Doctoral Dissertations and Master's Theses

Fall 2011

Numerical Analysis of Gun Barrel Pressure Blast Using Dynamic Mesh Adaption

Sathish Xavier
Embry-Riddle Aeronautical University

Follow this and additional works at: https://commons.erau.edu/edt

Part of the Aerodynamics and Fluid Mechanics Commons

NUMERICAL ANALYSIS OF GUN BARREL PRESSURE BLAST USING DYNAMIC MESH ADAPTION

by

SATHISH XAVIER

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Aerospace Engineering

Embry-Riddle Aeronautical University
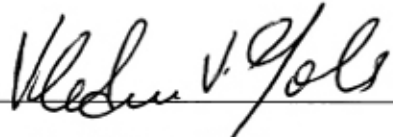
Daytona Beach, Florida

Fall 2011

Numerical Analysis of Gun Barrel Pressure Blast Using Dynamic Mesh Adaption

by

Sathish Xavier

This Thesis was prepared under the supervision of the Candidate's Thesis Advisor, Dr. Vladimir Golubev, Department of Aerospace Engineering, and has been approved by the members of Thesis Committee. This Thesis was submitted to the Department of Aerospace Engineering in partial fulfillment of the requirements for the Degree of Master of Science in Aerospace Engineering.

Thesis Committee:

Dr. Vladimir Golubev

Thesis Advisor

Dr. Eric Perrell

Member

Dr. Yongho Lee

Member

Dr. Howard Curtis

Department Chair, Aerospace Engineering

12/8/2011

Dr. Robert Oxley

Associate Vice President for Academics

ABSTRACT

Author: Sathish Xavier

Title: Numerical Analysis of Gun Barrel Pressure Blast Using Dynamic Mesh Adaption

Institution: Embry-Riddle Aeronautical University

Degree: Master of Science in Aerospace Engineering

Year: 2011

A Computational Fluid Dynamics (CFD) method has been applied to simulate the pressure blast of the projectile which is launched from a barrel and to investigate the pressure distribution and sound pressure level (dB) along different positions away from the gun axis and towards the fuselage of the aircraft. Fluent was employed to simulate the unsteady flow using dynamic mesh with moving boundary. Most CFD based ballistics-model requires additional thermodynamic functions which must be derived from the Noble-Abel equation of state. The unsteady, axi-symmetric Navier Stokes equation systems were numerically solved using the Advection Upstream Splitting Method (AUSM) scheme; with third-order Monotone Upstream centered Scheme for Conservation Laws (MUSCL) approach. The computed results reasonably capture the major flow features such as shock waves, blast waves, vortical flows, etc. which are generated in launching a projectile up to supersonic speed. The projectile mass and the initial conditions behind the projectile for inviscid cases have been varied to investigate its effect on the flow field and were compared with other available CFD results. Viscous effects of unsteady projectile aerodynamics were studied and conclude that the inclusion of viscous terms is essential for a more realistic mathematical model of the muzzle blast.

ACKNOWLEDGEMENTS

Table of Contents

List of Figures

List of Tables

1.0 INTRODUCTION

The full-scale experimental testing of the interior ballistics and the intermediate ballistics is very expensive and difficult; especially for the prevalent trend of gun systems that includes large caliber canon and multiple launch gun systems. The flow induced by a projectile, which is launched from a barrel is extremely difficult as it is associated with strong unsteadiness, compressibility and transient at the vicinity of the exit of the launch tube. The flow around a projectile will be developing and changing. The near-field flow structures of the launch tube are comparatively well known, but the near-field aerodynamics is understood only qualitatively, as almost all of the related data are based on the optical visualizations. On account of the very short test times, it is extremely difficult to experimentally investigate the unsteady projectile aerodynamics. Theoretical approach is also not easy due to the unsteadiness and three-dimensionalities, with inherent, strong compressibility effects involved in the aerodynamics of projectile. Much of the theoretical works carried out in the projectile aerodynamics fall short of providing useful data on the effect of unsteady aerodynamic loads on the projectile motion. As a result, the simulation of the projectile motion is highly desirable as an early design tool. Many attempts have been made to apply numerical methods to evaluate the high–pressure and temperature unsteady flow field. With the recent development in the computing capabilities and numerical schemes, Computational Fluid Dynamics (CFD) is being recognized as the most effective research tool to investigate the highly complicated unsteady, compressible flow phenomena.

