho_0 = 1.13$$

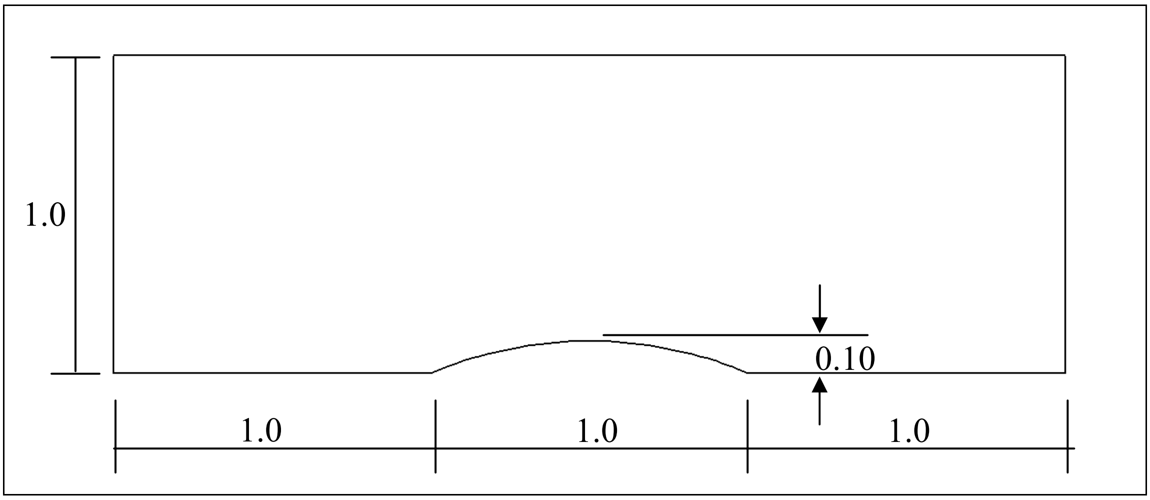

Figure 4.155 0.10 Circular arc domain

The solution domain is discretized using 90x30 bilinear quadrilateral elements with expansion ratio of 1.03 in the lateral direction. Forty elements have been placed on the bump surface. The value of 0.20 has been used for the DCF. Two data sets have been obtained for two different back pressures. The first case is subsonic/transonic case where a supersonic pocket has been generated over the bump surface, while the second one constitutes a completely subsonic flow.

4.4.2.1 Subsonic/Transonic Flow, $P_b = 2.48$

In order to have a supersonic pocket, the inlet Mach number should be relatively high and this asks for low back pressure. So the value of 2.48 has been used. Figure 4.156, Figure 4.157, Figure 4.158, and Figure 4.159 show the contours of the conservation variables $\rho, \rho u_1, \rho u_2, \rho e_t$, respectively. Figure

4.160 plots the Mach number contours, while Figure 4.161 and Figure 4.162 plot s_1 and s_2 contours, respectively.

From the results, it can be concluded that FDV/MFDV formulation succeeded in resembling the partial shock generated in the supersonic pocket. In this case both subsonic and transonic flows exist in the same domain of study and it is evident that FDV method could handle such a situation. This problem is harder than the nozzle case because the shock is not blocking the entire span. This solution matches the results obtained in [57] very well.

