DEVELOPMENT OF A NAVIER-STOKES SOLVER FOR MULTI-BLOCK APPLICATIONS

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

BY

ERİNÇ ERDOĞAN

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

SEPTEMBER 2004

Approval of the Graduate School of Natural and Applied Sciences

Prof. Dr. Canan ÖZGEN
Director

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

Prof. Dr. Kemal İDER
Head of Department

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

Dr. H. Tuğrul TINAZTEPE	Prof. Dr. M. Haluk AKSEL
Co-Supervisor	Supervisor

Examining Committee Members:

Prof. Dr. Kahraman ALBAYRAK (METU, ME)

Prof. Dr. M. Haluk AKSEL (METU, ME)

Dr. H. Tuğrul TINAZTEPE (ROKETSAN)

Asst. Prof. Dr. Tahsin ÇETİNKAYA (METU, ME)

Asst. Prof. Dr. Cüneyt SERT (METU, ME)

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

Name, Last name :

Signature :

ABSTRACT

DEVELOPMENT OF A NAVIER-STOKES SOLVER FOR MULTI-BLOCK APPLICATIONS

ERDOĞAN, Erinç
M. S., Department of Mechanical Engineering
Supervisor: Prof. Dr. M. Haluk AKSEL
Co-Supervisor: Dr. H. Tuğrul TINAZTEPE

September 2004, 161 pages

A computer code is developed using finite volume technique for solving steady two-dimensional and axisymmetric compressible Euler and Navier-Stokes equations for internal flows by "multi-block" technique. For viscous flows, both laminar and turbulent flow properties can be used. Explicit one step second order accurate Lax-Wendroff scheme is used for time integration.

Inviscid solutions are verified by comparing the results of test cases of a support project which was supported by ONERA/France for Turkey T-108, named "2-D Internal Flow Applications for Solid Propellant Rocket Motors". For laminar solutions, analytical flat plate solution is used for planar case and theoretical pipe flow solution is used for axisymmetric case for verification. Prandtl turbulent flow analogy is used in a flat plate solution to verify the turbulent viscosity calculation. The test cases solved with single-block code are compared with the ones solved with multi-block technique to verify the multi-block algorithm and good similarity is observed between single-block solutions and multi-block solutions. For the burning simulation of propellant of Solid Propellant Rocket Motors, injecting boundary is used. Finally, a segmented solid propellant rocket motor case is solved to show the multi-block algorithm' s flexibility in solving complex geometries.

Keywords : Multi-Block, Axisymmetric Flow, Internal Flow, Navier-Stokes Solver, Injection Boundary Condition

ÖZ

ÇOK BLOKLU UYGULAMALAR İÇİN NAVIER-STOKES ÇÖZÜCÜSÜ GELİŞTİRİLMESİ

ERDOĞAN, Erinç
Yüksek Lisans, Makine Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. M. Haluk AKSEL
Ortak Tez Yöneticisi: Prof. Dr. H. Tuğrul TINAZTEPE

Eylül 2004, 161 sayfa

Sıkıştırılabilir gazlar için sonlu hacimler yöntemi kullanılarak tek zaman için iki boyutlu ve eksenel-simetrik iç akış çözümlerinde kullanılmak üzere "Çok-blok" çözebilme yetisine sahip Euler ve Navier-Stokes denklemlerini çözebilen bir bilgisayar programı geliştirilmiştir. Sürtünmeli akışlar için laminar ve türbülanslı akış özellikleri kullanılabilmektedir. Zamanda entegrasyon için tek adım ilerlemeli ikinci dereceden hassas Lax-Wendroff yöntemi kullanılmıştır.

Sürtünmesiz akışların doğrulanması ONERA/Fransa tarafından desteklenen T-108 Türkiye destek projesi kapsamında elde edilen sonuçlar kullanılmıştır. Laminar akışlarda doğrulama için planer durumlarda düz plaka analitik sonuçları kullanılmış, eksenel-simetrik durumlar için de boru içerisindeki akış için teorik çözümler kullanılmıştır. Türbülans viskozitesi hesabının doğrulanması için düz plakada Prandtl türbülanslı akış analojisi kullanılmıştır. Tek bloklu çözülen örnekler Çok-blok yöntemi ile de çözülmüştür ve sonuçların uyum içerisinde olduğu görülmüştür. Katı yakıtlı roket motorlarının yakıtının yanmasının modellenmesi için enjeksiyon sınır koşulları kullanılmıştır. Son olarak, Çok-blok yönteminin karmaşık geometrilerdeki yeteneklerinin anlaşılması için birden çok parçadan oluşan bir roket motoru çözülmüştür.

Anahtar Kelimeler : Çok-blok, Eksenel-Simetrik Akış, İç Akış, Navier-Stokes Çözücü, Enjeksiyon Sınır Koşulu

NOMENCLATURE

LIST OF SYMBOLS

a	speed of sound
C_p	specific heat at constant pressure
e	internal energy per unit volume
G	mass flux
h_o	total enthalpy
H	axisymmetric flux term
i	unit normal vector
J	Jacobian of transformation
l	length
M	Mach number
n	normal
p	pressure
Pr	Prandtl number
q	heat flux
r	Cartesian coordinate position vector magnitude
R	gas constant
Re	Reynolds number
S	surface area
t	time
T	temperature
u	x-component of velocity vector
U	conservative variables vector
v	y-component of velocity vector
V	velocity vector magnitude
w	z-component of velocity vector
x	Cartesian space coordinate
y	Cartesian space coordinate
Y	normal distance from the wall for turbulence model
z	Cartesian space coordinate

Other Symbols:

k	thermal conductivity
ρ	density
ζ	shear stress
γ	specific heat ratio
μ	viscosity coefficient

ω	vorticity
ξ	local cell coordinate
η	local cell coordinate
∂	partial derivative operator
Ω	control volume
δ	boundary layer thickness, residual operator
σ	artificial viscosity coefficient

Superscripts:

n	quantity at discrete time level n
*	non-dimensional quantity
+	non-dimensional turbulence variable
l	direction
m	direction

Subscripts:

∞	freestream
$cell$	cell property
$exit$	inlet property
i	inviscid component of flux vector, cell number
inj	injection property
$inlet$	inlet property
lam	laminar
n	normal component
ref	reference
$turb$	turbulent
t	tangential component
θ	rotation direction on x-y plane
v	viscous component of flux vector
$wall$	inlet property
x	x component
y	y component

ACKNOWLEDGMENTS

I would like to express my deepest thanks and gratitude to Prof. Dr. M. Haluk AKSEL for his supervision, encouragement, understanding and constant guidance.

I would like to express my sincere appreciation to Dr. H. Tuğrul TINAZTEPE for providing me the base Navier-Stokes solver and his crucial advises and suggestions during this thesis study.

I would also like to thank Mine YUMUŞAK for sharing her experiences and her suggestions during the development of the solver.

I would like to thank to ROKETSAN for partially supporting this study. I would also like to thank my colleagues in ROKETSAN for their support and discussions which were very helpful for me.

Special thanks go to my friend Onur AVCI for his moral support from the beginning of this study to the end far beyond the oceans.

Finally, my greatest thanks go to my family, Fatma and Doğan, to whom this thesis is devoted. Without them, nothing would be possible.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

For the analysis of any fluid flow, solution of Navier-Stokes equations is needed to simulate all the proper behaviours of the motion of the fluid. Besides, for viscous flows, a relation between viscosity and heat conductivity with a thermodynamic property has to be established. The solution of this system of equations result in an accurate simulation of the fluid flows.

In solving the Navier-Stokes equations, Partial Differential Equations (PDE) come across which are quite complicated. Thus, mostly analytical solution of the Navier-Stokes equations needs some assumptions so that the equations become tractable. This means that analytical solution is possible only with some restrictions imposed on the fluid flow. Application of boundary conditions brings further restrictions to the solution procedure [1]. Dunlap, Willoughby and Hermsen [2] have solved the flow inside a solid propellant rocket motor (SPRM) with the inviscid and incompressible flow assumptions, which is a good example of the restrictions that an analytic solution requires.

While analytical methods are limited, experimental methods can give all the information that can be obtained from a fluid flow, but only for a specific problem.

Due to the limitations of data acquisition and experiment setups, because of increasing costs and physical restrictions, the experimental analysis is not very flexible and easily applicable. Nevertheless, the flow field information from experiments is valuable in validating the mathematical solutions of the governing equations [1].

Over the last forty years, a new form of solution method to handle the fluid motion equations has begun to be used, with an increasing field of application. This method is called as the Computational Fluid Dynamics (CFD). Actually, CFD is the art of replacing the governing PDE’ s of fluid flow with numbers, and advancing these numbers in space and/or time to obtain a final numerical description of the complete flow field of interest [3]. Since the solution procedure of the CFD methods requires a number of repetitive calculations, the use of computer is a must for these calculations. By the increasing computer speeds and memory usages, more detailed problems with higher accuracy can be handled by the CFD methods. Since computer speed and their memory usage capacity is increasing rapidly, CFD applications are being used in a wider field of interests day by day.

One of the fields of interests of the CFD, in fact one of the most commonly used one, is internal aerodynamics. Solution of an internal flow plays a great role in the design and performance prediction of a propulsion system, such as a SPRM. A simple SPRM can be said to be formed of two sections, combustion chamber where all the solid propellant inside it is burned during the flight of the rocket, and a nozzle section where the burned propellant particles are thrown out while thrust is being produced to create the motion of the rocket. The propellant inside the SPRM, which is called the grain, has a form that is variable for the desired burning area conditions, which specifies the thrust regime of the rocket. The shape of the grain is also important for the pressure variations inside the combustion chamber as well as the nozzle section. The nozzle gas dynamics calculations are sensitive to the flow entering the nozzle [4]. Besides the effect on combustion performance, accurate predictions of the internal flow field are also needed to properly estimate the motor case insulator requirements, and thereby avoid performance penalties due to excessive inert weight. Underestimation of insulator thickness has an even greater penalty, as it can result in burn-through of the motor case and failure of the rocket during flight [5]. Beyond these topics, effects of flow on structural loading of the rocket must also be well known in order to have a safe combustion, which is the most critical concept for a rocket designer [6].

Considering all of these factors, the flow inside the SPRM must be known as accurate as possible.

To simulate a SPRM, an injecting boundary is required where the propellant is burning. Even the simplified internal analysis is able to give reasonable accuracy on total performance prediction of a SPRM, which is performed with the assumption of one-dimensional flow [7] flow simulations performed at different injection directions [8,9] have shown that the flow in mass injected chambers is highly two-dimensional.

Thus, to obtain an efficient and accurate results, a two dimensional (2-D) Navier-Stokes solver should be used in the simulation of a SPRM.

1.1 THE NECESSITY OF A MULTI-BLOCK SOLVER

Before a flow simulation is to be performed, a grid should be created to define the physical domain of the flow. Accuracy of the solutions achieved by CFD methods often depends on the mesh accuracy of the grid. When mesh size decreases, the grid used represents the geometry better and as a result the errors coming from the numerical representation of the solution domain decrease. Thus, one needs to use a mesh size which is as small as possible. But this situation brings another problem. As the mesh size decreases, the number of nodes used increase. This increased node number results in a longer computational time and need of more memory usage, thus stands as a big disadvantage in sense of time and cost. Besides the disadvantages of time and money, there exist such cases for which an satisfactory grid can not be obtained or it is even not possible to obtain a grid which is defining the complete geometry of the physical domain.

Multi-block technique is a solution to these problems. Multi-block technique (also known as multiple zone approach) can be explained as a process of dividing the whole domain into subdomains, named as blocks, and solving each one independently while connecting their interfaces properly so that the physics of the problem is conserved. By doing this, it is possible to create independent grids for each block, thus complex geometries can easily be handled in the sense of grid generation. This simplicity also shortens the time needed to create a grid. In this sense, Multi-Block technique can be called to be cost-effective when compared to single block solvers. Use of structured meshes may bring some problems when a complex geometry is to be solved. Unstructured meshes are partially a solution for creating grids for complex geometries, but they are difficult to create when compared to structured meshes. Solution of unstructured grids are also more difficult when compared to structured grid solvers. Besides, there exist various cases that a grid with unstructured mesh does not give the desired accuracy. Multi-block technique can be said to bring the flexibility of creating a grid for complex geometries while preserving the simplicity of creating a structured mesh.

1.2 LITERATURE SURVEY

Multi-block technique recently has become popular with the developments in the computer technology which allows higher CPU speed and larger memory usage.

Lijewski [10] applied multi-block method for finned bodies, using only 2 and 3 blocks. Atkins [11] compared a single-block solver with its multi-block version for both inviscid and viscous flows about aircraft configurations. As stated in his study, the multi-block method preserves the convergence and accuracy properties of the original single-block flow solver.

Bush [12] also worked on both inviscid and viscous flows. A “Ni’ s bump” problem is used to test the solver. Consecutively, sharp and blunt bodies are solved with multi-block approach. These sharp and blunt geometries strongly emphasize the power of a multi-block flow solver.

Arthur, Blaylock and Anderson [13] used a multi-block algorithm for inviscid airfoil flows together with a multigrid algorithm. Jenssen [14] used large number of blocks with an implicit algorithm for both inviscid and viscous flow and concluded that thin blocks cause the algorithm to breakdown. Delta wing test cases are solved in this study. He concluded that the method is well suited for massively parallel computers.

A different approach with multi-block algorithm was studied by Sheng, Whitfield and Anderson [15] where unstructured grids were used instead of structured one which is not very common. Incompressible Reynolds-averaged Navier-Stokes equations are used to solve airfoil test cases in this study and good consistency has been observed with the single block solutions, with 73 % less memory usage with the multi-block solver

Madavan and Rai [16] used multi-block approach in turbomachinery applications.

Cambier, Ghazzi, Veuillot and Viviand [17] used different blocks with different flow properties. One of the blocks is solved with viscous flow assumption, while the other block, where inviscid flow dominates, is solved with the inviscid flow assumption. Also, regions with shocks, which are obtained with the single-block solver, are treated differently, all resulting in a total single viscous transonic flow.

Jenssen and Weinerfelt [18] investigated the influence of a coarse grid scheme for implicit multi-block calculations on parallel computers. As stated in their study, implicit flow solvers are often difficult to parallelize. Thus efficiency losses are present either computational or numerical, as the number of blocks increase. They reached a considerable reduction in the number of time steps for reaching convergence with the coarse grid correction scheme in subsonic and transonic flows.

Ghaffari, Luckring, Thomas, Bate and Biedron [19] carried out three dimensional thin-layer Navier-Stokes computations for the F/A-18 configuration. Their results are in good agreement with the flight test results.

A high-expansion ratio axisymmetric nozzle with and without longitudinal trailing-edge slots was studied by Carlson, Pao and Abdol-Hamid [20] using a three-dimensional simplified Reynolds-averaged Navier-Stokes computer code. Supercomputers were used in order to handle complex configurations in three-dimensions. Their results showed good agreement with the experimental data and they also concluded that slotted nozzles improves thrust ratio by 10 percent when compared to unslotted nozzles.

Giordano, Ivanov, Kashkovsky, Markelov, Tumino and Koppenwallner [21] applied the multi-block technique to the satellite thrusters to obtain their plume flow regime. They concluded that the application of their proposed methodology showed that optimized satellite design can benefit from a complete numerical study of nozzle and plume flows. This shows that the multi-block approach appears to be very effective in providing detailed information about the whole flowfield to a degree of accuracy that cannot be achieved by approximate engineering methods for nozzle flows and analytical plume models.

The alternating direction implicit (ADI) method was coupled with a multi-block approach in the study of Rosenfeld and Yassour, [22] which was presented as a novel approach for solving implicit systems of time-dependent two-dimensional partial differential equations in multi-block domains.

Another different approach combined with a multi-block algorithm is the usage of an implicit Total Variation Diminishing (TVD) method by Grasso and Marini [23] for hypersonic wind tunnel flows.

1.3 A BRIEF INTRODUCTION TO THE MULTI-BLOCK SOLVER

In the present study, an explicit two-dimensional planar single block external Navier-Stokes solver [24] has been modified into an explicit multi-block 2-D planar/axisymmetric Navier-Stokes solver for external or internal flow applications. The solver is also capable of solving single block applications. For viscous problems, Sutherland' s formula is used to calculate laminar viscosity and Baldwin-Lomax model is used to calculate turbulent viscosity. In the grid generation, non-over lapping blocks are used for multi-block solutions, having common node points from one face of both neighbouring nodes. Grids used in all flow solutions are of structured meshes, created using Tecplot 9.0 Mesh Generator. The solver in fact has a pointer system, which will be explained in following chapters, that allows also to solve unstructured meshes. But the existence of multi-block algorithm removes this need.

The burning of the propellant inside the rocket motor can be simulated by using an injecting surface from proper locations, as mentioned above, by which the burning propellant is introduced into the solution domain with a specified flame temperature and mass flux. Therefore, for these purposes, injection boundary conditions are used in the simulation of SPRM flows. For the validation of the solver, several test cases are used. For the 2-D axisymmetric cases with injection, a support project which was supported by ONERA (Office National D'Etudes et de Recherches Aerospatiales) /France for Turkey T-108 [25] and completed with the collaboration of Roketsan (RS) (Missile Industries, Ankara, Turkey) and ONERA.. The goal of the project was to modify and apply existing 2-D and axisymmetric flow solvers of RS to combustion chambers and nozzles of solid propellant rocket motors (SPRM). The test cases of the project will be explained in detail in Chapter 6. For the planar laminar case validation, Blasius solution for the flat plate is used. For the axisymmetric laminar case validation, theoretical fully developed laminar pipe flow velocity profile is used.

1.4 SCOPE AND OUTLINE OF THE THESIS

Nowadays, as the demand on parallel processing increases, which is widely used to decrease computer time and memory usage, the need of solvers with multi-block algorithms increases.

In the literature, although multi-block techniques are commonly used in various solvers, it is not widespread in internal flow solutions, especially SPRM solutions, most probably because of the secrecy policies in this technology. In this study, it is aimed to provide an accurate and efficient SPRM internal flow solver to the use of Turkish defense industry and academic studies performed in the similar topics.

The present study is explained in the following chapters, which are organized as follows:

Chapter 2 includes governing equations for the flow of viscous fluids. Chapter 3 explains the numerical discretization of the governing equations. Numerical calculation and time step calculation are also introduced in the same chapter. Initial and boundary conditions are explained in Chapter 4. Chapter 5 is devoted to the pointer system and multi-block technique used in the solver. The results of the test cases that are used in the validation of the present study are presented and discussed in Chapter 6. Finally, discussion of the results and future recommendations are given in Chapter 7.

CHAPTER 2

GOVERNING EQUATIONS

2.1 GENERAL

The fundamental equations of fluid dynamics are based on universal laws of conservation of mass, conservation of momentum and conservation of energy. In addition to the equations developed from these universal laws, it is necessary to establish relationships between fluid properties in order to close the system of equations. When combined with an equation of state, such as the perfect gas law, the constant Prandtl number assumption and an expression relating viscosity to temperature, such as Sutherland's law, a complete set of equations for laminar flow can be obtained.

2.2 AXISYMMETRIC NAVIER-STOKES EQUATIONS

The 2-D axisymmetric compressible Navier-Stokes equations written in conservative form for a Cartesian coordinate system are as follows:

$$\frac{\partial U}{\partial t}+\frac{\partial F}{\partial x}+\frac{\partial G}{\partial y}+\frac{\partial R}{\partial x}+\frac{\partial S}{\partial y}+\alpha(H_i-H_v)=0 \qquad (2.1)$$

where

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ e \end{bmatrix} \tag{2.2}$$

$$\mathrm{F} = \begin{bmatrix} \rho \mathrm{u} \\ \rho \mathrm{uu} + \mathrm{p} \\ \rho \mathrm{uv} \\ \rho \mathrm{uh}_0 \end{bmatrix} \tag{2.3}$$

$$\mathrm{G} = \begin{bmatrix} \rho \mathrm{v} \\ \rho \mathrm{uv} \\ \rho \mathrm{vv} + \mathrm{p} \\ \rho \mathrm{vh}_0 \end{bmatrix} \tag{2.4}$$

$$R = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ u\tau_{xx} + v\tau_{xy} + q_x \end{bmatrix} \tag{2.5}$$

$$S = \begin{bmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ u\tau_{xy} + v\tau_{yy} + q_y \end{bmatrix} \tag{2.6}$$

$$H_i = \frac{1}{y}\begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 \\ \rho v h_0 \end{bmatrix} \tag{2.7}$$

$$H_v = \frac{1}{y}\begin{bmatrix} 0 \\ -\tau_{xy} - \frac{2}{3} y \frac{\partial}{\partial x}\left(\mu \frac{v}{y}\right) \\ -\tau_{yy} + \tau_{\theta\theta} - \frac{2}{3}\mu\left(\frac{v}{y}\right) - y\frac{2}{3}\frac{\partial}{\partial y}\left(\mu\frac{v}{y}\right) \\ \rho v h_0 - u\tau_{xy} - v\tau_{yy} - q_y - \frac{2}{3}\mu\frac{v^2}{y} - y\frac{\partial}{\partial y}\left(\frac{2}{3}\mu\frac{v^2}{y}\right) - y\frac{\partial}{\partial x}\left(\frac{2}{3}\mu\frac{uv}{y}\right) \end{bmatrix} \tag{2.8}$$

where α=0 denotes that the governing equations are solved for a 2-D-planar domain and α=1 for a 2-D-axisymmetric domain. Since planar domains can also be solved within this thesis study, coordinates will be defined as (x,y) instead of (r,z).

The shear stress components are expressed as follows

$$\tau_{xx} = -\mu\left(\frac{4}{3}\frac{\partial u}{\partial x} - \frac{2}{3}\frac{\partial v}{\partial y}\right) \tag{2.9}$$

$$\tau_{yy} = -\mu\left(\frac{4}{3}\frac{\partial v}{\partial y} - \frac{2}{3}\frac{\partial u}{\partial x}\right) \tag{2.10}$$

$$\tau_{xy} = -\mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \tag{2.11}$$

$$\tau_{xy} = \tau_{yx} \tag{2.12}$$

$$\tau_{\theta\theta} = -\mu\left[-\frac{2}{3}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) + \frac{4}{3}\frac{v}{y}\right] \tag{2.13}$$

and the heat flux components are expressed as

$$q_x = -k\frac{\partial T}{\partial x} \tag{2.14}$$

$$q_y = -k\frac{\partial T}{\partial y} \tag{2.15}$$

In order to close the system of equations, thermodynamic pressure is written in terms of the conservative variables using equation of state for a perfect gas.

$$\mathrm{p} = (\gamma - 1)\left[\mathrm{e} - \frac{1}{2}\rho(\mathrm{u}^2 + \mathrm{v}^2)\right] \tag{2.16}$$

Also, total enthalpy per unit mass is defined as follows;

$$\mathrm{h}_0 = \frac{(\mathrm{e} + \mathrm{p})}{\rho} \tag{2.17}$$

where ρ is the density, u and v are the velocity components in the x and y directions, respectively, T is the temperature, e is the total internal energy per unit volume, μ is the total effective viscosity coefficient, k is the thermal conductivity, and γ is the ratio of specific heats.

2.2.1 NON-DIMENSIONALIZATION

In CFD applications, governing equations are often put in non-dimensional form. As a result of the non-dimensionalization, equations representing the flow are characterized by parameters as Reynolds number and Prandtl number instead of the actual values of viscosity and thermal conductivity [29].

The following non-dimensionalization procedure is applied for the present formulation.

$$x^* = \frac{x}{l_{ref}} \qquad y^* = \frac{y}{l_{ref}} \qquad u^* = \frac{u}{a_{ref}} \qquad v^* = \frac{v}{a_{ref}}$$

$$\rho^* = \frac{\rho}{\rho_{ref}} \qquad T^* = \frac{T}{T_{ref}} \qquad \mu^* = \frac{\mu}{\mu_{ref}} \qquad p^* = \frac{p}{a_{ref}^2} \tag{2.18}$$

$$\mathrm{h}_0^* = \frac{\mathrm{h}_0}{\mathrm{a}_{\mathrm{ref}}^2} \qquad e^* = \frac{e}{\rho_{ref} a_{ref}^2} \qquad t^* = \frac{t a_{ref}}{l_{ref}}$$

where l_{ref} is the reference length, T_{ref} is the reference temperature, ρ_{ref} is the reference density, μ_{ref} is the reference coefficient of viscosity and a_{ref} is the reference speed of sound at the reference temperature.

As a result of non-dimensionalization, following non-dimensional parameters appear in the governing equations:

$$Re_{ref} = \frac{\rho_{ref} a_{ref} l_{ref}}{\mu_{ref}} \tag{2.19}$$

$$Pr_{ref} = \mu_{ref} \frac{C_p}{k} \tag{2.20}$$

where Re_{ref} is the reference Reynolds number and Pr_{ref} is the reference Prandtl number. Relations given below also apply for the above equations.

$$a_{ref} = \sqrt{\gamma R T_{ref}} \tag{2.21}$$

$$C_p = \frac{\gamma}{\gamma - 1} R \tag{2.22}$$

This non-dimensionalization also yields the following useful relations:

$$p^* = \frac{\rho^* T^*}{\gamma} \qquad \text{(equation of state)} \tag{2.23}$$

$$a^* = \sqrt{T^*} \tag{2.24}$$

$$M^* = \frac{u^*}{a^*} \tag{2.25}$$

Throughout the present work, freestream stagnation conditions have been used , to determine reference values of pressure, speed of sound and temperature. The advantage of using stagnation conditions over many other choices of reference conditions is that the non-dimensional stagnation quantities reduce to

$p_{ref}=1/\gamma$ $\qquad$ $\rho_{ref}=1$ $\qquad$ $T_{ref}=1$

such that they are independent of the actual flow conditions.

Inserting non-dimensional variables defined in Equation (2.18) together with the given definitions given in Equation (2.19) and (2.20) into Equations (2.2) to Equation (2.8), non-dimensional form the governing equations can be obtained. From now on, all variables will be given in non-dimensional form and for convenience, superscript * will not be used. Since only the viscous shear terms and viscous axisymmetric source term have been changed, only their modified form are given below as follows:

$$H_v = \frac{1}{y}\begin{bmatrix} 0 \\ -\tau_{xy} - \frac{2}{3}\frac{y}{Re_{ref}}\frac{\partial}{\partial x}\left(\mu\frac{v}{y}\right) \\ -\tau_{yy} + \tau_{\theta\theta} - \frac{2}{3}\frac{\mu}{Re_{ref}}\left(\frac{v}{y}\right) - \frac{y}{Re_{ref}}\frac{2}{3}\frac{\partial}{\partial y}\left(\mu\frac{v}{y}\right) \\ \rho v h_0 - u\tau_{xy} - v\tau_{yy} - q_y - \frac{2}{3}\frac{\mu}{Re_{ref}}\frac{v^2}{y} - \frac{y}{Re_{ref}}\frac{\partial}{\partial y}\left(\frac{2}{3}\mu\frac{v^2}{y}\right) - \\ \frac{y}{Re_{ref}}\frac{\partial}{\partial x}\left(\frac{2}{3}\mu\frac{uv}{y}\right) \end{bmatrix} \tag{2.26}$$

$$\tau_{xx} = -\frac{\mu}{Re_{ref}}\left(\frac{4}{3}\frac{\partial u}{\partial x} - \frac{2}{3}\frac{\partial v}{\partial y}\right) \tag{2.27}$$

$$\tau_{yy} = -\frac{\mu}{Re_{ref}}\left(\frac{4}{3}\frac{\partial v}{\partial y} - \frac{2}{3}\frac{\partial u}{\partial x}\right) \tag{2.28}$$

$$\tau_{xy} = -\frac{\mu}{Re_{ref}}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \tag{2.29}$$

$$\tau_{\theta\theta} = -\frac{\mu}{Re_{ref}}\left[-\frac{2}{3}\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)+\frac{4}{3}\frac{v}{y}\right] \tag{2.30}$$

where

$$q = -\frac{\mu}{Re_{ref}(\gamma-1)Pr_{ref}}\frac{\partial T}{\partial x} \tag{2.31}$$

$$q = -\frac{\mu}{Re_{ref}(\gamma-1)Pr_{ref}}\frac{\partial T}{\partial y} \tag{2.32}$$

2.3 GOVERNING EQUATIONS FOR VISCOSITY

In the above form of governing Navier-Stokes equations μ is the effective viscosity coefficient. According to the eddy-viscosity concept due to Boussinesq [26] stresses resulting from turbulent fluctuations can be added to the mean flow stresses. Thus, the equation given below is used when the flow is turbulent

$$\frac{\mu}{\mu_{ref}} = \frac{\mu_{lam}}{\mu_{ref}} + \frac{\mu_{turb}}{\mu_{ref}} \tag{2.33}$$

where μ_{lam} corresponds to the laminar viscosity and μ_{turb} corresponds to the turbulent viscosity due to Boussinesq hypothesis. Thus, in the solution method, laminar viscosity and turbulent viscosity are calculated independently and then added to be used as the total effective viscosity coefficient.

Similarly Reynolds proposed an analogy between turbulent heat and momentum transfer [26] and introduced a turbulent Prandtl number concept as follows.

$$\frac{\mu}{\Pr} = \frac{\mu_{lam}}{\Pr_{lam}} + \frac{\mu_{turb}}{\Pr_{turb}} \tag{2.34}$$

where $\Pr_{lam}$ is the laminar Prandtl number and $\Pr_{turb}$ is the turbulent Prandtl number. In this study laminar and turbulent Prandtl numbers are assumed to be constant for each flow field analyzed and turbulent energy transport is included in Equation (2.1) by the approach given in Equation (2.34).

2.3.1 LAMINAR VISCOSITY MODEL

To complete the set of governing equations, an expression relating viscosity to temperature should be used. For solutions involving laminar flow, free stream laminar viscosity which is assumed to be constant for transonic flows is calculated from Sutherland' s formula [26] in SI units as follows:

$$\mu_{lam} = (1.45x10^{-6})\frac{T^{\frac{3}{2}}}{T+110} \tag{2.35}$$

where the temperature is specified in Kelvin degrees.

2.3.2 TURBULENT VISCOSITY MODEL

In order to calculate turbulent viscosity, the Baldwin-Lomax model [27] is used for turbulent flows. In this model, turbulent viscosity is defined by the equation given below

$$\mu_{turb} = \begin{cases} (\mu_{turb})_{inner}, Y \leq Y_{crossover} \\ (\mu_{turb})_{outer}, Y > Y_{crossover} \end{cases} \tag{2.36}$$

where Y is the normal distance from the wall and $Y_{crossover}$ is the smallest value of Y at which values from inner and outer formulas become equal. For inner layer eddy viscosity, $(\mu_{turb})_{inner}$, Prandtl mixing length theory is used as below

$$(\mu_{turb})_{inner} = \mathrm{Re}_{ref}\, \rho l_{inner}^2 |\vec{\omega}| \tag{2.37}$$

For this equation to be complete, l_{inner} should be known, which can be done by relating it to the flow conditions. For inner layer, l_{inner} is calculated by a formula derived by van Driest (1956) where the van Driest damping factor is applied, given as below

$$l_{inner} = \kappa Y \left(1 - e^{\frac{-Y^+}{A}}\right) \tag{2.38}$$

where the dimensionless damping constant A varies with flow conditions such as pressure gradient, wall roughness and blowing or suction. A≈26 for flat plate flow and κ, the Karman constant, is 0.41. The van Driest damping factor in brackets is derived from oscillating laminar flow near a wall. [28]

The non-dimensional wall distance Y^+ is defined as

$$Y^+ = \mathrm{Re}_{ref} \left(\frac{\rho u_\tau}{\mu_{lam}}\right)_{wall} Y \tag{2.39}$$

where Y is the distance from the wall and u_τ is called the friction velocity, calculated at wall for wall boundary layers.

$$u_\tau = \sqrt{\frac{\tau_{wall}}{\mathrm{Re}_{ref}\,\rho_{wall}}} \tag{2.40}$$

Substituting Equation (2.40) into Equation (2.39), Y^+ can be rewritten as

$$Y^+ = \sqrt{\frac{\mathrm{Re}_{ref}\,\rho_{wall}\,\tau_{wall}}{\mu_{wall}}}\,Y \tag{2.41}$$

To calculate inner layer eddy viscosity, ω, the vorticity should also be calculated. The formal definition of vorticity for a 2-D coordinate system is given as below,

$$\vec{\omega} = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \tag{2.42}$$

By the definitions given above, inner layer eddy viscosity can be calculated properly.

Now that the inner layer eddy viscosity is found, the outer layer eddy viscosity should also be calculated.

In the outer region where Y>Ycrossover, the turbulent viscosity is given as follows.

$$(\mu_{turb})_{outer} = \frac{\mathrm{Re}_{ref}\;0.016C_{cp}\rho F_{wake}F(Y)}{1+5.5\left(\dfrac{C_{Kleb}Y}{Y_{\max}}\right)^6} \tag{2.43}$$

where C_{kleb}, the Klebanoff parameter, is taken as 0.3 and F_{wake} is defined as

$$F_{wake} = \min \begin{cases} Y_{\max} F_{\max} \\ C_{WK} Y_{\max} \dfrac{V_{diff}^{\ 2}}{F_{\max}} \end{cases} \qquad (2.44)$$

where Y_{max} is the Y value corresponding to F_{max}. In the Equation (2.44), the upper formula is valid for attached wall boundary layers and the lower formula is valid for separated and attached wall boundary layers and wakes where C_{cp}=1.6 and C_{WK}=0.25 as proposed by Baldwin-Lomax model [27]. The velocity difference V_{diff} is calculated by the following equation,

$$V_{diff} = (V)_{\max} - (V)_{\min} \qquad (2.45)$$

where V denotes the magnitude of the velocity vector. In the above Equation (2.45) $(V)_{max}$ is the velocity value corresponding to F_{max} as it is in the definiton of Ymax. $(V)_{min}$ corresponds to the minimum velocity of the profile. The vorticity moment F(Y) is calculated from,

$$F(Y) = \begin{cases} Y|\vec{\omega}|\left(1 - e^{\frac{-Y^+}{A}}\right) \\ Y|\vec{\omega}| \end{cases} \qquad (2.46)$$

In this equation, the upper formula is applicable to wall boundary layers whereas the lower formula is applied to wakes. The denominator in Equation (2.43) is known as Klebanoff intermittency factor and causes the outer eddy viscosity vanish as Y goes to infinity, as expected.

2.4 GOVERNING EQUATIONS IN CURVILINEAR COORDINATES

The two dimensional axisymmetric Navier-Stokes equations in Cartesian coordinate system are transformed into a generalized body-conforming curvilinear coordinate system, preserving strong conservative law form by using metric transformation in order to facilitate the implementation of boundary conditions in arbitrary geometries [1,29,30,31]. A general transformation from Cartesian coordinates to curvilinear coordinates can be written as

$$(x,y) \rightarrow (\xi,\eta)$$

where

$$\xi=\xi(x,y)$$
$$\eta=\eta(x,y)$$

These transformations are made such that the grid spacing in the new coordinate system is uniform and of unit length and besides, there will be a one-to-one correspondence between the physical and computational domains. Cartesian derivatives can be obtained by chain rule as below:

$$\frac{\partial}{\partial x}=\xi_x\frac{\partial}{\partial \xi}+\eta_x\frac{\partial}{\partial \eta} \tag{2.47}$$

$$\frac{\partial}{\partial y}=\xi_y\frac{\partial}{\partial \xi}+\eta_y\frac{\partial}{\partial \eta} \tag{2.48}$$

The Jacobian of transformation can be written as

$$J = \frac{1}{x_\xi y_\eta - x_\eta y_\xi} \tag{2.49}$$

The Jacobian, J, is interpreted as the ratio of areas in the physical space and the computational space. Since the computational space has a uniform spacing of unity in both directions, it is the area of the cell in the physical space for the present formulation.

The metric terms can be written as:

$$\begin{aligned} \xi_x &= \frac{1}{J} y_\eta \\ \xi_y &= \frac{-1}{J} x_\eta \\ \eta_x &= \frac{-1}{J} y_\xi \\ \eta_y &= \frac{1}{J} x_\xi \end{aligned} \tag{2.50}$$

So far, equations required to make necessary transformations are derived. Strong conservation form of governing equations is now to be transformed from physical domain to computational domain while preserving the strong conservation form. The transformed form of Navier-Stokes equations (2.1) can be given as follows:

$$\begin{aligned} &\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} + \frac{\partial R}{\partial x} + \frac{\partial S}{\partial y} + \alpha(H_i - H_v) \rightarrow \\ &\qquad \frac{\partial U'}{\partial t} + \frac{\partial F'}{\partial \xi} + \frac{\partial G'}{\partial \eta} + \frac{\partial R'}{\partial \xi} + \frac{\partial S'}{\partial \eta} + \alpha(H_i' - H_v') \end{aligned} \tag{2.51}$$

where

$$U' = \frac{U}{J} \tag{2.52}$$

$$F' = \frac{1}{J}\left(\frac{\partial y}{\partial \eta}F - \frac{\partial x}{\partial \eta}G\right) \tag{2.53}$$

$$G' = \frac{1}{J}\left(\frac{\partial x}{\partial \xi}G - \frac{\partial y}{\partial \xi}F\right) \tag{2.54}$$

$$R' = \frac{1}{J}\left(\frac{\partial y}{\partial \eta}R - \frac{\partial x}{\partial \eta}S\right) \tag{2.55}$$

$$S' = \frac{1}{J}\left(\frac{\partial x}{\partial \xi}S - \frac{\partial y}{\partial \xi}R\right) \tag{2.56}$$

$$H_i' = \frac{H_i}{J} \tag{2.57}$$

$$H_v' = \frac{H_v}{J} \tag{2.58}$$

The gradients of velocity required for shear stress components are calculated as follows:

$$\frac{\partial u}{\partial x} = \frac{1}{\overline{\rho}}\left[\frac{\partial y}{\partial \eta}\frac{\partial \rho u}{\partial \xi} - \frac{\partial y}{\partial \xi}\frac{\partial \rho u}{\partial \eta} - \overline{u}\left(\frac{\partial y}{\partial \eta}\frac{\partial \rho}{\partial \xi} - \frac{\partial y}{\partial \xi}\frac{\partial \rho}{\partial \eta}\right)\right] \tag{2.59}$$

$$\frac{\partial v}{\partial y} = \frac{1}{\overline{\rho}}\left[\frac{\partial x}{\partial \eta}\frac{\partial \rho v}{\partial \xi} - \frac{\partial x}{\partial \xi}\frac{\partial \rho v}{\partial \eta} - \overline{v}\left(\frac{\partial x}{\partial \eta}\frac{\partial \rho}{\partial \xi} - \frac{\partial x}{\partial \xi}\frac{\partial \rho}{\partial \eta}\right)\right] \tag{2.60}$$

whereas the temperature gradients required for the heat flux components are:

$$\frac{\partial T}{\partial x} = \frac{\partial T}{\partial \xi}\frac{\partial \xi}{\partial x} \tag{2.61}$$

$$\frac{\partial T}{\partial y} = \frac{\partial T}{\partial \xi}\frac{\partial \xi}{\partial y} \tag{2.62}$$

The temperature definition can be given using Equations (2.16) and (2.23) as follows:

$$T = \gamma(\gamma - 1)\left\{\frac{e}{\rho} - \frac{1}{2\rho^2}\left[(\rho u)^2 + (\rho v)^2\right]\right\} \tag{2.63}$$

CHAPTER 3

NUMERICAL ALGORITHM

3.1 GENERAL

In this chapter, the numerical algorithm used for the solution of axisymmetric Navier-Stokes equations is explained. A single step explicit Lax-Wendroff type time marching method is used as the base solver. Finite volume method is used to discretize the governing equations. Application of the finite volume integration yields a "cell residual" for each cell, which represents the change in the conservative variables between the current and previous time steps for that control volume. This second order change of each cell is then distributed to the surrounding nodes of the cell by a distribution formula. While Lax-Wendroff type algorithms are known to have a significant amount of implicit artificial smoothing, transonic or supersonic flows with shocks need additional explicit smoothing procedure for a stable solution. Therefore, a first order smoothing algorithm is used to damp the oscillations during the solution. In order to have a fast convergence, local time stepping technique is employed.