1.1 THESIS OBJECTIVE

A realistic example of an analysis of pressure blast of a 20 mm gun that fires a 1500 grain (97.197 gram) projectile at a breech pressure and temperature of 3450 atm and 2500 K respectively for a 200 mm barrel. The objective of this project is to investigate the pressure distribution and sound pressure level (dB) along different positions away from the gun axis and towards the fuselage of the aircraft. The computed results were also used to capture the major flow features such as shock waves, blast waves, vortical flows, etc. which are generated in launching a projectile up to supersonic speeds.
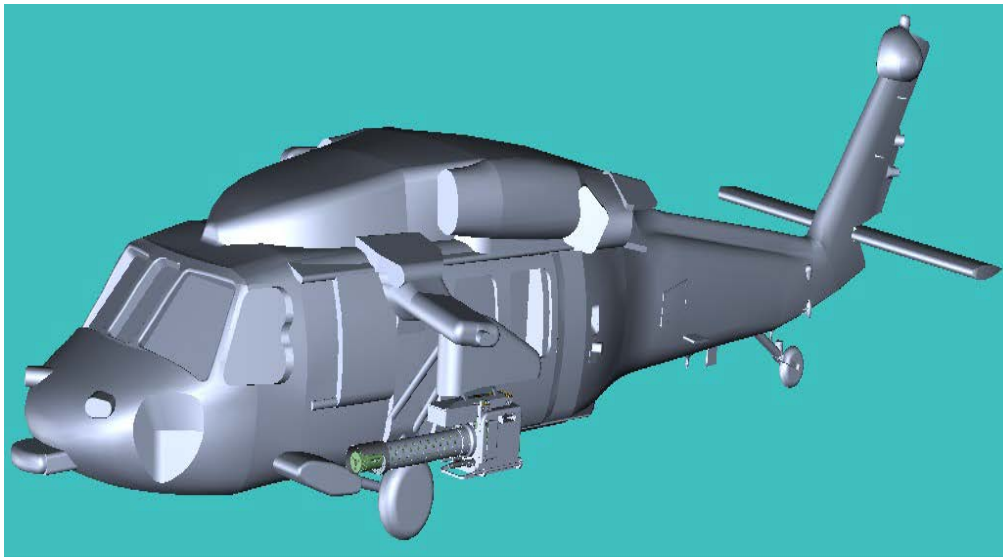
Figure 1. Aircraft with the gun mount

The acoustic environment produced by gun fire is of the impulse type with the muzzle being the primary source. There are two major effects of the induced gun fire noise environment that must be considered; namely, human tolerance and sonic aircraft structural fatigue.

The sound pressure level (dB) can be calculated by the following expression [20]

$$dB = 20 \ log \ \frac{P}{P_0}$$

(1)

where

log = logarithm to the base 10

P = perturbation pressure (calculated by subtracting the mean pressure, 1 atm from the obtained pressure)

P_0 = reference pressure (20μ Pa) is the pressure that can be heard by a person with normal hearing

A CFD method, Fluent has been applied to simulate the present computational problem. Fluent uses the cell-centered finite volume method to solve the governing equations. The unsteady, axi-symmetric Navier-Stokes equation systems were numerically solved using the Advection Upstream Splitting Method (AUSM) scheme; with third-order Monotone Upstream centered Scheme for Conservation Laws (MUSCL) approach. The flow field was simulated by using a dynamic mesh and moving boundary conditions with six-DOF solver. In order to simulate interior ballistic flow fields, the models require a description of the thermodynamic behavior of the propellant gas. To avoid the multiphase and mixing-flow considerations, the propellant is assumed to be a gas identical to the ambient air, and there are no chemical reactions inside or outside the barrel. The Noble-Abel equation provides a simple and reasonably accurate equation of state for propellant gases at the high densities and temperatures experienced in guns. Most Computational Fluid Dynamics (CFD) based ballistics models, require additional thermodynamic functions which must be derived from the equation of state. The additional