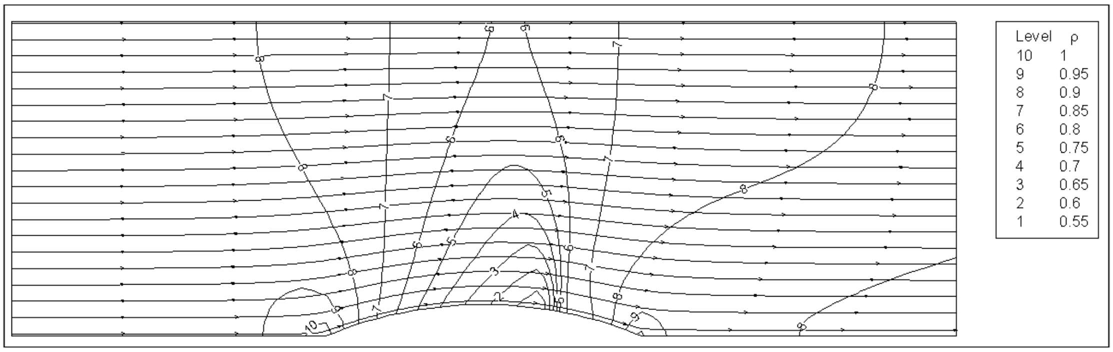

Figure 4.156 Steady state ρ contours, transonic circular arc

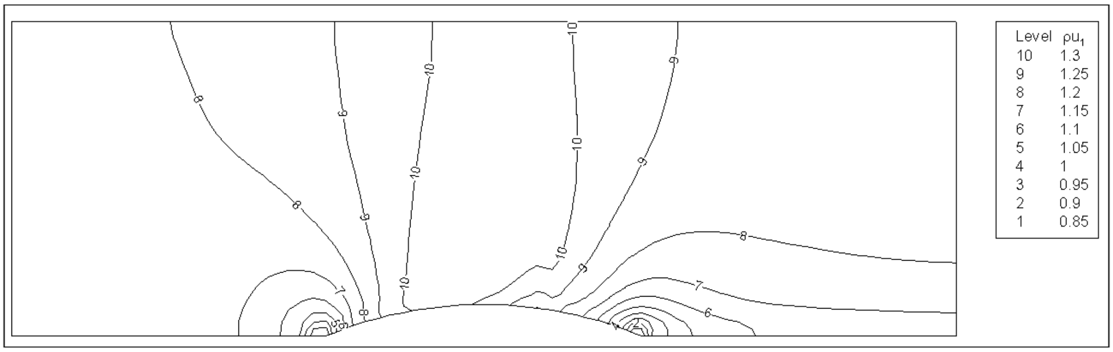

Figure 4.157 Steady state ρu_1 contours, transonic circular arc

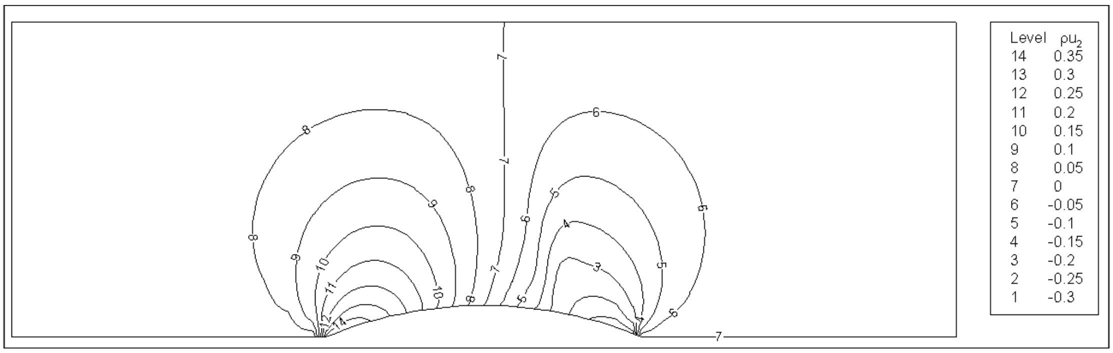

Figure 4.158 Steady state ρu_2 contours, transonic circular arc

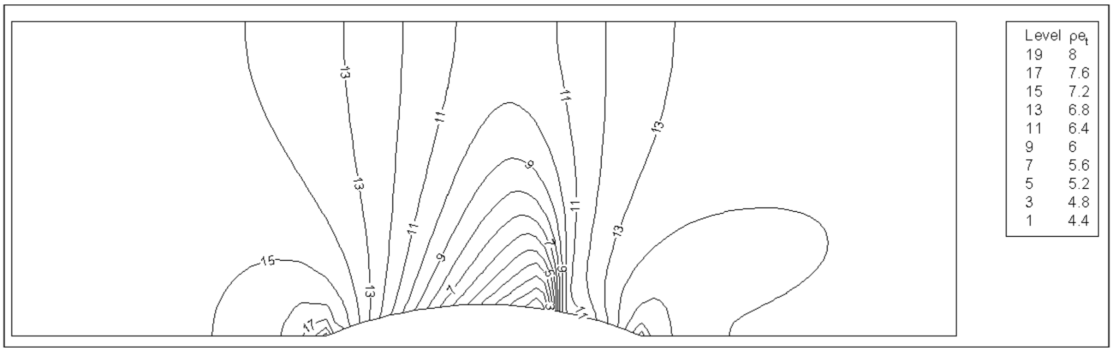

Figure 4.159 Steady state ρe_t contours, transonic circular arc

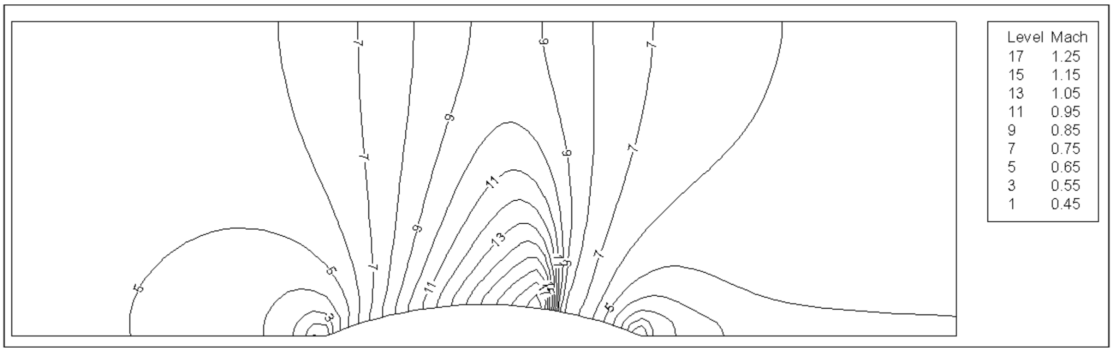

Figure 4.160 Steady state Mach No. contours, transonic circular arc

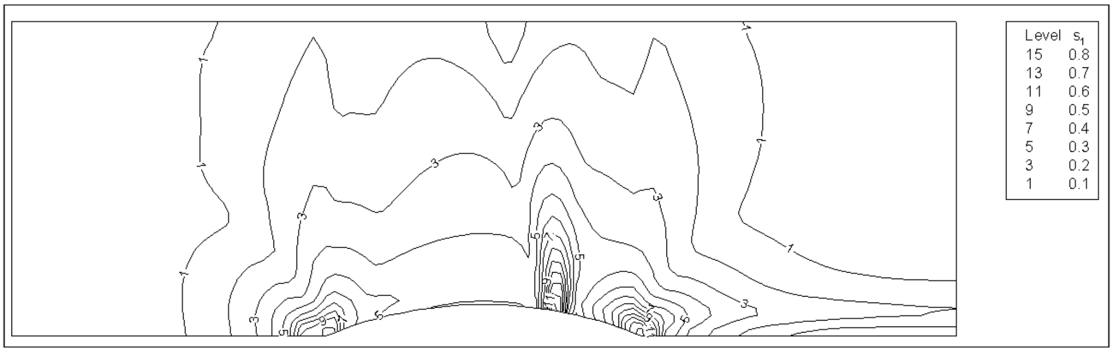

Figure 4.161 Steady state s_1 contours, transonic circular arc

Figure 4.162 Steady state s_2 contours, transonic circular arc

4.4.2.2 Completely Subsonic Flow, $P_b = 3.16$

In order to have a completely subsonic flow through out the circular arc domain, the back pressure should be relatively high. So the value of 3.16 is used. Figure 4.163, Figure 4.164, Figure 4.165, and Figure 4.166 plot the contours of the conservation variables $\rho, \rho u_1, \rho u_2, \rho e_t$, respectively. Figure 4.167 shows the Mach number contours, while Figure 4.168 and Figure 4.169 plot s_1 and s_2 contours, respectively.

Again the FDV/MFDV formulation succeeded in handling the completely subsonic flow with relatively low Mach number. The solution is symmetric over the circular arc. Figure 4.165 is of special importance because it indicates that the solution is symmetric about the bump. This solution matches the results obtained in [57] very well.