Finally, implementation of boundary conditions for wall, symmetry, injection, inflow and outflow boundaries is explained.

3.2 SOLUTION METHOD

Solution method is a cell-vertex finite volume integration technique of the governing equations. It is mainly based on the Ni' s method [32] for solving Euler equations. To solve the Navier-Stokes equations and axisymmetric problems, viscous flux terms and inviscid and viscous axisymmetric source terms are also implemented into the solution algorithm. For the finite volumes, cell-vertex formulation is used for the Euler equations and Lax-Wendroff method is applied for the time integration. Additional terms due to viscosity and axisymmetricity are added using cell-vertex formulation.

Expressing the conservative variable, *U,* term by using Taylor' s series expansion in time up to second order accuracy, following relation is obtained:

$$U^{n+1} = U^{n} + \Delta t \left(\frac{\partial U}{\partial t} \right)^{n} + \frac{\Delta t^{2}}{2} \left(\frac{\partial^{2} U}{\partial t^{2}} \right)^{n} \tag{3.1}$$

where the superscript '*n*' defines the time from which the iteration is being proceeded. Rearranging Equation (2.1) and defining $H=H_i-H_v$ for simplicity during the discretization procedure

$$\left(\frac{\partial U}{\partial t} \right)^{n} = -\left(\frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} + \frac{\partial R}{\partial x} + \frac{\partial S}{\partial y} + H \right)^{n} \tag{3.2}$$

For convenience, the cell residual 'δU' can be defined as

$$\delta U = U^{n+1} - U^{n} = \Delta t\left(\frac{\partial U}{\partial t}\right)^{n} + \frac{\Delta t^{2}}{2}\left(\frac{\partial^{2} U}{\partial t^{2}}\right)^{n} \tag{3.3}$$

Successively, Equation (3.1) becomes

$$\begin{aligned}\delta U = &-\Delta t\left(\frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} + \frac{\partial R}{\partial x} + \frac{\partial S}{\partial y} + H\right)^{n} \\ &-\frac{\Delta t^{2}}{2}\frac{\partial}{\partial t}\left(\frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} + \frac{\partial R}{\partial x} + \frac{\partial S}{\partial y} + H\right)^{n}\end{aligned} \tag{3.4}$$

Second term in the above equation can also be rewritten as

$$\begin{aligned}\frac{\partial}{\partial t}\left(\frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} + \frac{\partial R}{\partial x} + \frac{\partial S}{\partial y} + H\right)^{n} = &\frac{\partial}{\partial x}\left(\frac{\partial F}{\partial t}\right)^{n} + \frac{\partial}{\partial y}\left(\frac{\partial G}{\partial t}\right)^{n} + \frac{\partial}{\partial x}\left(\frac{\partial R}{\partial t}\right)^{n} \\ &+ \frac{\partial}{\partial y}\left(\frac{\partial S}{\partial t}\right)^{n} + \left(\frac{\partial H}{\partial t}\right)^{n}\end{aligned} \tag{3.5}$$

or

$$\begin{aligned}\frac{\partial}{\partial t}\left(\frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} + \frac{\partial R}{\partial x} + \frac{\partial S}{\partial y} + H\right)^{n} = &\frac{\partial}{\partial x}\left(\frac{\partial F}{\partial U}\frac{\partial U}{\partial t} + \frac{\partial R}{\partial U}\frac{\partial U}{\partial t}\right)^{n} \\ &+ \frac{\partial}{\partial y}\left(\frac{\partial G}{\partial U}\frac{\partial U}{\partial t} + \frac{\partial S}{\partial U}\frac{\partial U}{\partial t}\right)^{n} + \left(\frac{\partial H}{\partial t}\right)^{n}\end{aligned} \tag{3.6}$$

in quasi-linear form. Substituting Equation (3.6) into Equation (3.4), one can obtain

$$\delta U = -\Delta t\left(\frac{\partial F}{\partial x}+\frac{\partial G}{\partial y}+\frac{\partial R}{\partial x}+\frac{\partial S}{\partial y}+H\right)^n - \frac{\Delta t^2}{2}\left[\begin{array}{l}\dfrac{\partial}{\partial x}\left(\dfrac{\partial F}{\partial U}\dfrac{\partial U}{\partial t}+\dfrac{\partial R}{\partial U}\dfrac{\partial U}{\partial t}\right)^n+\dfrac{\partial}{\partial y}\left(\dfrac{\partial G}{\partial U}\dfrac{\partial U}{\partial t}+\dfrac{\partial S}{\partial U}\dfrac{\partial U}{\partial t}\right)^n\\ +\left(\dfrac{\partial H}{\partial t}\right)^n\end{array}\right] \tag{3.7}$$

Separating viscous, inviscid and axisymmetric terms and rearranging Equation (3.7), it is possible to obtain

$$\begin{aligned}\delta U = &-\Delta t\left(\frac{\partial F}{\partial x}+\frac{\partial G}{\partial y}\right)^n - \frac{\Delta t^2}{2}\left[\frac{\partial}{\partial x}\left(\frac{\partial F}{\partial U}\frac{\partial U}{\partial t}\right)^n+\frac{\partial}{\partial y}\left(\frac{\partial G}{\partial U}\frac{\partial U}{\partial t}\right)^n\right]\\ &-\Delta t\left(\frac{\partial R}{\partial x}+\frac{\partial S}{\partial y}\right)^n - \frac{\Delta t^2}{2}\left[\frac{\partial}{\partial x}\left(\frac{\partial R}{\partial U}\frac{\partial U}{\partial t}\right)^n+\frac{\partial}{\partial y}\left(\frac{\partial S}{\partial U}\frac{\partial U}{\partial t}\right)^n\right]\\ &-\Delta t H^n - \frac{\Delta t^2}{2}\left(\frac{\partial H}{\partial t}\right)^n\end{aligned} \tag{3.8}$$

Defining

$$\Delta U_i^{\,n} = -\Delta t\left(\frac{\partial F}{\partial x}+\frac{\partial G}{\partial y}\right)^n \tag{3.9}$$

$$\Delta U_v^{\,n} = -\Delta t\left(\frac{\partial R}{\partial x}+\frac{\partial S}{\partial y}\right)^n \tag{3.10}$$

$$\Delta U_T^{\,n} = \Delta U_i^{\,n} + \Delta U_v^{\,n} - \Delta H^n \tag{3.11}$$

$$\Delta F^n = \left(\frac{\partial F}{\partial U}\right)\Delta U_T^{\,n} \tag{3.12}$$

$$\Delta G^n = \left(\frac{\partial G}{\partial U}\right)\Delta U_T{}^n \tag{3.13}$$

$$\Delta R^n = \left(\frac{\partial R}{\partial U}\right)\Delta U_T{}^n \tag{3.14}$$

$$\Delta S^n = \left(\frac{\partial S}{\partial U}\right)\Delta U_T{}^n \tag{3.15}$$

$$\frac{\partial U}{\partial t} = \frac{\Delta U_T{}^n}{\Delta t} \tag{3.16}$$

$$\frac{\partial F}{\partial U}\frac{\partial U}{\partial t} = \left(\frac{\partial F}{\partial U}\right)\left[\frac{\Delta U_T{}^n}{\Delta t}\right] = \frac{\Delta F^n}{\Delta t} \tag{3.17}$$

Similarly,

$$\frac{\partial G}{\partial U}\frac{\partial U}{\partial t} = \frac{\Delta G^n}{\Delta t} \tag{3.18}$$

$$\frac{\partial R}{\partial U}\frac{\partial U}{\partial t} = \frac{\Delta R^n}{\Delta t} \tag{3.19}$$

$$\frac{\partial S}{\partial U}\frac{\partial U}{\partial t} = \frac{\Delta S^n}{\Delta t} \tag{3.20}$$

Inserting Equation (3.17) to (3.20) into Equation (3.8) formulation, one can obtain

$$\delta U = \Delta U_i^{\,n} - \frac{\Delta t}{2}\left[\frac{\partial}{\partial x}\left(\Delta F^n\right) + \frac{\partial}{\partial y}\left(\Delta G^n\right)\right] + \Delta U_v^{\,n}$$
$$- \frac{\Delta t}{2}\left[\frac{\partial}{\partial x}\left(\Delta R^n\right) + \frac{\partial}{\partial y}\left(\Delta S^n\right)\right] - \Delta t H^n - \frac{\Delta t^2}{2}\left(\frac{\partial H}{\partial t}\right)^n \tag{3.21}$$

Since the viscous terms include second order derivatives of the conservative variables, $\left[\frac{\partial}{\partial x}\left(\Delta R^n\right) + \frac{\partial}{\partial y}\left(\Delta S^n\right)\right]$ term is generally neglected [33], [34], [35], [36] but for a complete derivation, these terms will be retained.

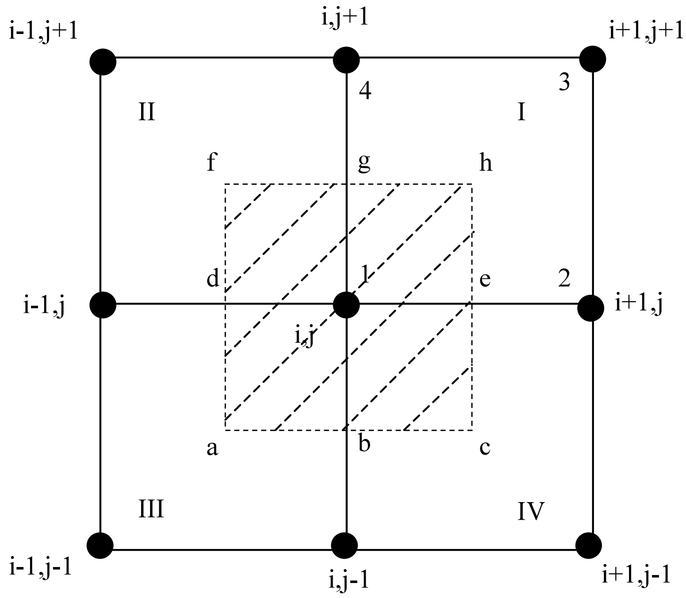


Figure 3.1 Control Volume

Integrating Equation (3.21) over a control volume given as a-c-h-f in Figure 3.1, one can obtain

$$
\begin{aligned}
\iiint \delta U d\forall = & \iiint_{achf} \Delta U_i^{\,n} d\forall - \iiint_{achf} \frac{\Delta t}{2}\left[\frac{\partial}{\partial x}\left(\Delta F^n\right)+\frac{\partial}{\partial y}\left(\Delta G^n\right)\right] d\forall + \iiint_{achf} \Delta U_v^{\,n} d\forall \\
& - \iiint_{achf} \frac{\Delta t}{2}\left[\frac{\partial}{\partial x}\left(\Delta R^n\right)+\frac{\partial}{\partial y}\left(\Delta S^n\right)\right] d\forall - \iiint_{achf} \Delta t H^n d\forall \\
& - \iiint_{achf} \frac{\Delta t^2}{2}\left(\frac{\partial H}{\partial t}\right)^n d\forall
\end{aligned}
\tag{3.22}
$$

Equation (3.22) can be rewritten, expanding U_i and U_v terms and integrating the term on the left hand side, as follows:

$$
\begin{aligned}
\delta U^n = & -\frac{\Delta t}{\Delta V} \iiint_{achf} \left(\frac{\partial F}{\partial x}+\frac{\partial G}{\partial y}\right)^n d\forall - \frac{\Delta t}{2\Delta V} \iiint_{achf} \left[\frac{\partial}{\partial x}\left(\Delta F^n\right)+\frac{\partial}{\partial y}\left(\Delta G^n\right)\right] d\forall \\
& -\frac{\Delta t}{\Delta V} \iiint_{achf} \left(\frac{\partial R}{\partial x}+\frac{\partial S}{\partial y}\right)^n d\forall - \frac{\Delta t}{2\Delta V} \iiint_{achf} \left[\frac{\partial}{\partial x}\left(\Delta R^n\right)+\frac{\partial}{\partial y}\left(\Delta S^n\right)\right] d\forall \\
& -\frac{\Delta t}{\Delta V} \iiint_{achf} H^n d\forall - \frac{\Delta t^2}{2\Delta V} \iiint_{achf} \left(\frac{\partial H}{\partial t}\right)^n d\forall
\end{aligned}
\tag{3.23}
$$

First, third and fifth terms in Equation (3.23) can be handled as four separate volume integrals around the portions of the control volume a-c-h-f. That is, volume integral of a-c-h-f can be written as the summation of integrals of a-b-l-d, b-c-e-l, l-e-h-g and d-l-g-f. Moreover, these can be approximated as one quarter of the volume integral of the cells which Node 1 belongs to. Thus, δU_1 can be expressed as below

$$
\delta U_1 = \delta U_{1,I} + \delta U_{1,II} + \delta U_{1,III} + \delta U_{1,IV}
\tag{3.24}
$$

Here, $\delta U_{1,I}$ is the portion of correction of Node 1 due to the changes taking place in Cell I.

Using Equation (3.23) and applying Gauss' Divergence theorem, $\delta U_{1,I}$ can be rewritten as

$$\begin{aligned}
\delta U_{1,I} = & -\frac{\Delta t}{4\Delta\forall}\oint_{Cell\ I}\left(F\vec{i}_x + G\vec{i}_y\right).(dS_x\vec{i}_x + dS_y\vec{i}_y) \\
& -\frac{\Delta t}{2\Delta\forall}\oint_{lehg}\left[\left(\Delta F\vec{i}_x\right)+\left(\Delta G\vec{i}_y\right)\right].(dS_x\vec{i}_x + dS_y\vec{i}_y) \\
& -\frac{\Delta t}{4\Delta\forall}\oint_{Cell\ I}\left(R\vec{i}_x + S\vec{i}_y\right).(dS_x\vec{i}_x + dS_y\vec{i}_y) \\
& -\frac{\Delta t}{2\Delta\forall}\oint_{lehg}\left[\left(\Delta R\vec{i}_x\right)+\left(\Delta S\vec{i}_y\right)\right].(dS_x\vec{i}_x + dS_y\vec{i}_y) \\
& -\frac{\Delta t}{\Delta\forall}\iiint_{Cell\ I} H d\forall - \frac{\Delta t^2}{2\Delta\forall}\iiint_{Cell\,I}\left(\frac{\partial H}{\partial t}\right)d\forall
\end{aligned} \tag{3.25}$$

where S_x and S_y represent the areas perpendicular to x and y directions, respectively. In this equation, the first, third and fifth terms are first order corrections while the second, fourth and sixth terms are second order corrections. Thus, $\delta U_{1,I}$ can be expressed as the summation of these two terms as below

$$\delta U_{1,I} = \Delta U_I + \Delta\Delta U_{1,I} \tag{3.26}$$

where ΔU_I denoting first order changes and $\Delta\Delta U_{1,I}$ denoting second order changes. Similarly, 1^{st} and 2^{nd} order correction contributions from other neighbouring cells of Node 1 can be defined. These two terms contributing to $\delta U_{1,I}$ are expressed in detail in the below two step.

1st Step:

For the calculation of surface integrals in Equation (3.25), surface vectors are defined according to Figure 3.3. Since the viscous terms contain derivatives of the conservative variables, they are calculated at the cell center and then distributed to the nodes of corresponding cells. Accordingly, first order corrections for the nodes of Cell I can be written as

$$
\begin{aligned}
\Delta U_{1,I} = \frac{\Delta t}{4\Delta\forall}\Big[\; & 0.5(F_1+F_2)(y_2-y_1)-0.5(G_1+G_2)(x_2-x_1) \\
& -0.5(F_3+F_4)(y_3-y_4)+0.5(G_3+G_4)(y_3-y_4) \\
& +0.5(G_1+G_4)(x_4-x_1)-0.5(F_1+F_4)(y_4-y_1) \\
& -0.5(G_2+G_3)(x_3-x_2)+0.5(F_2+F_3)(y_3-y_2)\;\Big] \\
& +\frac{\Delta t}{\Delta\forall}(-\delta r_I-\delta s_I)-\frac{\Delta t}{4\Delta\forall}H_I
\end{aligned}
\tag{3.27}
$$

$$
\begin{aligned}
\Delta U_{2,I} = \frac{\Delta t}{4\Delta\forall}\Big[\; & 0.5(F_1+F_2)(y_2-y_1)-0.5(G_1+G_2)(x_2-x_1) \\
& -0.5(F_3+F_4)(y_3-y_4)+0.5(G_3+G_4)(y_3-y_4) \\
& +0.5(G_1+G_4)(x_4-x_1)-0.5(F_1+F_4)(y_4-y_1) \\
& -0.5(G_2+G_3)(x_3-x_2)+0.5(F_2+F_3)(y_3-y_2)\;\Big] \\
& +\frac{\Delta t}{\Delta\forall}(-\delta r_I+\delta s_I)-\frac{\Delta t}{4\Delta\forall}H_I
\end{aligned}
\tag{3.28}
$$

$$
\begin{aligned}
\Delta U_{3,I} = \frac{\Delta t}{4\Delta\forall}\Big[\; & 0.5(F_1+F_2)(y_2-y_1)-0.5(G_1+G_2)(x_2-x_1) \\
& -0.5(F_3+F_4)(y_3-y_4)+0.5(G_3+G_4)(y_3-y_4) \\
& +0.5(G_1+G_4)(x_4-x_1)-0.5(F_1+F_4)(y_4-y_1) \\
& -0.5(G_2+G_3)(x_3-x_2)+0.5(F_2+F_3)(y_3-y_2)\;\Big] \\
& +\frac{\Delta t}{\Delta\forall}(\delta r_I+\delta s_I)-\frac{\Delta t}{4\Delta\forall}H_I
\end{aligned}
\tag{3.29}
$$

$$
\begin{aligned}
\Delta U_{4,I} = \frac{\Delta t}{4\Delta\forall}\big[\ & 0.5(F_1+F_2)(y_2-y_1) - 0.5(G_1+G_2)(x_2-x_1) \\
& - 0.5(F_3+F_4)(y_3-y_4) + 0.5(G_3+G_4)(y_3-y_4) \\
& + 0.5(G_1+G_4)(x_4-x_1) - 0.5(F_1+F_4)(y_4-y_1) \\
& - 0.5(G_2+G_3)(x_3-x_2) + 0.5(F_2+F_3)(y_3-y_2)\ \big] \\
& + \frac{\Delta t}{\Delta\forall}(\delta r_I - \delta s_I) - \frac{\Delta t}{4\Delta\forall} H_I
\end{aligned}
\tag{3.30}
$$

where

$$\delta r_I = (R_I \Delta y^l - S_I \Delta x^l) \tag{3.31}$$

$$\delta g_I = (S_I \Delta x^m - R_I \Delta x^m) \tag{3.32}$$

This step is a first order approximation to the governing equations on a cell volume whose shape is independent of time.

2nd Step:

This step is the calculation of second order corrections of Node 1 due to the changes taking place within Cell I during time step *Δt*.

A sample cell over which the second order finite volume calculations are carried out is shown in Figure 3.2 together with its surface normal vectors. Node definitions are the same for each cell since a pointer system is used to define the whole mesh structure.

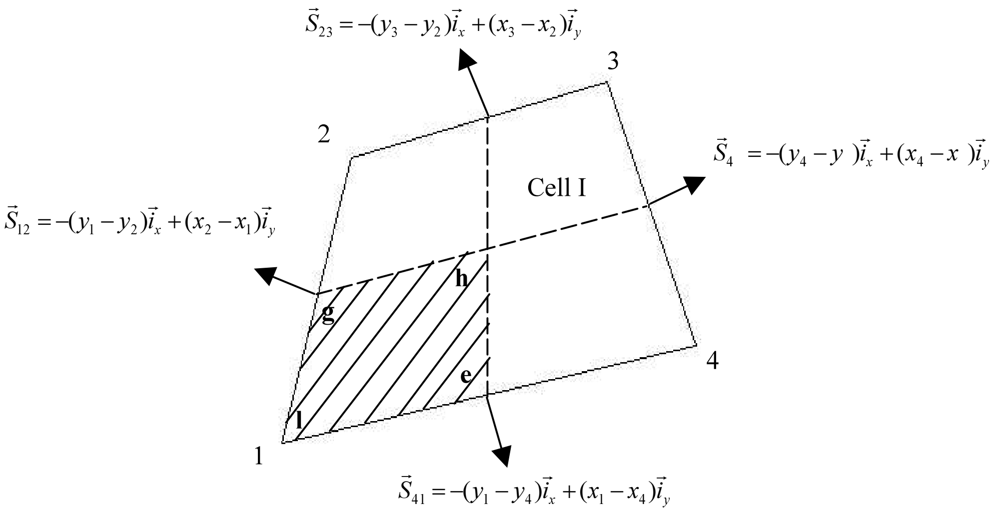


Figure 3.2 Sample cell for finite volume calculations

The second order changes are due to fluxes passing through l-g and l-e of Figure 3.3 for Node 1. Second order corrections in Equation (3.25) can be discretized as below

$$\Delta\Delta U_{1,I} = \frac{\Delta t}{2\Delta\forall}(-\delta f_I - \delta g_I) - \frac{\Delta t}{8\Delta\forall}\Delta H_I \tag{3.33}$$

where

$$\delta f_I = (\Delta F_I \Delta y^l - \Delta G_I \Delta x^l) \tag{3.34}$$

$$\delta g_I = (\Delta G_I \Delta x^m - \Delta F_I \Delta x^m) \tag{3.35}$$

and

$$\Delta F_I = \left(\frac{\partial F}{\partial U}\right)_I \Delta U_I \tag{3.36}$$

$$\Delta G_I = \left(\frac{\partial G}{\partial U}\right)_I \Delta U_I \tag{3.37}$$

$$\Delta H_I = \left(\frac{\partial H}{\partial U}\right)_I \Delta U_I \tag{3.38}$$

$$\Delta x^l = \frac{1}{2}(x_2 + x_3 - x_1 - x_4) \tag{3.39}$$

$$\Delta x^m = \frac{1}{2}(x_3 + x_4 - x_1 - x_2) \tag{3.40}$$

$$\Delta y^l = \frac{1}{2}(y_2 + y_3 - y_1 - y_4) \tag{3.41}$$

$$\Delta y^m = \frac{1}{2}(y_3 + y_4 - y_1 - y_2) \tag{3.42}$$

One should note that while $\frac{\partial H}{\partial t}$ is approximated as $\frac{\Delta H}{\Delta t}$ and used accordingly in Equation (3.33) rewritten during the discretization of second order corrections of Equation (3.25).

For nodes 2,3 and 4 of Cell I, corrections are different than the one given in Equation (3.33) because of different second order changes.

$$\Delta\Delta U_{2,I} = \frac{\Delta t}{2\Delta\forall}(-\delta f_I + \delta g_I) - \frac{\Delta t}{8\Delta\forall}\Delta H_I \tag{3.43}$$

$$\Delta\Delta U_{3,I} = \frac{\Delta t}{2\Delta\forall}(\delta f_I + \delta g_I) - \frac{\Delta t}{8\Delta\forall}\Delta H_I \tag{3.44}$$

$$\Delta\Delta U_{4,I} = \frac{\Delta t}{2\Delta\forall}(\delta f_I - \delta g_I) - \frac{\Delta t}{8\Delta\forall}\Delta H_I \tag{3.45}$$

Derivative terms included in the formulation of flux terms (*ΔF*, *ΔG* and *ΔH* terms) appeared above are called Jacobians. Calculation of inviscid and axisymmetric source term Jacobians are given in Appendix A.

Now that both first and second order corrections are calculated for all nodes of Cell I, total correction at these nodes can be found as

$$\delta U_{1,I} = \Delta U_{1,I} + \Delta\Delta U_{1,I} \tag{3.46}$$

$$\delta U_{2,I} = \Delta U_{2,I} + \Delta\Delta U_{2,I} \tag{3.47}$$

$$\delta U_{3,I} = \Delta U_{3,I} + \Delta\Delta U_{3,I} \tag{3.48}$$

$$\delta U_{4,I} = \Delta U_{4,I} + \Delta\Delta U_{4,I} \tag{3.49}$$

Finally, after the calculation and distribution steps to every node of every cell and boundary condition implementation, the solution is updated at each node according to the following procedure for a sample node, Node 1, referring to Figure 3.1:

$$U_1^{\,n+1} = U_1^{\,n} + \delta U_{1,I} + \delta U_{1,II} + \delta U_{1,III} + \delta U_{1,IV} \tag{3.50}$$

The solution procedure in the code can be summarized briefly as below:

Before the solution process, the mesh is swept by node initializing all cell residuals to zero (i.e. δU=0).

After the initiation, first and second order changes are calculated and added to find the residual term sweeping all the nodes along the domain.

After the contributions to the nodes are calculated and added, boundary condition implementation and updating of boundary cell residuals completes the solution cycle for a single time step.

3.3 NUMERICAL SMOOTHING

Most of the explicit formulations like Lax-Wendroff scheme require additional artificial smoothing to filter the non-physical oscillations near discontinuities. The necessity arises in two ways. First, the grid resolution or quality may not be sufficient to resolve the physics of the flow in discrete domain. Second, the damping mechanism of the numerical method may be insufficient to damp the unnecessary numerical oscillations especially in the inviscid regions where the viscous dissipation vanishes. While the presence of numerical smoothing is essential, it should be kept as small as possible to minimize degradation of the solution accuracy.

For the present algorithm, below smoothing mechanism is used in the solutions after the distribution of cell changes to the nodes. The smoothing procedure can be applied as follows to each node. For example, referring to Cell I of Figure 3.1:

$$\delta U_1 = \delta U_1 + \frac{1}{4}\mu(U_I - U_1) \tag{3.51}$$

$$\delta U_2 = \delta U_2 + \frac{1}{4}\mu(U_I - U_2) \tag{3.52}$$

$$\delta U_3 = \delta U_3 + \frac{1}{4}\mu(U_I - U_3) \tag{3.53}$$

$$\delta U_4 = \delta U_4 + \frac{1}{4}\mu(U_I - U_4) \tag{3.54}$$

where

$$\mu = \sigma \frac{\Delta t(\Delta l + \Delta m)}{\Delta V} \tag{3.55}$$

with

$$\Delta l = \sqrt{(\Delta x^l)^2 + (\Delta y^l)^2} \tag{3.56}$$

$$\Delta m = \sqrt{(\Delta x^m)^2 + (\Delta y^m)^2} \tag{3.57}$$

and

$$U_I = \frac{1}{4}(U_1 + U_2 + U_3 + U_4) \tag{3.58}$$

where σ is the artificial viscosity coefficient which depends on the problem considered and can be specified by trial-and-error and U_I is the average of the conservative variables at the four nodes of a cell.

This completes the formulation of the solver on the solution domain except the calculation of time step for each cell.

3.4 TIME STEP CALCULATION

The time step limitation is determined by the stability limit of Ni scheme applied to the governing equations. Unfortunately, since the governing equations are non-linear, a stability analysis can not be done directly for this system. Therefore, CFL number is not calculated during the calculations but it is specified by noting the condition CFL≤1.

For the time step calculation, inviscid stability limit formula is used as given below:

$$\Delta t = CFL \cdot \min\left\{\frac{\Delta V}{\left|u\Delta x^{l} - v\Delta y^{l}\right| + a\Delta l}, \frac{\Delta V}{\left|u\Delta x^{m} - v\Delta y^{m}\right| + a\Delta m}\right\} \tag{3.59}$$

where $CFL \leq 1$.

No modification is done for viscous solutions since the time step limitation of equation (3.59) provide enough stability with proper CFL number usage.

In order to accelerate convergence, a new time step is calculated for each cell locally. This step is called as the local time step. If accuracy in time is needed without considering the convergence rate, a single minimum time step is used in every cell of the whole domain. This time step is called as global time step. The global time step usage is more appropriate for identical mesh elements. In this thesis work, all the computations are made using local time step since different mesh sizes are used in either each block or between the blocks.

CHAPTER 4

INITIAL AND BOUNDARY CONDITIONS

Boundary conditions have considerable importance for any numerical solution since the flow's limits are defined by means of boundary conditions. Besides, stability and convergence rate of the numerical solution also depends on the implementation of the boundary conditions. Consequently, proper boundary condition treatment is a strongly required condition for a proper numerical solution.

To start the numerical algorithm, a set of initial conditions should be given as input to the computer program so that the solution procedure initiates. Better initial conditions doubtlessly decrease the convergence time for the solution. In all of the test cases, the flow is initialized by using the inflow conditions of the internal flow, which can be counted as a convenient initiation.

In this chapter, boundary conditions that are implemented to the solution will be explained. These conditions can be classified in five subtitles as; injecting (inflow) boundary condition, symmetry axis boundary condition, solid wall boundary condition for inviscid flow, solid wall boundary condition for laminar flow and exit (outflow) boundary condition.

4.1 INJECTION AND INFLOW CONDITION

4.1.1 INJECTION BOUNDARY CONDITIONS

In solid rocket motors, generally there exists an empty cylindrical channel section, which is surrounded by the propellant that is to burn and flow through the empty channel. Through this process, propellant is continuously enters the empty channel, just as if it is injected. To simulate this condition of injection, injection boundary condition is implemented on the numerical algorithm. By this implementation, a rocket motor with grain burnback is simulated at a certain instance which have specific injection conditions (mass flux and injection temperature most generally).

Injecting boundary condition implementation starts with the assumption of zero pressure gradient at the injecting surface, that is

$$p_{inj} = p_{inj-1} \tag{4.1}$$

with p_{inj} being the pressure value at the point neighbouring the boundary in the normal direction and p_{inj} being the pressure value of the boundary itself.

To find injection density, following equations are used:

$$T_o = T + \frac{V^2}{2C_p} \tag{4.2}$$

$$p = \rho RT \tag{4.3}$$

$$T_o = \frac{p_{inj}}{\rho_{inj} R_{inj}} + \frac{(G/\rho_{inj})^2}{2C_{p_{inj}}} \tag{4.4}$$

where T_o is the flame temperature, G is the injection mass flux both being the input parameters for the injection flow.

The only unknown in the above equation is the injection density. Thus, from these given equations, injection density can be calculated by solving a second-order polynomial equation with one unknown. Solid propellant is burning over the surface in the normal direction to the injection surface, therefore tangential velocity is taken as zero [37] Thus, mass flux divided by the calculated density gives the normal velocity component, while the tangential velocity component is enforced to be zero.

$$\vec{V}_{n_{inj}} = \frac{G}{\rho_{inj}} \tag{4.5}$$

4.1.2 INFLOW BOUNDARY CONDITIONS

For the flat plate and pipe flow solutions, injection boundary conditions are not used but external flow boundary conditions are used. Inflow boundary conditions are computed from the known values of the free stream flow which are free stream Mach number and free stream angle.

4.2 SYMMETRY AXIS BOUNDARY CONDITION

For the boundary conditions on the symmetry axis, a simple method is applied without having to solve any of the governing equations. In this method, velocity component perpendicular to the boundary is enforced to be zero and all other flow variables are equated to the value of neighbouring non-boundary point (that is in the normal direction to the boundary).

$$
\begin{aligned}
&V_n = 0 \\
&\frac{\partial V_t}{\partial \vec{i}_n} = 0 \\
&\frac{\partial \rho}{\partial \vec{i}_n} = 0 \qquad (4.6) \\
&\frac{\partial e}{\partial \vec{i}_n} = 0
\end{aligned}
$$

where the subscript "n" denotes normal direction, "t" denotes the tangential direction and "i" is the unit normal vector.

4.3 SOLID WALL BOUNDARY CONDITION FOR THE INVISCID FLOW

For an inviscid flow, there is no flux through the surface, that is, normal component of the velocity is zero. To find the tangential velocity, total velocity component of neighbouring interior point is projected to the wall surface. Density and pressure values are directly interpolated from the interior region. This formulation shows similarity with the symmetry axis boundary condition because of zero viscosity.

4.4 SOLID WALL BOUNDARY CONDITION FOR VISCOUS FLOW

For the viscous flow, boundary conditions for a solid wall are called the "no-slip" condition, where both the normal and tangential components of the velocity vector is set to zero, i.e. relative velocity between the wall and the fluid is zero. The wall temperature or the heat flux at the wall can be specified as

$$T = T_{wall} \tag{4.7}$$

or

$$-k\frac{\partial T}{\partial n} = \frac{\partial T_{wall}}{\partial n} = q_{wall} \tag{4.8}$$

In this work, the wall is assumed to be adiabatic, i.e. q_{wall}=0. So that Equation (4.8) becomes

$$\frac{\partial T}{\partial n} = 0 \tag{4.9}$$

Besides, momentum equation for the solid wall with no-slip boundary condition reduces to such a form that pressure gradient becomes zero. Thus, the pressure value can be taken from the interior neighbouring points.

$$\frac{\partial p}{\partial n} = 0 \qquad \Rightarrow \qquad p_{wall} = p_{wall-1} \tag{4.10}$$

Using Equations (4.8) and (4.9), it appears that $\rho_{\mathrm{wall}} = \rho_{\mathrm{wall}-1}$. With these equations implemented into the solver, solid wall boundary conditions for the viscous flow is defined.

4.5 EXIT (OUTFLOW) BOUNDARY CONDITIONS

In solid propellant rocket motors, a supersonic diverging-converging nozzle should exist in order to obtain high momentum. Thus, exit boundary conditions are always supersonic for a solid propellant rocket motor. But to be general, the computer program can handle both supersonic and subsonic exit conditions. Test case 1 of reference [25], which will be explained in the next chapter, has subsonic exit flow which allows to verify the subsonic exit boundary conditions.

4.5.1 SUBSONIC OUTFLOW

For subsonic outflow, the flow should reach an ambient pressure where the code converges, that is, the flow interacts with the ambient conditions. Thus, pressure value is enforced to the ambient pressure at outflow boundary, and other parameters should be interpolated from interior points.

$$
\begin{aligned}
p_{exit} &= p_{ambient} \\
\vec{V}_{exit} &= 2\vec{V}_{exit-1} - \vec{V}_{exit-2} \qquad (4.11) \\
\rho_{exit} &= 2\rho_{exit-1} - \rho_{exit-2}
\end{aligned}
$$

4.5.2 SUPERSONIC OUTFLOW

For supersonic flow, contrary to the subsonic flow, outflow conditions do not interact with ambient conditions, i.e. they are independent of the region that they are exposed to. Thus, all the exit flow parameters are interpolated from interior points.

$$p_{exit} = 2p_{exit-1} - p_{exit-2}$$

$$\vec{V}_{exit} = 2\vec{V}_{exit-1} - \vec{V}_{exit-2} \tag{4.12}$$

$$\rho_{exit} = 2\rho_{exit-1} - \rho_{exit-2}$$

4.6 SMOOTHING FORMULATION AT THE BOUNDARIES

In addition to physical boundary condition implementations, special care must be taken at the boundaries to apply the smoothing procedure. Smoothing procedure for a specific cell for the main solver requires the existence of 4 nodes of the cell, which is not applicable for the boundary points since there will always be missing nodes at the boundaries. However, a one-dimensional smoothing procedure clears out this ambiguity. Considering the boundary cell shown in Figure 4.1, the smoothing formulation is

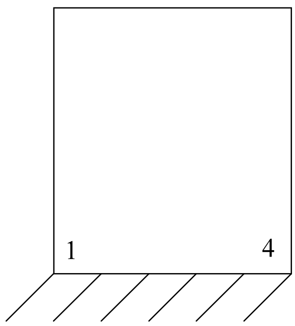


Figure 4.1 Sample boundary cell for smoothing formulation

$$\delta U_{I} = \delta U_{1} + \frac{1}{4}\mu(U_{4} - U_{1}) \tag{4.13}$$

$$\delta U_{I} = \delta U_{1} + \frac{1}{4}\mu(U_{4} - U_{1}) \tag{4.14}$$

where

$$\mu = \sigma \frac{\Delta t(\Delta l + \Delta m)}{\Delta V} \tag{4.15}$$

$$\Delta l = \sqrt{\Delta x^{l\,2} + \Delta y^{l\,2}} \tag{4.16}$$

$$\Delta m = \sqrt{\Delta x^{m\,2} + \Delta y^{m\,2}} \tag{4.17}$$

CHAPTER 5

POINTER SYSTEM AND MULTI-BLOCK ALGORITHM

The solver, as it is explained before, includes a pointer system for the definition of mesh structure and boundaries. Also, a multi-block solver system is adapted into the code to make the solver capable of solving different domains inside a single solver, adding a great capability of solving complex domains easier for the current solver. In this chapter, both the pointer system and multi-block algorithm are explained, by using the same definitions that are used in the code, in order to introduce the solver algorithm more closely.

5.1 POINTER SYSTEM

When a problem is to be solved, all solvers must first recognize the geometry and properties of the geometry (i.e. location and type of the boundary conditions, location of nodes) that is to be solved before the application of the solution algorithm. This can either be done by defining the structure inside the solver or by another program that only defines the geometry and gives required information to the solver in a single file. The former is not as flexible as the latter, since when the geometry is defined as an integral part of the code, new geometries or even modifications in the same geometry needs the modification of the solver for each geometry. Also, different element shapes can not be handled without an unstructured mesh structure which requires a pointer algorithm. In the latter case, the general solver stays the same, no matter how different or complex is the geometry, the solver acts like a big subroutine, taking the required geometry information from the pointer file and solving it appropriately.

Besides, the multi-block algorithm, which is implemented to the solver has different domains with different physical properties together with the different boundary conditions. These many blocks can be defined one by one in the pointer program as single cases, thus the preparation time for a new problem with N blocks is much less than that of a program that carries out the meshing processes in it.

So, the suitable solution for the recognition of mesh structure by the solver is a pointer system, which defines the mesh structure in a way similar to that of a connectivity array which defines the mesh structure for a finite element system. Using the pointer algorithm, changes in the grid structure will change the pointer system while remaining the solver unchanged.