3

thermodynamic functions for the Noble-Abel gases will be discussed in Chapter 3. A User Defined Function (UDF) for Noble-Abel equation of state with the thermodynamic properties of air was provided by Ansys Technical Support and the UDF was built in to Fluent. The present computations were validated with other available CFD results [1] for the effects of projectile mass and various initial conditions behind the projectile. The initial conditions and dimensions of the projectile were different for validation cases and realistic case. Time histories of projectile velocity and acceleration were plotted for different conditions. Static pressures were plotted at different locations in the flow field to capture the flow field characteristics in accordance with the projectile aerodynamics. The validations were done for all inviscid cases. For different muzzle-blast cases, various numerical solutions have been reported in the past decade for inviscid cases (they do not include the effects of viscosity). In order to study the viscous effects on muzzle pressure blast, the author ran two different cases and made a comparison with the corresponding inviscid cases. It is conclusive that, for a more realistic mathematical model of the muzzle blast, the inclusion of the viscous term is essential. Once the present computational results were in good agreement with the bench mark case [1], the author proceeds with the realistic problem.

2.0 LITERATURE REVIEW

The blow down or emptying of a gun barrel after the projectile leaves the barrel is similar to the free jet expansion process. The flow field around the muzzle of a barrel can be complicated and include flow phenomena such as expansion waves, compression waves, shocks, shear layers and blast waves. As the flow exits the barrel and begins to expand around the sharp corner of the gun barrel, a Prandtl-Meyer expansion fan forms at the muzzle plane and spreads radially away from the axis and then terminates at the free-shear jet-boundary of the muzzle flow. These expansion waves reflect off the jet boundary forming a series of weak compression waves that coalesce to form a barrel or intercepting shock. In addition, these waves can propagate into the flow field toward the main blast wave [2].

Downstream of the barrel, a Mach disk forms across which the flow decelerates from supersonic to subsonic velocities. The Mach disk and intercepting shock enclose a volume known as the shock-bottle. Initially the Mach disk is constrained by a blast-wave front or primary shock. This constraining action can be seen as a deformation in the plume boundary [2]. When the projectile moves within the launch tube, a series of compression waves are formed ahead of it. These compression waves coalesce in to a shock wave called precursor shock, depending on the acceleration of the projectile the precursor shock wave propagates towards the end of the launch tube. The precursor shock wave propagates further downstream of the barrel and develops in to the first blast wave and a jet flow developing in the vicinity of the exit of the launch tube. Later, when the projectile emerges from the launch tube and moves in to the big chamber, the high pressure gas behind the projectile rushes out and the second blast wave develops. Moreover, once the projectile moves out of the supersonic flow region bounded by a barrel shock and Mach disk, a bow shock wave is generated in front of the projectile. This bow shock wave can catch up