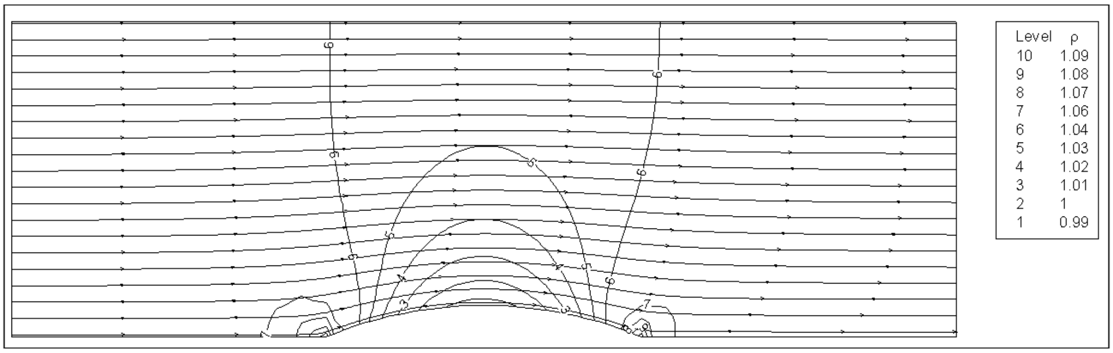

Figure 4.163 Steady state ρ contours, subsonic circular arc

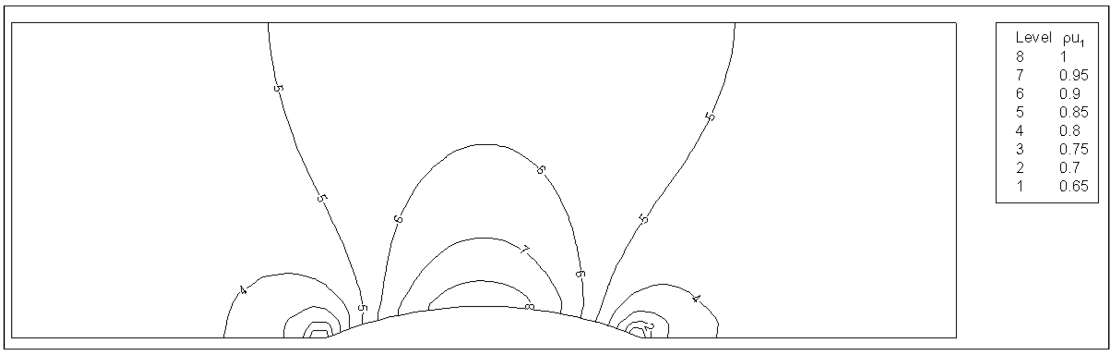

Figure 4.164 Steady state ρu_1 contours, subsonic circular arc

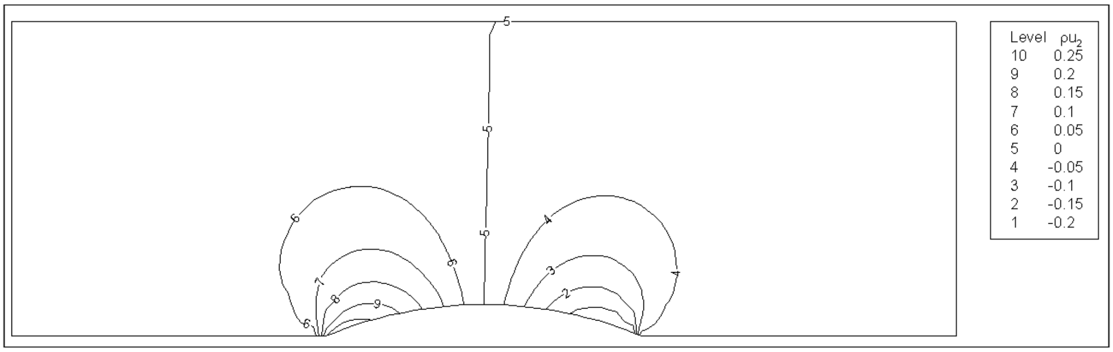

Figure 4.165 Steady state ρu_2 contours, subsonic circular arc

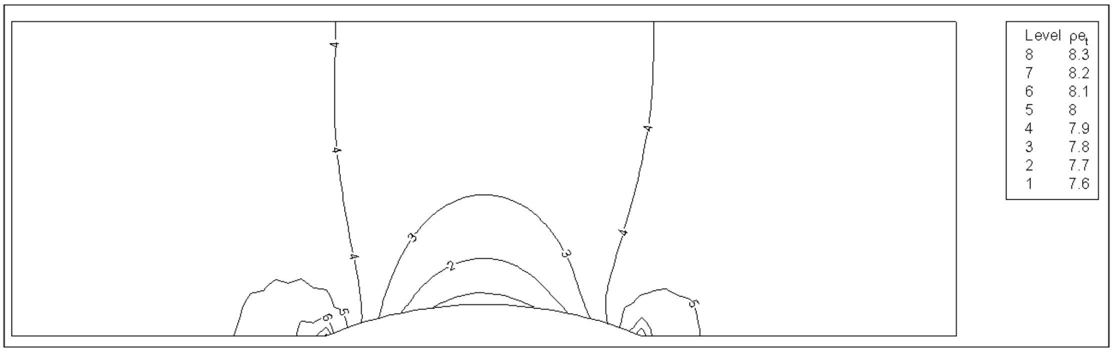

Figure 4.166 Steady state ρe_t contours, subsonic circular arc

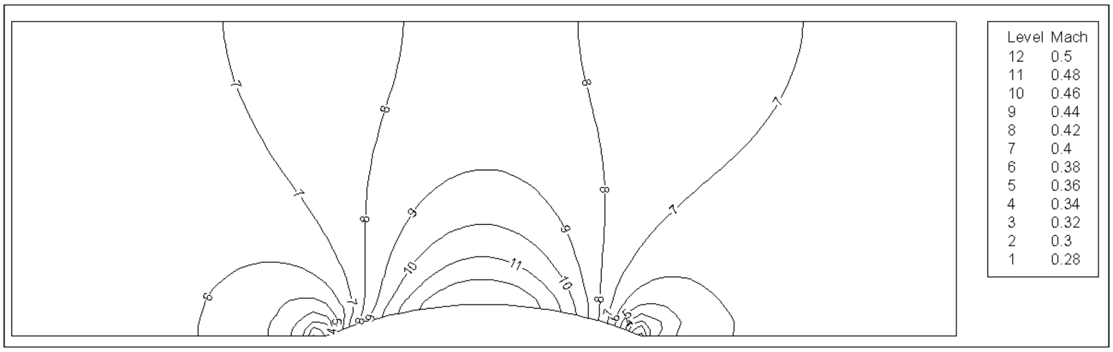

Figure 4.167 Steady state Mach No. contours, subsonic circular arc

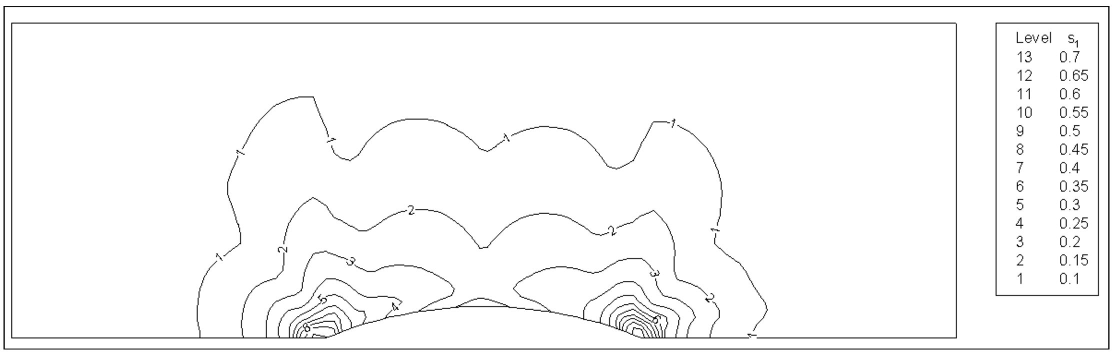

Figure 4.168 Steady state s_1 contours, subsonic circular arc

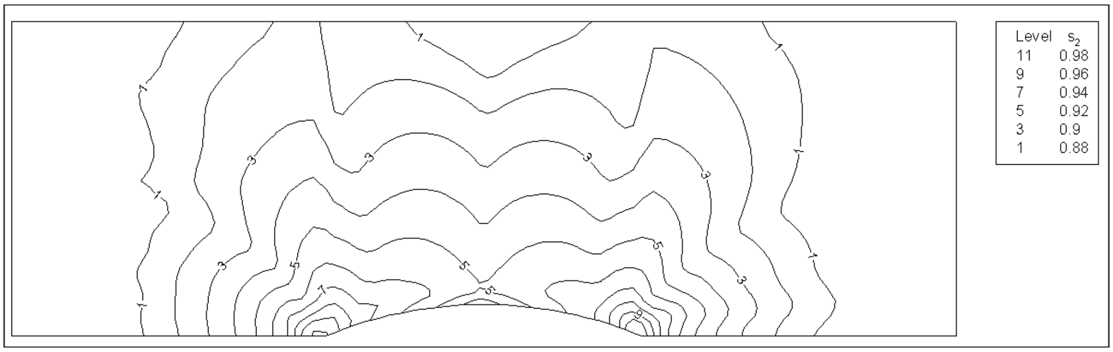

Figure 4.169 Steady state s_2 contours, subsonic circular arc

4.5 Conclusions and Future Research

The aim of the thesis as stated in section 1.5 was to develop a time-accurate flow solver for Euler equations. The flowfield dependent variation method (FDV) has been investigated to verify the claim that this method is capable of operating on all flow regimes. A modification of the theory has been developed and published in [56] during this research. Another paper has been submitted and accepted for publication in [60] that verifies the ability of the MFDV method to handle supersonic internal flow problems.

As a result of the theoretical analysis, a FDV-FEM compressible Euler flow solver has been developed. The characteristics of this solver have been stated in 1.5 and they have been carefully met in this research. Each characteristic will be discussed now to prove that it has been implemented.

- The solver is able to operate on virtually all flow regimes, starting from subsonic, transonic, and to the completely supersonic flow. No special treatment is required to handle any case. The same algorithm and computer program have been used in all the results in sections 4.3 and 4.4 for supersonic and subsonic/transonic flows, respectively.
- The time accuracy of the developed solver has been proved in section 4.2, where the analytical solution of the shock tube problem has been obtained with remarkable accuracy.

- As has been discussed in each of the proposed test cases in sections 4.2, 4.3, and 4.4, various boundary conditions that could simulate many problems have been included in the developed algorithm.
- Another benefit from the finite element implementation is the ability of the solver to handle unstructured grids and this has been demonstrated by freely discretizing all the domains in the proposed test cases without restrictions and without the need to allocate large transformation matrices.
- As demonstrated from the problems in sections 4.2, 4.3, and 4.4, when discontinuities and/or shockwaves are generated the proposed combination of FDV/MFDV algorithm with discontinuity capturing operator could resemble a solution that is free from the Gibbs type errors.
- The developed computer program has been written in FORTRAN 90 format. This program uses the preconditioned GMRES iterative solver discussed in Appendix B .
- Each computational step in the computer program has been carefully programmed and no external and/or built in functions have been used in the program code.