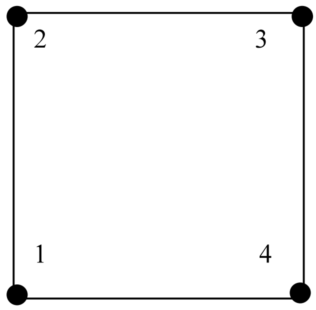
When the solution shows that one or more blocks has inappropriate or coarse grids, it is quite easy to change that particular block or blocks and leaving the other ones as they are. Thus, somehow an adaptive mesh system can be implemented with the absolute control of the user.

5.1.1 THE 2-D CELL POINTER SYSTEM

The pointer system used for the solver can be explained as follows. A two dimensional array is assigned to every cell in the domain in the following form:

IP = IP[m,n]

where n denotes the cell number itself, m denotes cell node number of the n^{th} cell. Maximum value of m is 4 since every cell used in the grid consists of 4 nodes, shown in the Figure 5.1.



gure 5.1 Node numbering for a sample cell

The structure of pointer system is not altered for the multi-block application, to maintain its simplicity.

By defining this IP array, all the points inside the domain are defined in a single file, with the cells including them and coordinates of them except the boundary points. Boundary points are indexed with different names in the solver according to their types. The following arrays are used for each type of the boundary, having different names:

IPBB = IPBB [i,j]   (stands for solid wall boundary)

IPBU = IPBU [i,j]   (stands for far field boundary)

IPBTS = IPBTS [i,j]   (stands for symmetry axis boundary)

IPBINJ = IPBINJ [i,j]   (stands for mass injection boundary)

where j represents the boundary point number. Index i is used for two purposes for the boundary arrays, that is: for i = 1 and i = 2, i represents the cell number of the j^{th} boundary, i = 3 represents the orientation of the cells 1 and 2. For better understanding, four possible boundary cell orientations are given in Figures 5.2, 5.3, 5.4 and 5.5. After all the IP values are assigned in the pointer program, according to their physical positions, boundaries are defined one by one. Also, since the given names for the boundary points are unique for the main solver and pointer program, it is very easy to follow the iterations. Besides, assigning different boundary conditions to the desired boundaries can be rapidly accomplished.

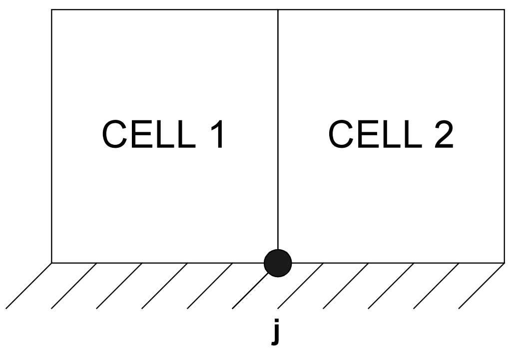


Figure 5.2 Boundary type 1

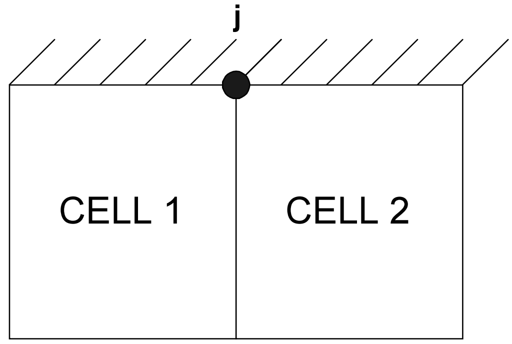


Figure 5.3 Boundary type 2

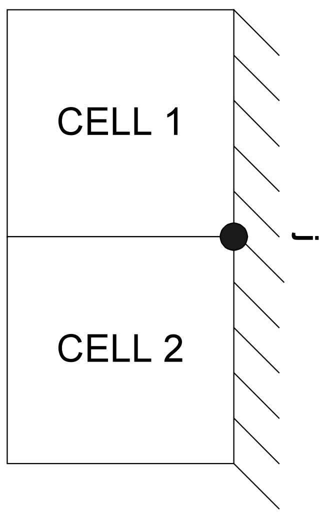


Figure 5.4 Boundary type 3

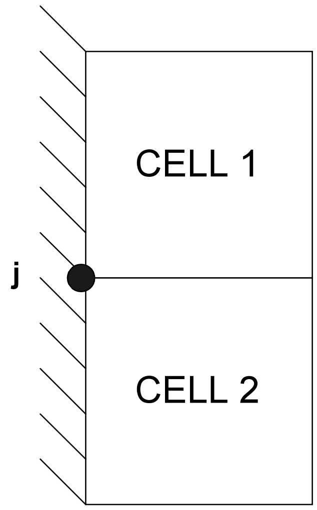


Figure 5.5 Boundary type 4

By the help of internal points and boundary descriptions, whole domain (or domains) is defined in such a way that the main solver can make computations in the domain. This is a very flexible approach for dealing especially with complex grids. All the calculations and applications inside the main solver is applied according to this system without any extra data requirement for the main solver.

5.2 MULTI-BLOCK ALGORITHM

5.2.1 INTRODUCTION

Before starting, it should be noted that, the code is already capable of solving single block, conventional internal mesh structures without requiring any multi-block algorithm. Besides, after the addition of the multi-block algorithm, the code is still capable of solving single domains. The main goal, which is explained also in Chapter

1, is to create independent mesh structures for a desired number of subdomains of a problem, thus eliminate the difficulty in creating meshes for complex geometries.

To explain the multi-block system clearly, first, the arrays that define the flow parameters used in the solver will be explained.

In the formulation of the solver, U and δU values must be stored for the present time step at each grid point in order to obtain a complete and converged solution. Thus, including the coordinates of a two dimensional domain, the array must consist of 10 rows to include all these information.

$$Q=Q[i,j]$$

where i denotes the variable type and j denotes the node number. These 10 rows of the Q array are given as follows respectively from i=1 to i=10 : x, y, ρ, ρu, ρv, e, d(ρ), d(ρu), d(ρv), d(e). The first two variables are for coordinates of the j^{th} node, next four is for conservation variables of the same node and the last four for the changes in the conservation variables of the same node. With this Q array, all the required flow parameters can be obtained at any point in the computational domain. Therefore, in the multi-block algorithm, interchange of this Q array in a proper algorithm is quite enough to solve a whole multi-blocked computational domain with the given boundary conditions.

Given the information about the equations used in the solver together with the numerical algorithm used in the solver so far, now the multi-block algorithm can be explained.

5.2.2 IMPLEMENTATION OF MULTI-BLOCK ALGORITHM

As it is explained in the previous chapters, the solver has the capability of solving different subdomains (blocks) as a single domain while preserving their independency in terms of mesh size, distribution and flow parameters without making any modifications in the main solver The multi-block algorithm needs a different input file for each domain so that even completely different problems can be combined if the result is applicable and logical.

If one wants to solve a specific problem in N blocks, N mesh files and N input files should be given as inputs to the code. The code first solves Block-1 as the first block, for a single time step, then consecutive blocks are solved until Block-*N*, which is the last block, is solved for again a single time step. While this process is running, appropriate boundary conditions are implemented to each block, either conventional ones or boundary conditions that are coming from previous blocks, which will be explained in detail later. Then, after the solution of *N* blocks, the procedure repeats itself until convergence is reached for every block. For a single block, the solution algorithm is given below:

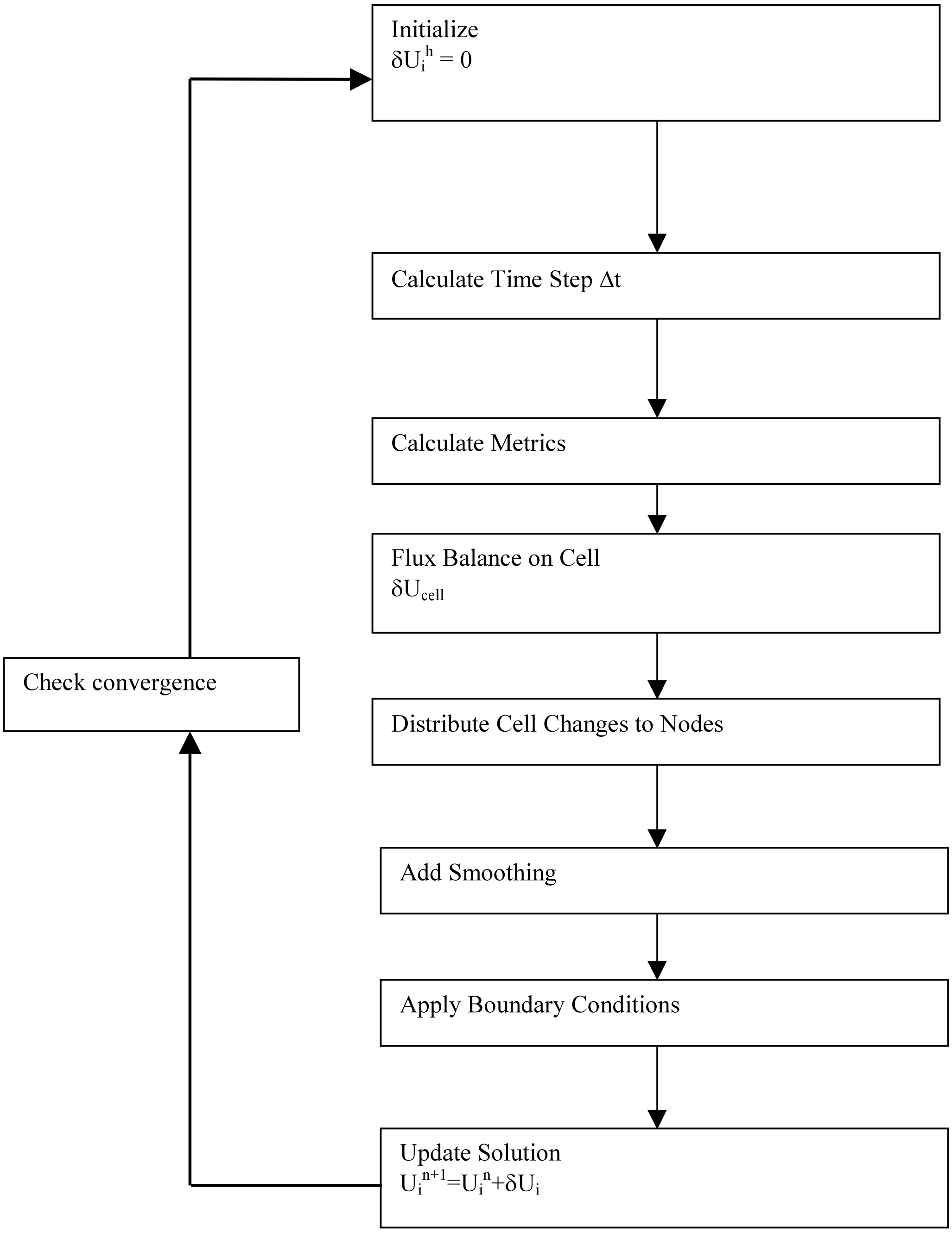


Figure 5.6 Solution algorithm of a single block

Of course, this block division process introduces a problem of matching one domain to another since there are no pre-defined boundary types for interface surfaces.

The basic idea behind solving multi-block problems is making the blocks interact (communicate) with each other in such a way that the final solution represents one complete problem as if it is solved as a single block problem.

To explain the matching process at block interfaces, a simple example consisting of two domains with uniform rectangular cells of different sizes in x-direction is given in as shown in Figure 5.7.

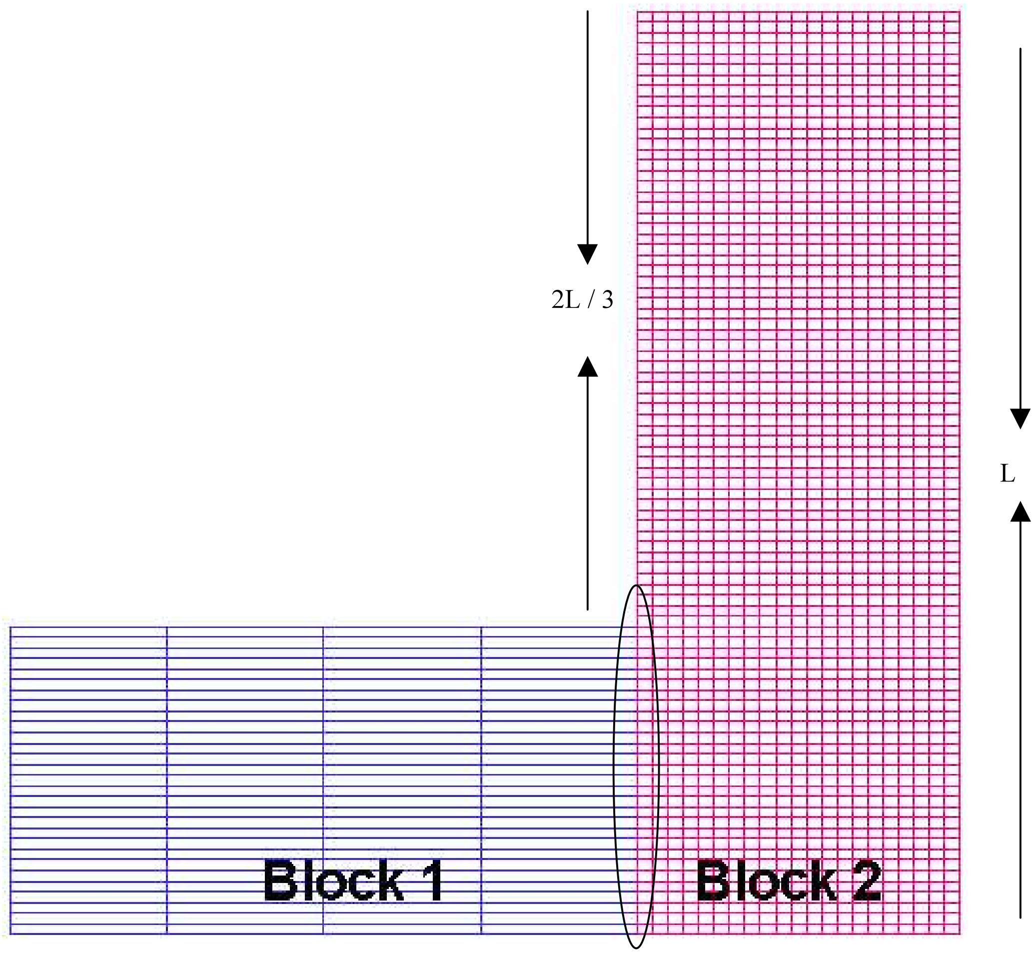


Figure 5.7 Sample two-blocked geometry

When the common surface of two subdomains is zoomed, (the region inside the elliptic circle of Figure 5.7), mesh structure at the interface is as shown in Figure 5.8

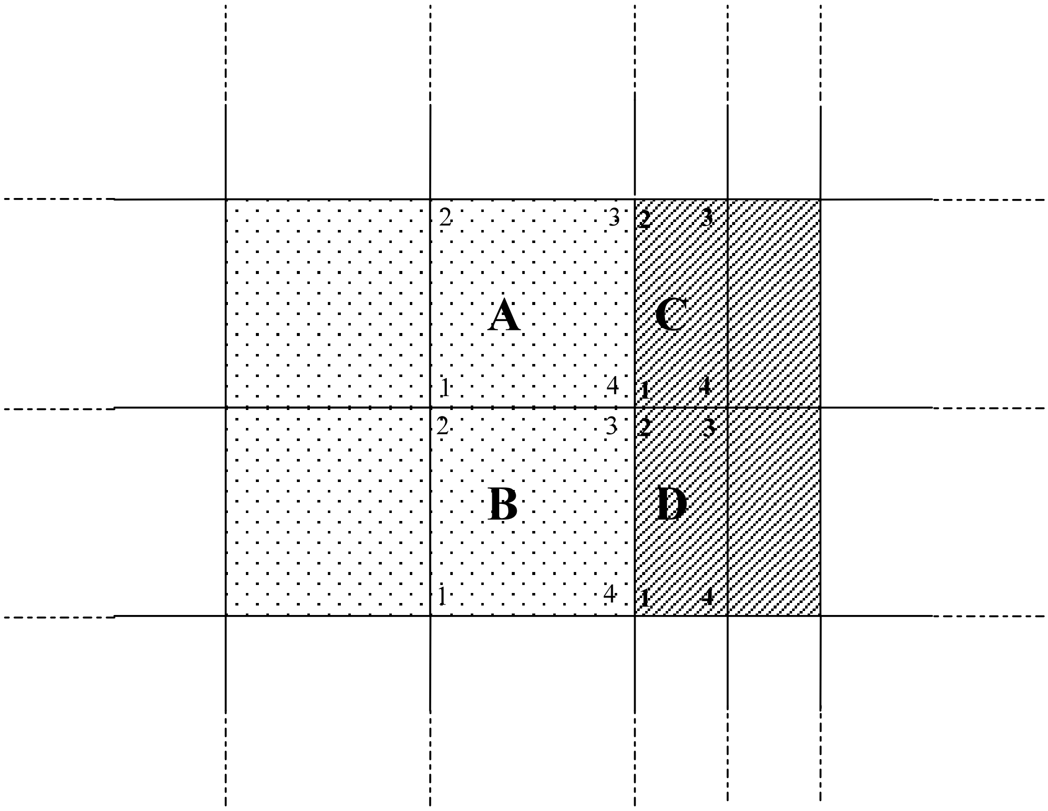


Figure 5.8 Sample cells on the interface of two blocks

Consider four neighbouring cells which of two belong to one block (A&B, in the dotted region) and the other two (C&D, in the cross-hatched region) belong to the other.

For Block-1, left (1-2), bottom (1-4) and top (2-3) faces can be defined as what they simulate without any difference in definition or application. But for the right face (3-4), it is obvious that any predefined boundary conditions will be inconvenient. Therefore, a new type of boundary condition must be defined for this faces. The same situation is valid for Block-2' s left face interacting with the right face of the other block this time. To obtain a converged solution, these two faces are expected to have the same value at the final stage.

Basically, when the solution converges, below statements should be reached in order to obtain a smooth and continuous interface transition between the blocks.

Q(K,A3,Block1) = Q(K,C2,Block2)

Q(K,A4,Block1) = Q(K,C1,Block2)

Q(K,B3,Block1) = Q(K,D2,Block2)

Q(K,B3,Block1) = Q(K,D1,Block2)

where K is flow parameter index for ρ, ρu, ρv, ρe.

Now, to obtain these equalities, the algorithm that is applied inside the solver will be explained.

First of all, to remove any discontinuity at the interface nodes as convergence is achieved, these two faces should interact with each other such that:

Block-1' s right face is forced to be the inflow boundary condition for Block-2 when Block-2 is being solved.

Block-2' s left face is the outflow boundary condition for Block-1 when Block-1 is being solved.

Thus, during the solution of each block, the nodes at the interface boundaries do not go under a special boundary treatment (like solid wall, symmetry axis etc.), but instead they are left with the calculated results coming from distribution step of the previous block.

This 2-block example can be generalized such that any physically interacting N blocks can be solved, with the given flow chart & equations. Note that this chart is the same chart as the chart given in Figure 5.1 if a single block is solved.

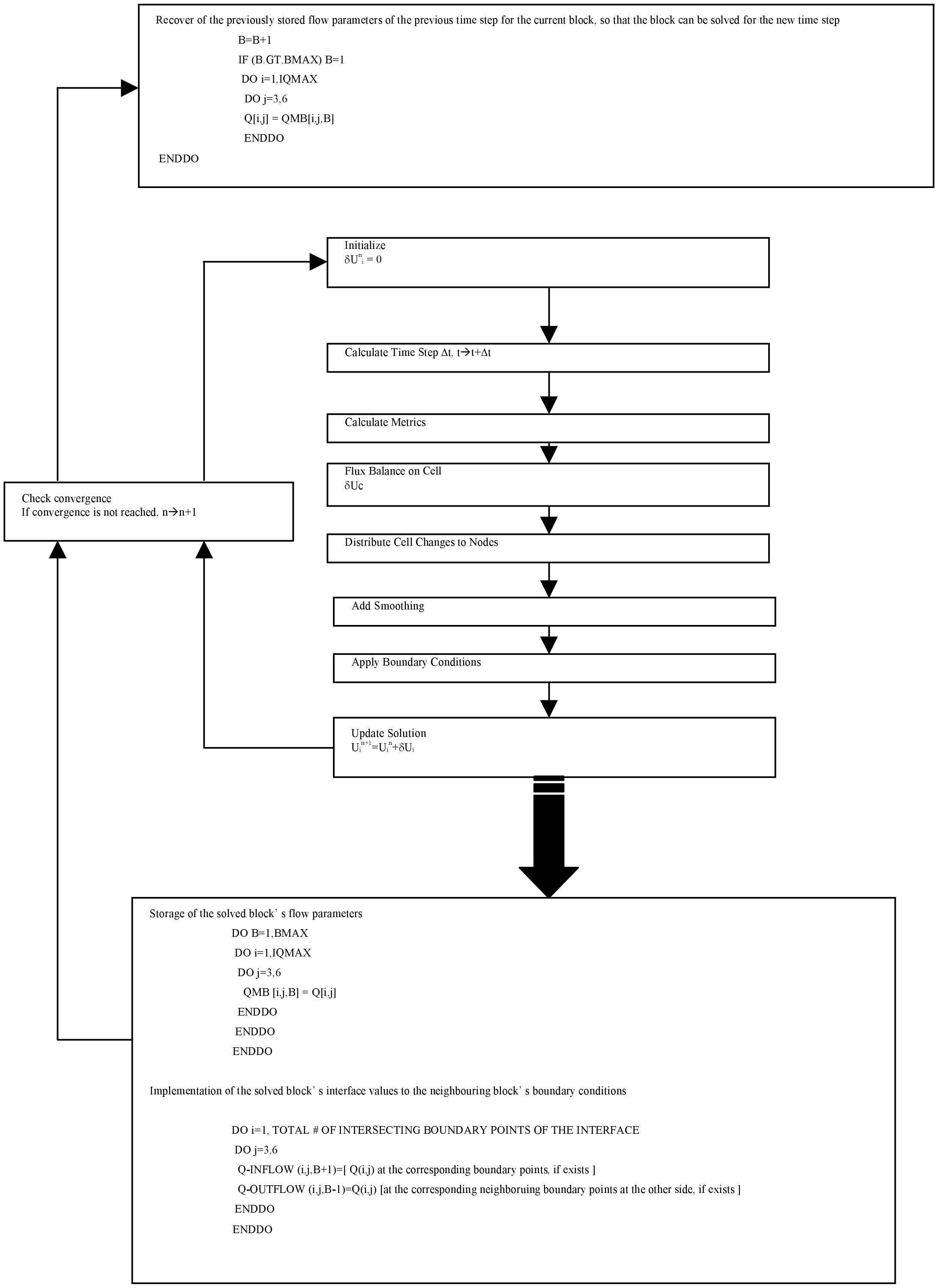


Figure 5.9 Flowchart of multi-block algorithm

where i represents the node number, j represents the flow parameter (ρ, ρu, ρv, ρe), QMB is used to store the parameters of the solved block, B is used for the block number and Q is the general flow parameter array.

When convergence is reached, the values of intersecting nodes which are at the exact same location but belonging to different are expected to have the same value, such that there is no discontinuity through the interface. When this is achieved, it can be said that multi-block algorithm works correctly, but of course general convergence must be checked.

In solving multiple number of blocks together, there exists a problem in the distribution step of the solver algorithm for some special interface arrangements. This can be shown in Figure 5.10, an enlarged appearance of the intersection region of two blocks shown in Figure 5.7 .

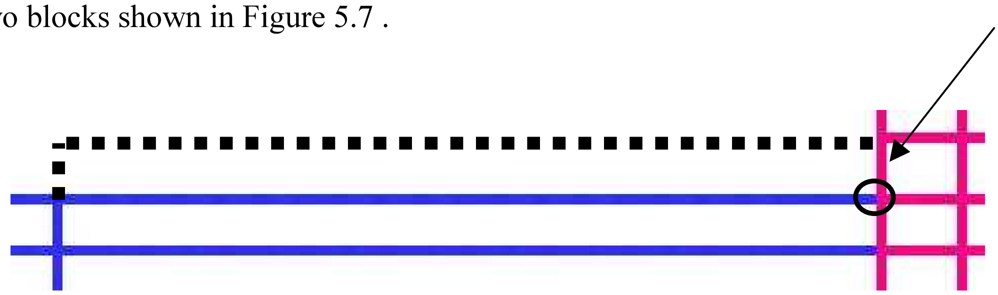


Figure 5.10 Enlarged view of the intersection of two cells on the corner

As it can be seen, in the distribution step, one cell is missing (the cell drawn in dashed line, which is not there actually) to complete the distribution process of the node shown in the circle. So, special attention is needed for this specific cell. On can note that this situation is for the cases where the specified missing cell situation is present if the corner node is not a boundary node. If it is a boundary node, the modification explained below is not applied.

Considering equations (3.26-3.29), δU calculation of this specific node, named here as δU^*, will be as follows:

$$\delta U^* = \delta U^* + \frac{1}{3}(\delta U_c - \delta f_c - \delta g_c - \delta r_c - \delta s_c - \frac{\Delta t}{2}\Delta H_c) \tag{5.1}$$

to compensate the lack of one node in the distribution step. Of course, this kind of situation can occur in different orientations than this block interface. All the possibilities for this special condition is given in Figures (5.10) in circles.

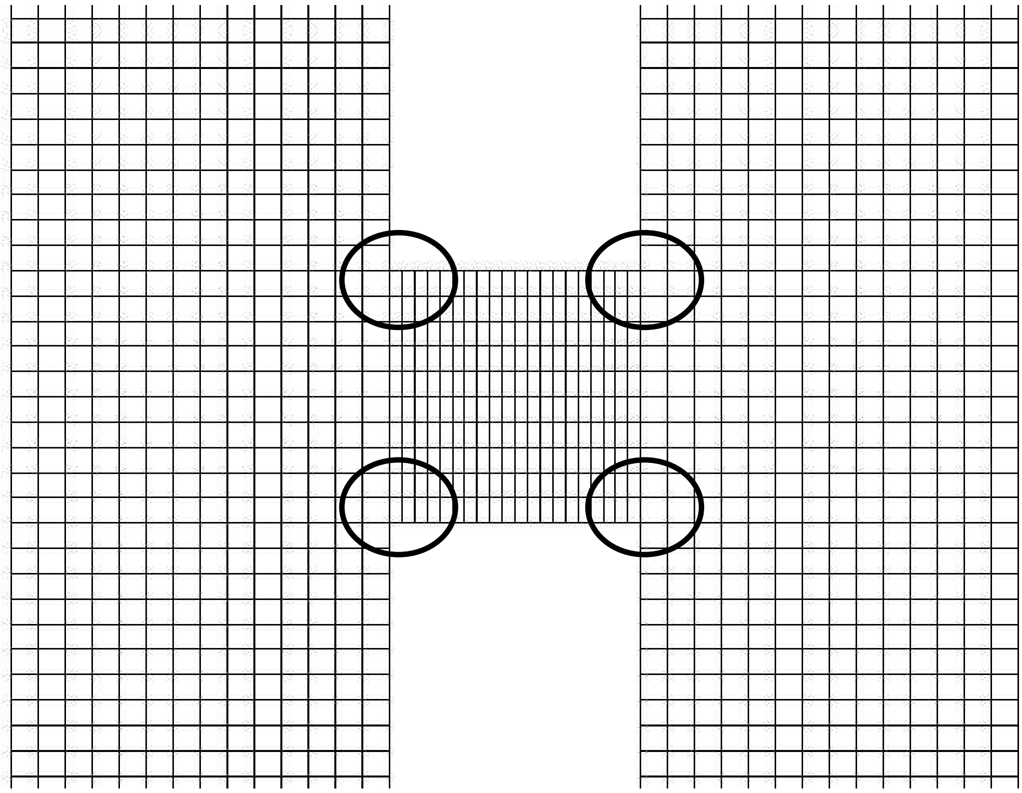


Figure 5.11 Possible intersection points subject to Equation (5.1)

Equation (5.1) holds for all these four possibilities and calculates the true δU value of these special nodes.

CHAPTER 6

RESULTS AND DISCUSSION

This chapter is dedicated to the computational results of various test cases which are used to validate the present multi-block Navier-Stokes solver. Test cases used are a superposition of different flow situations for the verification of the internal/external 2-D planar/axisymmetric multi-block Navier-Stokes solver.

Simulation on a computer is done to design, operate or investigate existing or proposed processes (or systems) and to determine their performance under various conditions [38]. Especially in missile industry, during the design of a rocket motor or modification of an existing configuration need some experiments of validation of the new configuration in the most efficient procedure. For this purpose, although experiments can provide quite useful information about the rocket motor, they are either quite expensive or sometimes impossible or very restricted due to substructural problems. Besides, safety requirements may sometimes prevent the designer from performing experiments. In this situation, simulation by a computer program is the best choice, if there exists a validated computer code. The practitioners and users of simulation results, the decision makers arriving at decisions based on these results, and those affected by these decisions are all justly concerned with whether the results are credible or whether the level of credibility of the results is acceptable for the purposes for which they are being used [38].

The main objective of this study is to develop a reliable Navier-Stokes solver to be used in SPRM flow simulations. For these purposes, result of RTO-AVT T-108 support project [25] is used for validation of SPRM simulation by the present solver. Since this project includes only inviscid flow applications, Blasius flat plate solution is used to verify the planar laminar flow solutions. For the verification of the axisymmetric laminar flow solution, theoretical velocity profile in a pipe flow is used. [28] For the verification of turbulent viscosity calculation, turbulent 1/7 power law is used. After the single-block results are presented, the same test cases are solved by multi-block algorithm.

To check the flexibility of the multi-block algorithm, a segmented SPRM, which is frequently used for large SPRM' s, is solved finally.

6.1 SINGLE BLOCK SOLUTIONS OF T-108 TEST CASES

6.1.1 TEST CASE 1

This test case is not a SPRM simulation but it is used in reference [25] to verify the injecting boundary conditions and compare the results with the analytic results along the symmetry axis. It is also the first test case in reference [25], divided into two parts, one being planar and the other being axisymmetric, both of which are solved by the present solver. The obtained results are compared with the results in reference [25].

The geometry and the boundaries are given in Figure 6.1. The grid used in the solution of this problem is given in Figure 6.2.

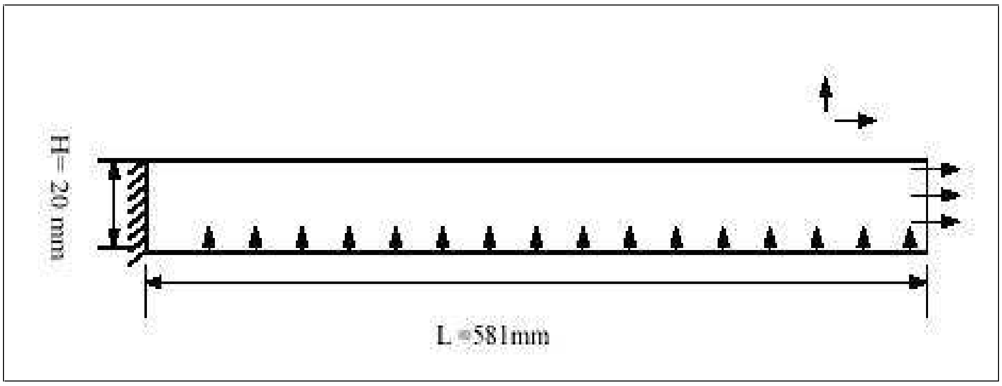


Figure 6.1 Geometry of Test Case 1

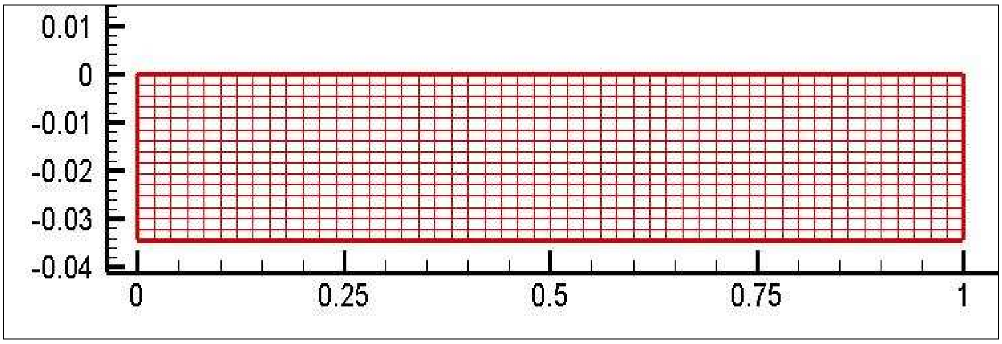


Figure 6.2 Grid of Test Case 1

The physical properties of the flow and grid properties are given in Table 6.1

Table 6.1 Geometric properties and input flow parameters used for Test Case 1

	2-D Planar (Test Case 1-a)	2-D Axisymmetric (Test Case 1-b)
Geometric Properties		
Total Length	581 mm	581 mm
Inside Radius	20 mm	20 mm
Grid Size	51 x 16	51 x 16
Flow Parameters		
Mass flux	2.42 $kg/m^2/s$	2.42 $kg/m^2/s$
Exit Pressure	150000 Pa	150000 Pa
Gamma	1.4	1.4
Gas Constant	299.5 J/kgK	299.5 J/kgK
Reference pressure	100000 Pa	100000 Pa
Flame Temperature	3387 K	3387 K

The solution is started from the subsonic initial condition of which the injection boundary condition imposes for both cases. Front end is inert wall with slip velocity, bottom wall is an injecting wall with constant temperature (flame temperature of the propellant), top wall is the symmetry boundary and the aft end is the pressure outlet boundary. Mach contours, pressure contours and streamlines, also the Mach and pressure profiles along the symmetry axis obtained from the present solver is compared with the results in reference [25] for the planar test case (TC1-a).

6.1.1.1 TC1-A

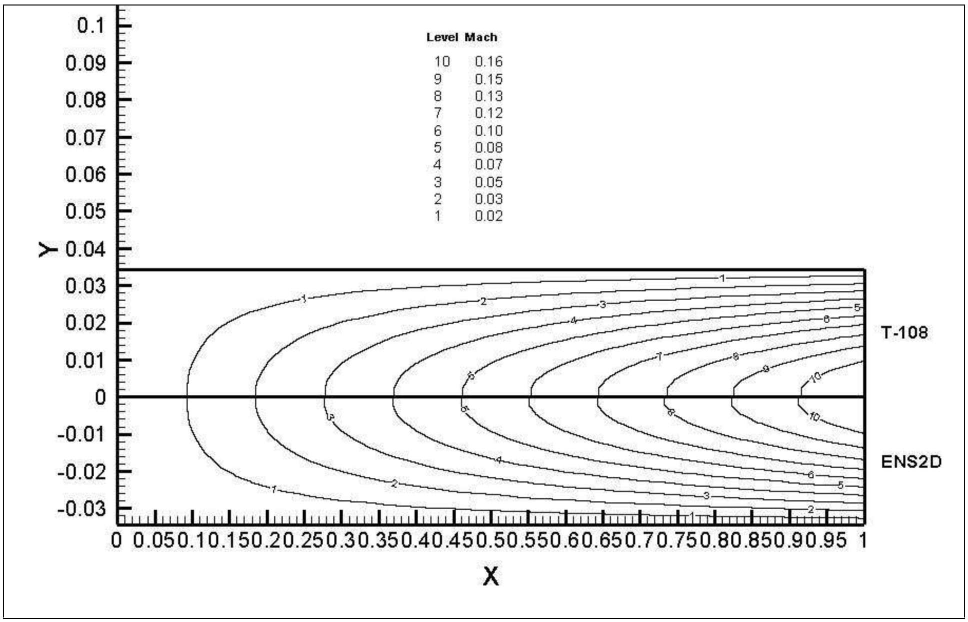


Figure 6.3 Mach Contours of the present solver and T-108 for TC1-a

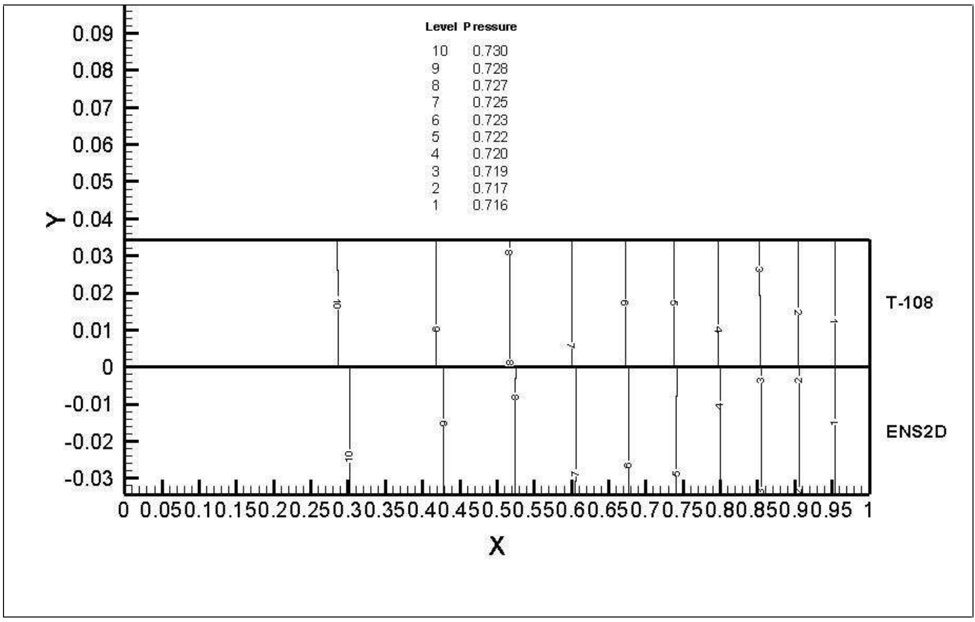


Figure 6.4 Pressure Contours of the present solver and T-108 for TC1-a

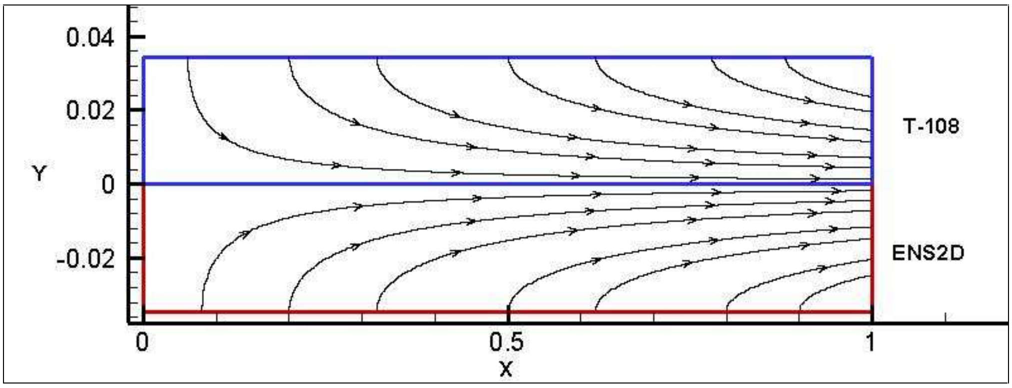


Figure 6.5 Streamlines of the present solver and T-108 for TC1-a

As it can be observed from the iso-Mach, iso-pressure lines and streamlines in Figure 6.3, 6.4 and 6.5 respectively, results obtained by the present solver shows great similarity with the results in reference [25]. It should be noted that no pressure gradient is observed in the transverse direction.