5

with the precursor shock wave and will also interact with the shock waves reflected from the rigid wall. [3]
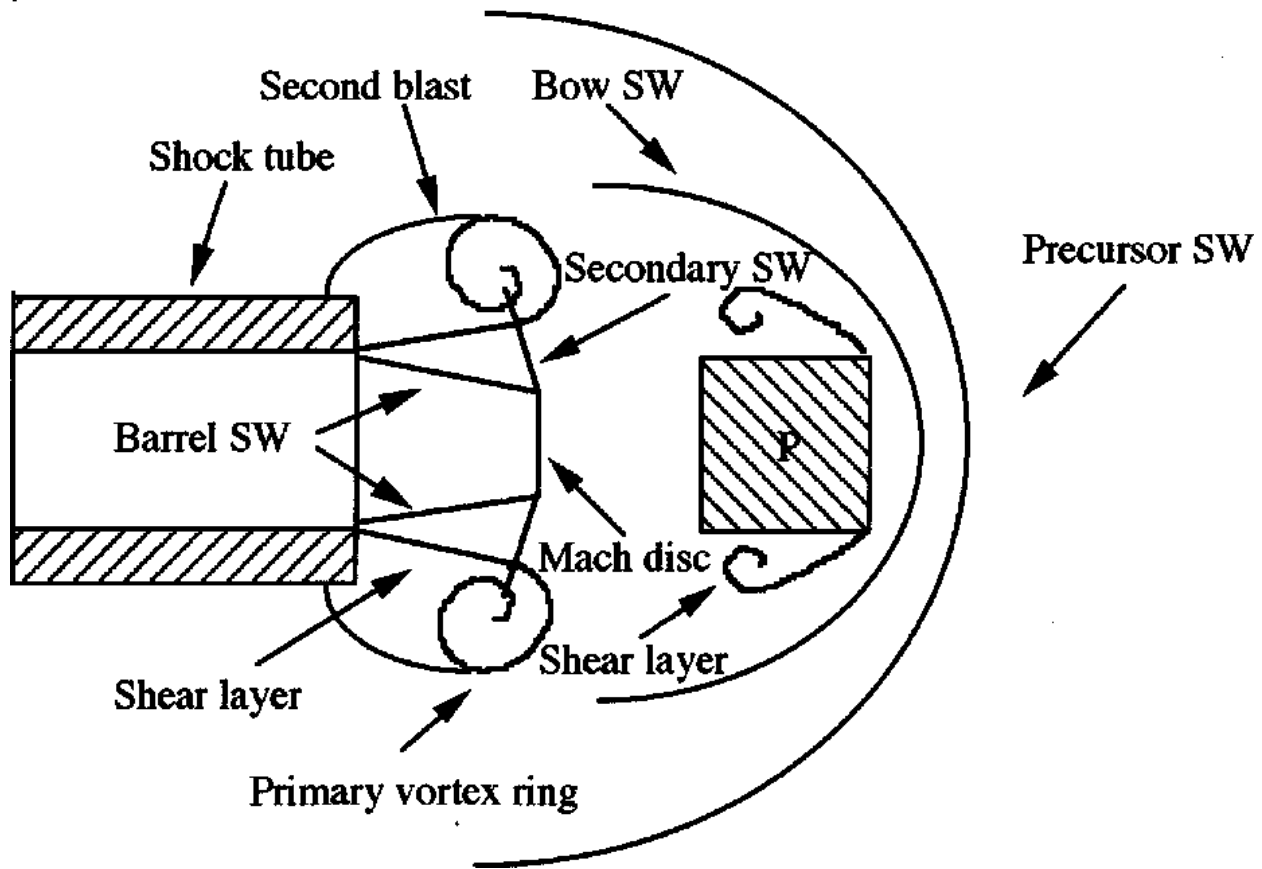
Figure 2. Schematic of wave dynamic processes occuring in muzzle blast flows [3]

The unsteady flow structures mentioned above are typically observed in the near-field of the launch tube. Accordingly, much of the research works are mainly concentrated on capturing the near-field structures of such flows. Glass [4] has shown detailed near-field structures of a bullet discharged from a rifle at Mach number 2, using several visualization pictures. Schmidt and Shear [5] have optically measured the muzzle blast waves using spark-shadow graph technique and observed strong coupling between the propellant gas flow and the blast wave. Erdos and Del

Guidice [6] have performed some theoretical studies to investigate the blast wave and the propellant gas flow. Later, Merline and Dyment [7] have tried to model the blast wave flow using a similarity law, based on visualization pictures.

Jiang et al. [8] numerically solved the blast flow fields of a projectile discharged from a shock tube using unsteady, axi-symmetric, Euler equations which were discretized by the dispersion-controlled scheme, with moving boundary conditions. In their simulations, a precursor shock wave was assumed at the exit of the shock tube. After the discharge of the precursor shock, the projectile was also assumed to move at the post-shock velocity. Two blast waves have been observed in their works at the vicinity of the shock tube exit, and they have argued that the release time between the precursor shock wave and the projectile has a negligible effect on the flow structures except in the region just behind the projectile. Though their work could provide the characteristic flow features in the near-field, it did not explain the effects of projectile mass, configuration and the initial launch conditions on the projectile aerodynamics. In another work, Jiang [3] investigated the effect of friction between the projectile and the shock tube on the near-field flow structures. He analyzed the near-field flow structures for various pressure ratios across the projectile due to frictional effect and concluded that even though the major flow features hardly exhibit any difference with varying friction, the shock interactions and the contact discontinuity become more intensive, which in turn, turns the flow field in to a more complex one.