As final conclusion, the FDV/MFDV formulations appear to be promising for simulating all flow regimes. The finite element discretization gave the method more physical meaning and more power in discretizing any problem. Most of the encountered discontinuities such as shock waves have been simulated with outstanding ability to be captured with at maximum two elements. It is suggested for any future work that would be carried by any other interested researcher to look at the following points:

- The extension of this work to include the boundary conditions that is required for external flow problems.

- The computer program can be enhanced by supporting parallel computing algorithms.
- The sparse matrix GMRES solver can be replaced with combination of the GMRES iterative algorithm with element by element or edge by edge technique.
- The extension of this work for Navier-Stokes is a logical step because of the proved robustness of the FDV/MFDV formulations.

APPENDICES

Appendix A
Jacobian Matrices

A.1 Convection Jacobian $\mathbf{a}_i$

The convection flux vector is given in 2-D space in terms of the primitive variables by:

$$\mathbf{F}_i = \begin{Bmatrix} \rho u_i \\ \rho u_i u_1 + P\delta_{1i} \\ \rho u_i u_2 + P\delta_{2i} \\ \rho u_i e_t + P u_i \end{Bmatrix} = \begin{bmatrix} \rho u_1 & \rho u_2 \\ \rho u_1^2 + P & \rho u_2 u_1 \\ \rho u_1 u_2 & \rho u_2^2 + P \\ \rho u_1 e_t + P u_1 & \rho u_2 e_t + P u_2 \end{bmatrix} \tag{A.1}$$

In order to calculate the convection Jacobian for quasi-linear Euler equations, the convection flux vector should be written in terms of the conservation variables. Rewriting in terms of the conservation variable we get:

$$\mathbf{F}_i = \begin{bmatrix} U_2 & U_3 \\ \dfrac{U_2^2}{U_1} + P & \dfrac{U_2 U_3}{U_1} \\ \dfrac{U_2 U_3}{U_1} & \dfrac{U_3^2}{U_1} + P \\ \dfrac{U_2\left(U_4 + P\right)}{U_1} & \dfrac{U_3\left(U_4 + P\right)}{U_1} \end{bmatrix} \tag{A.2}$$

where;

$$P = \left(\gamma - 1\right)\left(U_4 - \frac{\left(U_2^2 + U_3^2\right)}{2U_1}\right) \tag{A.3}$$

The convection Jacobian is defined by:

$$\mathbf{a}_i = \frac{\partial \mathbf{F}_i}{\partial \mathbf{U}} \tag{A.4}$$

It is apparent from the definition that this Jacobian is a third order tensor. In 2-D space, this tensor will have the dimensions of $[2,4,4]$. A MATLAB script that have been used to drive this Jacobian in 2-D space is presented in A.1.1, followed with the outputs written in both of the conservation (subsection A.1.2) and primitive (subsection A.1.3) variables form. These results have been double checked with [1], [3], and [34] to assure their correctness.

A.1.1 Derivation by MATLAB

```
clear all
syms U1 U2 U3 U4 real

syms U1X1 U1X2 real
syms U2X1 U2X2 real
syms U3X1 U3X2 real
syms U4X1 U4X2 real
syms gam R real

U = [U1, U2, U3, U4];
DU = [U1X1, U1X2;
  U2X1, U2X2;
  U3X1, U3X2;
  U4X1, U4X2];

P = (gam-1)*(U4-(U2^2+U3^2)/(2*U1));
F2d = [U2         , U3      ;
  U2^2 /U1+P      , U2*U3/U1;
  U2*U3/U1        , U3^2 /U1+P;
  U2*U4/U1+P*U2/U1, U3*U4/U1+P*U3/U1];

for n = 1:4,
  for m = 1:4,
    for i = 1:2,

      a2d(n,m,i) = simplify(diff(F2d(n,i), U(m)));


    end
  end
end
echo on

pretty(expand(a2d(:,:,1)))
pretty(expand(a2d(:,:,2)))
```

```
echo off
```

A.1.2 Conservation Variables Form

$$\mathbf{a}_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ a_{21}^1 & (3-\gamma)\dfrac{U_2}{U_1} & (1-\gamma)\dfrac{U_3}{U_1} & (\gamma-1) \\ \dfrac{-U_2U_3}{U_1^2} & \dfrac{U_3}{U_1} & \dfrac{U_2}{U_1} & 0 \\ a_{41}^1 & a_{42}^1 & (1-\gamma)\dfrac{U_2U_3}{U_1^2} & \gamma\dfrac{U_2}{U_1} \end{bmatrix} \tag{A.5}$$

$$a_{21}^1 = \frac{(\gamma-3)U_2^2 + (\gamma-1)U_3^2}{2U_1^2}$$

$$a_{41}^1 = \frac{-U_2}{U_1^3}\left((1-\gamma)\left(U_2^2 + U_3^2\right) + \gamma U_4 U_1\right)$$

$$a_{42}^1 = \frac{2\gamma U_4 U_1 + (1-\gamma)\left(3U_2^2 + U_3^2\right)}{2U_1^2}$$

$$\mathbf{a}_2 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ \dfrac{-U_2U_3}{U_1^2} & \dfrac{U_3}{U_1} & \dfrac{U_2}{U_1} & 0 \\ a_{31}^2 & (1-\gamma)\dfrac{U_2}{U_1} & (3-\gamma)\dfrac{U_3}{U_1} & (\gamma-1) \\ a_{41}^2 & (1-\gamma)\dfrac{U_2U_3}{U_1^2} & a_{43}^2 & \gamma\dfrac{U_3}{U_1} \end{bmatrix} \tag{A.6}$$

$$a_{31}^2 = \frac{(\gamma-3)U_3^2 + (\gamma-1)U_2^2}{2U_1^2}$$

$$a_{41}^2 = \frac{-U_3}{U_1^3}\left((1-\gamma)\left(U_2^2 + U_3^2\right) + \gamma U_4 U_1\right)$$

$$a_{43}^2 = \frac{2\gamma U_4 U_1 + (1-\gamma)\left(U_2^2 + 3U_3^2\right)}{2U_1^2}$$

A.1.3 Primitive Variables Form

$$\mathbf{a}_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ a_{21}^1 & (3-\gamma)u_1 & (1-\gamma)u_2 & (\gamma-1) \\ -u_1u_2 & u_2 & u_1 & 0 \\ a_{41}^1 & a_{42}^1 & (1-\gamma)u_1u_2 & \gamma u_1 \end{bmatrix} \tag{A.7}$$

$$a_{21}^1 = \frac{1}{2}\left((\gamma-3)u_1^2 + (\gamma-1)u_2^2\right)$$

$$a_{41}^1 = -u_1\left((1-\gamma)\left(u_1^2 + u_2^2\right) + \gamma e_t\right)$$

$$a_{42}^1 = \gamma e_t + \frac{(1-\gamma)}{2}\left(3u_1^2 + u_2^2\right)$$

$$\mathbf{a}_2 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ -u_1u_2 & u_2 & u_1 & 0 \\ a_{31}^2 & (1-\gamma)u_1 & (3-\gamma)u_2 & (\gamma-1) \\ a_{41}^2 & (1-\gamma)u_1u_2 & a_{43}^2 & \gamma u_2 \end{bmatrix} \tag{A.8}$$

$$a_{31}^2 = \frac{1}{2}\left((\gamma-3)u_2^2 + (\gamma-1)u_1^2\right)$$

$$a_{41}^2 = -u_2\left((1-\gamma)\left(u_1^2 + u_2^2\right) + \gamma e_t\right)$$

$$a_{43}^2 = \gamma e_t + \frac{(1-\gamma)}{2}\left(u_1^2 + 3u_2^2\right)$$

A.2 Entropy Variables' Jacobians

In order to obtain the entropy variables Jacobian $\mathbf{C}$, the conservation variables should be written in terms of the entropy variables. The relation between the two variables' sets is given by:

$$\mathbf{U} = \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \rho e \begin{bmatrix} -V_4 \\ V_2 \\ V_3 \\ 1 - \dfrac{\left(V_2^{\,2} + V_3^{\,2}\right)}{2V_4} \end{bmatrix} \tag{A.9}$$

where;

$$\mathbf{V} = \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix} \tag{A.10}$$

$$\rho e = \left(\frac{(\gamma - 1)}{(-V_4)^{\gamma}} \right)^{\left(\frac{1}{(\gamma-1)}\right)} \exp\left(\frac{(s_0 - s)}{(\gamma - 1)} \right) \tag{A.11}$$

$$s = \gamma - V_1 + \frac{\left(V_2^{\,2} + V_3^{\,2}\right)}{2V_4} \tag{A.12}$$