This test case shows that the injection boundary condition in the present solver is valid for injecting flow test cases. Also, wall, symmetry and subsonic exit boundary conditions are valid for subsonic inviscid flows. The Mach number value at the aft end of the geometry at the axis of symmetry is found as 0.1811 by the present solver and the value obtained in reference [25] solution is 0.1804, with a deviation of 0.3%.

The residual used is the maximum local residual, defined as the largest difference for the δU values of all the nodes of the solution domain for that time step. About orders of decrease is observed for the residual in the current test case, as shown in Figure 6.6. CFL number and artificial viscosity are 0.3 and 0.001, respectively. Since no divergent or convergent section is present in this test case, the residual decreases with a good slope in 75000 iterations and reaches to steady value.

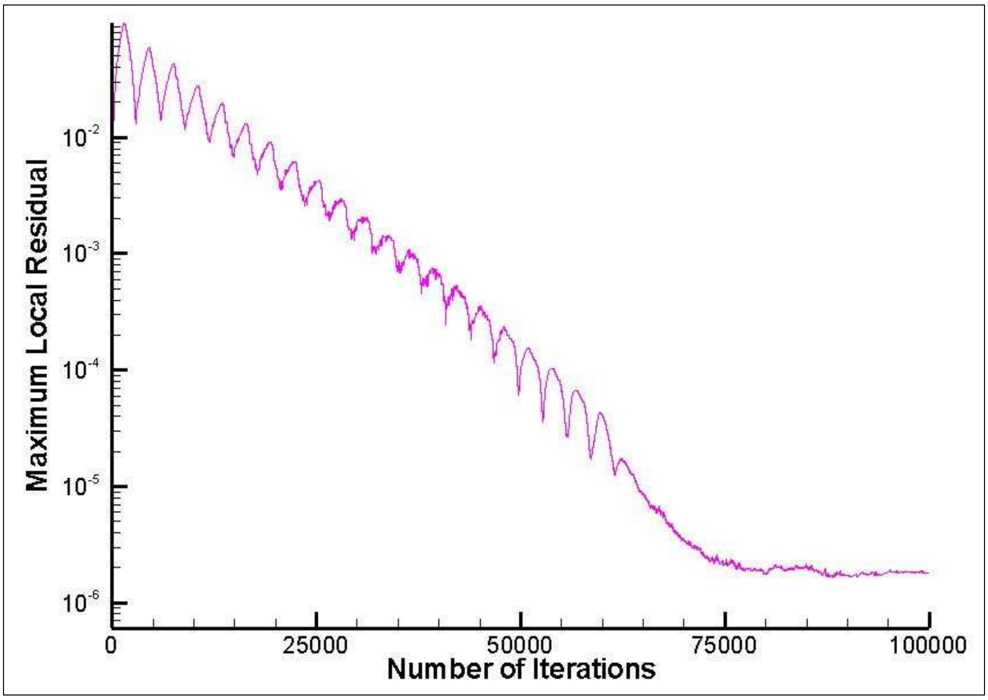


Figure 6.6 Convergence History of TC1-a

Also, the variations of Mach number and non-dimensional pressure along the symmetry plane are compared with the results in reference [25] as shown in Figure 6.7 and 6.8, respectively. The results are quite comparable and verify the results of the current solver.

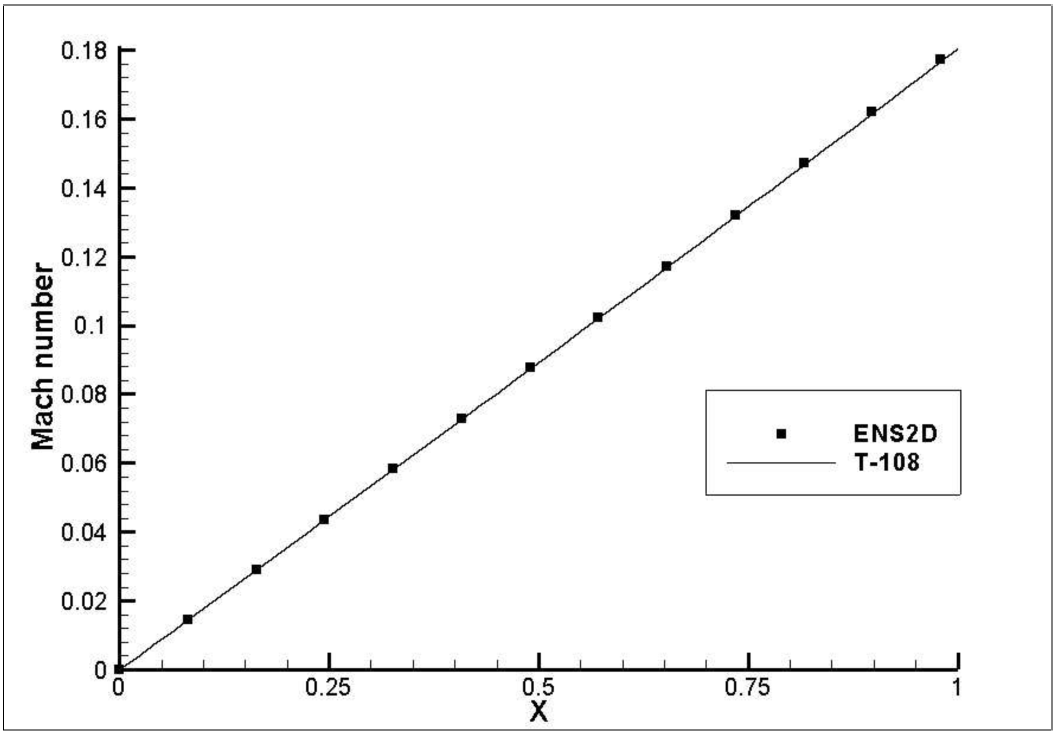


Figure 6.7 Mach profile along the symmetry axis

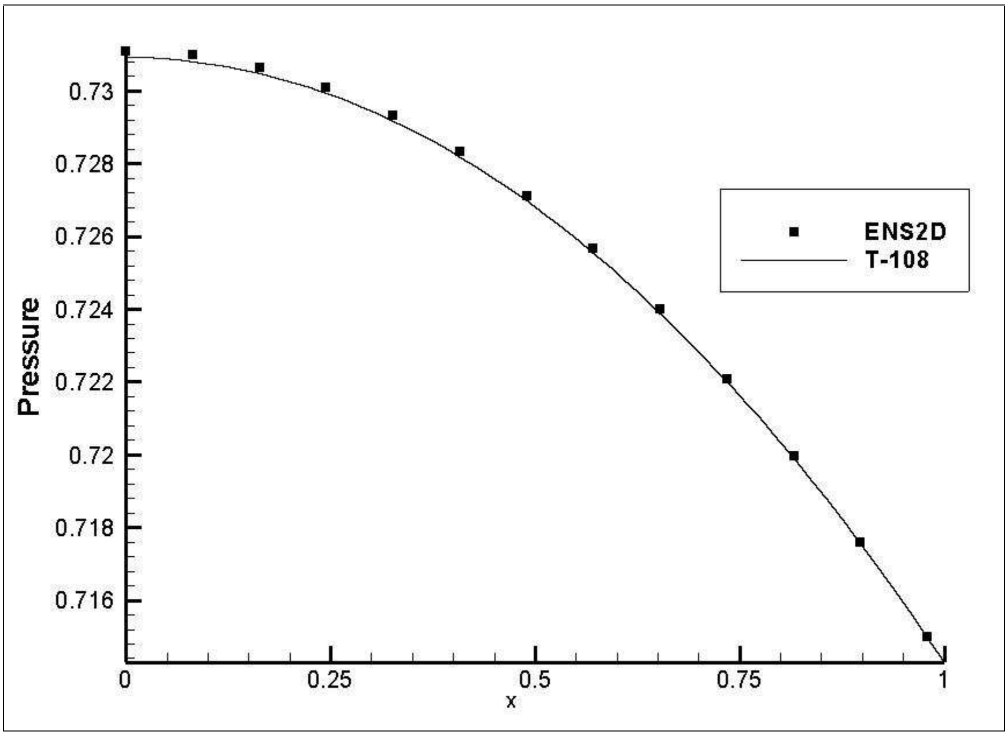


Figure 6.8 Non-dimensional pressure profile along the symmetry axis

6.1.1.2 TC1-B

To check axisymmetric terms, the same test case is solved with axisymmetricity. All the boundary conditions are the same, except that the top plane is no longer a symmetry plane but it is an axis of symmetry boundary. Again, Mach contours, pressure contours and streamlines, also the Mach and pressure profiles along the symmetry axis obtained from the present solver is compared with the results of TC1-b in reference [25].

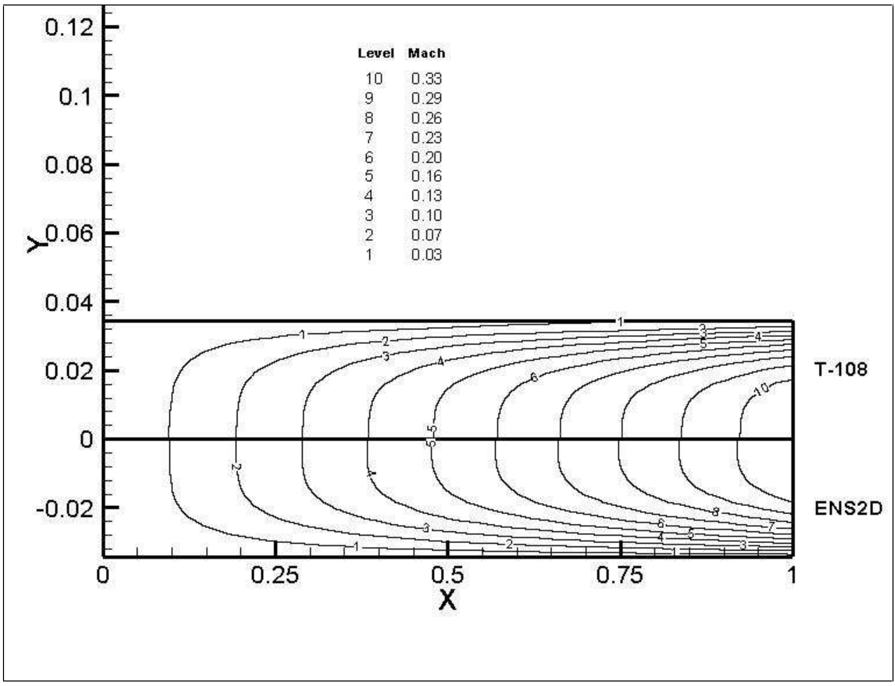


Figure 6.9 Mach contours of the present solver and T-108 for TC1-b

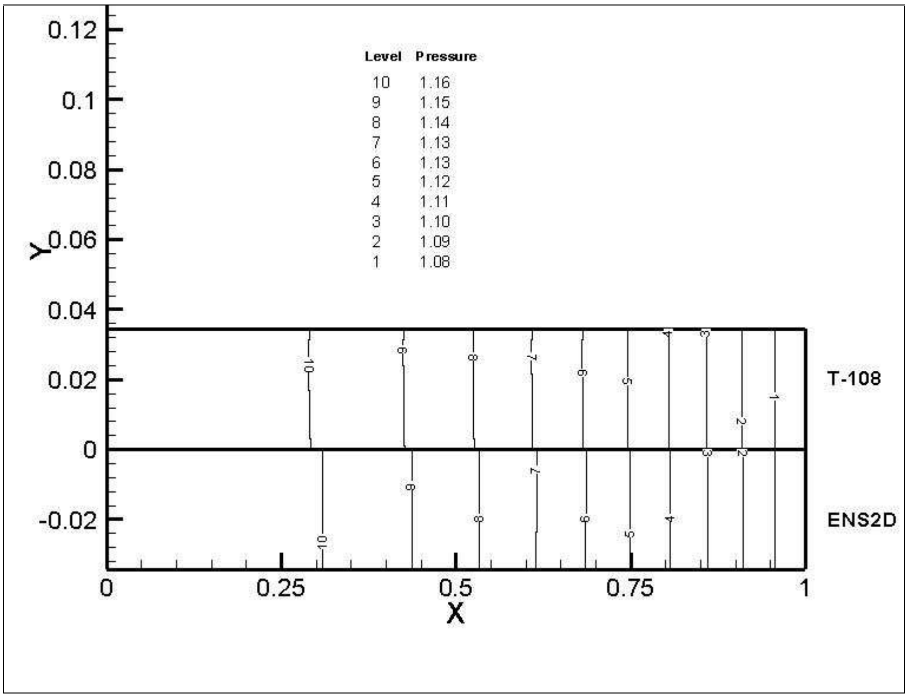


Figure 6.10 Pressure contours of the present solver and T-108 for TC1-b

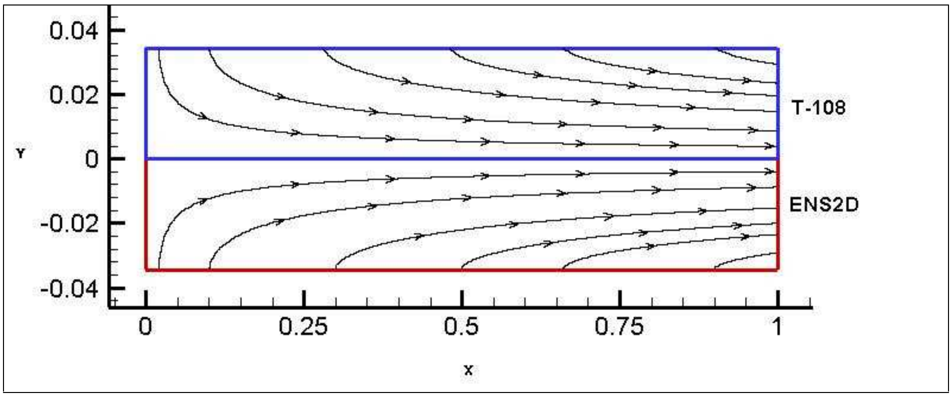


Figure 6.11 Streamlines of the present solver and T-108

The comparisons given in Figure 6.9, 6.10 and 6.11 imply that the results obtained with the present solver are satisfactory, verifying the fact that the axisymmetric terms added to the present solver are valid for inviscid flows. Pressure gradients are only seen in the axial direction, which is also observed also in the planar test case.

The Mach number and non-dimensional pressure variations on the axis of symmetry are given in Figure 6.12 and 6.13. The results are quite similar and comparable. Maximum difference in both Mach number and pressure values at the axis of symmetry is less than 1%. Small variations for Mach number in the head and aft ends of the symmetry axis are most probably due to differences in the boundary conditions at the corners where two boundaries intersect.

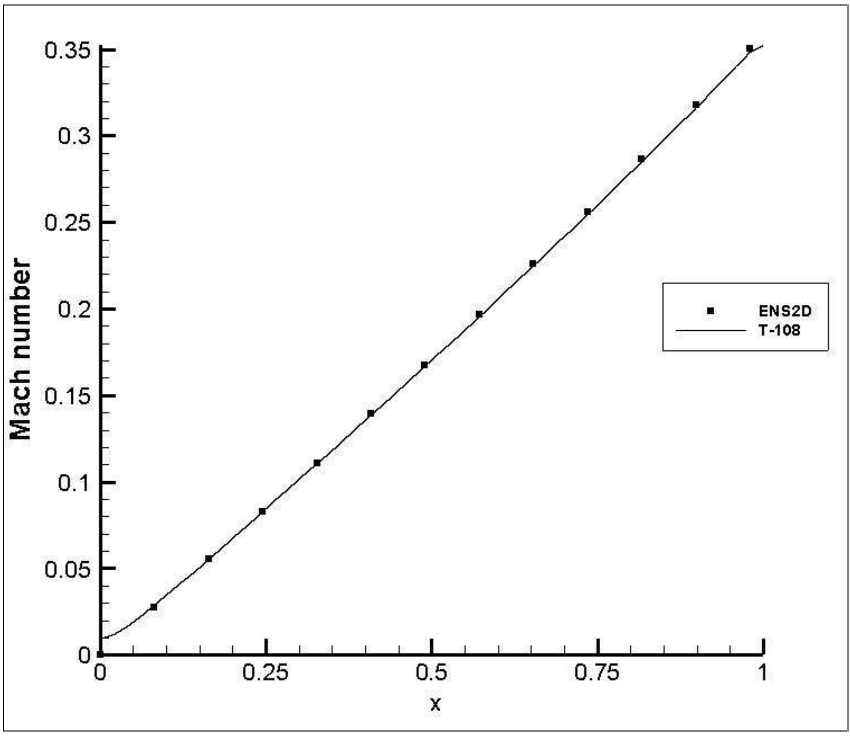


Figure 6.12 Mach profile along the symmetry axis

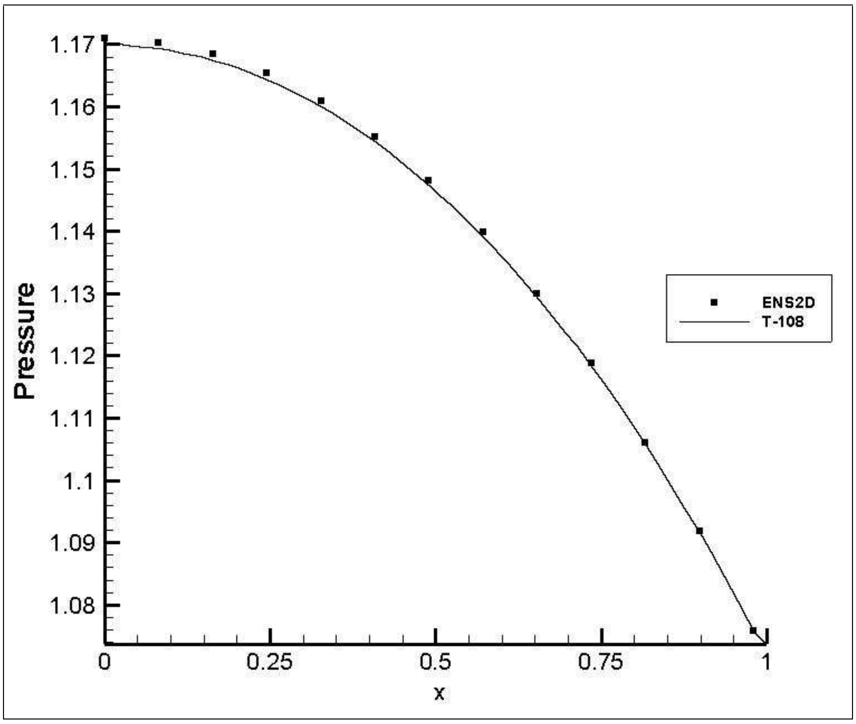


Figure 6.13 Non-dimensional pressure profile along the symmetry axis

For the axisymmetric solution, the maximum local residual decrease is more than 2 order of magnitudes. For this test case, artificial viscosity is 0.001 and CFL number is 0.1. This test case converges slower than the planar test case. Axisymmetric terms are responsible for this situation since no other modification is done on the solver and the flow properties are also same. A lower CFL number is used for this test case and the response of the solver in sense of decrease in residual is slower for the axisymmetric case as expected. Finally, the results obtained are quite satisfactory when the variations in flow variables are considered.

For higher artificial viscosity values, a faster convergence can be obtained but theoretically. However, it should be kept as small as possible to get less smoothed gradients. In all cases, the artificial viscosity is kept as small as possible in order to have minimum smoothening which certainly slows down the convergence.

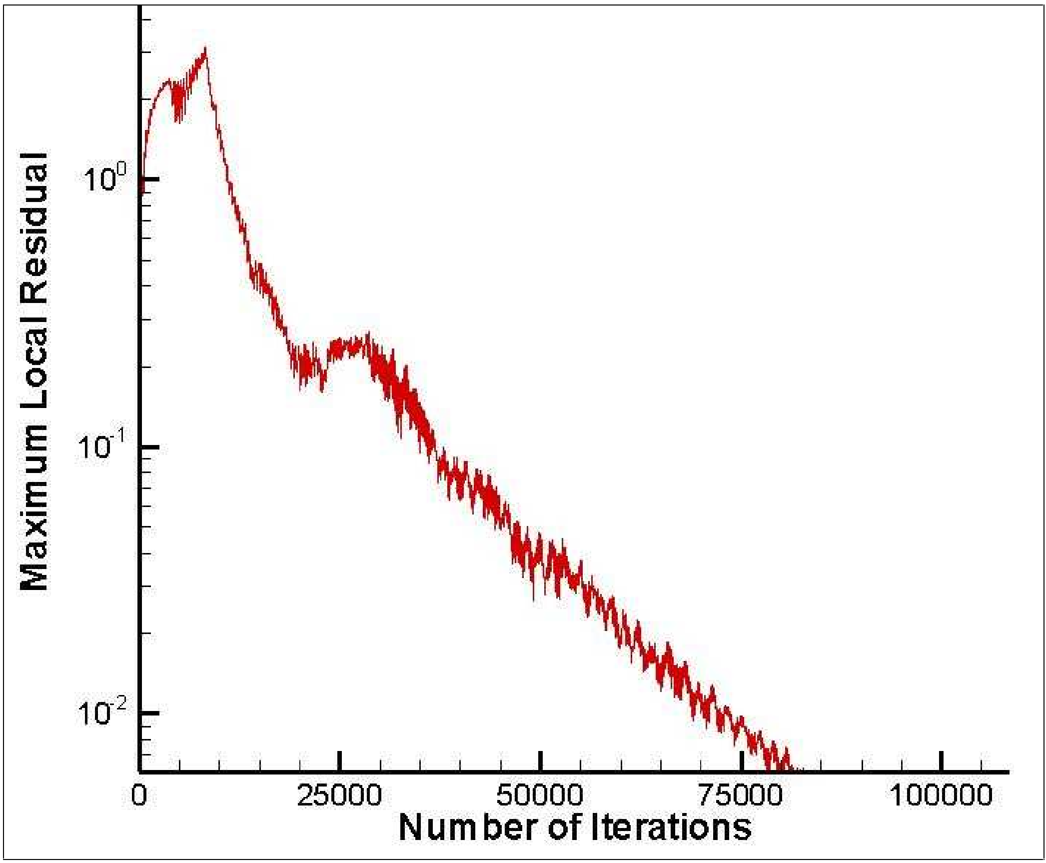


Figure 6.14 Convergence History of TC1-b

6.1.2 TEST CASE 2

Second test case is a lab-scale-motor for the verification of mass injection boundary condition in an axisymmetric geometry. This test case is used in reference [25] to validate the stability of 2-D computations. The burnt propellant is injected from the bottom wall at a constant flame temperature. Length of cylindrical part of combustion chamber is 170 mm while the inner radius is 45 mm. The converging-diverging nozzle has a throat radius of 16.77 mm with a radius of curvature of 30 mm. Overall motor length is L = 270 mm. The physical properties of the flow and grid properties are given in Table 6.2.

Table 6.2 Geometric properties and input flow parameters used for Test Case 2

2-D Axisymmetric (Test Case 2)	
Geometric Dimensions	
Total Length	270 mm
Inside Radius	20 mm
Grid Size	99 x 16
Flow Parameters	
Mass flux	11.39 kg/m^2/s
Reference Pressure	100000 Pa
Gamma	1.14
Gas Constant	299.5 J/kgK
Flame Temperature	3387 K

The head end of the geometry is again inert wall with slip velocity, the top plane is an axis of symmetry, the bottom plane is injecting wall with fixed temperature and the exit plane (aft end) is free supersonic boundary.

The geometry and boundaries are given in Figure 6.15 while the grid for this test case is presented in Figure 6.16.

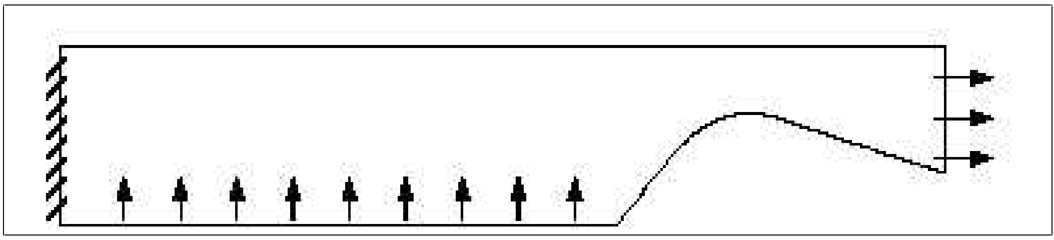


Figure 6.15 Geometry of Test Case 2

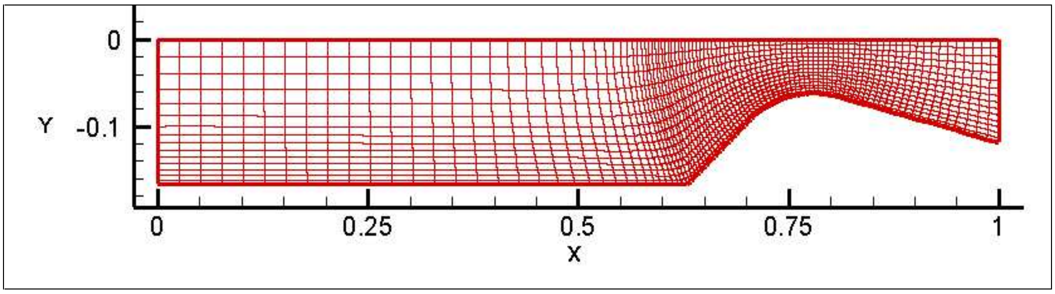


Figure 6.16 Grid of Test Case 2

Mach number and non-dimensional pressure contours and streamlines demonstrating the motion of burnt gases in the rocket motor for the current problem are given in Figure 6.17, 6.18 and 6.19 respectively.

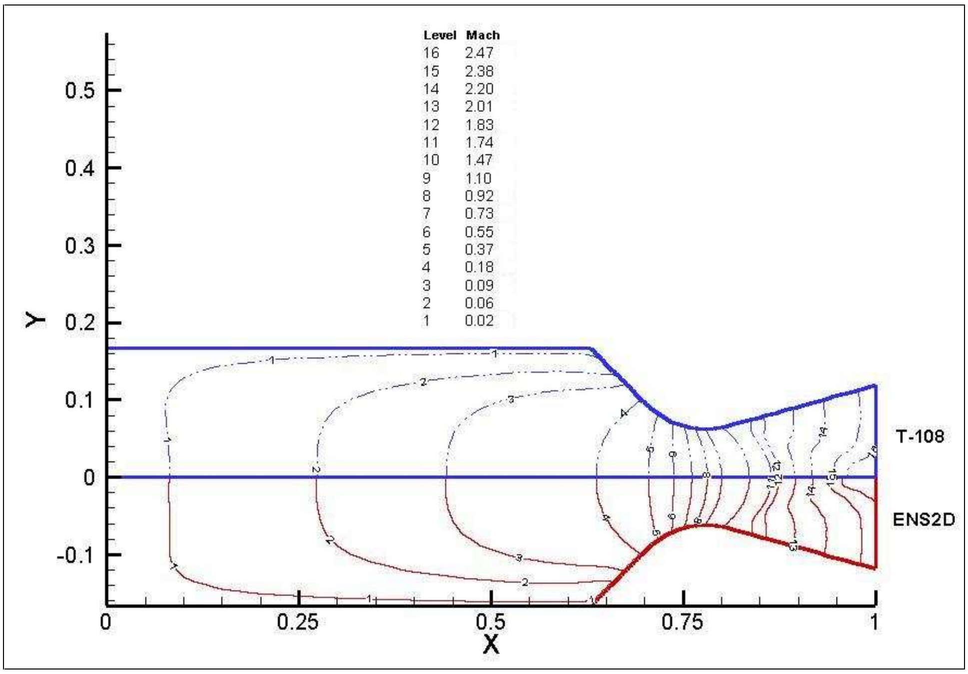


Figure 6.17 Comparison of Mach contours of Test Case 2 with T-108

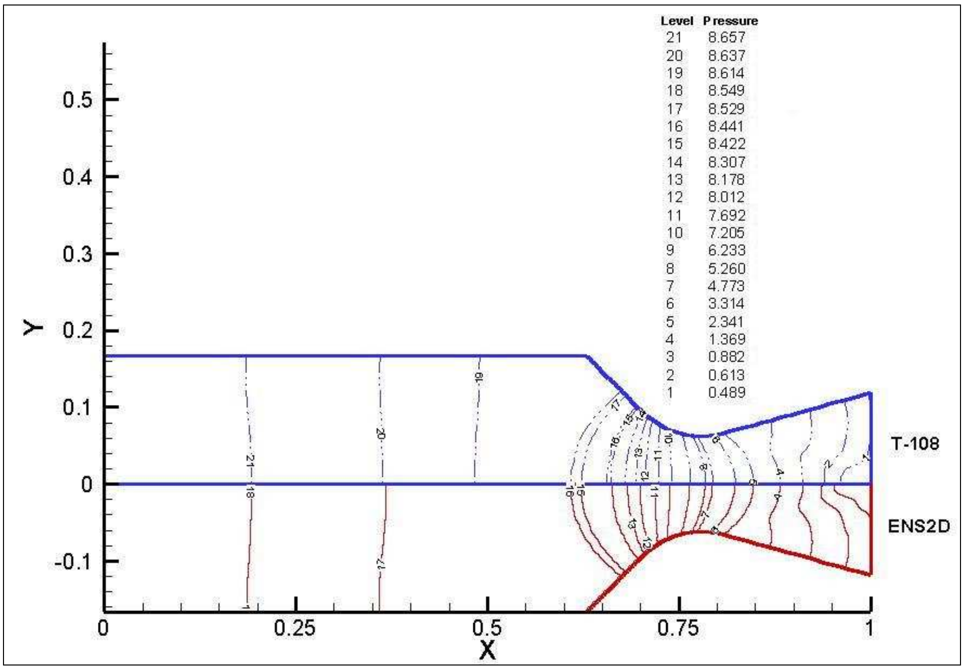


Figure 6.18 Comparison of non-dimensional pressure contours of Test Case 2 with T-108

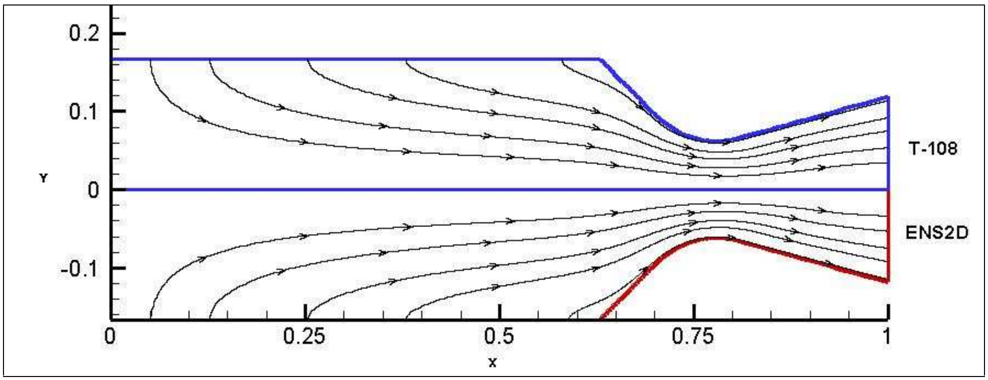


Figure 6.19 Comparison of streamlines of Test Case 2 with T-108

As it can be seen from the above figures, Figure 6.17, 6.18 and 6.19, the results obtained by the present solver are quite compatible to the results of reference [25]. At the converging section of the nozzle, Mach number starts to increase as expected and speed of sound is reached at the throat section of the nozzle. After the throat, supersonic flow is observed and at the point of intersection of symmetry axis and exit plane, Mach number reaches 2.50. Supersonic boundary conditions are also validated by this test case.

Mach number and non-dimensional pressure variation along the symmetry axis are given in Figure 6.20 and 6.21, respectively. Non-dimensional pressure value at the subsonic region is almost constant and reaches its maximum value at the head end which is 8.553. Even though there are differences in velocity profiles, the maximum local difference is approximately 1.1% and this is most probably due to the symmetry plane boundary condition.

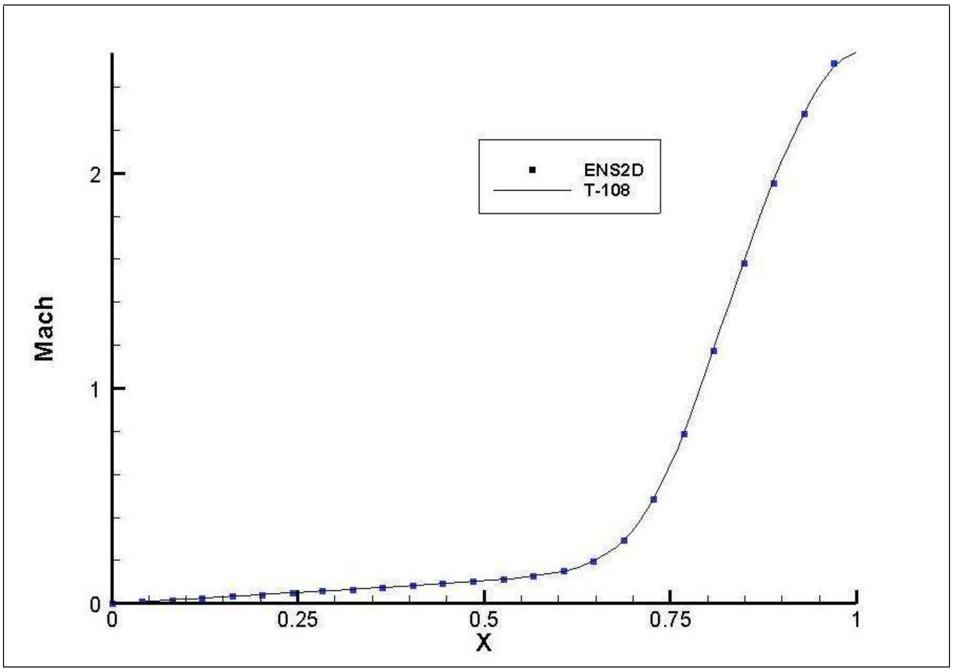


Figure 6.20 Mach variation along the symmetry axis

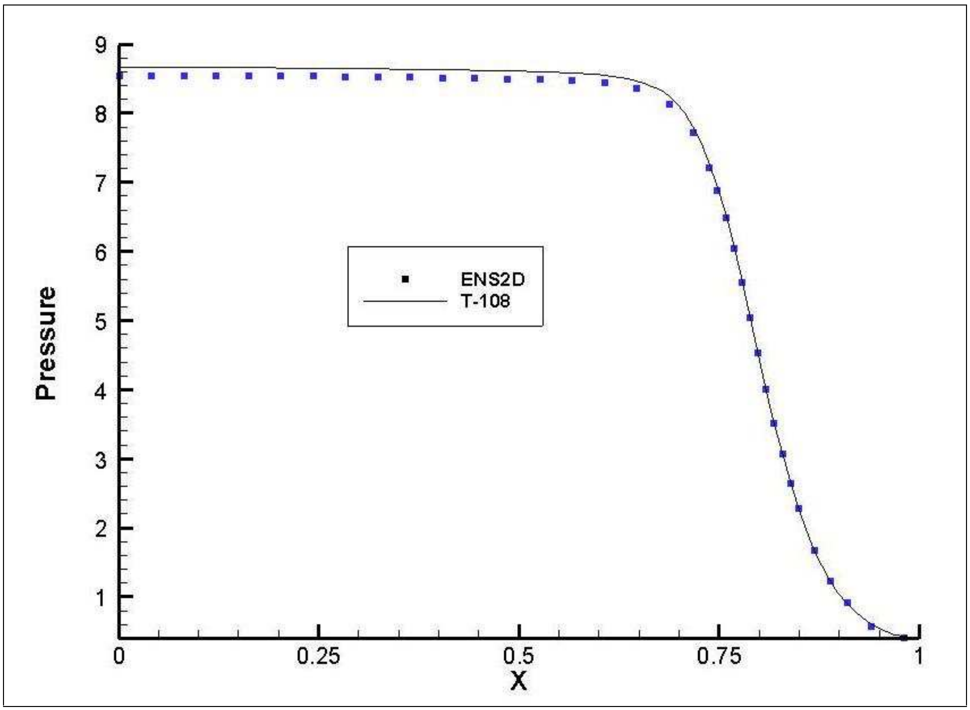


Figure 6.21 Non-dimensional pressure variation along the symmetry axis

For the supersonic case, some difficulties are observed during the initiation of the solution for low artificial viscosity values. When the solution is initiated, the flow inside the geometry is subsonic everywhere and according to CFL number used, some iterations are required to obtain a supersonic flow beyond the nozzle throat. During these iterations, an artificial shock wave travels through the divergent part of the nozzle and may cause some instabilities till the throat Mach number reaches unity.

Above results are obtained for a CFL number of 0.2. For this value of the CFL number, 1000 iterations are required with 0.02 artificial viscosity to overcome these instabilities and artificial viscosity value is decreased to 0.001 after 1000 iterations. The residual history is given in Figure 6.22.

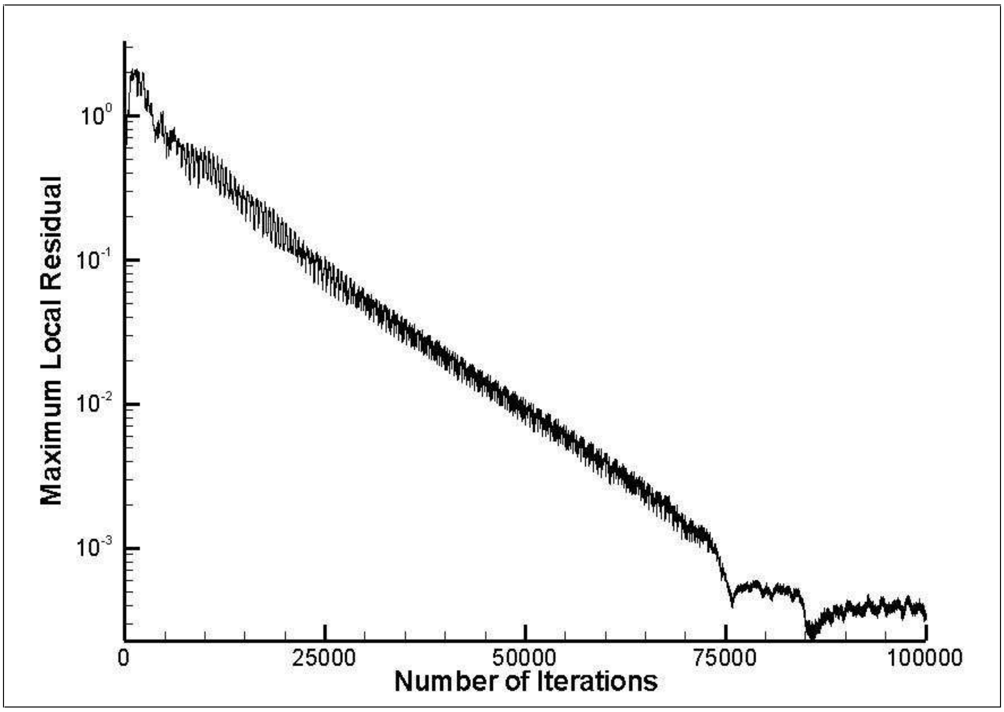


Figure 6.22 Residual history of Test Case 2

6.1.3 TEST CASE 3

This test case is used to simulate an actual motor with a narrow throat. Physical dimensions and flow properties are given in Table 6.3. Geometry and grid used for the solution of this test case are given in Figure 6.23 and 6.24, respectively.

Table 6.3 Geometric properties and input flow parameters used for Test Case 3

Geometric Dimensions	
Total Length	270 mm
Inside Radius	20 mm
Grid Size	121 x 16
Flow Parameters	
Mass flux	16 kg/m^2/s
Reference Pressure	100000 Pa
Gamma	1.14
Gas Constant	299.5 J/kgK
Flame Temperature	3387 K

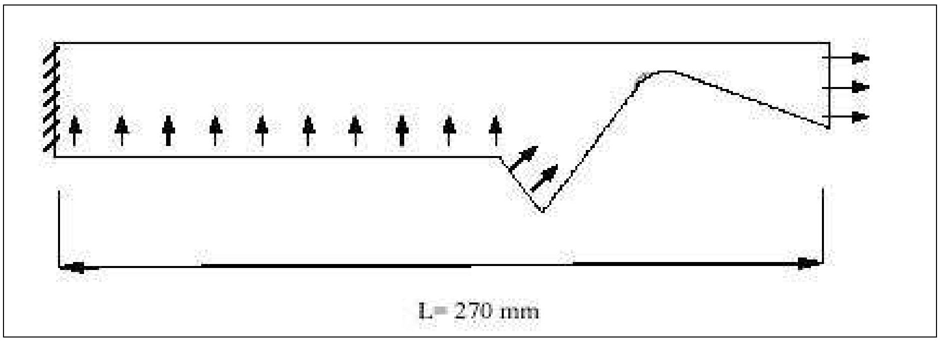


Figure 6.23 Geometry of Test Case 3

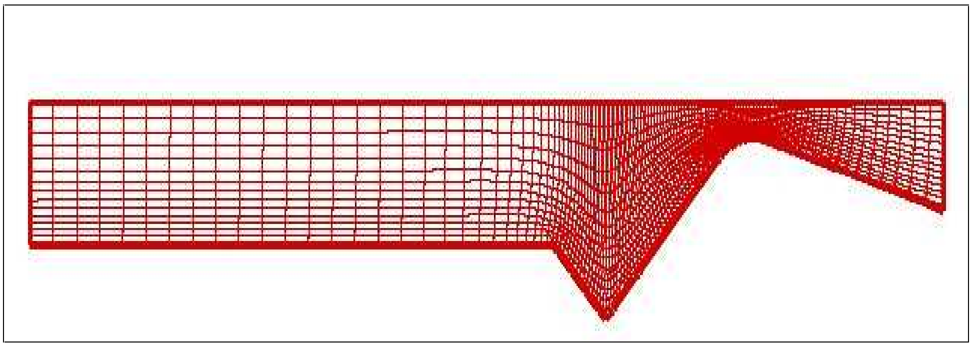


Figure 6.24 Grid of Test Case 3

This test case is a modified form of test case 2, with injection from the inclined surface in front of the convergent part of the convergent-divergent nozzle. Also, the throat is narrower when compared to the previous test case. Due to this inclined injection and narrower throat, this test case was the most challenging test case in sense of convergence. The definition of the point of intersection of inclined injection surface and horizontal injection surface can also effect the results especially in the convergent section.

The boundary conditions can be given as follows: the bottom plane starts with mass injection boundary with a constant flame temperature till the convergent section of the nozzle and then wall type boundary condition is used along the nozzle. Wall boundary conditions are also used for the left plane (head end). Axis of symmetry boundary condition is used for the top plane while supersonic boundary condition is used for the exit plane.

Mach number and non-dimensional pressure contours and streamlines demonstrating the motion of burnt propellant in the rocket motor are given in Figure 6.25, 6.26 and 6.27, respectively.

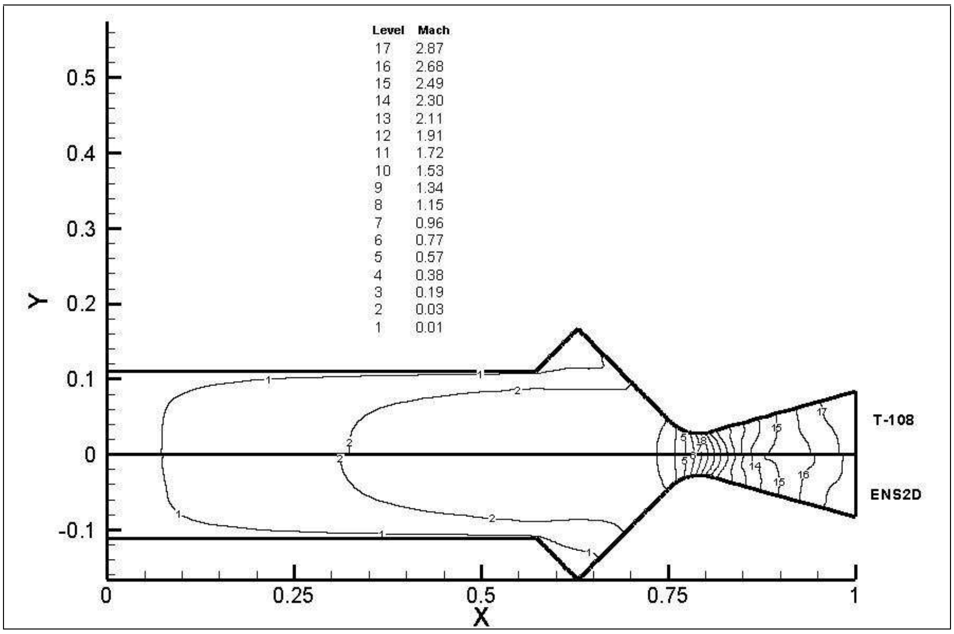


Figure 6.25 Comparison of Mach contours of Test Case 3 with T-108

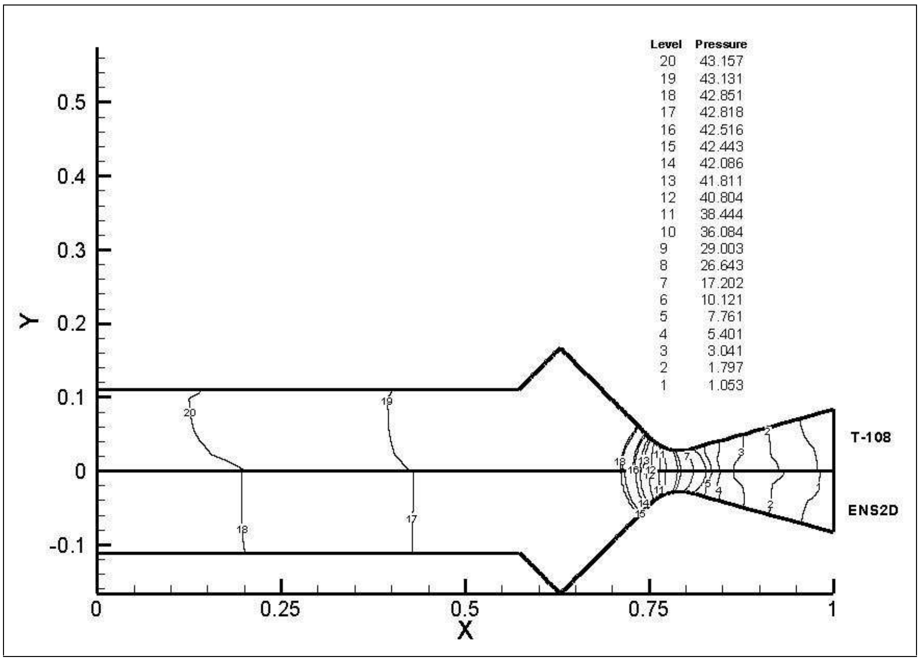


Figure 6.26 Comparison of non-dimensional pressure contours of Test Case 3 with T-108

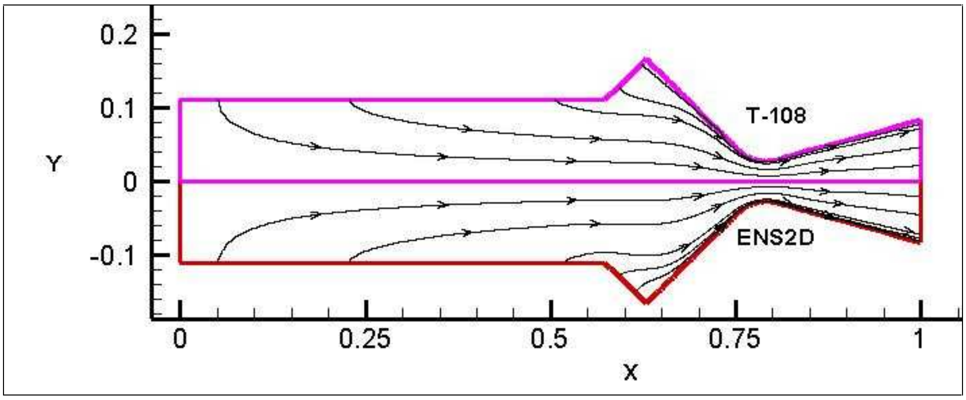


Figure 6.27 Comparison of streamlines of Test Case 3 with T-108

As it can be observed from the figures, results obtained by the present solver show great similarity to the results obtained in reference [25]. The slopes of iso-Mach lines change in the divergent section of the nozzle in both solvers, due to the compression wave reflecting from the symmetry plane. Pressure contours obtained by ENS2D show less variation in transverse direction in the subsonic region where the diameter is constant. This is more reasonable when subsonic flow with fixed diameter is considered and the previous test case demonstrates this behaviour properly.

Besides these contours, the Mach number and non-dimensional pressure variations along the symmetry axis are given in Figure 6.28 and 6.29, respectively.

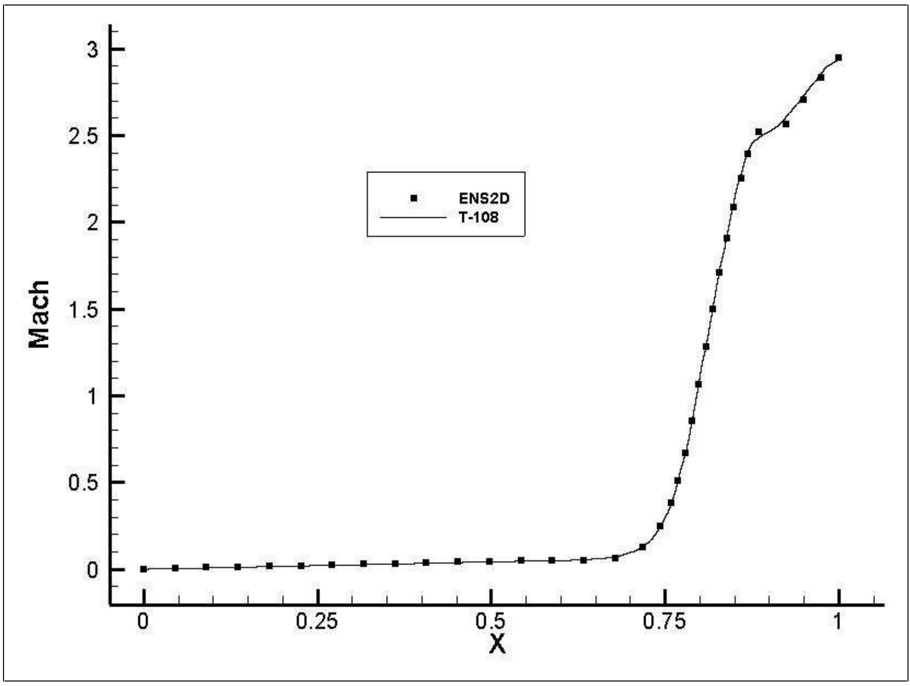


Figure 6.28 Mach profile along the symmetry axis

Mach number profile seems well fit to the results obtained in reference [25]. At the longitudinal location where a compression wave reflects barely, a difference of 1.2 % is observed but the profiles tend to fit well again after the reflection. This difference may again be due to the implementation of symmetry plane boundary conditions.

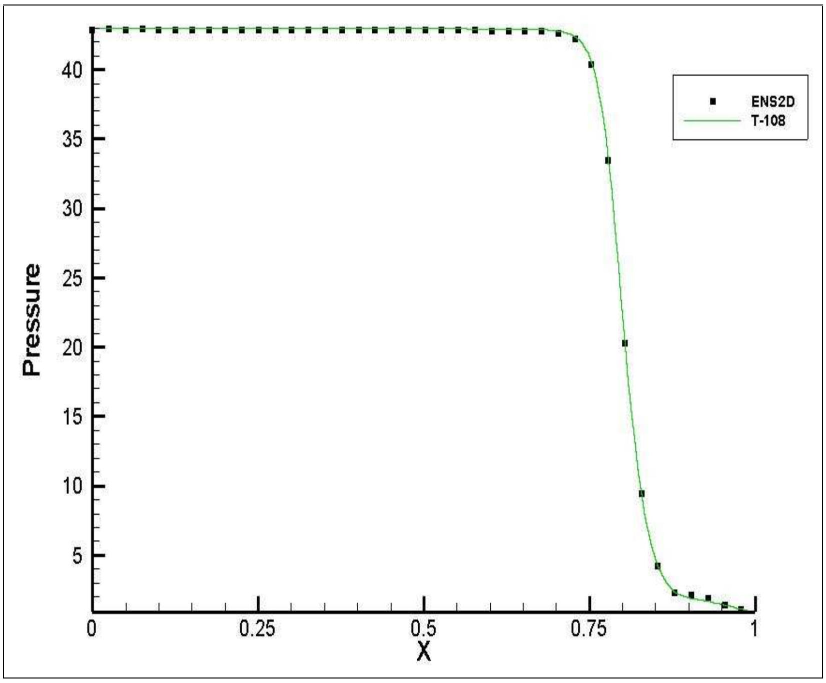


Figure 6.29 Non-dimensional pressure variation along the symmetry axis

Non-dimensional pressure profile of the present solver is also similar to one obtained from the results of reference [25]. Again, as it is in the previous test case, pressure profile in the subsonic region is almost constant and reaches its maximum value of 42.853 at the head end.

The difficulty encountered during the solution of the previous test case is observed also for this test case. This time, since the geometry and injection directions are more complex, the initial artificial viscosity value is taken as 0.2. This is a quite high value, which allows an initial CFL value of 0.5. When a Mach number of unity is obtained at the nozzle throat after 5000 iterations, artificial viscosity is decreased to 0.05 which allows a CFL number of 0.15 at most. Lower artificial viscosity values are tested to obtain proper results but even with low CFL numbers, instabilities are observed. Besides, the results obtained by this CFL and artificial viscosity values are quite satisfactory and no artificial effects are observed in the solution.

Maximum local residual decreased only 2 orders in 100000 iterations which was lower than the ones in other cases, most probably due to the complexity of the flow, but this situation does not bring any problem again in sense of comparing the results. After 100000 iterations, the residual stayed constant and it is assumed that the convergence is reached after 100000 iterations with 3 orders of decrease in the residual.

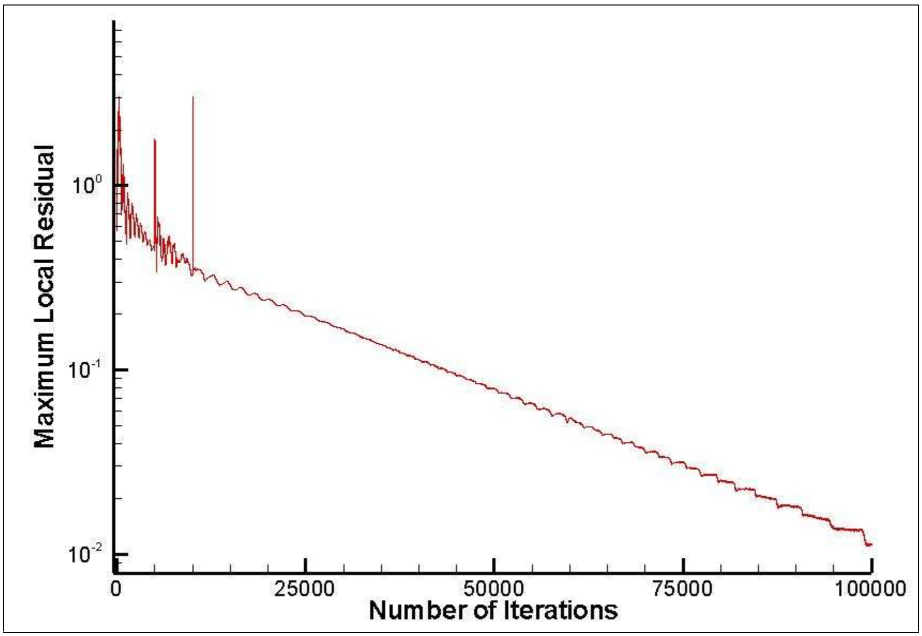


Figure 6.30 Residual history of Test Case 3

6.1.4 TEST CASE 4

This test case is a simulation of a full SPRM with two inclined injection surfaces. Physical dimensions and flow parameters are given in Table 6.4. The geometrical representation of the test case and the grid used in the solution are given in Figure 6.31 and 6.32, respectively.

Table 6.4 Geometric properties and input flow parameters used for Test Case 4

Geometric Dimensions	
Total Length	270 mm
Inside Radius	20 mm
Grid Size	176 x 15
Flow Parameters	
Mass flux	19.6 $kg/m^2/s$
Reference Pressure	100000 Pa
Gamma	1.22
Gas Constant	287 J/kgK
Flame Temperature	3149 K

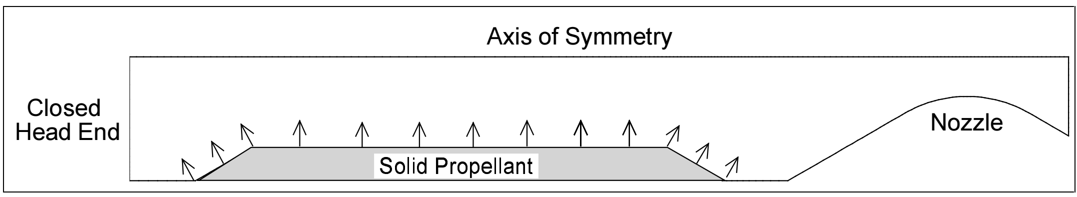


Figure 6.31 Geometry of Test Case 4

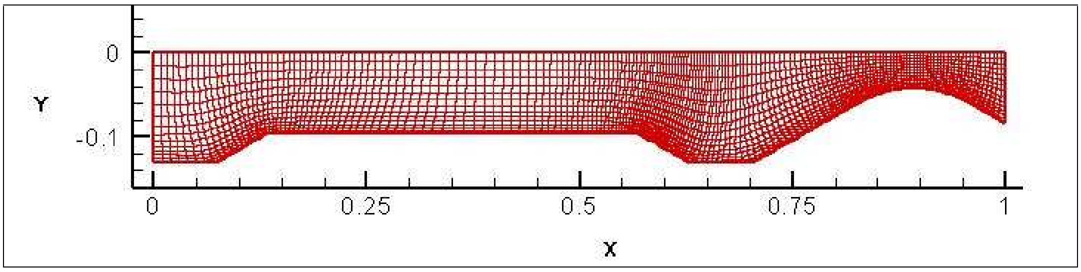


Figure 6.32 Grid of Test Case 4

This test case is the best test case in sense of resembling a SPRM. Generally, to obtain the required thrust regime for the mission of the missile, special grain (propellant) shapes are used. During the burnback of these grain shapes, there are injections in different directions or propellant thickness varies as it can be observed from Figure 6.31. The geometrical configuration has been retained corresponding to mid-combustion. The combustion chamber is 50 mm in length, which contains an axisymmetric grain formed of two conical ends with inclined injection linked to a cylindrical port with vertical injection.

The bottom plane starts with a wall, then continues with mass injection at constant flame temperature up to the divergent section of the nozzle. Thereafter, there is a wall boundary. Left plane can also be treated as a wall. Top plane is an axis of symmetry and the exit plane has supersonic boundary conditions.

The Mach number, non-dimensional pressure contours and the streamlines demonstrating the motion of the burnt propellant are given in Figure 6.33, 6.34 and 6.35, respectively.

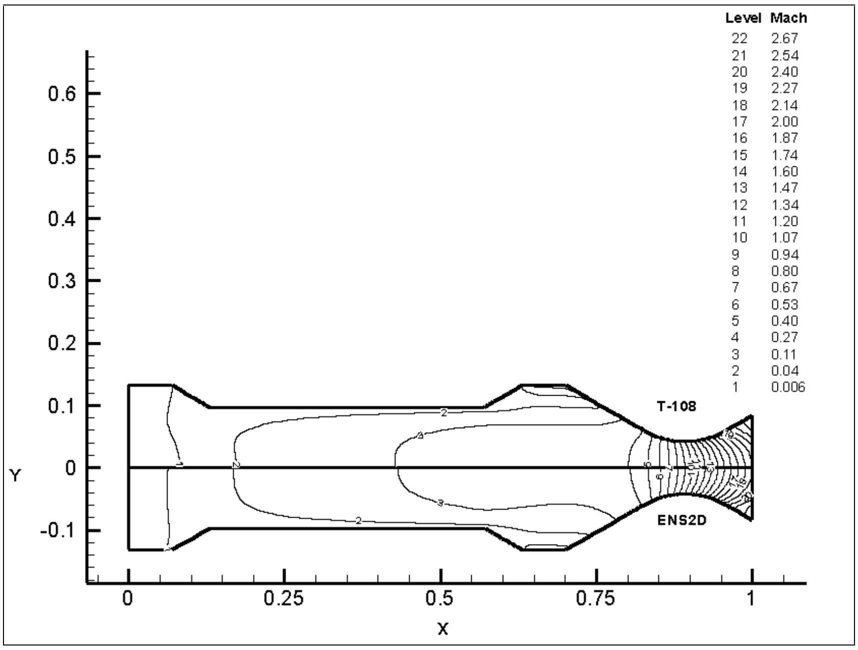


Figure 6.33 Comparison of Mach contours of Test Case 4 with T-108

Mach number contours of the two solutions have very small differences, as it can be seen from Figure 6.33. Even if there are differences in the subsonic region at the head end, ENS2D gives almost constant Mach values at the transverse direction which is reasonable just like the present test case. Besides, the Mach number has quite small value at the head end. For the supersonic section, the Mach number contours are almost identical.

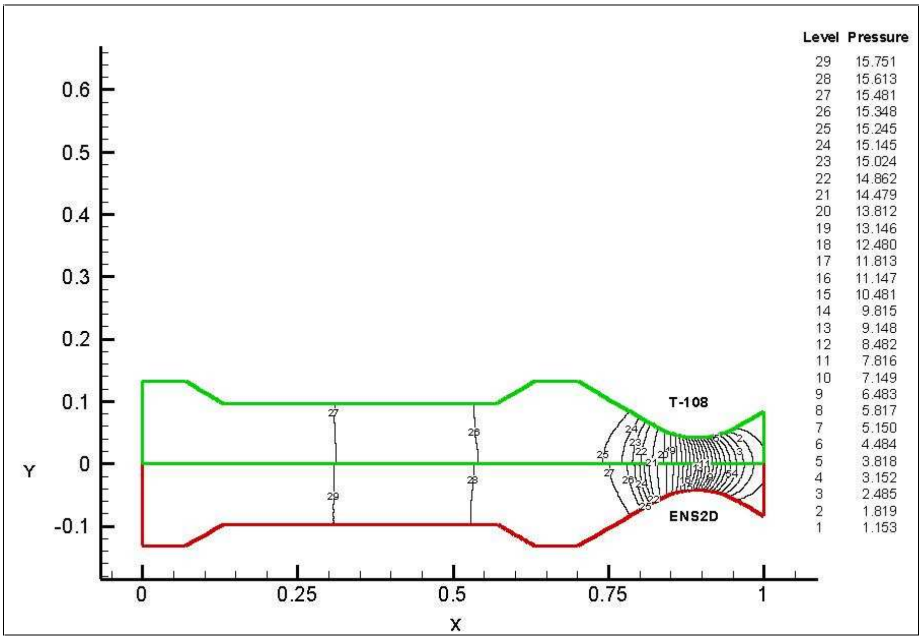


Figure 6.34 Comparison of non-dimensional pressure contours of Test Case 4 with T-108

The pressure contours also show great similarity, with some small differences in the subsonic region. This difference is locally 1 percent at most and most probably due to the small differences in the implementation of symmetry boundary condition and because of the handling of the intersection points of two sides of the inclined surface near to the head end.

In Figure 6.35, the turning of the flow from inclined surface at the head end can be clearly seen from the streamlines.

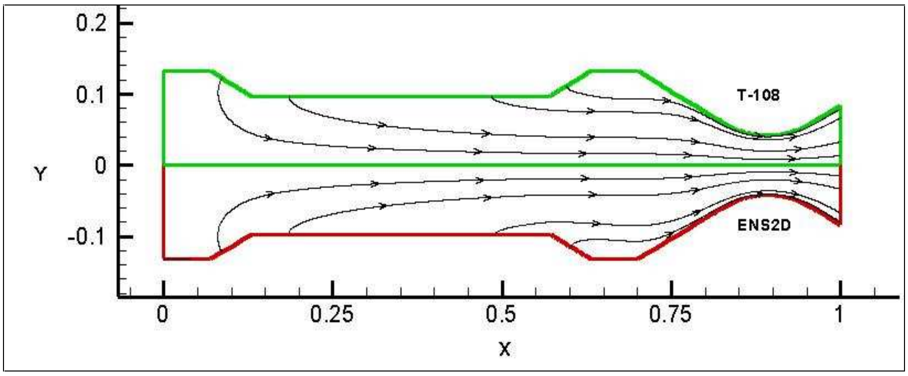


Figure 6.35 Comparison of streamlinesof Test Case 4 with T-108

Below, Mach and non-dimensional pressure profiles of two solutions are compared along the symmetry axis in Figure 6.36 and 6.37, respectively.

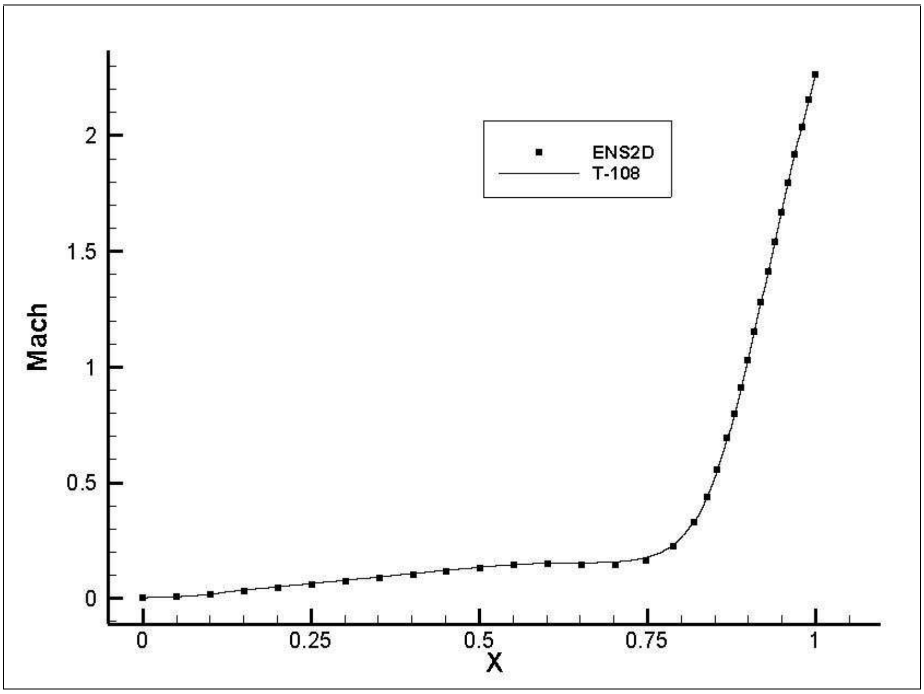


Figure 6.36 Mach profile along the symmetry axis

As it is clearly seen from Figure 6.36, the Mach profile along the symmetry axis obtained by the present solver compares well with reference [25]. The maximum difference is about 0.6%.

Pressure profiles also show great similarity and pressure remains almost constant in the subsonic region, having its maximum value at the head end which is 15.808.

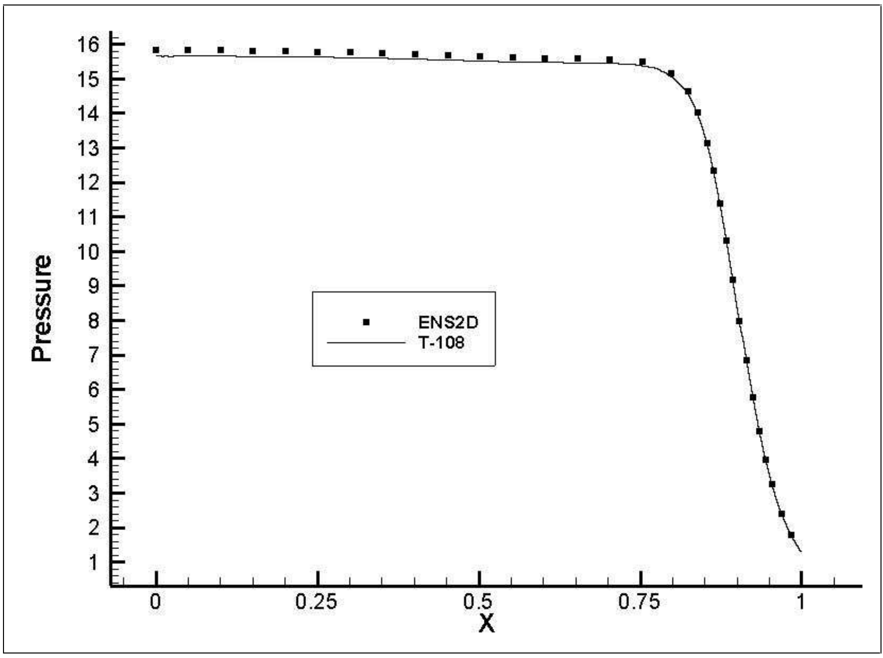


Figure 6.37 Non-dimensioanl pressure profile along the symmetry axis

Residual history for the current test case is given in Figure 6.38. The maximum local error decreases about 3 orders which the solution is assumed to be converged.

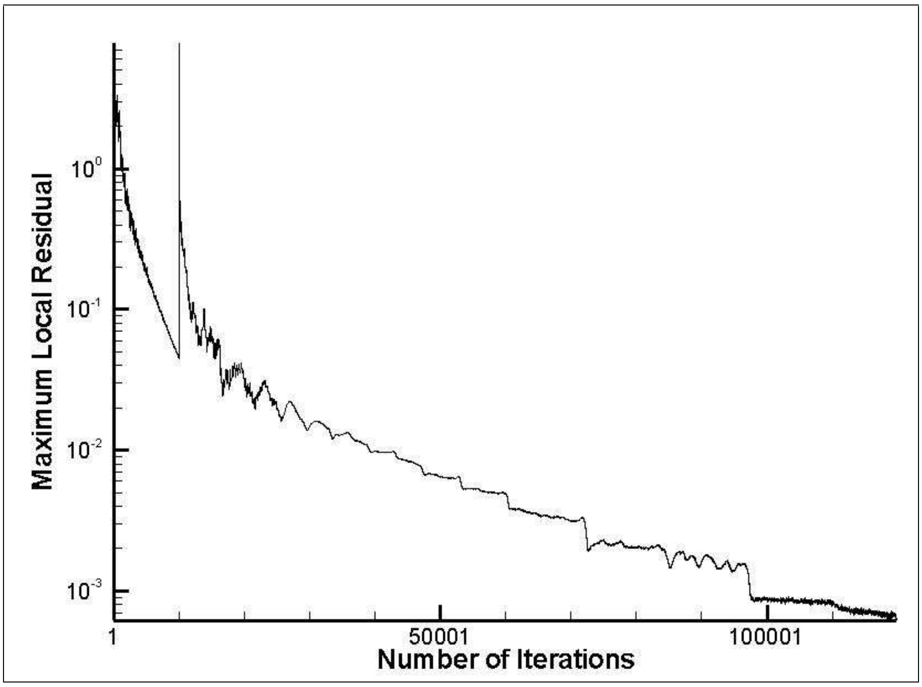


Figure 6.38 Residual history of Test Case 4

By these 4 test cases, inviscid axisymmetric solver with injection boundary conditions is verified.

6.2 VISCOUS FLOW TEST CASES

6.2.1 LAMINAR FLAT PLATE

After the solver is verified for inviscid internal axisymmetric flows, viscous flux terms and also additional axisymmetric terms coming from the nature of viscous flow should also be verified. First, for the verification of viscous flux terms, the laminar over a flat plate is solved and the results are compared with the self-similar Blasius solution.

The grid used is given in Figure 6.39 where the plate begins at x=0 and ends at x=1. Different than the test cases solved above, this test case has only two different kinds of boundary conditions. One of them is viscous adiabatic wall boundary and the other is far field boundary, where the Mach number of the flow is defined on the left, top and right planes, to provide a free flow over the flat plate. Even though these boundary conditions are not used for internal flows, a flat plate solution gives a good indication of whether the viscous flux terms are correct and the viscous adiabatic wall boundary conditions are properly applied.

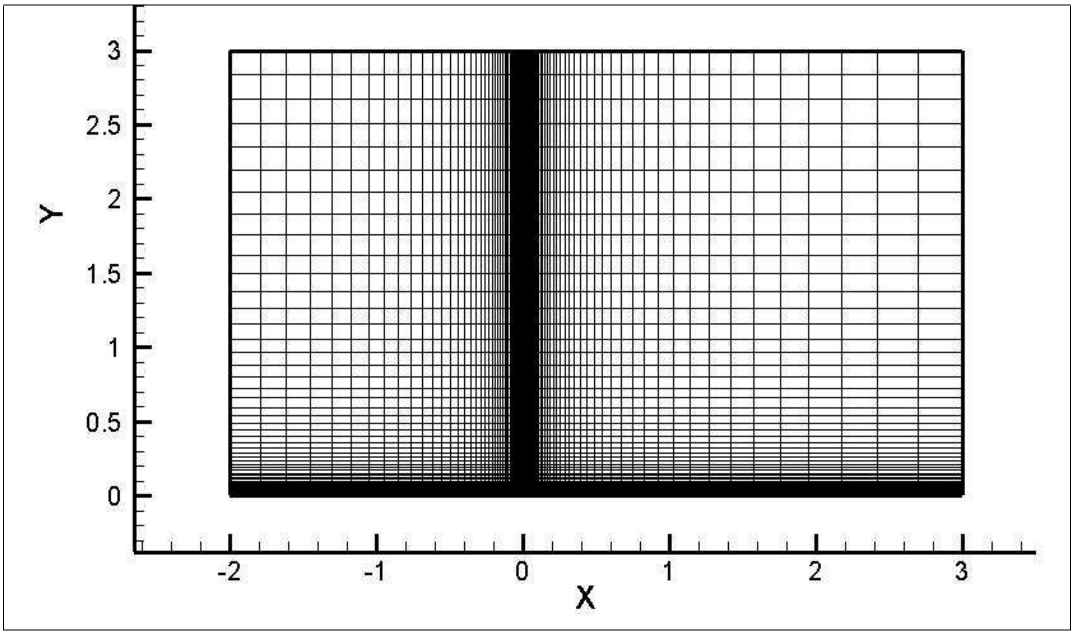


Figure 6.39 Grid used for the solution of laminar flat plate

The Mach number, Reynolds number, Prandtl number and reference temperature are 0.3, 35000, 0.72 and 288 K, respectively. The velocity profile of the solution obtained by ENS2D is compared with the self similar Blasius solution and the obtained non-dimensional velocity profile is given in Figure 6.40 where η is defined as $\eta = y/5x/Re_x$.

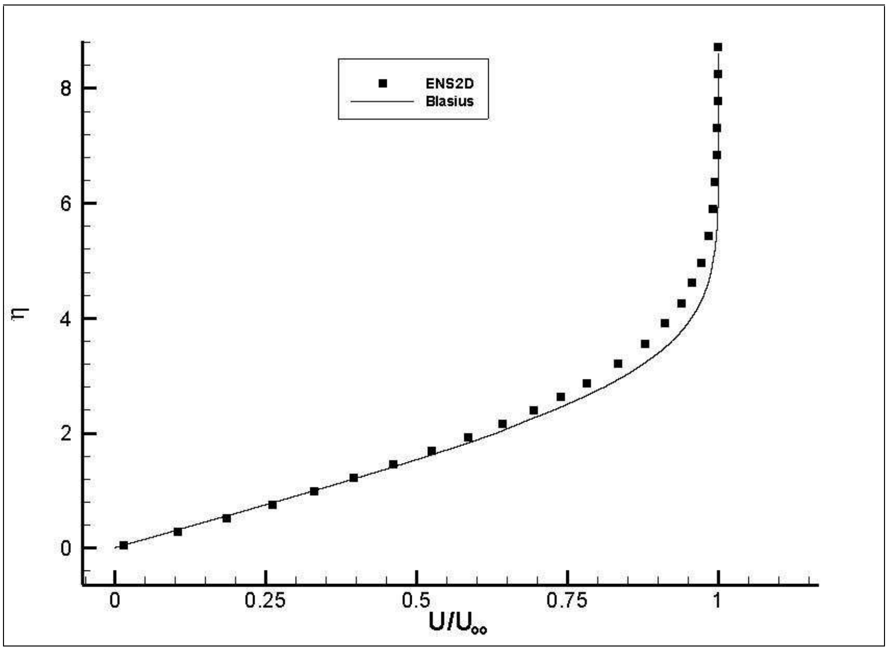


Figure 6.40 Axial velocity profile of the laminar flow over a flat plate

It should be noted that the profile above is taken in the longitudinal location where the fully developed flow is already observed, which is at *x/L*=0.5. As it can be seen from Figure 6.40, the obtained axial velocity profile compares well with the self-similar Blasius solution. The CFL number and the artificial viscosity are 0.3 and 0.004, respectively. With these values, the solution is assumed to be converged with the decrease of maximum local residual decreased by 3 orders. The residual history obtained from the solution is given in Figure 6.41.