Such that; $\mathbf{V}$ is the entropy variables vector. The entropy variables Jacobian is defined by:

$$\mathbf{C} = \frac{\partial \mathbf{U}}{\partial \mathbf{V}} \tag{A.13}$$

and the inverse Jacobian definition is defined by:

$$\mathbf{C}^{-1} = \frac{\partial \mathbf{V}}{\partial \mathbf{U}} \tag{A.14}$$

It is apparent from the definitions that these two Jacobians are second order tensors. In 2-D space, the entropy variables Jacobian $\mathbf{C}$ and its inverse

$\mathbf{C}^{-1}$ will have the size of $[4,4]$. The MATLAB script used in the derivation of these Jacobians is presented in A.2.1, followed with the outputs written in the entropy variables form in A.2.2. These results have been double checked with [1], [3], and [34].

A.2.1 Derivation by MATLAB

```
clear all
syms U1 U2 U3 U4 V1 V2 V3 V4 so real

syms gam R muu k real

U = [U1, U2, U3, U5];
V = [V1, V2, V3, V5];

s = gam-V1+(V2^2+V3^2)/(2*V5);
rhoe = ((gam-1)/(-V5)^gam)^(1/(gam-1))*exp((-s+so)/(gam-1));
UinV2d = rhoe*[-V5;
    V2;
    V3;
    1-(V2^2+V3^2)/(2*V5)];

for n = 1:4,
    for m = 1:4,
        C2d(n,m) = simplify(diff(UinV2d(n), V(m)));
    end
end

C2dInv = simplify(C2d^-1);

C2dSimple = simplify(C2d/rhoe*(gam-1)*V5);
C2dInvSimple = simplify(C2dInv*rhoe*-V5);

echo on

pretty(expand(C2dSimple))
pretty(expand(C2dInvSimple))

echo off
```

A.2.2 Entropy Variables Form

$$\bar{\gamma} = \gamma - 1$$

$$k_1 = \frac{V_2^2 + V_3^2}{2V_4}, \qquad k_2 = k_1 - 1, \qquad k_3 = k_1^2 - 2\gamma k_1 + \gamma$$

$$c_1 = \bar{\gamma} V_4 - V_2^2, \qquad c_2 = \bar{\gamma} V_4 - V_3^2, \qquad d_1 = -V_2 V_3$$

$$e_1 = V_2 V_4, \qquad e_2 = V_3 V_4$$

$$\mathbf{C} = \frac{\rho e}{\bar{\gamma} V_4} \begin{bmatrix} -V_4^2 & e_1 & e_2 & V_4 (1 - k_1) \\ & c_1 & d_1 & V_2 k_2 \\ & & c_2 & V_3 k_2 \\ & symm. & & -k_3 \end{bmatrix} \tag{A.15}$$

$$\mathbf{C}^{-1} = \frac{-1}{\rho e V_4} \begin{bmatrix} k_1^2 + \gamma & k_1 V_2 & k_1 V_3 & (k_1 + 1) V_4 \\ & V_2^2 - V_4 & -d_1 & e_1 \\ & & V_3^2 - V_4 & e_2 \\ & symm. & & V_4^2 \end{bmatrix} \tag{A.16}$$

Appendix B
Preconditioned Sparse Matrix GMRES Solver

In most, if not all, FEMs a global system contribution matrix results from the assembly of the element equations in the form:

$$\mathbf{A}_{n,n}\mathbf{X}_{n,1} = \mathbf{F}_{n,1} \tag{B.1}$$

Where; $\mathbf{A}$ is the global contribution matrix of size $[n,n]$, $\mathbf{F}$ is the right hand side vector $[n,1]$, $\mathbf{X}$ is the unknowns vector $[n,1]$, n is the order of the system.

To get the finite element solution this system has to be solved after enforcing the boundary conditions. Exact solvers, like Gauss elimination, are considered to be a good choice for small finite element applications. But, for actual finite element applications, where the size of the system becomes very large, an iteration-based matrix solver is due.

The matrix storage should also be reduced, because many of the matrix entries are essentially zeros. These zeros come from the naturally unconnected nodes. The sparse matrix storage is considered to be a good solution. Many other storage formats exists like Element By Element (EBE) storage and edge by edge. But these methods have been developed mainly to help in parallel computing strategies. Because parallel computing facilities are not an option, only the sparse matrix format is considered.

For large time steps and/or badly meshed domains (with very large and very small elements), the global contribution matrix becomes ill-conditioned. Such systems can slow the convergence of any iterative solver. It can even cause failure of convergence in some cases. Many preconditioning techniques have been developed to improve the convergence of these systems by using a certain transformation. This transformation clusters the matrix bandwidth, and hence accelerates the convergence. The reader is referred to [61]-[66] for further readings on this subject.

One of the most successful iterative solvers is the Generalized Minimal RESidual (GMRES). This solver has the ability to work on unsymmertic matrices with superior performance compared to other conventional solvers. In this appendix the pseudo code of the GMRES solver is introduced. Also the implemented preconditioning techniques are introduced. The sparse matrix format is left as a programming option.

B.1 GMRES Algorithm

From [64] the GMRES algorithm can be summarized as follows:

- Choose initial value for $X_{\beta}^{(0)}$ and compute:

$$E_{\alpha}^{(0)} = F_{\alpha} - A_{\alpha\beta} X_{\beta}^{(0)}$$
$$\overline{E}_{\alpha}^{(1)} = E_{\alpha}^{(0)} \Big/ \left\| E_{\alpha}^{(0)} \right\|$$

- Loop for a selected integer r (Krylov space dimension), and compute:

$$\begin{aligned}
&for\ i = 1,2,......,r \\
&\quad \widetilde{E}_{\alpha}^{(i)} = A_{\alpha\beta} \overline{E}_{\beta}^{(i)} \\
&\quad for\ j = 1,2,......,i \\
&\quad\quad H\left(j,i\right) = \widetilde{E}_{\alpha}^{(i)} \overline{E}_{\alpha}^{(i)} \\
&\quad\quad \widetilde{E}_{\alpha}^{(i)} = \widetilde{E}_{\alpha}^{(i)} - H\left(j,i\right) \overline{E}_{\beta}^{(j)} \\
&\quad end\ j \\
&\quad H\left(i+1,i\right) = \left\| \widetilde{E}_{\alpha}^{(i)} \right\| \\
&\quad \overline{E}_{\alpha}^{(i+1)} = \widetilde{E}_{\alpha}^{(i)} \Big/ H\left(i+1,i\right) \\
&end\ i
\end{aligned}$$

- Using Krylov space:

$$\mathbf{A}_{n,n} \mathbf{B}_{n,r} = \mathbf{B}_{n,r} \mathbf{H}_{r,r} \tag{B.2}$$