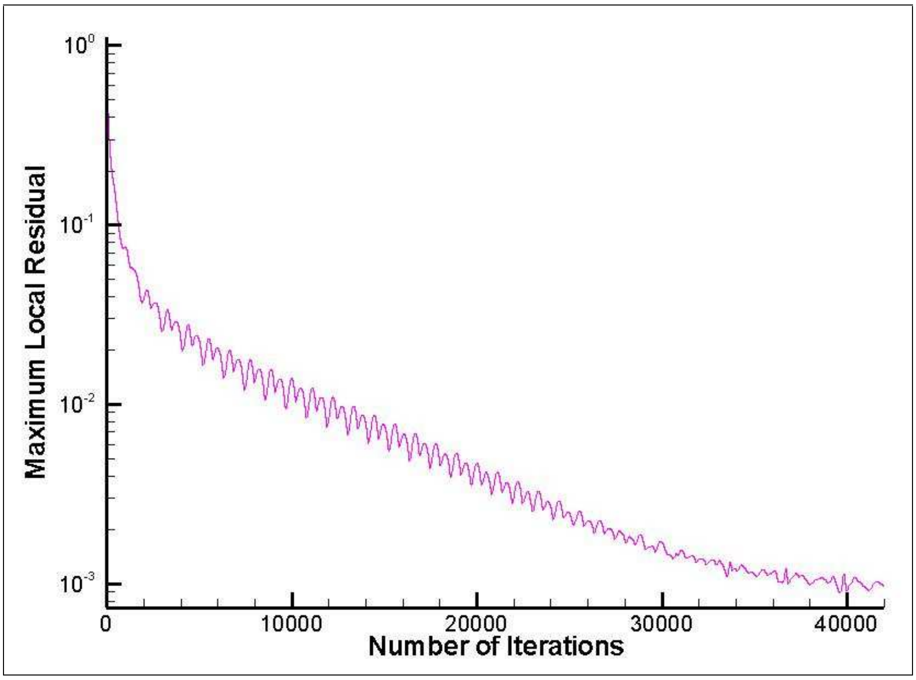


Figure 6.41 Residual history for the laminar flow over a flat plate

With this test case, the viscous flux terms for 2-D planar flows are verified.

The same problem is also solved by decomposing the solution domain into two blocks, to verify the solution of laminar problems with the multi-block technique and also to see the effects of this technique on convergence. The geometry is divided at x=0, where the flat plate just starts. It can be said that x=0 is relatively most complex x location in sense of domain decomposition along the flat plate, where one side of the point is free flow while the other side is a solid wall. The results obtained for the two-block domain are exactly the same as the single block solution. Therefore, only the convergence history of the two-block solution is given in Figure 6.42.

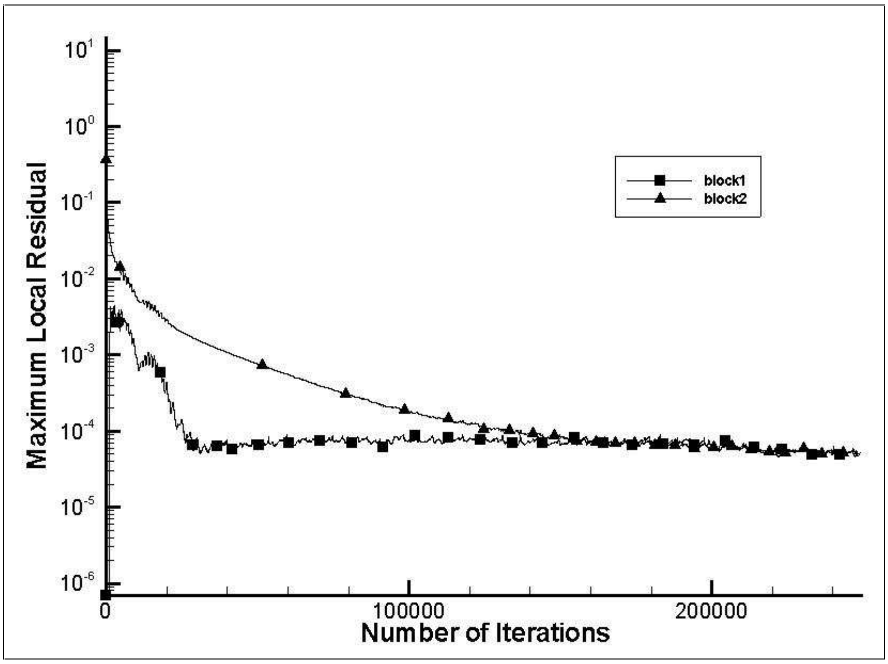


Figure 6.42 Residual history of the multi-block solver for the laminar flow over a flat plate

It can be observed from Figure 6.42 that block 1 shows no progress in convergence while block 2, which has the flat plate has 3 orders of decrease of residual, where the solution is assumed to be converged. Since block 1 has no solid boundary and all the boundaries except the interface boundary are free flow, just as it is in the initiation, the residual of block 1 starts from a very low value. At the end, it converges to the value that block 2 converges. This convergence is the stabilization of flow variables at the interface which are disturbed by the leading edge of the flat plate.

6.2.2 LAMINAR PIPE FLOW

Since the scope of this thesis is mainly SPRM flow solutions as also mentioned above, most of the problems that are to be solved are axisymmetric problems. Therefore, besides the 2-D viscous flux terms, axisymmetric viscous terms must be verified for an accurate laminar Navier-Stokes solver.

A laminar pipe flow is used for the verification of the axisymmetric viscous terms. The grid used in the solution is given as below in Figure 6.43. At the inlet, free stream Mach number is 0.0023. Top plane is handled as axis of symmetry, the bottom plane is adiabatic wall and the exit plane is subsonic outflow.

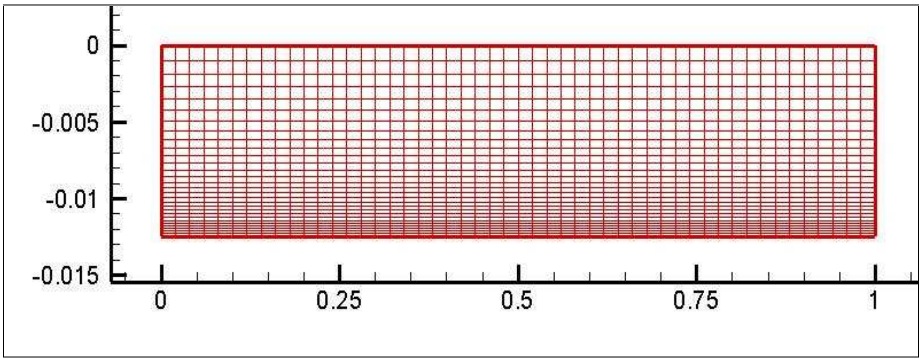


Figure 6.43 Grid used forlaminar flow in a pipe

The theoretical axisymmetric pipe flow velocity profile for laminar flow is given as:

$$u(r) = V_c\left[1-\left(\frac{2r}{D}\right)^2\right] \tag{6.1}$$

where V_c being the centerline velocity, r being radial position and D is the diameter of the pipe.For the equation (6.1), the velocity profile obtained from ENS2D and the corresponding theoretical velocity profile from Equation 6.1at x/L=0.5 are given in Figure 6.44.

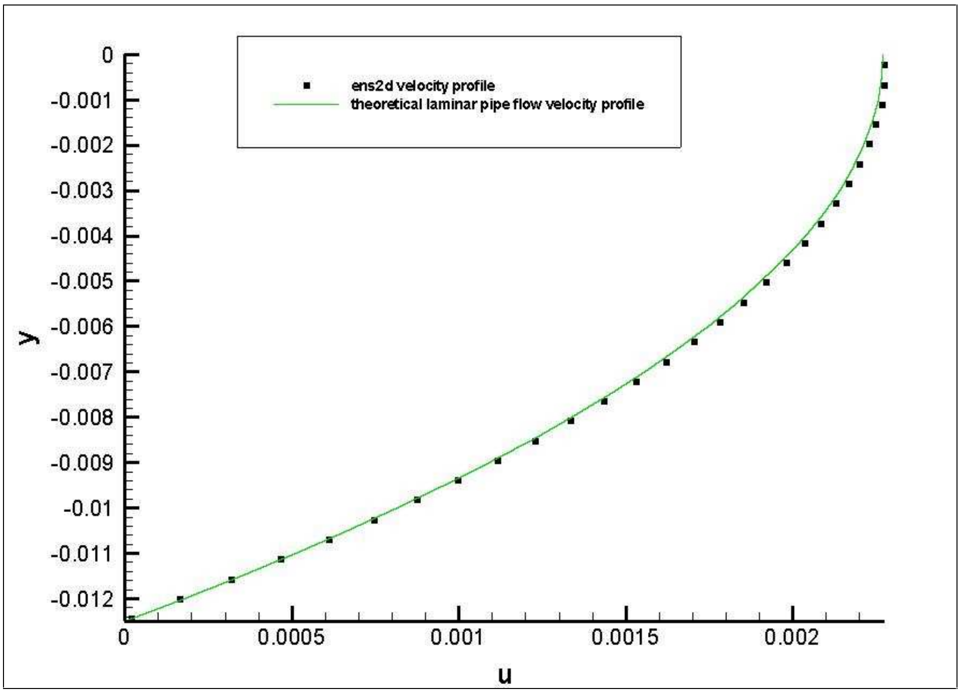


Figure 6.44 Velocity profile at x=0.5 where the fully developed laminar profile is achieved

The velocity profile of ENS2D compares quite well with the theoretical profile, indicating that the handling of viscous axisymmetric terms is acceptable. The velocity vectors for the pipe flow for the fully developed section is given in Figure 6.45.

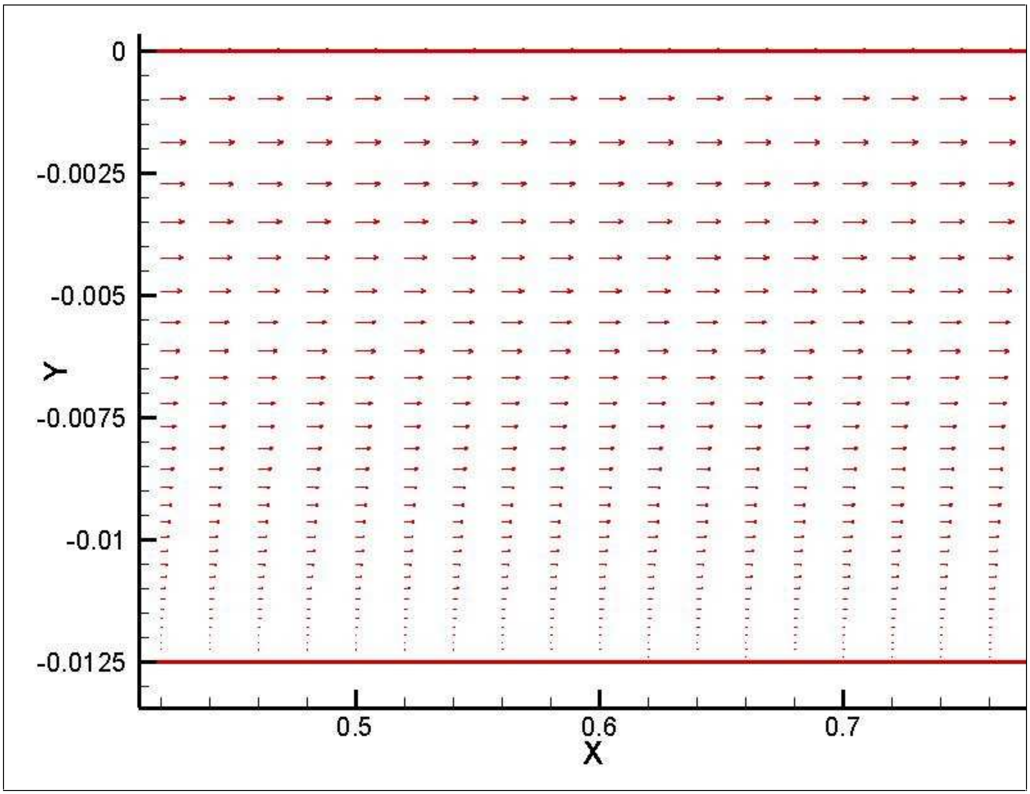


Figure 6.45 Velocity vectors in the fully developed region of axisymmetric pipe flow

Residual history for the current problem is given in Figure 6.46. As it can be seen from the figure, the residual decreases about 4 orders where the solution is assumed to be converged.

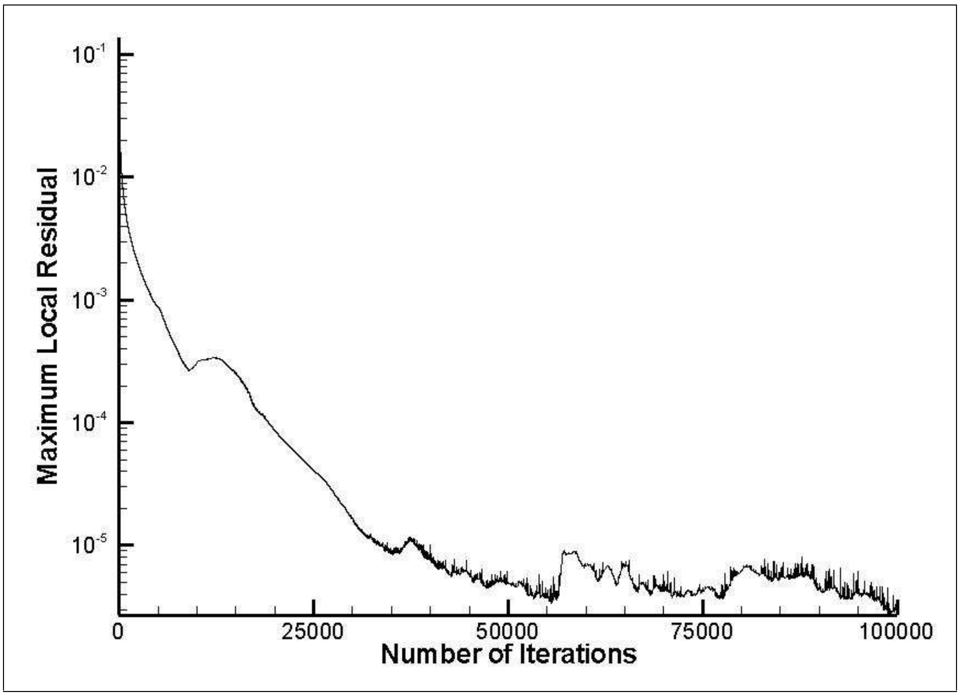


Figure 6.46 Residual history for the laminar flow in a pipe

As it is also done in the laminar flat plate solution, the pipe domain is decomposed into two blocks in order to obtain a multi-block solution for this problem. The pipe is divided from the mid-section, at x=0.5, and the solution is obtained using same boundary and free stream conditions. The results obtained with the multi-block solver again is the same results obtained with single block solver. Therefore, again, only residual history of the two-block solution is given in Figure 6.47.

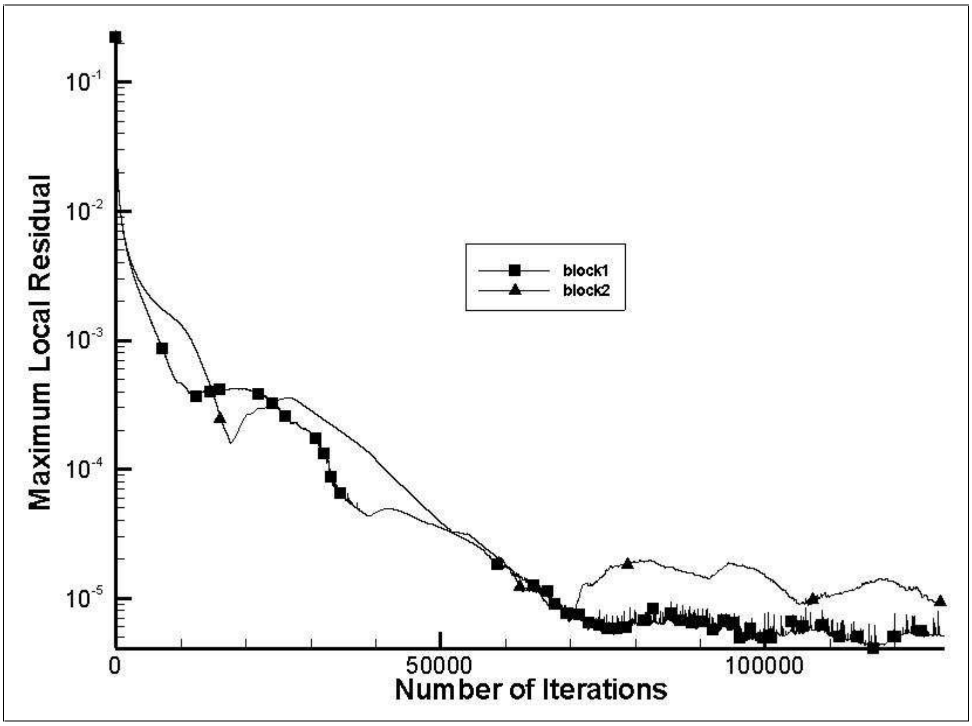


Figure 6.47 Residual history of multi-block solver

The residual of both blocks drop more than 3 orders, where the convergence is assumed to be obtained.

6.2.3 TURBULENT FLAT PLATE

For the verification of turbulent viscosity calculation, the same flat plate which is used for the laminar flow in Figure 6.39 is used. Free stream Reynolds number is taken as $6x10^6$ and free stream Mach number is 0.3. The transition location is 0.054

and the axial velocity profiles at several locations beyond this transition location are given in Figure 6.48 together with turbulent 1/7 power law $\frac{u}{U_\infty} = \left(\frac{y}{\delta}\right)^{\frac{1}{7}}$.

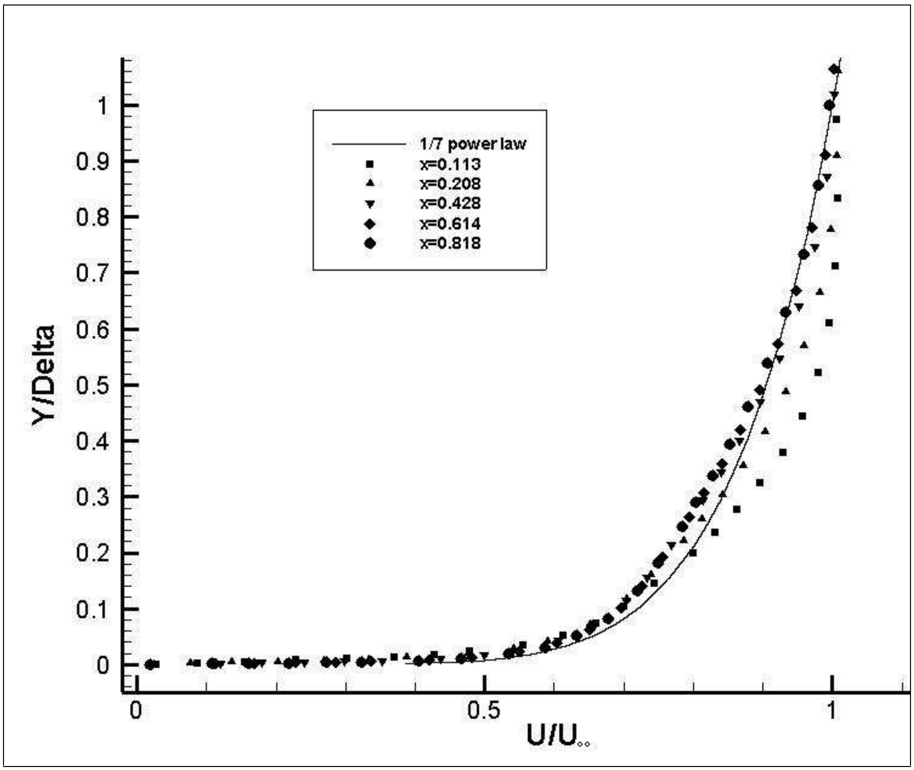


Figure 6.48 Non-dimensional velocity versus delta for turbulent flow over the flat plate at several axial locations

When Figure 6.48 is examined, it can be seen that the velocity profile is in harmony with the power law, emphasizing the Baldwin-Lomax turbulent model implemented to the solver works properly. Universal profiles together with the calculated turbulent similarity profiles are given in Figure 6.49, where laminar sublayer, logarithmic and wake regions obtained by the present solver' s solution can be easily seen for several axial locations along the flat plate.

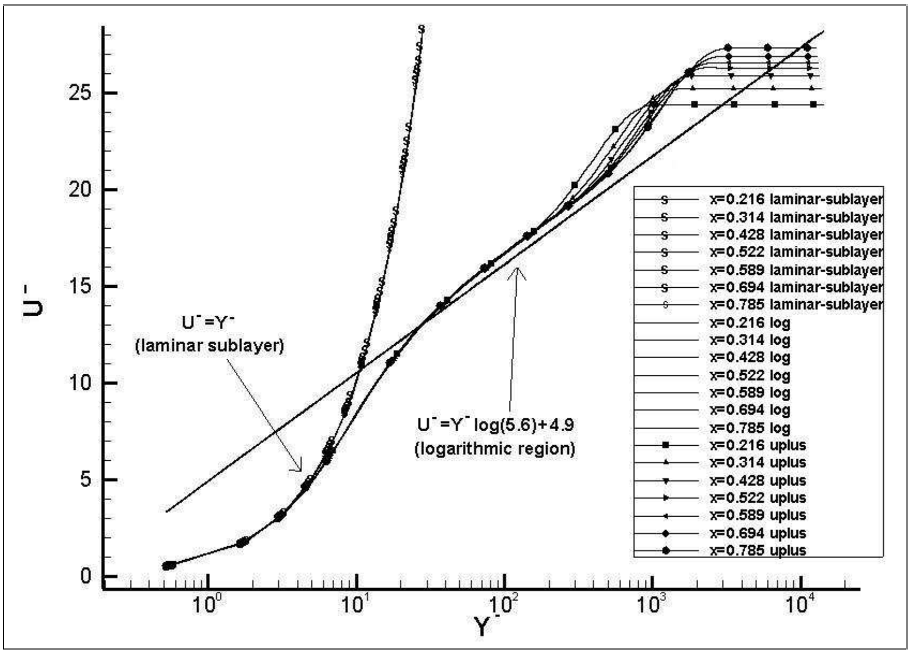


Figure 6.49 Turbulent similarity profiles compared with universal profiles

The residual history for the problem is given in Figure 6.50.

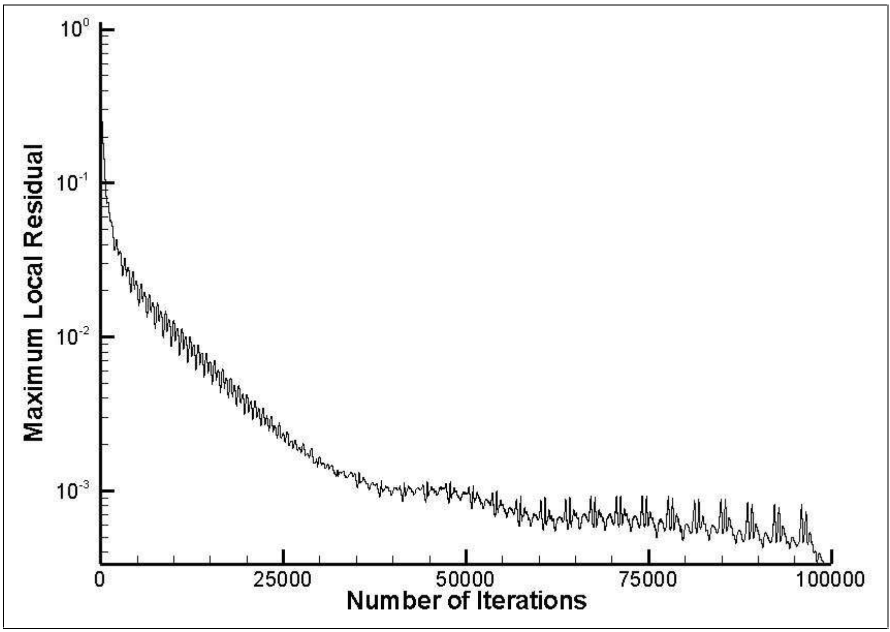


Figure 6.50 Residual history for the turbulent flow over the flat plate

The residual drops 3 orders where the solution is assumed to be converged. The same problem is also solved with multi-block algorithm and almost exactly the same velocity profiles are obtained. Blocks used are also exactly the same with the ones used in laminar flat plate solution. The residual history for the multi-block algorithm is given in Figure 6.51.

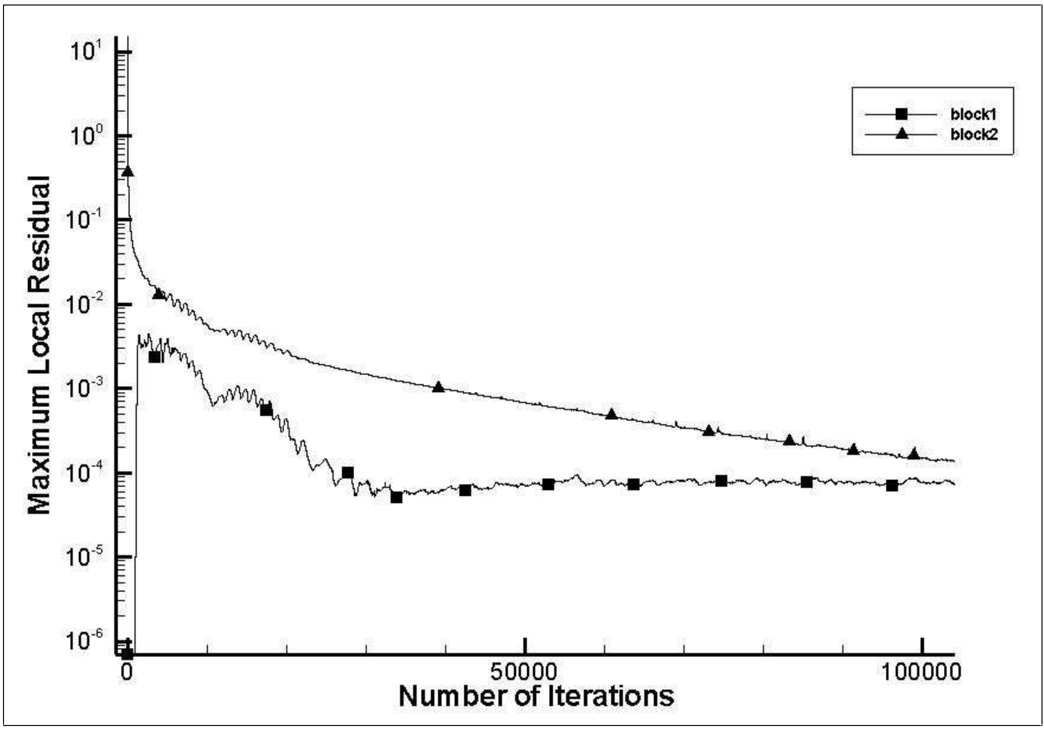


Figure 6.51 Residual history of multi-block solver for the turbulent flow over the flat plate

The same situation seen in the residual history of laminar flat plate can also be observed in Figure 6.51. Block 2, which has the boundary layer shows residual drop of 4 orders while Block 1 only converges to the residual value of Block 2.

As a comment for viscous flow solutions with multi-block algorithm, it can be said that multi-block algorithm works properly for laminar and turbulent problems either planar or axisymmetric.

6.3 MULTI-BLOCK SOLUTIONS OF T-108 TEST CASES

The results obtained by the present solver for several test cases involving inviscid and viscous flows for planar and axisymmetric geometries using single block show that the solution method for the governing equations is valid and provides satisfactory results in sense of engineering. Besides, viscous flow problems are also solved with multi-block algorithm and verified. To test the multi-block algorithm for internal flows, T-108 test cases solved with single block are resolved using multi-block approach.

Besides these test cases, a completely different problem is solved using multi-block algorithm, to test the solution method's robustness and to demonstrate the flexibility of the algorithm more clearly. This problem involves a section from a segmented motor. For large SPRM (i.e. having large diameter and longer combustion chamber), generally the combustion chamber includes multi-grain geometries to keep the burn area neutral, thus to acquire a neutral thrust-time curve. This situation brings a space between adjacent propellant grain structures. The space in between often requires special care in sense of fluid flow since they may cause instabilities [39], [40], [41] or even may change the flow regime improperly if their influence on the combustion chamber is not analyzed with enough accuracy.

For this reason, the solution of segmented SPRM' s generally need denser grids than other sections of the combustion chamber. When a complete combustion chamber is to be solved, this segment forms a small part of the complete domain and it is often challenging job to adapt a mesh to the segment or even impossible for generic mesh generator tools (e.g. Tecplot's Mesh Generator). The difficulty seen in creating a single-block mesh for such a geometry is tried to be illustrated by several attempts by

Tecplot 9.0' s Mesh Generator where the grids of multi-block geometries are created. At this point, multi-block algorithm steps forward and can use desired grid size for desired geometries.

The last test case to be solved includes a segment section together with a combustion chamber section and a converging-diverging nozzle. Since the goal is to demonstrate the flexibility of the multi-block algorithm for this kind of geometries and because of lack of experimental or numerical data, the solution is analyzed in the sense of convergence and general flow profile without comparing the results with any other data.

6.3.1 TEST CASE 1-B

To prevent repetition and since the goal in this section is to emphasize the effects of multi-block algorithm, planar flow solution of Test Case 1 (Test Case 1-a) is not solved but instead, axisymmetric flow solution is given. This geometry is divided into two sections, just through the middle of the complete geometry. The results obtained with the multi-block solver are compared with the single block solutions, again to be able to demonstrate the possible differences or problems rising from multi-block solver, if any. Since the Mach contours and pressure contours are sufficient for comparison for the results of the same solver, Mach and pressure variations along symmetry axis are not presented. It should be noted that same comparison method is used for all test cases except for Test Case 4 which is the most similar geometry to a real SPRM. In Test Case 4, the streamlines are also given to show the continuity between the blocks together with the Mach and pressure variations.

Mesh used for the current test case is given in Figure 6.52. As a common notation, blocks are numbered starting from the most left one as 1 and is increased as the axial position of blocks increase. The first block used in the solution has 60x21 grid points while the second one has 51x21 grid points.

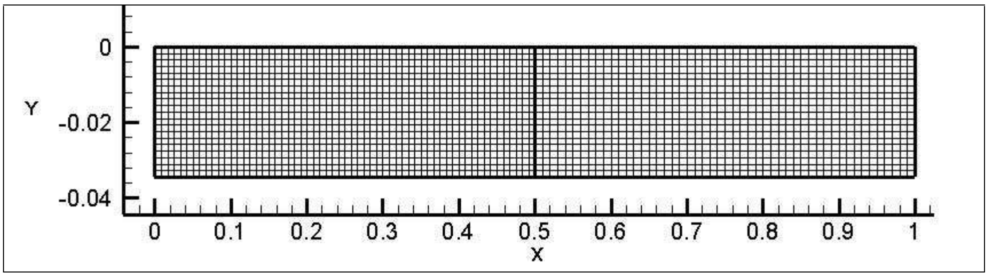


Figure 6.52 Blocks of Test Case 1-b

Figure 6.53 and 6.54 demonstrate the Mach and pressure contours of Test Case 1b, respectively. The contours on the top are the results of single block solver while the contours on the bottom are the results of multi-block solver.

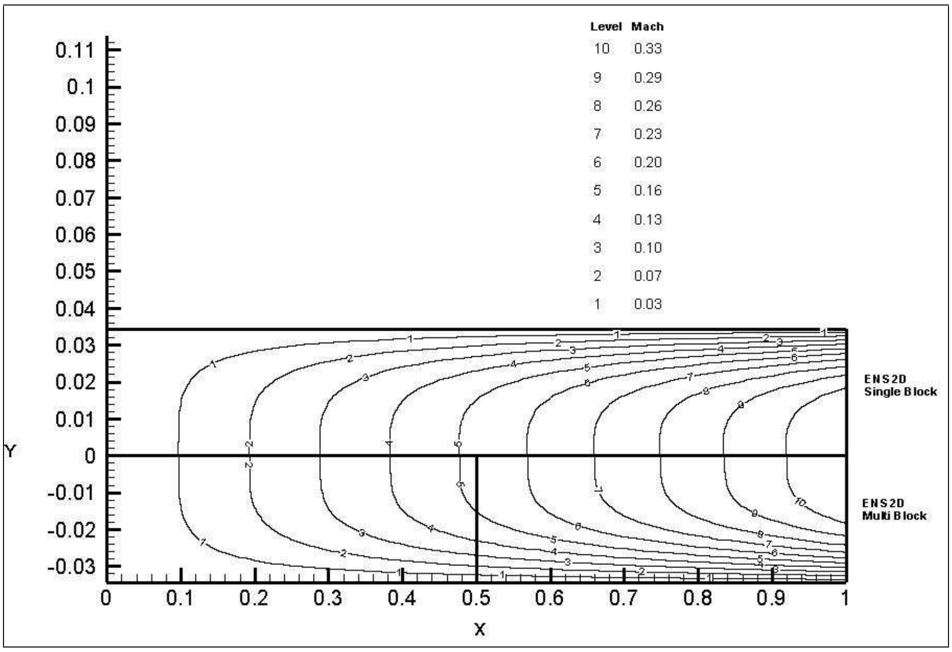


Figure 6.53 Mach number contours of single and two-block solutions for Test Case 1-b

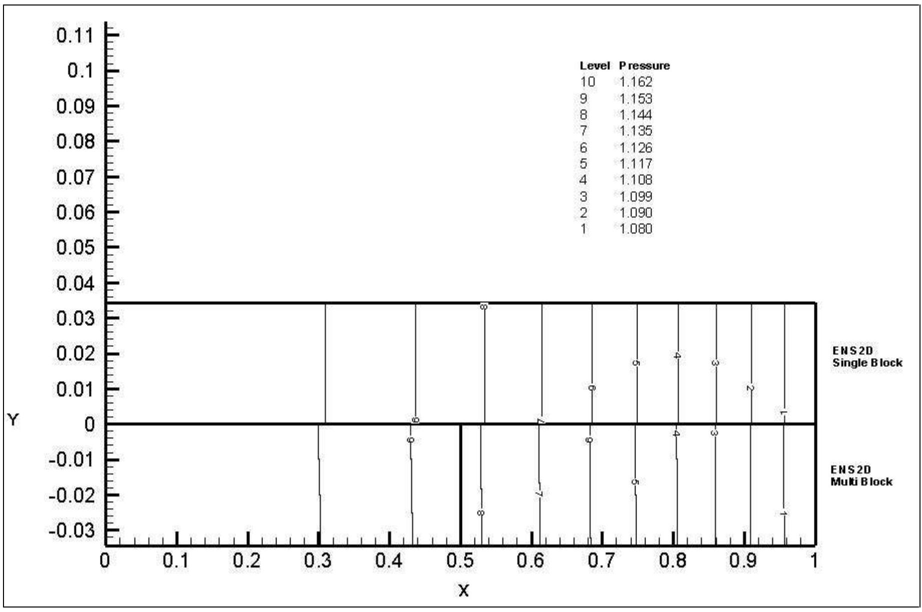


Figure 6.54 Non-dimensional pressure contours of single and two-block solutions for Test Case 1-b

As it can be observed from these figures, both the pressure and Mach contours show great similarities. For the two-block solution, the maximum value of Mach number is 0.358 on the top of the exit plane where maximum pressure value is 1.170 on the top of the wall. The same values are 0.359 and 1.171, respectively for the single block solution.

No discontinuity is observed through the adjacent boundaries of the two blocks, verifying that the multi-block algorithm is working properly. Since the results obtained by the single block solver are also in accordance with the results of reference [25], no obvious improvements in the sense of accuracy of the solver with the multi-block algorithm is observed, naturally.

Residual history for the present test case is given in Figure 6.55.

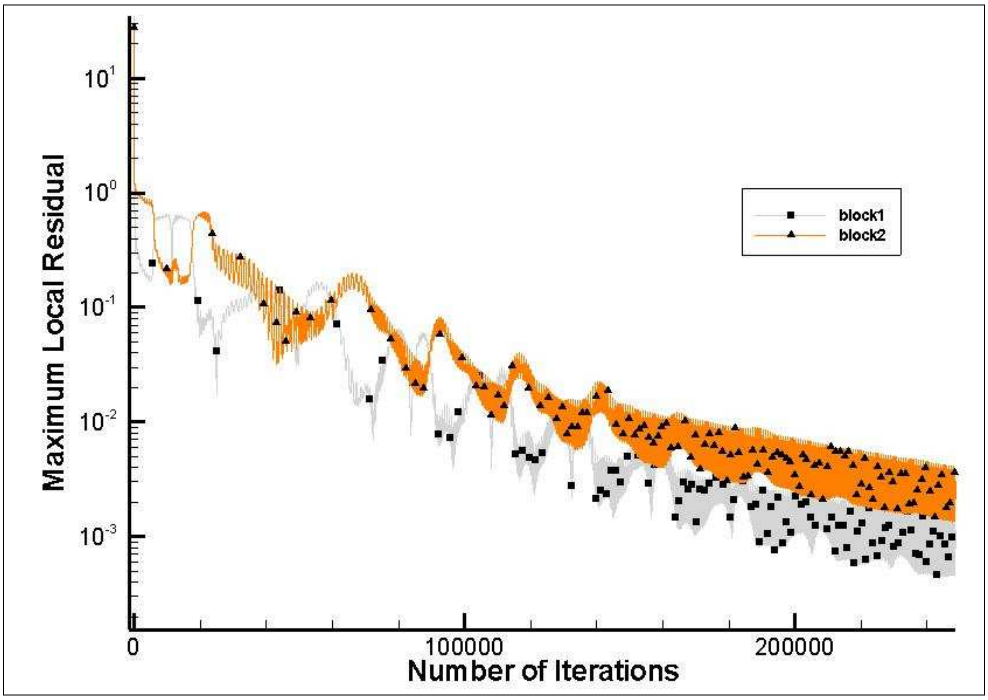


Figure 6.55 Residual history of the multi-block solution for Test Case 1-b

Comparing the residual histories of the single and multi-block solvers, it is clear that the residual of the multi-block solver drops more than that of the single block solver. Even though the slopes are different and it seems that convergence of multi-block solver is slower, it should be noted that the iteration number for one block is the half of what is being observed from the convergence history since two blocks are solved. Therefore, with this information, it can be said that the residual drops about 3 orders at the end of 75000 iterations and the convergence rate is quite close to that of the single block solver.

During the iterations, CFL number and artificial viscosity values have been changed, as in the some test cases solved with single block. Initially, both CFL and artificial viscosity are taken as 0.1, then both dropped to 0.05 after 12500 iterations.

After 30000 iterations, artificial viscosity is dropped to 0.01 and finally after 70000 iterations, CFL number is increased to 0.1 and artificial viscosity is dropped to 0.001. The final CFL and artificial viscosity values, which were used in the single block solver throughout the solution, cause instabilities in the intersecting boundaries of the two blocks and diverge the solution. Even the convergence rates are close, the multi-block solver is not as robust as single block solver in the sense of convergence behaviour for subsonic flows.

6.3.2 TEST CASE 2

Meshes used for the multi-block solution of Test Case 2 are given in Figure 6.56. Block 1 has a mesh size of 51x16 where Block 2 has 66x16.

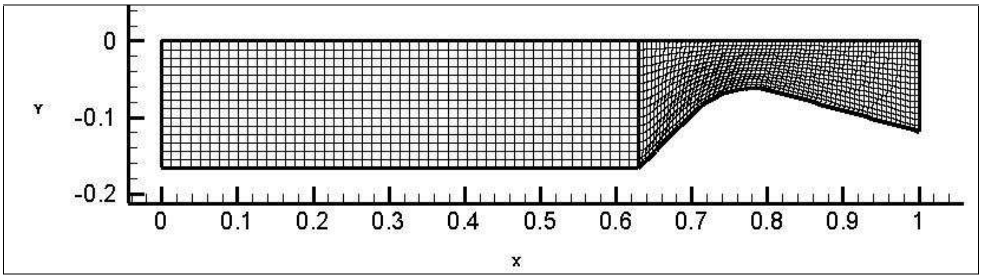


Figure 6.56 Grids of two blocks of Test Case 2

For Test Case 2, Mach number and pressure contours are compared with the results single block solver in Figure 6.57 and 6.58, respectively, with the one on the top being single block solution while the one on the bottom being multi-block solution.

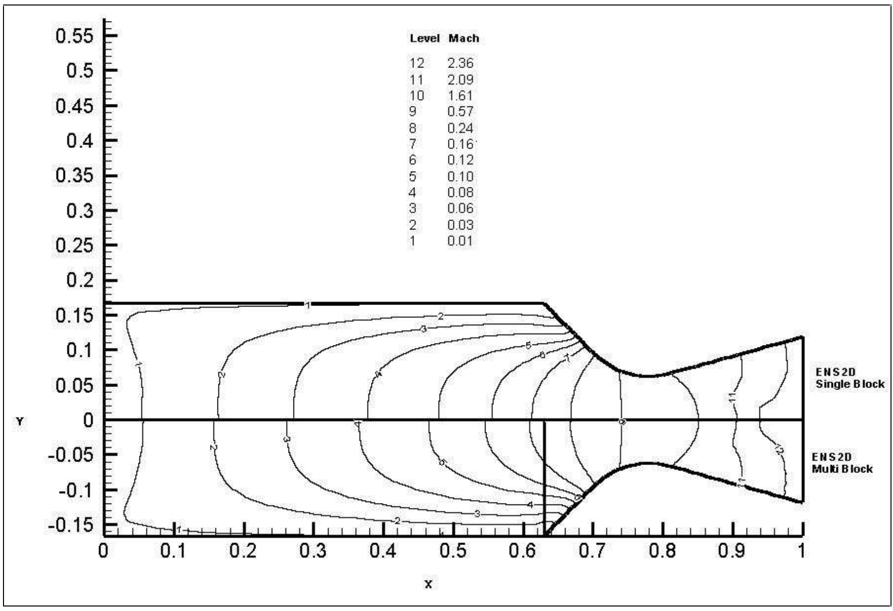


Figure 6.57 Mach number contours of single and multi-block solutions for Test Case 2

As observed from Figure 6.57, it is clear that the Mach contours are quite similar to that of the single block solution. In the subsonic region, even though there seems that there are some differences in the contours, it should be noted that the Mach number values in these regions are below 0.1, being quite low. This kind of difference can be expected for very slow flows in the subsonic region. The maximum Mach number in the whole geometry is 2.498, which is quite close to the value obtained by single block solver which is 2.503.

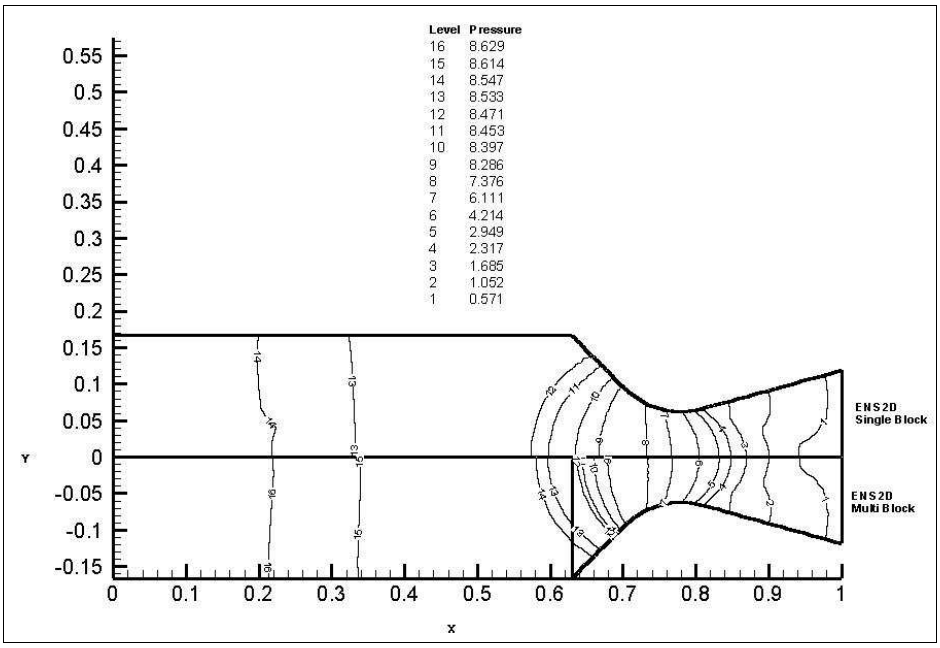


Figure 6.58 Non-dimensional pressure contours of single and multi-block solutions for Test Case 2

When the pressure contours are examined, contours are also quite close to each other. In the supersonic region, the results are much closer than that of the ones in the subsonic region. Minimum pressure value of 0.447 is obtained at the top section of exit plane while the maximum pressure value of 8.640 is obtained at the most left of the symmetry plane. The same values for single block solver' s results are 0.438 and 8.553, respectively for the single block solution.

One of the most important issues when applying a multi-block algorithm is the transition of the values from one block to the other, i.e. continuity of the solution along the interfaces. As it can be observed from Figure 6.57 and 6.58, this continuity has been conserved between the two blocks. No distinguishable gradients can be observed between the blocks, indicating that the multi-block solver can solve the whole geometry as a single block solver.

Convergence history for this test case is given in Figure 6.59.

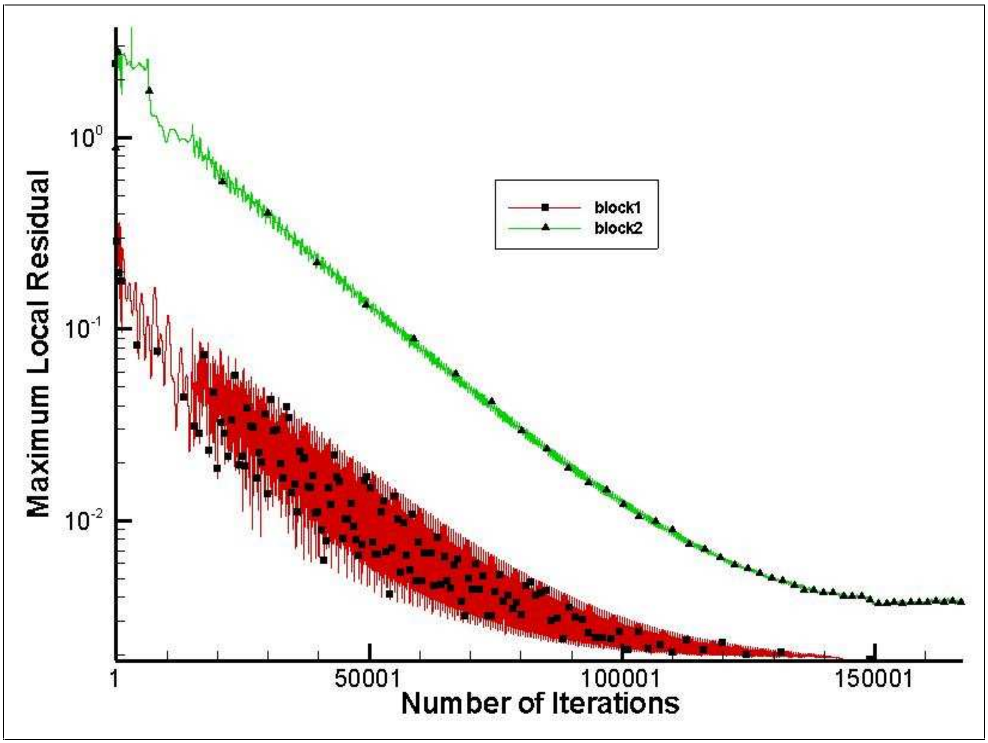


Figure 6.59 Residual history of multi-block solution for Test Case 2

When Figure 6.59 is compared with Figure 6.22, it can be observed that the convergence rates of single and multi-block solutions are again close to each other. However, the residual of the single block solution drops more than 3 orders while the residual of the multi-block solver drops about 3 orders for each block. Besides, one can observe that there are some oscillations in the residual history of Block 1. But these oscillations start to fade out after 50000 iterations for each block. This difference most probably rises from the interface of the blocks. In the end the reached convergence state is quite satisfactory.

During iterations, block 1, which is always subsonic, goes through some disturbances because of the existence of interfaces. This is an expected situation since the continuity is not completed till convergence is reached, thus both block has some incomplete interface boundaries in the sense of continuity until convergence is obtained. Block 1 carries these disturbances inside its domain, which is subsonic, and this effects the residual history in this block. On the otherhand, since Block 2 is mostly supersonic, the effect of this problem to Block 2 is much smaller than that of Block 1 and Block 2 seems to have a smoother convergence. Also, it is interesting that block1 reaches a lower residual value when compared to block 2 but the difference is about a quarter order, which not so significant.

CFL number and artificial viscosity are initially taken as 0.3 and 0.05, respectively. After 15000 iterations, they are both decreased to 0.1 and 0.02, respectively. It should be noted that this test case converged easier than Test Case 1 most probably due to the existence of supersonic flow.

6.3.3 TEST CASE 3

Being solved Test Case 1 and Test Case 2 by dividing them into two blocks, number of blocks used for Test Case 3 and 4 are increased to three. Furthermore, one of the interfaces of Test Case 3 is at the throat section of the converging-diverging nozzle to test the effects of interface boundaries on transition regions.

The grid of Test Case 3 is given in Figure 6.60. The grid sizes in the first, second and third blocks are 21x16, 101x16 and 41x16, respectively. In the second block, denser grid has been used since the inclined surface can effect the solution. Because of the interface at the throat and the narrowness of the throat, this test case was the most challenging test case in the sense of the capabilities of the multi-block algorithm within the test cases in reference [25]. This test case is solved to test the effect of the location of the interface on the solution.

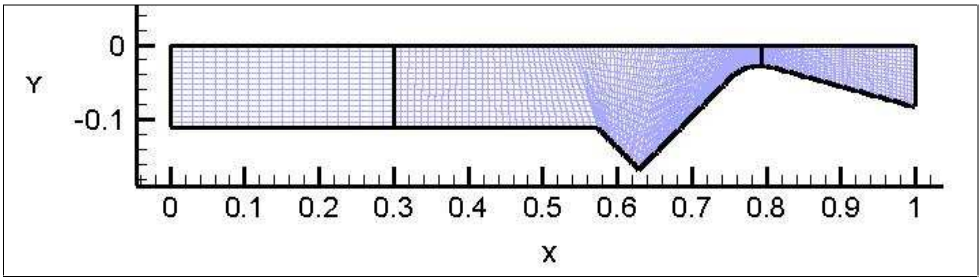


Figure 6.60 Blocks of Test Case 3

Mach and pressure contours obtained for Test Case 3 by the multi-block solver are given in Figure 6.61 and 6.62, respectively.

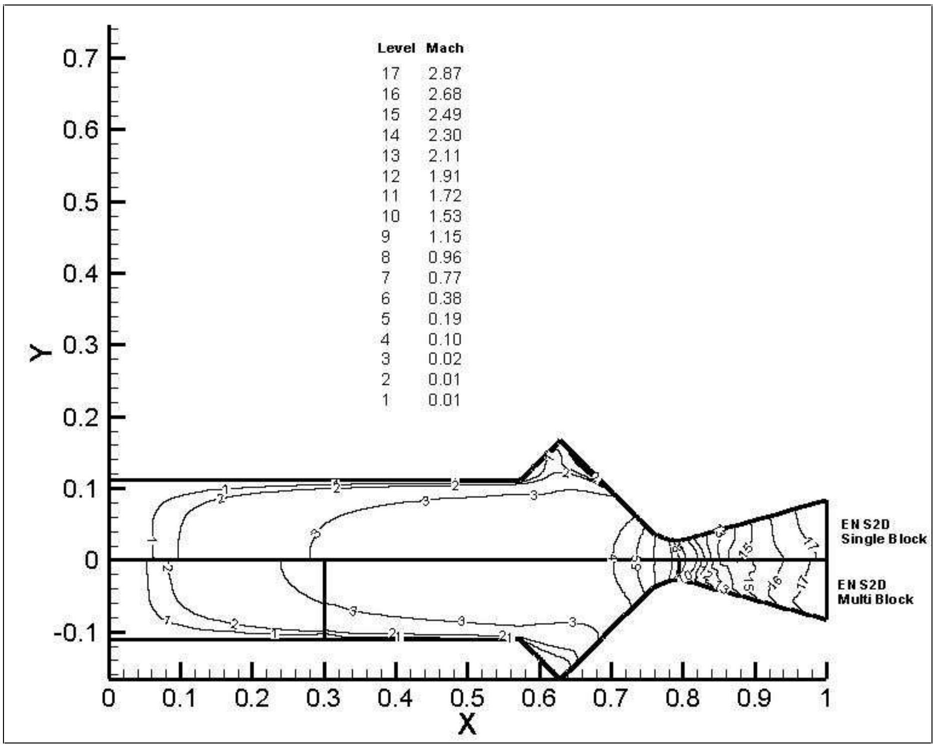


Figure 6.61 Mach number contours of single and multi-block solutions for Test Case 3

When the Mach contours are examined, differences are observed in subsonic region similar to Test Case 2. These differences are for the Mach number values below 0.1 and these kind of small deviations are expected for very slow flows in the subsonic region. In the diverging part of the nozzle, a compression wave reflects from the symmetry plane which is most probably carried out by the small discontinuities created at the throat interface but fades out as the wave reflects from the symmetry plane. Maximum Mach number value obtained with the multi-block solver is 2.99 while it is 2.94 for the single block solver, with a difference of 1.57%.

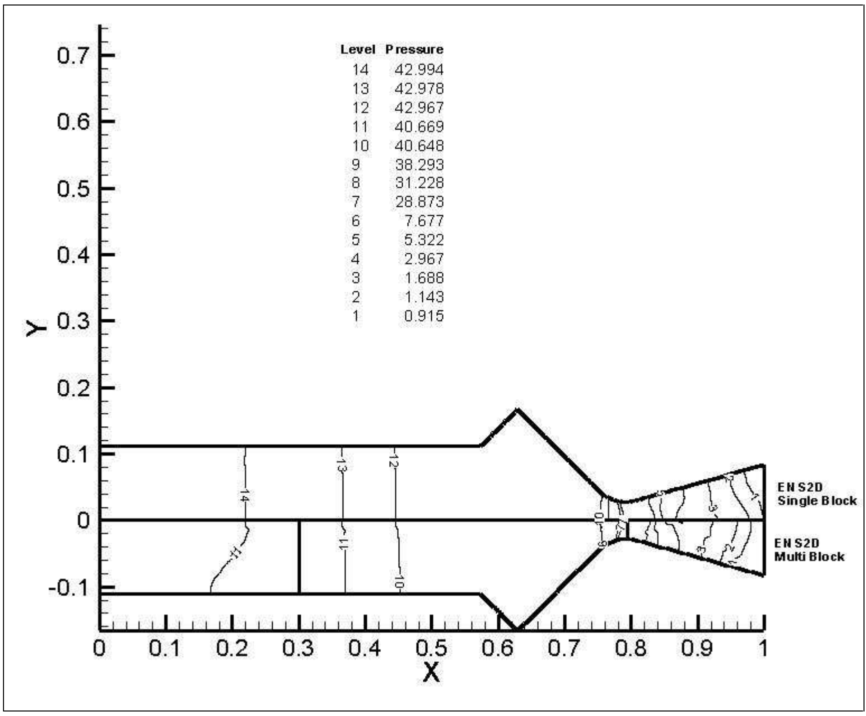


Figure 6.62 Non-dimensional pressure contours of single and multi-block solutions for Test Case 3

Pressure contours are in better agreement than the Mach number contours. Maximum pressure value obtained by the multi-block solver is 41.602 while it is 42.835 for the single block solution. The difference is about 2.83%, which is higher when compared to previous test case results. When compared to other test cases, it is obvious that dividing the geometry from the critical axial locations can effect the solution and increase the error. The small perturbations on the interface boundaries can magnify since the flow is affected in large extent from these critical boundaries.

The residual history for this test case is given in Figure 6.63. As it can be observed, residuals for Block 2 and 3 drop only about 2 orders while residual of Block 1 drops 3 orders. Several values for CFL and artificial viscosity values are tested and the best results are obtained for 0.2 and 0.04, respectively. Initial values of CFL number and artificial viscosity were 0.04 and 0.3 respectively, and 0.09 and 0.35 were used after 13000 iterations. After the supersonic flow is obtained at the diverging part, the artificial viscosity is decreased to 0.04 and CFL number is increased to 0.2. Artificial viscosity is tried to be kept as minimum as possible, in order to not to apply smoothing to the results.

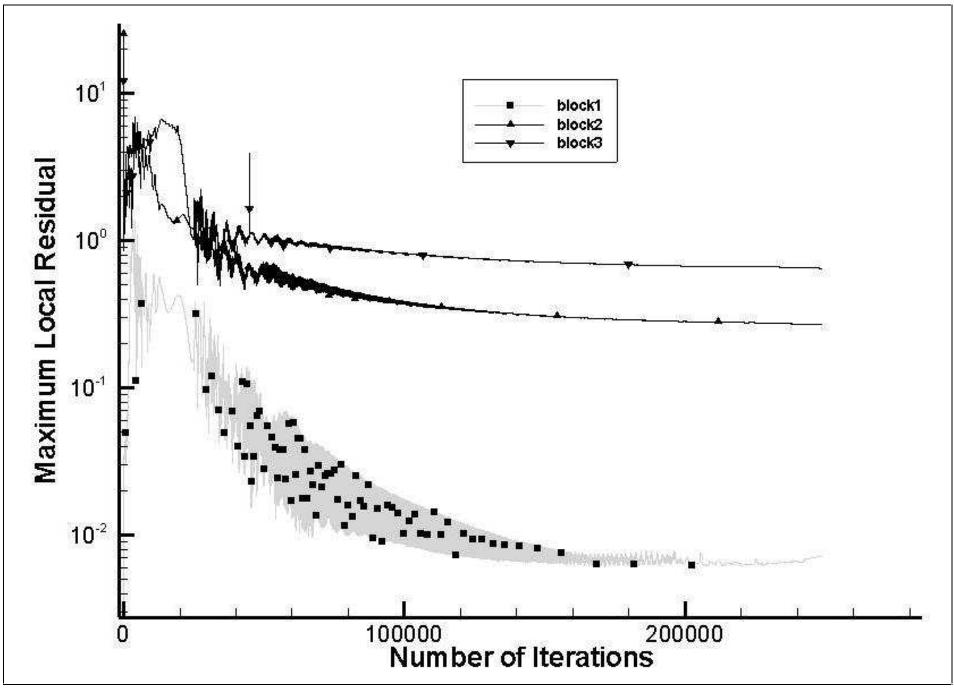


Figure 6.63 Residual history of multi-block solver for Test Case 3

This test case was the most challenging test case among the test cases in reference [25]. Although the results obtained show some deviations, the solver was able to handle the problem and also able to converge the solution to some extent. When the results are examined for the whole domain, contours lines are as expected except some numerical differences. These differences can be corrected by an algorithm which increases the convergence and decreases instabilities of the residual history.

6.3.4 TEST CASE 4

The entire geometry is divided into 3 blocks and each block is solved separately. The gird used is given in Figure 6.64. The grids for Block1, Block 2 and Block 3 are 21x16, 71x16 and 97x16, respectively.

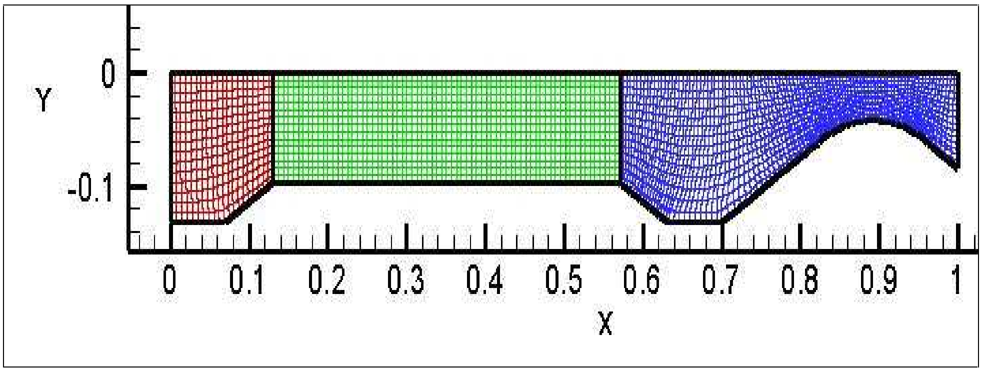


Figure 6.64 Blocks of Test Case 4

Mach number and pressure contours for Test Case 4 are given in Figure 6.65 and 6.66, respectively.

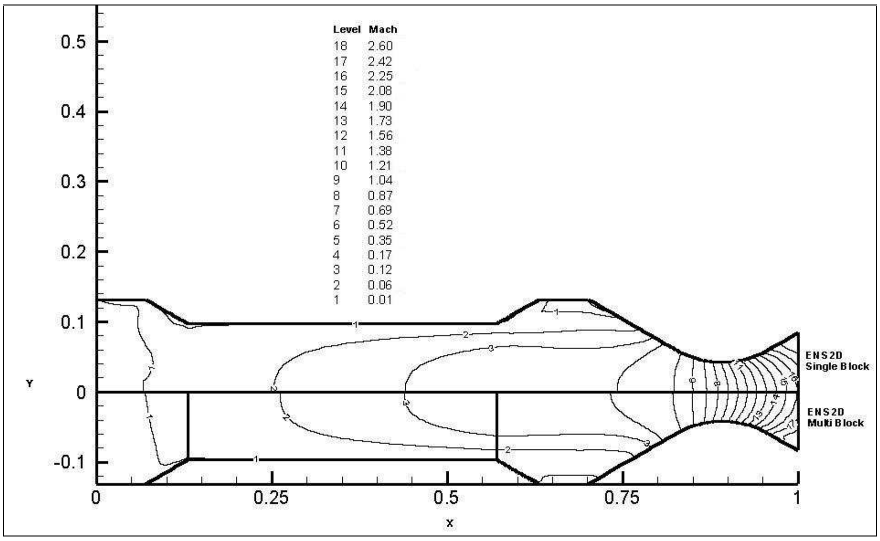


Figure 6.65 Mach number contours of single and multi-block solutions for Test Case 4

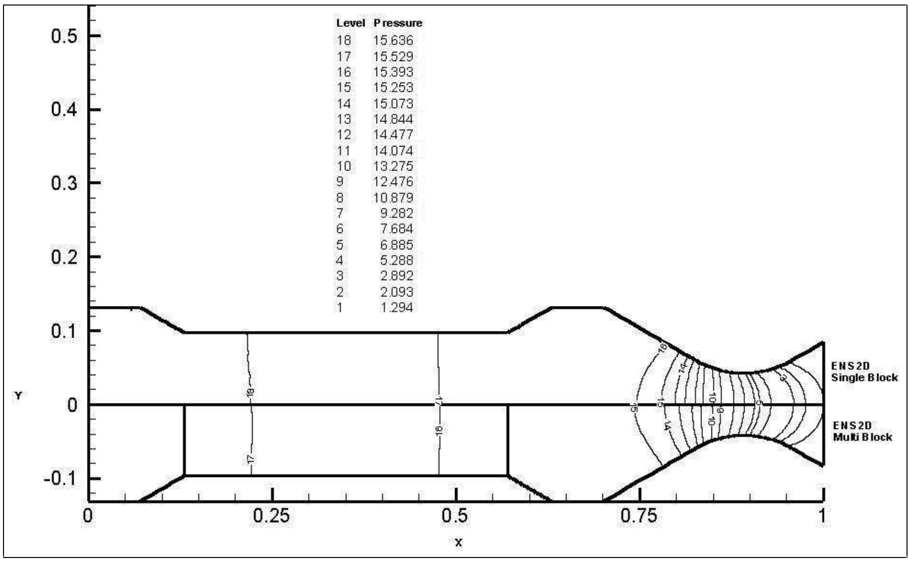


Figure 6.66 Non-dimensional pressure contours of single and multi-block solutions for Test Case 4

When Figure 6.65 and 6.66 are examined, it can clearly be observed that the solutions are quite similar. Maximum Mach number values obtained for the multi block solver and single block solver are 2.2652 and 2.2674 respectively, with a difference below 0.1%. The maximum pressure values are 15.55 and 15.65 for the multi-block and single block solver, respectively, with a difference of 0.6%. Analyzing the differences, this test case is solved as successful as by the single block solution. Mesh generation quality of the Tecplot 9.0 Mesh Generator was sufficient and convergence is obtained faster than Test Case 3, most probably due to quality of the mesh. To illustrate the continuity between the blocks, streamlines are given for both solutions in Figure 6.67. It can be clearly seen that the solution is complete with no discontinuities. Since this test case is the most SPRM-like test case and the three block solution is satisfactory, the multi-block code can be used for real SPRM applications.

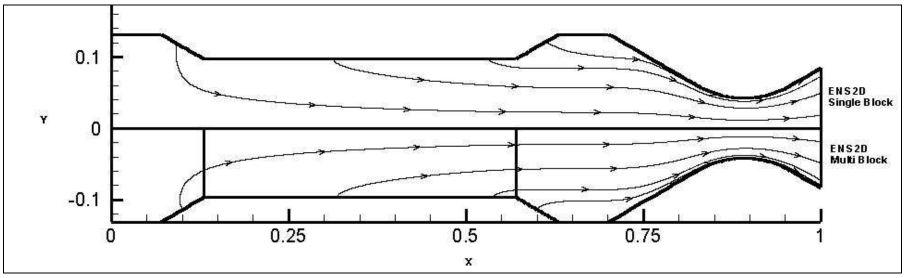


Figure 6.67 Streamline comparison of single and multi-block solutions for Test Case 4

The residual history of the Test Case 4 is given in Figure 6.68.

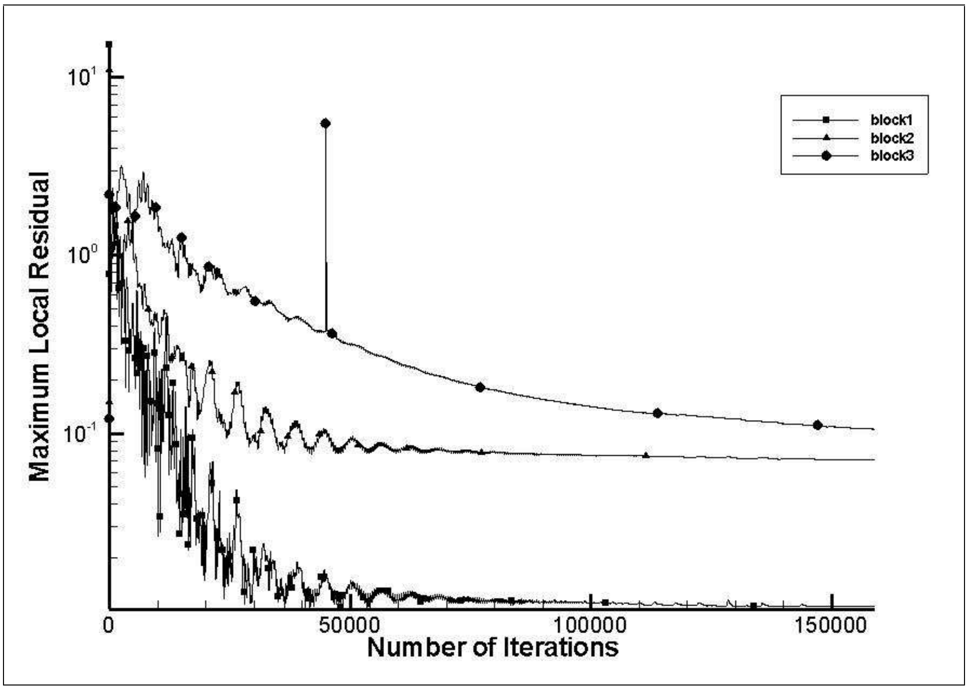


Figure 6.68 Residual history of multi-block solver for Test Case 4

When the residual history of single block solver is compared with the multi-block solver, it is obvious that single block solver has one more order of decrease. But in the overall, both three blocks has about 2 orders of decrease in residual after 50000 iterations.

6.4 SEGMENTED MOTOR

As a final problem, a SPRM of two segments is presented. In SPRM production, improvements are most probably exposed to dimensional restrictions in the sense of propellant casting or case production. Solving these restrictions by increasing the geometrical capabilities of the production facilities is not always a good problem since the investment cost would increase causing the improvement to be inefficient. Instead, a multi-segment motor can be produced with mechanical connections between these segments with the present capabilities. This will both increase the efficiency while decreasing the production cost and also will provide the flexibility of choosing desired motor length and size.

Today, most of the long-range SPRM' s which are used both for military and aerospace missions are formed from several segments (eg. Ariane Space Program of European Union). But these multi-segmented motors need special attention in the sense of analysis of the internal flow because of the space present between the segments which can effect the flow regime and connection items. This test case is a shifted version of Test Case 2, where it is assumed that the motor is formed of two segments with a space between them, with the same flow conditions. The main goal in solving this problem is to demonstrate the efficiency of the multi-block solver in solving these kind of problems. The geometry of the problem with injection boundaries is given in Figure 6.69 and 6.70:

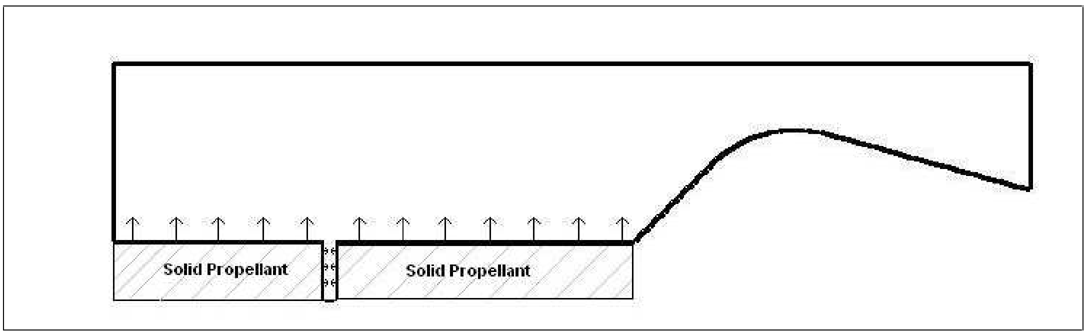


Figure 6.69 Injection boundary of segmented SPRM

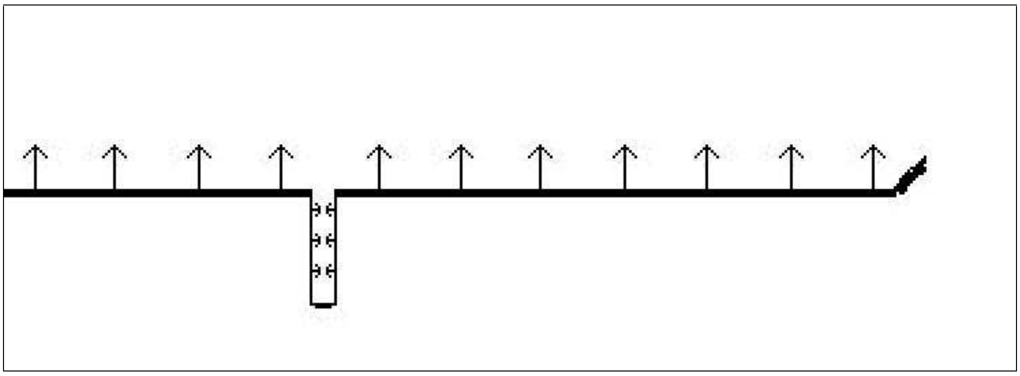


Figure 6.70 Enlarged view of the space between two segments of the segmented SPRM

The blocks with their grids on them are given in Figure 6.71.

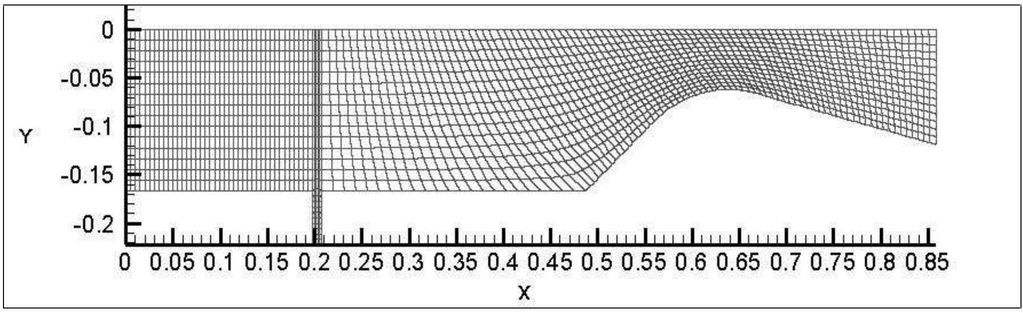


Figure 6.71 Geometry of segmented SPRM

Because of the gap between the two segments as shown in Figure 6.71, it is quite difficult to create a structured mesh. Several attempts with Tecplot 9.0 Mesh

Generator were not sufficient to obtain a satisfactory mesh structure. Examples of these unsuccessful attempts can be seen in Figure 6.72 to 6.74.

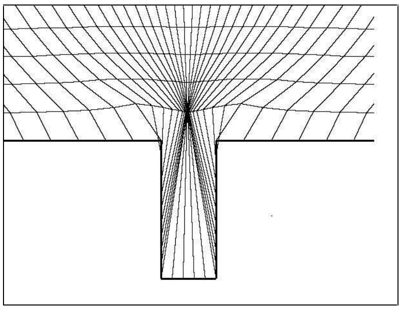


Figure 6.72 Grid generation attempt 1 for segmented SPRM

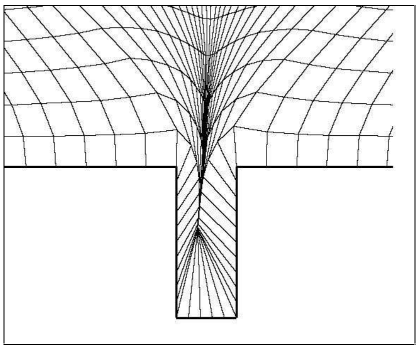


Figure 6.73 Grid generation attempt 2 for segmented SPRM

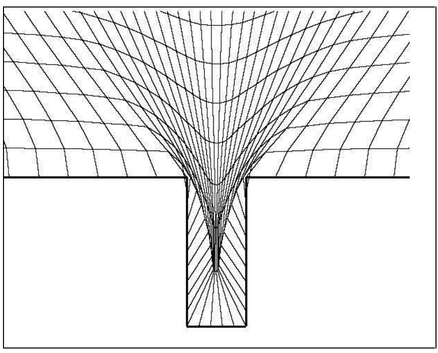


Figure 6.74 Grid generation attempt 3 for segmented SPRM

When Figure 6.72, 6.73 and 6.74 are examined, it can be seen that these meshes which are impossible to use, are created or meshes with high length-to-diameter are obtained which is not preferable to use in a CFD solver.

After the implementation of multi-block algorithm, it is quite easy to create a structured mesh for these kinds of geometries, in fact almost any kind of geometries. As it can be observed from Figure 6.75, the mesh in the middle block is created separately as the other two blocks. The connection of this block with the two blocks on the right and left hand sides are set within the solver.

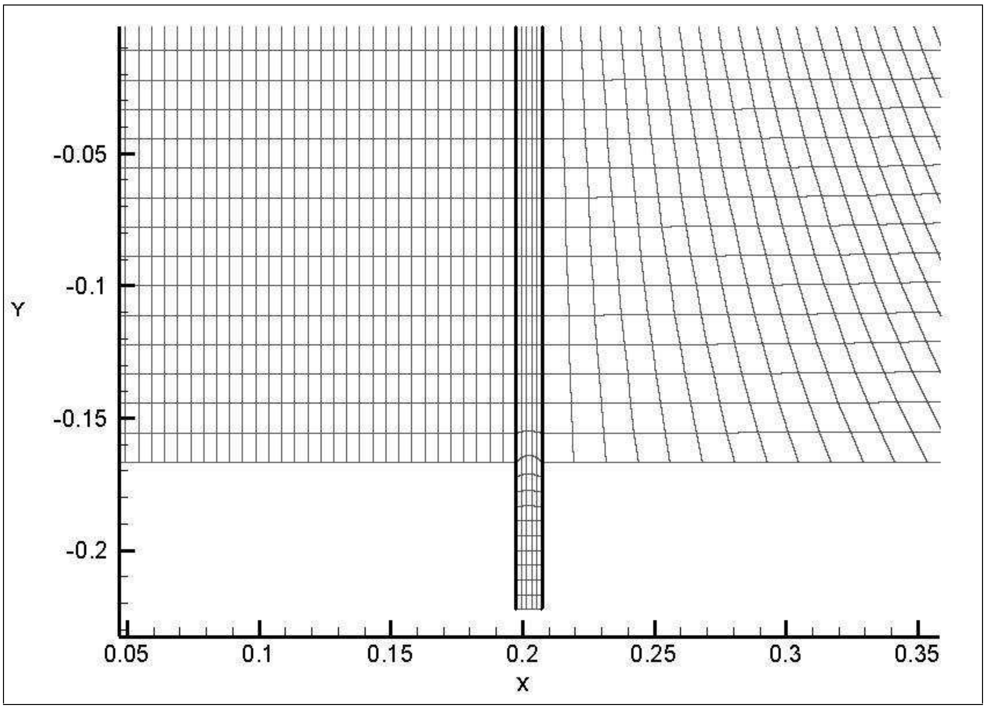


Figure 6.75 Enlarged view of mesh structure in Block 2

After defining the geometry and required connections within the solver, the solution of the problem is not different than the solution of a single cylinder. Mach contours obtained by using the same flow properties of Test Case 2 are given in Figure 6.76 and 6.77, involving contour of the whole geometry and an enlarged view of the space between two segments.