Where;

$$\mathbf{B}=\begin{bmatrix}\overline{E}_1^{(1)} & \overline{E}_1^{(2)} & . & \overline{E}_1^{(r)}\\ \overline{E}_2^{(1)} & \overline{E}_2^{(2)} & . & \overline{E}_2^{(r)}\\ . & . & . & .\\ \overline{E}_n^{(1)} & \overline{E}_n^{(2)} & . & \overline{E}_n^{(r)}\end{bmatrix} \tag{B.3}$$

- Get y_ξ that minimizes η, such that:

$$\eta=\min\left\|F_\alpha-A_{\alpha\beta}\left(X_\beta^{(0)}+z_\beta\right)\right\|=\min\left\|E_\alpha^{(0)}-A_{\alpha\beta}B_{\beta\xi}y_\xi\right\| \tag{B.4}$$

- Substituting from Krylov space definition:

$$\eta=\min\left\|E_\alpha^{(0)}-B_{\alpha\beta}H_{\beta\xi}y_\xi\right\|$$

$$\because E_\alpha^{(0)}=\left\|E_\alpha^{(0)}\right\|\overline{E}_\alpha^{(1)}=\left\|E_\alpha^{(0)}\right\|B_{\alpha\beta}\{1,0,0,...,0\}^{\mathrm{T}}$$

where;

$$e_\beta=\left\{\left\|E_\alpha^{(0)}\right\|,0,0,...,0\right\}^{\mathrm{T}}$$

$$\therefore E_\alpha^{(0)}=B_{\alpha\beta}e_\beta$$

- Substituting:

$$\eta=\min\left\|B_{\alpha\beta}\left(e_\beta-H_{\beta\xi}y_\xi\right)\right\|$$

So to have a minimum the following relation should hold:

$$e_\beta-H_{\beta\xi}y_\xi\cong 0$$

$$y_\xi=H_{\beta\xi}^{-1}e_\beta$$

$$X_\alpha^{(1)}=X_\alpha^{(0)}+B_{\alpha\beta}y_\beta$$

- Restart:

$$E_\alpha^{(1)}=F_\alpha-A_{\alpha\beta}X_\beta^{(1)}$$

$$\overline{E}_\alpha^{(1)}=E_\alpha^{(1)}\Big/\left\|E_\alpha^{(1)}\right\|$$

- Loop until convergence.

B.2 Q-R Modification

Using the same steps illustrated before except for the inversion of the Hessenberg. This modification mainly aims at avoiding the inversion of the Hessenberg matrix by using a system of successive rotations in the Krylov space. This transforms the Hessenberg matrix into an upper triangular matrix of the form:

$$\overline{H}_{\beta\xi} = R_{\beta\gamma} H_{\gamma\xi} = \begin{bmatrix} \overline{H}_{1,1} & \overline{H}_{1,2} & . & \overline{H}_{1,r-1} & \overline{H}_{1,r} \\ 0 & \overline{H}_{2,2} & . & . & \overline{H}_{2,r} \\ 0 & 0 & . & . & . \\ . & . & 0 & \overline{H}_{r-1,r-1} & . \\ 0 & 0 & 0 & 0 & \overline{H}_{r,r} \end{bmatrix} \tag{B.5}$$

$$\overline{e}_{\beta} = R_{\beta\gamma} e_{\gamma} = \begin{bmatrix} \overline{e_1} & \overline{e_2} & . & \overline{e_r} \end{bmatrix}^{\mathrm{T}} \tag{B.6}$$

where; $R_{\beta\gamma}$ is the rotation matrix given by:

$$R_{\beta\gamma} = \mathbf{R}^{r} \mathbf{R}^{r-1} .. \mathbf{R}^{1} \tag{B.7}$$

Such that:

$$\mathbf{R}^{k} = \begin{bmatrix} 1 & & & & & & & \\ & . & & & & & & \\ & & 1 & & & & & \\ & & & c_k & s_k & & & \\ & & & -s_k & c_k & & & \\ & & & & & 1 & & \\ & & & & & & . & \\ & & & & & & & 1 \end{bmatrix} \tag{B.8}$$

$$c_k = \frac{H(k,k)}{\sqrt{\left(H(k,k)\right)^2 + \left(H(k+1,k)\right)^2}} \tag{B.9}$$

$$s_k = \frac{H(k+1,k)}{\sqrt{\left(H(k,k)\right)^2 + \left(H(k+1,k)\right)^2}} \tag{B.10}$$

For more details the reader is referred to [64]-[65]. Then the system will be:

$$\bar{e}_{\beta} - \overline{H}_{\beta\xi} y_{\xi} = 0 \qquad \text{(B.11)}$$

and this system can be solved using backward substitution.

B.3 Preconditioning

In order to improve the convergence of the GMRES solver, a preconditioning technique may be used. This technique aims mainly to decrease the matrix bandwidth (the ratio between the maximum and minimum Eigen values) which leads to rapid convergence of the iterative solver. Y. Saad proposed a flexible inner-outer preconditioned GMRES algorithm in [65] (FGMRES). This algorithm has been adapted here for the use with sparse matrices. In what follows, $M_{n,n}$ represents the preconditioning matrix. As close as $M_{n,n}$ to $A_{n,n}$, the preconditioning technique will be efficient. Many approximations for the preconditioning matrix have been introduced in literature. All these matrices share the property of easily calculated inverse. The most famous preconditioning matrices are:

- The diagonal-preconditioning matrix. Where only the diagonal of the original matrix is kept.
- The tri-diagonal matrix. Where only the diagonal element, the first left element, and the first right element are kept.
- The Incomplete LU decomposition of zero order (ILU0).
- The Incomplete LU decomposition of first order (ILU1).

From [65] the FGMRES algorithm can be summarized in the following steps:

- Choose initial value for $X_{\beta}^{(0)}$ and compute:

$$E_{\alpha}^{(0)} = F_{\alpha} - A_{\alpha\beta} X_{\beta}^{(0)}$$

$$\overline{E}_{\alpha}^{(1)} = E_{\alpha}^{(0)} \Big/ \left\| E_{\alpha}^{(0)} \right\|$$

- Loop for a selected integer r :

$$
\begin{aligned}
& for\ i = 1,2,......,r \\
& \quad \widetilde{Z}_{\alpha}^{(i)} = M_{\alpha\beta}^{-1} \overline{E}_{\beta}^{(i)} \\
& \quad \widetilde{E}_{\alpha}^{(i)} = A_{\alpha\beta} \widetilde{Z}_{\alpha}^{(i)} \\
& \quad for\ j = 1,2,......,i \\
& \quad\quad H(j,i) = \widetilde{E}_{\alpha}^{(i)} \overline{E}_{\alpha}^{(i)} \\
& \quad\quad \widetilde{E}_{\alpha}^{(i)} = \widetilde{E}_{\alpha}^{(i)} - H(j,i) \overline{E}_{\beta}^{(j)} \\
& \quad end\ j \\
& \quad H(i+1,i) = \left\| \widetilde{E}_{\alpha}^{(i)} \right\| \\
& \quad \overline{E}_{\alpha}^{(i+1)} = \widetilde{E}_{\alpha}^{(i)} \big/ H(i+1,i) \\
& end\ i
\end{aligned}
$$

- Using Krylov space:

$$\mathbf{A}_{n,n} \mathbf{M}_{n,n}^{-1} \widetilde{\mathbf{Z}}_{n,r} = \widetilde{\mathbf{Z}}_{n,r} \mathbf{H}_{r,r} \tag{B.12}$$

where;

$$\widetilde{\mathbf{Z}}_{n,r} = \begin{bmatrix} \widetilde{Z}_1^{(1)} & \widetilde{Z}_1^{(2)} & . & \widetilde{Z}_1^{(r)} \\ \widetilde{Z}_2^{(1)} & \widetilde{Z}_2^{(2)} & . & \widetilde{Z}_2^{(r)} \\ . & . & . & . \\ \widetilde{Z}_n^{(1)} & \widetilde{Z}_n^{(2)} & . & \widetilde{Z}_n^{(r)} \end{bmatrix} \tag{B.13}$$