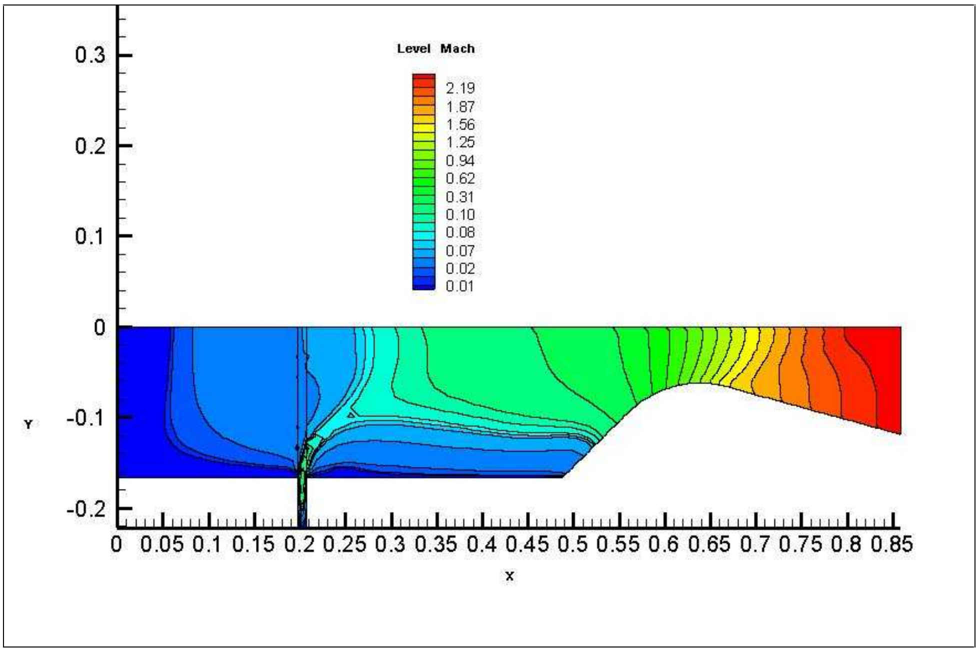


Figure 6.76 Mach contours of segmented SPRM

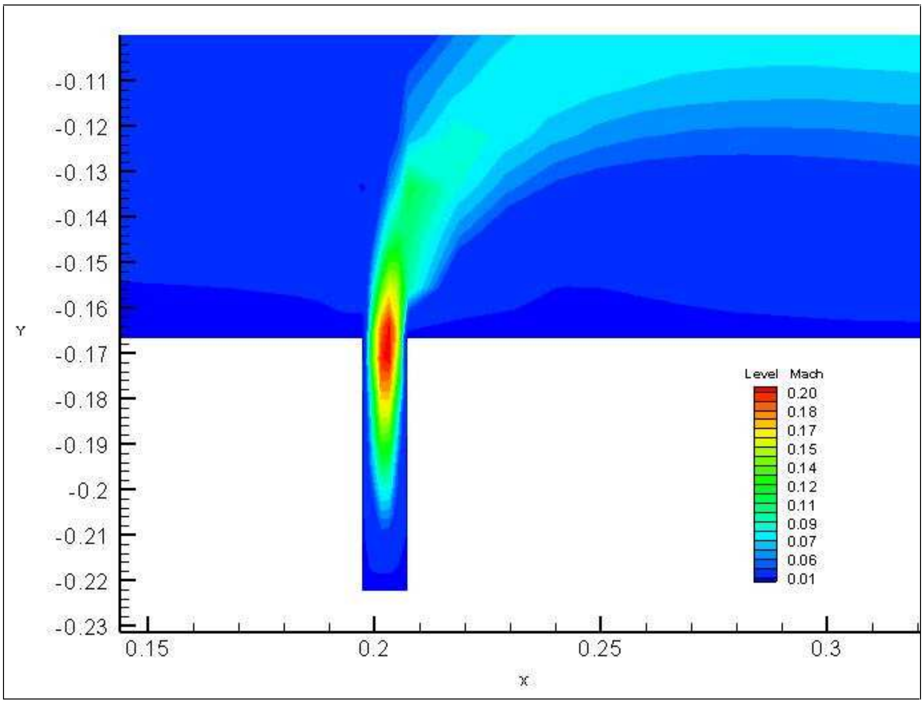


Figure 6.77 Enlarged view of Mach contours between two segments

Analyzing Figure 6.76 and 6.77, one can see that a faster flow is observed from the space between two blocks, block 2, relative to the injection surface perpendicular to the symmetry axis. This affects the flow regime but the maximum Mach number obtained at the exit plane is 2.498 where the same value for Test Case 2 was found as 2.503. The main difference is the pressure obtained at the gap which is 8.840 at the bottom of block 2, which is higher than the pressure value obtained at the intersection of symmetry plane and left wall, 8.153. This indicates that a relatively higher pressure is obtained in block 2, which may change the manufacturing requirements for the case and especially insulation thickness. The designer can rearrange the erosion models of insulation after the accomplishment of the internal flow simulation of SPRM, which is a situation that emphasizes the importance of an internal flow analysis of a SPRM.

Streamlines for this Test Case are given in Figure 6.78. The enlarged view of the block 2 where the flow turns from injection direction to the exit direction is shown in Figure 6.79.

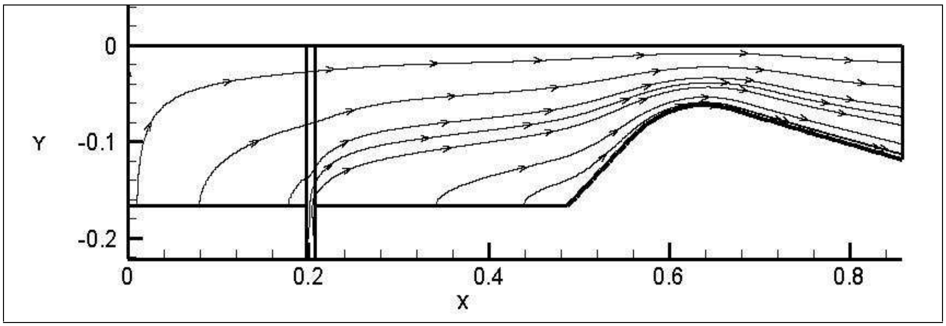


Figure 6.78 Streamlines of segmented SPRM

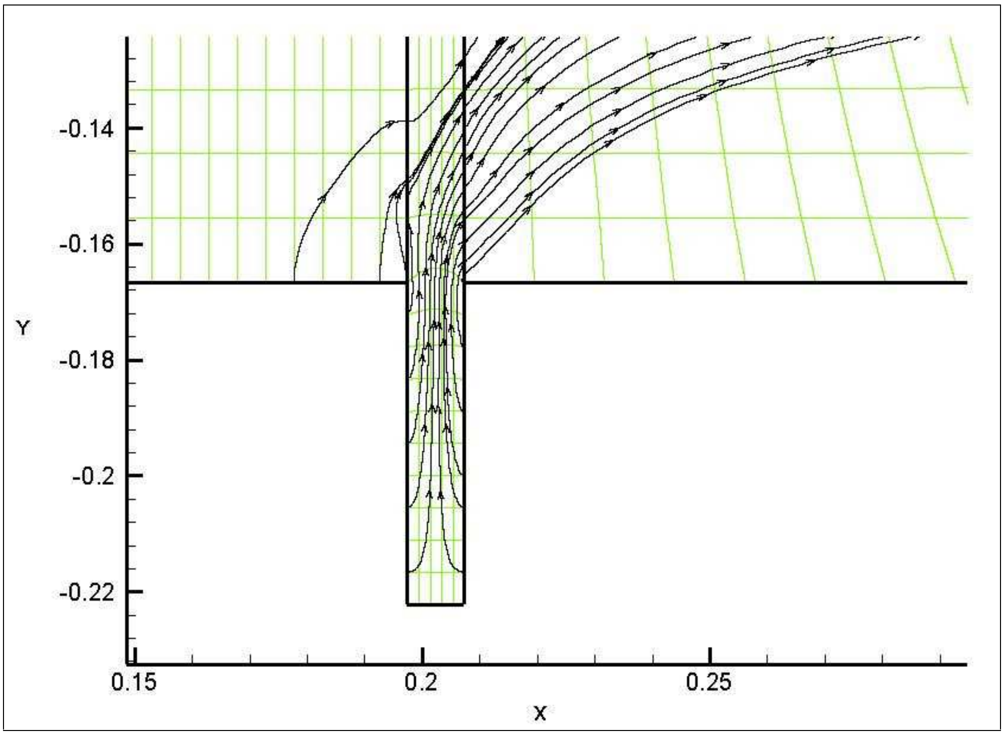


Figure 6.79 Streamlines of block 2 of segmented SPRM, enlarged view

Examining Figure 6.79, it can be seen clearly that the turning of the flow is achieved successfully. No discontinuity is observed along the interface of the blocks even though different mesh sizes are used for different blocks.

Residual history of this problem is given in Figure 6.80.

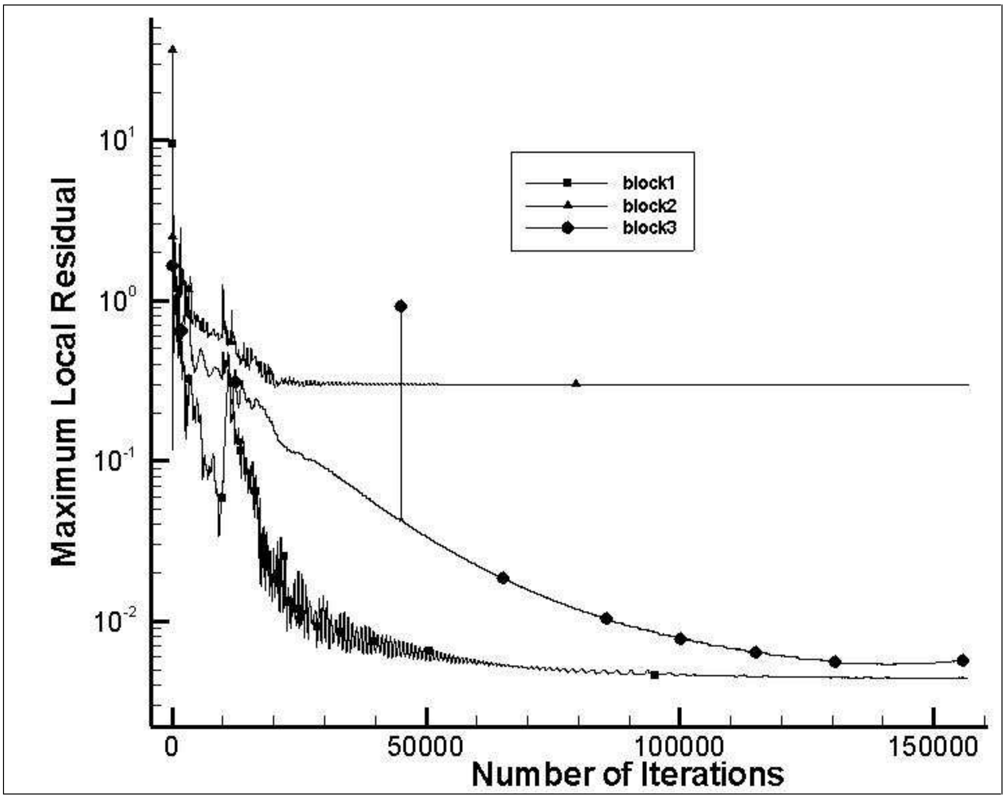


Figure 6.80 Residual history of segmented SPRM

Analyzing the residual history of the problem, it can be seen that residuals of all blocks dropped about 2 orders. For such a problem which involves a complex flow with a gap between the segments, the residual histories are quite satisfactory, especially that of block 2. This problem shows that the multi-block algorithm can be used for segmented SPRM' s quite easily and rapidly with sufficient accuracy.

6.5 AN OVERLOOK TO THE MULTI-BLOCK ALGORITHM

The observations and differences from the single block solver observed during the solution of the problems can be summarised as below:

- Each block has to be defined separately by the pointer algorithm, each having its own boundary condition and flow properties if there is a difference (inviscid, viscous etc.)

- The order of block definition to the solver is important since the interface matching is done with a fixed order.

- For the boundaries corresponding to the block interfaces, one should be careful to use indifferent boundary conditions.

- The possible transition sections should not be the interface boundary section if possible since this can create instabilities during the solution or even affect the results.

- Computational time naturally increases since in completion of every time step for one block, the block parameters are stored in a separate array and the stored array is assigned to arrays of the incoming block. Thus an extra computational effort is necessary for every step. Besides, the number of blocks used in the solution increases the memory usage. Also with the block number check points in various points of the solver which is inevitable, the computational time needed for a test case also increases with the present multi-block algorithm.

- The convergence is not influenced dramatically either positively or negatively. The code preserves its convergence behaviour after the implementation of multi-block algorithm. Different artificial viscosities can be applied to accelerate convergence as long as they are equated after a certain point if necessary. Provided that they are equated and the difference is not big, even different CFL numbers can be used during the iterations. These can be taken as advantages of the multi-block algorithm.

CHAPTER 7

CONCLUSION

7.1 SUMMARY OF THE WORK

A two-dimensional axisymmetric compressible laminar or turbulent Navier-Stokes solver is developed for multi-block applications especially to be used for the simulation of internal flow of SPRM’ s. Discretization is done by finite volume method in which the inviscid flux terms are based on cell-vertex formulation whereas the viscous flux terms are based on cell-centered formulation. A pointer system is used for the definition of grid structure, geometry and boundary condition assignments which makes the solver independent of the geometry or grid structure. For laminar viscosity, Sutherland’ s formula is used and for turbulent viscosity, one equation Baldwin-Lomax turbulence model is used. Artificial viscosity is used to stabilize the solution. For the simulation of fuel injection in the SPRM’ s, injection boundary conditions are used with the assumption of constant flame temperature.

A multi block algorithm is added to the single block solver to enable the solution of multiple domains. The division process of the whole domain is done without any loss or addition in the geometrical dimensions and even at the interfaces of the blocks, the cell-vertex formulation is completely applied. One constraint of this method is that the interface boundaries should have same coordinates in order to preserve the continuity. But this situation does not effect the mesh generation since one can create any number of blocks to define the desired geometry.

The newly developed code is tested with several test cases for different purposes. For inviscid axisymmetric formulation with injection boundary implementation, test cases which were used as test cases in reference [25] are solved and the results are in close agreement with the compared solutions. Especially last test case of these test cases is a good example for internal flows in SPRM' s which is of primary importance for the present work. For viscous flows, laminar and turbulent flow assumptions are tested by comparing the results with the self-similar Blasius solution and turbulent 1/7 power law analogy respectively. Results are in good agreement and the laminar and turbulent viscosity as well as the planar viscous flux terms are verified by these comparisons. For the verification of axisymmetric flux terms, laminar pipe flow is solved and the results show excellent agreement with the theoretical velocity profile in an axisymmetric pipe flow. After the verification of the code for single block applications, some of the test cases solved by single block solver are resolved by multi-block algorithm, dividing the geometries into blocks from several locations. The results of multi-block solver are in good agreement with that of the single block solver, signifying the verification and accuracy of the multi-block algorithm. The effects of multi-block algorithm to the convergence regime are also analyzed. Finally, a segmented SPRM is solved emphasizing the flexibility and robustness of the multi-block solver and the results obtained are satisfactory for the analysis of the internal flow in a SPRM.

7.2 RECOMMENDATIONS FOR FUTURE WORK

The present solver is quite satisfactory for the global features of internal flow in SPRM' s with complex geometries. For future developments, an implicit residual smoothing model can be implemented to the solver to accelerate the convergence.

The present solver may need user input to prevent instabilities or to have a properly decreasing convergence history.

A two-equation turbulence model can be implemented in order to eliminate the directional properties rising from algebraic Baldwin-Lomax model. Besides, due to lack of test cases involving SPRM' s with turbulent flow assumption, experimental data should be obtained to verify the developed solvers.

Since the solver is developed mainly for SPRM applications, coupling the present solver with a grain burnback model can provide a complete SPRM solver which can impose the effects of internal pressure deviations to the burnback model. By this coupling, a quasi-unsteady SPRM solver can be obtained, thus creating an accurate and complete internal flow simulation for the complete burning time of SPRM' s.

Finally, since this solver can solve blocks separately with the implemented multi-block algorithm, it can be used for parallel processing to save computational time.

REFERENCES

[1] Hoffman, K. A., "Computational Fluid Dynamics for Engineers", Engineering Education System, Texas, 1989, (ISBN 0-9623731-4-1).

[2] Dunlap, R., Willoughby, P. G. and Hermsen, R. W., "Flowfield in the Combustion Chamber of a Solid Propellant Rocket Motor, AIAA Journal, 12, 10, 1440.

[3] Anderson, D.A., Tannehill, J.C., and Pletcher, R.H., "Computational Fluid Mechanics and Heat Transfer", Hemisphere Publishing Corporation, McGraw-Hill, 1984.

[4] Sabnis, J. S., Gibeling, H. J. and McDonald, H., "Internal Fluid Mechanics of Solid Propellant Rocket Motors", CPIA Publ., Vol 1, No. 412, (October 1984).

[5] Sabnis, J. S., Gibeling, H. J. and McDonald, H., "Navier-Stokes Analysis of Two- And Three-Dimensional Flow Field in Solid Rocket Motors with Segment Joints", AIAA Paper 87-1804, (1987).

[6] Murdock, J. W., Johnston, W. A., "Flow-Structural Interaction in Solid Rocket Motors", Internal Aerodynamics in Solid Rocket Propulsion, Public. Von-Karman Institute for Fluid Dynamics, 2002.

[7] Traineau J. C., Hervat P., Kuentzmann P., "Cold Flow Simulation of a Two Dimensional Nozzleless Solid Rocket Motor", AIAA Paper, No.86-1147, Alabama, June 1986.

[8] Vuillot F., "Numerical Computation of Acoustic Boundary Layers in Large Solid Propellant Space Booster", AIAA Paper, No.91-0206, Reno, January 1991.

[9] Vuillot, F., Dupays, J., Lupoglazoff, N., "2-D Navier-Stokes Stability Computations for Solid Rocket Motors: Rotational, Combustion and Two-Phase Flow Effects", AIAA Paper, No.97-3326, Seattle, July 1997.

[10] Lijewski, L. E., "Transonic Euler Solutions on Mutually Interfering Finned Bodies", <u>AIAA Journal</u>, 28, 982, (July 25, 1989).

[11] Atkins, H. L., "A Multi-Block Multigrid Method for the Solution of the Euler and Navier-Stokes Equations for Three-Dimensional Flows, <u>AIAA Paper</u> 91-0101.

[12] Bush, R.H., "External Compression Inlet Predictions Using an Implicit Upwind, Multiple Zone Approach", <u>AIAA Paper</u> 85-1521.

[13] Arthur, M. T., Blaylock, T. A. and Anderson, J. M., "Cell-Vertex, Multigrid Euler Scheme for Use with Multiblock Grids", <u>AIAA Journal</u>, 29, 4, 507 (1990).

[14] Jenssen, Carl. B., "Implicit Multiblock Euler and Navier-Stokes Calculations"" <u>AIAA Journal</u>, 32, 9, 1808, (1994).

[15] Sheng, C., Whitfield, D. L. and Anderson, W. K., "Multiblock Approach for Calculating Incompressible Fluid Flows on Unstructured Grids, AIAA Journal, 37, 2, 169, (1999).

[16] Madavan, N. K. and Rai, M. M., "Computational Analysis of Rotor Stator Interaction in Turbomachinery Using Zonal Techniques", Applied Computational Aerodynamics, Progress in Astronautics and Aeronautics, 125, 481, (1990).

[17] Cambier L., Ghazzi W., Veuillot J. P. and Viviand H., "A Multi-Domain Approach for the Computation of Viscous Transonic Flows by Unsteady Type Methods", Computational Methods in Viscous Flows, pp 513-539, Pineridge, 1984.

[18] Jenssen C.B., and Weinerfelt P. Å., "Coarse Grid Correction Scheme for Implicit Multiblock Euler Calculations, AII Journal, 33, 10, 1816 (October 1995).

[19] Ghaffari F., Luckring J.M., Thomas J.L., Bates B.L. and Biedron R.T., "Multiblock Navier-Stokes Solutions About the F/A-18 Wing-LEX-Fuselage Configuration", Journal of Aircraft, 30, 3, 293, (May-June 1993).

[20] Carlson J.R., Pao P.S. and Abdol-Hamid K.S., "Computational Analysis of Vented Supersonic Exhaust Nozzle Using a Multiblock/Multizone Strategy", Journal of Propulsion and Power, 9, 6, 834, (Nov.-Dec. 1993).

[21] Giordano D., Ivanov M., Kashkovsky A., Markelov G. and Koppenwallner G., "Application of Numerical Multizone Approach to the Study of Satellite

Thruster Plumes”, Journal of Spacecraft and Rockets, 35, 4, 502, (July-August 1998).

[22] Rosenfeld M. and Yassour Y., “Fully Implicit Solution of Time-Dependent Partial Differential Equations in Arbitrary Multizone Domains”, AIAA Journal, 33, 5, 859, (May 1995).

[23] Grasso, F. and Marini, M., “Multiblock Implicit Total Variation Diminishing Solution of High-Speed Internal Flows, AIAA Journal, 9, 2, 255 (1993).

[24] Usab, W.J., "Embedded Mesh Solutions of the Euler Equation Using a Multiple-Grid Method", Ph.D. Thesis in Aeronautics and Astronautics, Massachusetts Institute of Technology, 1983.

[25] Yumuşak M., Taşkınoğlu E., and Tınaztepe T., “ 2-D Internal Flow Applicaitions for Solid Propellant Rocket Motors”, Technical Report of NATO-RTO T108 Support Project, 1999.

[26] Schlichting, H., "Boundary-Layer Theory", McGraw-Hill Book Company, 7th edition, 1987, (ISBN 0-07-055334-3).

[27] Baldwin, B.S. and Lomax, H., "Thin-Layer Approximation and Algebraic Model for Separated Turbulent Flows", AIAA Paper, 78-257, 1978.

[28] White, F.M., “Viscous Fluid Flow”, McGraw-Hill Book Company, 2nd edition, 1991, (ISBN 0-07-100995-7).

[29] Tınaztepe, H. T., "A Finite Volume Method for Compressible Viscous Flows”, Ph.D. Thesis in Mechanical Engineering, Middle East Technical University, 1997.

[30] Yumuşak, M., "Non-reacting and Reacting Flow Analysis in Propulsion Systems”, Ph.D. Thesis in Mechanical Engineering, Middle East Technical University, 2000.

[31] Taşkınoğlu, E. Ş., "Total Variation Diminishing(TVD) Methods for Internal Flows”, M. S. Thesis in Aeronautical Engineering, Middle East Technical University, 1997.

[32] Ni, R.H., "A Multiple Grid Scheme for Solving the Euler Equations", AIAA Journal , Vol. 20, No. 11, pp. 1565-1571, 1982.

[33] Crumpton, P.I., Mackenzie, J.A., and Morton, K.W., "Cell Vertex Algorithms for the Compressible Navier-Stokes Equations", Journal of Computational Physics, Vol. 109, pp. 1-15, 1993.

[34] Davis, R.L., Ni, R.H. and Carter, J.E., "Cascade Viscous Flow Analysis Using the Navier-Stokes Equations", AIAA Paper, 86-0033, 1986.

[35] Kallinderis, Y, "Algebraic Turbulence Modeling for Adaptive Unstructured Grids", AIAA Journal Vol. 30, No. 3, pp. 631-639. 1992.

[36] Radespiel, R., "A Cell-Vertex Multigrid Method for the Navier-Stokes Equations", NASA Technical Memorandum 101557, 1989.

[37] Tinaztepe, H. T., Yumusak, M. E. and Vuillot, F., “Validation of Internal Flow Prediction Codes for Solid Propellant Rocket Motors, RTO Meeting on “Advances in Rocket Propellant Performance, Life and Disposal for Improved System Performance and Reduced Cost, 2002: Aalborg, Denmark

[38] Mehta, U.B, ”Guide to Credible Computer Simulations of Fluid Flows”, AIAA Journal of Propulsion and Power, Vol. 12, No. 5, pp-940-948, 1996.

[39] Kourta, A., ”Vortex Shedding in Segmented Solid Rocket Motors”, AIAA Journal of Propulsion and Power, Vol. 12, No. 2, pp-371-376, 1996.

[40] Dotson, K.W., Koshigoe, S. and Pace, K. K., ”Vortex Shedding in a Large Solid Rocket Motor Without Inhibitors at the Segment Interfaces”, AIAA Journal of Propulsion and Power, Vol. 13, No. 2, pp-197-206, 1997.

[41] Prezoff, M., Vuillot, F. and Traineau, J.C., ”Vortex Shedding Driven Oscillations in Subscale Motors for the Ariane 5 MPS P230”, AIAA Paper, 96-3247, 1996.

APPENDIX A

JACOBIAN CALCULATION

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ e \end{bmatrix} = \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix}$$

and

$$u = \frac{\rho u}{\rho} = \frac{U_2}{U_1}$$

$$v = \frac{\rho v}{\rho} = \frac{U_3}{U_1}$$

$$p = (\gamma - 1)(e - \frac{1}{2}\rho(u^2 + v^2)) = (\gamma - 1)(U_4 - \frac{1}{2U_1}(U_2{}^2 + U_3{}^2))$$

$$H_T = \frac{(p+e)}{\rho} = \gamma \frac{U_4}{U_1} - (\gamma - 1)\frac{1}{2U_1{}^2}(U_2{}^2 + U_3{}^2)$$

$$F=\begin{bmatrix}\rho u\\ \rho uu+p\\ \rho uv\\ \rho uH_T\end{bmatrix}=\begin{bmatrix}U_2\\ \frac{U_2^{\,2}}{U_1}+(\gamma-1)(U_4-\frac{1}{2U_1}(U_2^{\,2}+U_3^{\,2}))\\ \frac{U_2U_3}{U_1}\\ \gamma\frac{U_2U_4}{U_1}-(\gamma-1)\frac{U_2}{2U_1^{\,2}}(U_2^{\,2}+U_3^{\,2})\end{bmatrix}$$

$$\frac{\partial F}{\partial U}=\begin{bmatrix}0 & 1 & 0 & 0\\ -\frac{U_2^{\,2}}{U_1^{\,2}}+\frac{(\gamma-1)}{2U_1^{\,2}}(U_2^{\,2}+U_3^{\,2}) & \frac{2U_2}{U_1}-\frac{(\gamma-1)U_2}{U_1} & -\frac{(\gamma-1)U_3}{U_1} & (\gamma-1)\\ -\frac{U_2U_3}{U_1^{\,2}} & \frac{U_3}{U_1} & \frac{U_2}{U_1} & 0\\ -\frac{\gamma U_2U_4}{U_1^{\,2}}+(\gamma-1)\frac{U_2}{U_1^{\,3}}(U_2^{\,2}+U_3^{\,2}) & \frac{\gamma U_4}{U_1}-\frac{(\gamma-1)}{2U_1^{\,2}}(3U_2^{\,2}+U_3^{\,2}) & -(\gamma-1)\frac{U_2U_3}{U_1^{\,2}} & \frac{\gamma U_2}{U_1}\end{bmatrix}$$

$$G=\begin{bmatrix}\rho v\\ \rho uv\\ \rho vv+p\\ \rho vH_T\end{bmatrix}=\begin{bmatrix}U_3\\ \frac{U_2U_3}{U_1}\\ \frac{U_3^{\,2}}{U_1}+(\gamma-1)(U_4-\frac{1}{2U_1}(U_2^{\,2}+U_3^{\,2}))\\ \gamma\frac{U_3U_4}{U_1}-(\gamma-1)\frac{U_3}{2U_1^{\,2}}(U_2^{\,2}+U_3^{\,2})\end{bmatrix}$$

$$\frac{\partial G}{\partial U}=\begin{bmatrix}0 & 0 & 1 & 0\\ -\frac{U_2U_3}{U_1^{\,2}} & \frac{U_3}{U_1} & \frac{U_2}{U_1} & 0\\ -\frac{U_3^{\,2}}{U_1^{\,2}}+\frac{(\gamma-1)}{2U_1^{\,2}}(U_2^{\,2}+U_3^{\,2}) & -\frac{(\gamma-1)U_2}{U_1} & \frac{2U_3}{U_1}-\frac{(\gamma-1)U_3}{U_1} & (\gamma-1)\\ -\frac{\gamma U_3U_4}{U_1^{\,2}}+(\gamma-1)\frac{U_3}{U_1^{\,3}}(U_2^{\,2}+U_3^{\,2}) & -(\gamma-1)\frac{U_2U_3}{U_1^{\,2}} & \frac{\gamma U_4}{U_1}-\frac{(\gamma-1)}{2U_1^{\,2}}(U_2^{\,2}+3U_3^{\,2}) & \frac{\gamma U_3}{U_1}\end{bmatrix}$$

Inviscid and viscous axisymmetric terms' Jacobeans are calculated by the same procedure used to calculate inviscid flux terms. The axisymmetric H term is divided into two parts, inviscid(H_i) and viscous(H_v), to give the solver flexibility of solving either inviscid Euler equations or viscous Navier-Stokes equations according to input statement. Resulting Jacobeans are given as below:

$$H_i = \frac{1}{y}\begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 \\ \rho v H_T \end{bmatrix} = \frac{1}{y}\begin{bmatrix} U_3 \\ \hline \dfrac{U_2 U_3}{U_1} \\ \hline \dfrac{U_3^{\ 2}}{U_1} \\ \hline \gamma \dfrac{U_3 U_4}{U_1} - (\gamma - 1)\dfrac{U_3}{2U_1^{\ 2}}(U_2^{\ 2} + U_3^{\ 2}) \end{bmatrix}$$

$$\Delta H_i = \frac{1}{y}\begin{bmatrix} \Delta \rho v \\ u(\Delta \rho v) + v(\Delta \rho u) \\ v(\Delta \rho v) + v(\rho \Delta v) + \Delta p \\ h_0(\Delta \rho v) + v(\rho \Delta h_0) \end{bmatrix}$$

$$H_v = \frac{1}{y}\begin{bmatrix} 0 \\ \hline -\tau_{xy} - \dfrac{2}{3}\dfrac{y}{\mathrm{Re}_\infty}\dfrac{\partial}{\partial x}\left(\mu \dfrac{v}{y}\right) \\ \hline -\tau_{yy} + \tau_{\theta\theta} - \dfrac{2}{3}\dfrac{\mu}{\mathrm{Re}_\infty}\left(\dfrac{v}{y}\right) - \dfrac{y}{\mathrm{Re}_\infty}\dfrac{2}{3}\dfrac{\partial}{\partial y}\left(\mu \dfrac{v}{y}\right) \\ \hline -u\tau_{xy} - v\tau_{yy} - q_y - \dfrac{2}{3}\dfrac{\mu}{\mathrm{Re}_\infty}\dfrac{v^2}{y} - \dfrac{y}{\mathrm{Re}_\infty}\dfrac{\partial}{\partial y}\left(\dfrac{2}{3}\mu \dfrac{v^2}{y}\right) - \dfrac{y}{\mathrm{Re}_\infty}\dfrac{\partial}{\partial x}\left(\dfrac{2}{3}\mu \dfrac{uv}{y}\right) \end{bmatrix}$$

$$
\frac{\partial H_v}{\partial U}=\frac{1}{y}\left[\begin{array}{c|c|c|c}
0 & 0 & 0 & 0 \\
\hline
\begin{array}{l}
\frac{\mu}{\mathrm{Re}_\infty}\left[-\frac{4}{3}\frac{\partial}{\partial x}\left(\frac{U_2}{U_1^2}\right)+\frac{2}{3}\frac{\partial}{\partial y}\left(\frac{U_3}{U_1^2}\right)\right] \\
+\frac{2}{3}\frac{\mu}{\mathrm{Re}_\infty}\frac{\partial}{\partial x}\left(\frac{U_3}{U_1^2}\right)
\end{array}
&
\frac{4\mu}{3\mathrm{Re}_\infty}\frac{\partial}{\partial x}\left(\frac{1}{U_1}\right)
&
\begin{array}{c}
\frac{-2\mu}{3\mathrm{Re}_\infty}\frac{\partial}{\partial y}\left(\frac{1}{U_1}\right) \\
-\frac{2\mu}{3\mathrm{Re}_\infty}\frac{\partial}{\partial x}\left(\frac{1}{U_1}\right)
\end{array}
& 0 \\
\hline
\begin{array}{l}
\frac{\mu}{\mathrm{Re}_\infty}\left[\frac{2}{3}\frac{\partial}{\partial x}\left(\frac{U_2}{U_1^2}\right)\right] \\
-\frac{\mu}{\mathrm{Re}_\infty}\frac{2}{3}\left[\frac{\partial}{\partial y}\left(\frac{U_2}{U_1^2}\right)-\frac{\partial}{\partial x}\left(\frac{U_3}{U_1^2}\right)\right] \\
-\frac{2\mu}{3\mathrm{Re}_\infty y}\left(\frac{U_3}{U_1^2}\right)-\frac{2\mu}{3\mathrm{Re}_\infty}\frac{\partial}{\partial y}\left(\frac{U_3}{U_1^2}\right)
\end{array}
&
\begin{array}{l}
\frac{-2\mu}{3\mathrm{Re}_\infty}\frac{\partial}{\partial x}\left(\frac{1}{U_1}\right) \\
+\frac{2\mu}{3\mathrm{Re}_\infty}\frac{\partial}{\partial y}\left(\frac{1}{U_1}\right)
\end{array}
&
\begin{array}{l}
\frac{4\mu}{3\mathrm{Re}_\infty}\frac{\partial}{\partial y}\left(\frac{1}{U_1}\right) \\
+\frac{2\mu}{3\mathrm{Re}_\infty}\frac{\partial}{\partial x}\left(\frac{1}{U_1}\right) \\
+\frac{2\mu}{3\mathrm{Re}_\infty}\left(\frac{1}{yU_1}\right)-\frac{2\mu}{3\mathrm{Re}_\infty}\frac{\partial}{\partial y}\left(\frac{1}{U_1}\right)
\end{array}
& 0 \\
\hline
\begin{array}{l}
-\frac{U_2}{U_1^2}\frac{\mu}{\mathrm{Re}_\infty}\left[\frac{\partial\left(\frac{U_2}{U_1}\right)}{\partial x}+\frac{\partial\left(\frac{U_3}{U_1}\right)}{\partial y}\right] \\
+\frac{U_2}{U_1}\frac{\mu}{\mathrm{Re}_\infty}\left[\frac{\partial}{\partial x}\left(-\frac{U_2}{U_1^2}\right)+\frac{\partial}{\partial y}\left(-\frac{U_3}{U_1^2}\right)\right] \\
-\frac{U_3}{U_1^2}\frac{\mu}{\mathrm{Re}_\infty}\left[\frac{4}{3}\frac{\partial\left(\frac{U_3}{U_1}\right)}{\partial y}-\frac{2}{3}\frac{\partial\left(\frac{U_2}{U_1}\right)}{\partial x}\right] \\
+\frac{U_3}{U_1}\frac{\mu}{\mathrm{Re}_\infty}\left[-\frac{4}{3}\frac{\partial}{\partial y}\left(\frac{U_3}{U_1^2}\right)+\frac{2}{3}\frac{\partial}{\partial x}\left(\frac{U_2}{U_1^2}\right)\right] \\
-\frac{\mu}{\mathrm{Re}_\infty R\Pr}\frac{\partial}{\partial y}\left(\frac{1}{U_1^2}\left(U_4-\frac{(U_2^2+U_3^2)}{U_1}\right)\right) \\
-\frac{4\mu}{3\mathrm{Re}_\infty}\frac{U_3^2}{U_1^3 y}+\frac{4\mu}{3\mathrm{Re}_\infty}\frac{\partial}{\partial y}\left(\frac{U_3^2}{U_1^3}\right) \\
+\frac{4\mu}{3\mathrm{Re}_\infty}\frac{\partial}{\partial x}\left(\frac{U_2U_3}{U_1^3}\right)
\end{array}
&
\begin{array}{l}
\frac{1}{U_1}\frac{\mu}{\mathrm{Re}_\infty}\left[\frac{\partial\left(\frac{U_2}{U_1}\right)}{\partial x}+\frac{\partial\left(\frac{U_3}{U_1}\right)}{\partial y}\right] \\
+\frac{U_2}{U_1}\frac{\mu}{\mathrm{Re}_\infty}\frac{\partial}{\partial x}\left(\frac{1}{U_1}\right) \\
-\frac{U_3}{U_1}\frac{\mu}{\mathrm{Re}_\infty}\left[\frac{2}{3}\frac{\partial}{\partial x}\left(\frac{1}{U_1}\right)\right] \\
-\frac{\mu}{\mathrm{Re}_\infty R\Pr}\frac{\partial}{\partial y}\left(\frac{U_2}{U_1^2}\right) \\
\frac{-2\mu}{3\mathrm{Re}_\infty}\frac{\partial}{\partial x}\left(\frac{U_3}{U_1^2}\right)
\end{array}
&
\begin{array}{l}
\frac{U_2}{U_1}\frac{\mu}{\mathrm{Re}_\infty}\frac{\partial}{\partial y}\left(\frac{1}{U_1}\right) \\
+\frac{1}{U_1}\frac{\mu}{\mathrm{Re}_\infty}\left[\frac{4}{3}\frac{\partial\left(\frac{U_3}{U_1}\right)}{\partial y}-\frac{2}{3}\frac{\partial\left(\frac{U_2}{U_1}\right)}{\partial x}\right] \\
+\frac{U_3}{U_1}\frac{\mu}{\mathrm{Re}_\infty}\left[\frac{4}{3}\frac{\partial}{\partial y}\left(\frac{1}{U_1}\right)\right] \\
-\frac{\mu}{\mathrm{Re}_\infty R\Pr}\frac{\partial}{\partial y}\left(\frac{U_3}{U_1^2}\right) \\
-\frac{4\mu}{3\mathrm{Re}_\infty}\frac{U_3}{yU_1^2}-\frac{4\mu}{3\mathrm{Re}_\infty}\frac{\partial}{\partial y}\left(\frac{U_3}{U_1^2}\right) \\
-\frac{2\mu}{3\mathrm{Re}_\infty}\frac{\partial}{\partial x}\left(\frac{U_2}{U_1^2}\right)
\end{array}
&
\frac{\mu}{\mathrm{Re}_\infty R\Pr}\frac{\partial}{\partial y}\left(\frac{1}{U_1}\right)
\end{array}\right]
$$