- Get y_{ξ} that minimizes η, such that:

$$
\begin{aligned}
\eta &= \min \left\| F_{\alpha} - A_{\alpha\beta} \left(X_{\beta}^{(0)} + z_{\beta} \right) \right\| \\
&= \min \left\| F_{\alpha} - A_{\alpha\beta} M_{\beta,\gamma}^{-1} M_{\gamma,\xi} \left(X_{\xi}^{(0)} + z_{\xi} \right) \right\| \\
&= \min \left\| \widetilde{Z}_{\alpha}^{(0)} - A_{\alpha\beta} M_{\beta,\gamma}^{-1} \widetilde{Z}_{\gamma}^{(i)} y_{\xi} \right\|
\end{aligned}
\tag{B.14}
$$

- Substituting from Krylov space definition:

$$\eta = \min \left\| \widetilde{Z}_{\alpha}^{(0)} - \widetilde{Z}_{\alpha,r} H_{r,\xi} y_{\xi} \right\| \tag{B.15}$$

- Complete as regular GMRES.

Appendix C
Error Indicators

Error estimators for finite element methods have been well developed since the early 1970s. There are two different finite element computational errors norm forms. These forms are:

- Pointwise error norms.
- Spacewise error norms.

The type of norms are many and range in the complexity and the mathematical base for which a certain norm has been designed. From these norms we report the following:

- Sobolev-Space norm error.
- Hilbert-Space norm error.
- Energy norm error.
- p norm error.
- L_2 norm error.
- L_∞ norm error.

The most used norm error in practice is the L_2 norm. It can be casted in both the pointwise and spacewise forms. To define this error norm we need to define the global node error ε_α as:

$$\varepsilon_\alpha = \psi_\alpha - \hat{\psi}_\alpha \tag{C.1}$$

where; $\psi_\alpha, \hat{\psi}_\alpha$ are the numerical and exact solutions, respectively. The spacewise L_2 norm error will take the form:

$$\|\varepsilon\|_{\text{Space}\,L_2} = \left(\int \varepsilon^2 dx \right)^{\frac{1}{2}} \tag{C.2}$$

while the pointwise counterpart will be:

$$\|\varepsilon\|_{\text{Point}\,L_2} = \left(\varepsilon_\alpha \varepsilon_\alpha\right)^{\frac{1}{2}} \tag{C.3}$$

and the percent may be computed by:

$$\|\varepsilon\|_{\text{Point}\,L_2\%} = \frac{\|\varepsilon\|_{\text{Point}\,L_2}}{\left(\psi_\beta \psi_\beta\right)^{\frac{1}{2}}} \tag{C.4}$$

Both relations given in (C.2) and (C.3) can be used to calculate the pointwise and spacewise L_2 norm error, respectively, whenever an analytical solution is known for a certain problem.

REFERENCES

[1] T. J. Chung, Computational Fluid Dynamics. Cambridge: Cambridge University Press, 2002.

[2] C. Hirsch, Numerical Computation of Internal and External Flows, Vol. 1: Fundamentals of Numerical Discretization. New York: John Wiley & Sons, 1988.

[3] C. Hirsch, Numerical Computation of Internal and External Flows, Vol. 2: Computational Methods for Inviscid and Viscous Flows. New York: John Wiley & Sons, 1990.

[4] J. D. Anderson, Jr., Modern Compressible Flow, with Historical Perspective. New York: McGraw-Hill, 1990.

[5] J. D. Anderson, Jr., Computational Fluid Dynamics, the Basics with Applications. New York: McGraw-Hill, 1995.

[6] K. A. Hoffman, S. T. Chiang, Computational Fluid Dynamics, Vol. 1 Fourth Edition. Texas, Austin: Engineering Education System, 2000.

[7] K. A. Hoffman, S. T. Chiang, Computational Fluid Dynamics, Vol. 2 Fourth Edition. Texas, Austin: Engineering Education System, 2000.

[8] H. Schichting, K. Gersten, Boundary-Layer Theory. New York: Springer-Verlag, 2003.

[9] B. R. Munson, D. F. Young, and T. H. Okiishi, Fundamentals of Fluid Mechanics, Third Edition. New York: John Wiley & Sons, 1990.

[10] W. H. Mason, Applied Computational Aerodynamics. Online Book, in Progress since 1997.

[11] J. Katz and A. Plotkin, Low-Speed Aerodynamics, from Wing Theory to Panel Methods. New York: McGraw-Hill, 1991.

[12] J. N. Reddy, An Introduction to the Finite Element Method, Second Edition. New York: McGraw-Hill, 1993.

[13] R. Lohner, K. Morgan, and O. C. Zienkiewiez, The solution of non linear hyperbolic equation systems by the finite element method. Int. J. Num. Meth. Fluids, 4, 1043-1063, 1984.

[14] J. Donea, A Taylor Galerkin method for .convective transport problems. Int. J. Num. Meth. Eng., 20, 101-119, 1984.

[15] A. J. Baker and J. W. Kim, A Taylor weak statement algorithm for hyperbolic conservation laws. Int. J. Num. Meth. Fluids, 7, 489-520, 1987.

[16] J. T. Oden, T. Strouboulis, and P. Devloo, Adaptive finite element methods for the analysis of inviscid compressible flow: Part I. fast refinement/unrefinement and moving mesh methods for unstructured meshes. Comp. Meth. Appl. Mech. Eng., 59, 327-362, 1986.

[17] J. T. Oden, T. Strouboulis and P. Devloo, Adaptive finite element methods for high speed compressible flows. Int. J. Num. Meth. Fluids, 7, 1211-1228, 1987.

[18] A. Harten, High resolution schemes for hyperbolic conservation laws. J. Comp. Phys., 49, 357-393, 1983.

[19] P. K. Sweby, High resolution schemes using flux limiters for hyperbolic conservation laws. SIAM J. Num. Anal., 21, 955-984, 1984.

[20] P. Colella and P. R. Woodward, The piecewise-parabolic method (PPM) for gas-dynamical simulation. J. Comp. Phys., 54, 174-201, 1984.

[21] S. Osher and S. Chakavarthy, High resolution schemes and the entropy condition. SIAM J. Num. Anal., 21, 955-984, 1984.

[22] H. C. Yee and A. Harten, Implicit TVD schemes for hyperbolic conservation laws in curvilinear coordinates. AIAA J., 25, 266-274, 1987.

[23] J. Y. Yang, Y. Liu, and H. Lomax, Computations of shock-wave reflection by circular cylinders, AIAA J., 25, 683-689, 1987.

[24] A. Harten and S. Osher, Uniformly high order accurate non-oscillatory schemes, I. SIAM J. Num. Anal., 24, 279-309, 1987.

[25] A. Harten, B. Engquist, S. Osher, and S. Chakavarthy, Uniformly high order accurate essentially non-oscillatory schemes, II. J. Comp. Phys., 71, 231-303, 1987.

[26] J. Y. Yang and C. K. Lombard, Uniformly second order ENO schemes for the Euler equations. AIAA Paper 87-1166-CP, 1987.

[27] C. W. Shu, TVB uniformly high order schemes for the conservation laws. Math. Comp., 49, 105-121, 1987.

[28] A. L. Brooks and T. J. R. Hughes, Streamline upwind Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. Comp. Meth. Appl. Mech. Eng., 32, 199-259, 1982.

[29] T. E. Tezduyar and T. J. R. Hughes, Finite element formulations for convection dominated flows with particular emphasis on the compressible Euler equations. Proceedings, AIAA 21st Aerospace Sciences Meeting, Reno, Nevada, AIAA Paper 83-0125, 1983.

[30] T. J. R. Hughes, L. P. Franca, and G. H. Hulbert, A new finite element formulation for computational fluid dynamics: I. Symmetric forms of the compressible Euler and Navier Stokes equations and the second law of thermodynamics. Comp. Meth. Appl. Mech. Eng., 54, 223-234, 1986.

[31] T. J. R. Hughes, M. Mallet and A. Mizukami, A new finite element formulation for computational fluid dynamics: II. Beyond SUPG. Comp. Meth. Appl. Mech. Eng., 54, 341-355, 1986.

[32] T. J. R. Hughes and M. Mallet, A new finite element formulation for computational fluid dynamics: III. The generalized streamline operator for multidimensional advective diffusive systems. Comp. Meth. Appl. Mech. Eng., 58, 305-328, 1986.

[33] T. J. R. Hughes and M. Mallet, A new finite element formulation for computational fluid dynamics: IV. A discontinuity capturing operator for multidimensional advective diffusive systems. Comp. Meth. Appl. Mech. Eng., 58, 329-339, 1986.

[34] F. Shakib, T. J. R. Hughes, and Z. Johan, A new finite element formulation for computational fluid dynamics: X. The compressible Euler and Navier-Stokes equations. Comp. Meth. Appl. Mech. Eng., 89, 141-219, 1991.

[35] J. F. Polk and P. P. Lynn, A least squares finite element approach to unsteady gas dynamics. Int. J. Num. Meth. Eng., 12, 3-10, 1978.

[36] H. Nguyen and J. Reynen, A Space-time least square finite element scheme for advection diffusion equations. Comp. Meth. Appl. Mech. Eng., 42, 331-342, 1984.

[37] C. A. J. Fletcher, A primitive variable finite element formulation for inviscid compressible flow. J. Comp. Physics, 33, 301-312, 1979.

[38] C. H. Bruneau, J. Laminie, and J. J. Chattot, Computation of 3-D vortex flows past a flat plate at incidence through a variational approach of the full steady Euler equations. Int. J. Num. Meth. Fluids, 9, 305-323, 1989.

[39] B. N. Jiang and L. A. Povinelli, Least squares finite element method for fluid dynamics. Comp. Meth. Appl. Mech. Eng., 81, 13-37, 1990.

[40] D. Lefebvre and J. Paraire, Finite element least squares solution of the Euler equations using linear and quadratic approximations. Int. J. Comp. Fluid Dyn., 1, 1-23, 1993.

[41] W. H. Reed and T. R. Hill, Triangular mesh methods for the Neutron transport equation. Los Alamos Scientific Laboratory Report LA-UR-73-479, 1973 (unpublished).

[42] P. Lesaint, Sur la resolution des systemes hyperboliques due premier ordre par des methods d'elements finis. Ph.D. thesis, Universite Paris, 1975 (unpublished).

[43] P. Lesaint and P. A. Raviart, On a finite element method to solve the Neutron transport equation, In Mathematical Aspects of Finite elements in partial Differential Equations, edited by C. de Boor. New York: Academic Press, 1974, pp. 89.

[44] B. Cockburn and C. W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: General framework. Math. Comp., 52, 411-435, 1989.

[45] B. Cockburn and C. W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: One dimensional systems. J. Comp. Phys., 84, 90-113, 1989.

[46] B. Cockburn and C. W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: The multidimensional case. Math. Comp., 54, 545-581, 1990.

[47] Z. Chen, B. Cockburn, J. Jerome, and C. W. Shu, Mixed-RKDG finite element method for the drift-diffusion semiconductor device equation. VLSI Design 3, 145-158, 1995.

[48] F. Bassi, S. Rebay, and M. Savini, A high resolution discontinuous Galerkin method for hyperbolic problems on unstructured grids. Proceedings, 3rd ICFD, Reading, April 1992, Num. Meth. Fluid Dyn., Clarendon, Oxford, 1993, pp. 345.

[49] F. Bassi, S. Rebay, and M. Savini, Discontinuous finite element Euler solutions on unstructured adaptive grids. Proceedings, 3rd ICNMFD, Rome, July 1992, Lecture Notes in Phys., Vol. 414, New York/Berlin: Springer-Verlag, 1993, pp. 245.

[50] F. Bassi and S. Rebay, Accurate 2-D Euler computations by means of a high order discontinuous finite element method. Proceedings, 14th ICNMFD, Bangalore, July 1994, Lecture Notes in Phys., New York/Berlin: Springer-Verlag, 1994.

[51] F. Bassi and S. Rebay, A higher-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations. J. Comp. Physics, 131, 267-279, 1997.

[52] T. J. Chung, Transitions and interactions of inviscid/viscous, compressible/incompressible and laminar/turbulent flows. Int. J. Num. Meth. Fluids, 31, 223-246, 1999.

[53] G. Schunk, F. Gunabal, G. Heard, and T. J. Chung, Unified CFD methods via flowfield-dependent variation theory. Proceedings, AIAA 30th Fluid Dyn. Conf., Norfolk, Virginia, AIAA Paper 99-3715, 1999.

[54] K. T. Yoon and T. J. Chung, Three dimensional mixed explicit-implicit generalized Galerkin spectral elements methods for high speed turbulent compressible flows. Comp. Meth. Appl. Mech. Eng., 135, 343-367, 1996.

[55] K. T. Yoon, S. Y. Moon, S. A. Garcia, G. W. Heard, and T. J. Chung, Flowfield-dependent mixed explicit (FDMEI) methods for high and low speed and compressible and incompressible flows. Comp. Meth. Appl. Mech. Eng., 151, 75-104, 1998.

[56] A. A. Megahed, M. W. El-Mallah, and B. R. Girgis, A modified flowfield dependent variation method applied to the compressible Euler flow equations. 8^{th} Int. Con. Fluid Dyn. and Prop., ICFDP8-EG-102, 2006.

[57] F. Moussaoui, A unified approach for inviscid compressible and nearly incompressible flow by least-squares finite element method. Appl. Num. Math., 44, 183-199, 2003.

[58] G. J. Le Beau and T. E. Tezduyar, Finite element computations of compressible flows with the SUPG formulation. Advances in Finite Element Analysis in Fluid Dynamics, Edt. M. N. Dhaubhadel, M. S. Engelmann and J. N. Reddy, FED-Vol. 123, ASME, New York, 1991, pp. 21-27.

[59] F. Taghaddosi, W. G. Habashi, W. Guevremont, D. Ait-Ali-Yahia, An adaptive least-squares method for the compressible Euler equations. Int. J. Num. Meth. Fluids, 31, 1121-1139, 1991.

[60] A. A. Megahed, M. W. El-Mallah, and B. R. Girgis, Solution of supersonic internal flow problems using MFDV. World Con. Eng., ICAEM-23, 2007.

[61] Y. Saad, Iterative Methods for Sparse Linear Systems. 2^{nd} edition, SIAM, Philadelphia, PA, 2003.

[62] X. Xu, N. Qin, and B. E. Richards, α-GMRES: A new parallelizable iterative solver for the large sparse non-symmetric linear systems arising from CFD. Int. J. Num. Meth. Fluids, 15, 613-623, 1992.

[63] L. B. Wigton, N. J. Yu, and D. P. Young, GMRES Acceleration of Computational Fluid Dynamics Codes. Proceeding, 7^{th} Comp. Fluid Dyn. Conf., Cincinnati, Ohio, July 1985, Technical Papers A85-40926 19-34, New York, 1985, pp. 67-74.

[64] Y. Saad and M. Schultz, GMRES: A new generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM, 7, 856-869, 1986.

[65] Y. Saad, A flexible inner-outer preconditioning GMRES algorithm. SIAM, 14, 461-469, 1993.

[66] Y. Saad, Preconditioning techniques for indefinite and nonsymmetric linear systems. J. Comp. Appl. Math., 24, 89-105, 1988.