The maximum projectile velocity is one of the most important design goals of the interior ballistic process and also is the initial condition for the exterior ballistic process. It is very important for studies and applications of weapons design. Most researchers take the muzzle velocity as the maximum projectile velocity. In reality, after the projectile exits the muzzle, the

projectile velocity will increase further due to the influence of high pressure gas behind the projectile. [9]

The projectile mass can be one of the most important parameters in determining the unsteady aerodynamics, since it is directly related to the acceleration and deceleration of the projectile. Moreover, the unsteady flow field of projectile discharged under various initial conditions also needs special attention as it has a deterministic effect on the blast wave development which subsequently affects the projectile motion. Rajesh et al. [1] applied the moving coordinate scheme to simulate the projectile aerodynamics at both the near-field. The unsteady, axi-symmetric Euler equation systems were numerically solved using the third order Chakravarthy-Osher total variation diminishing scheme, with Monotone Upstream centered Scheme for Conservation Laws (MUSCL) approach. Several cases with various initial conditions behind the projectile, projectile masses and configurations were analyzed to investigate its effect on the flow structures which affect the unsteady aerodynamics of the projectile in the near-field. They concluded that the projectile mass has a strong influence on the acceleration of the projectile inside the launch tube and deceleration of the projectile outside the launch tube. The projectile configuration affects the flow field and hence the unsteady aerodynamic loads on the projectile during its flight in the near-field. Reduced initial pressure and temperature conditions behind the projectile significantly affect the flow field as these conditions reduce the strength of the blast wave and the rate at which it is expanding. The author selects the reference [1] as a benchmark case for his comparison purpose.

3.0 THE NOBLE-ABEL GAS EQUATIONS

The ideal gas equation of state which is used for gases at moderate to low density, can be written as

$$P\vartheta = RT \tag{2}$$

where P is the absolute pressure, ϑ is the specific volume (volume per mass), R is the specific gas constant, T is the absolute temperature. The ideal gas equation of state is accurate providing that the mean free path of the gas is very large compared to the size of the gas molecules and the intermolecular forces are weak. For many practical flow regimes, both of these assumptions are true. However, the high gas densities and temperatures during combustion render the ideal gas equation of state inaccurate, especially for interior ballistic modeling. Due to the high temperature and densities experienced in the gun barrel for the present simulation, the ideal gas equation of state cannot be used.

While still an approximation, the Van der Waals equation of state provides an improvement in accuracy for high density gases. It is of the form [11],

$$\left(P + \frac{a}{\vartheta^2}\right)(\vartheta - b) = RT \tag{3}$$

where a and b are Van der Waals constants. The co-volume b, compensates for the finite volume occupied by the gas molecules, while the term $\frac{a}{\vartheta^2}$ accounts for intermolecular attraction forces. It is important to note here that if $a=b=0$, equation (3) reverts to the ideal gas equation of state. In ballistics applications, the high propellant gas temperature means that intermolecular attraction energy is small in comparison to molecular kinetic energy [11]. Thus the intermolecular

9

attraction energy term can be removed without significant loss of accuracy, resulting in the so-called Noble-Abel equation of state [10]

$$P(\vartheta - b) = RT. \tag{4}$$

In order to simulate interior ballistic flow fields, the models require a description of the thermodynamic behavior of the propellant gas. The Noble-Abel equation provides a simple and reasonably accurate equation of state for propellant gases at the high densities and temperatures experienced in guns. Most CFD-based ballistics models require additional thermodynamic functions which must be derived from the equation of state. Although the derivations are geared toward the functional requirements of the commercial FLUENT code, the results are equally applicable to all CFD solvers.

3.1 EQUATION OF STATE

The Noble-Abel equation of state has already been defined in equation (4). However FLUENT requires the equation of state in the form of $\rho(T, P)$. Hence the equation (4) is recast as

$$\rho = \frac{P}{(RT + bP)}. \tag{5}$$

3.2 OTHER THERMODYNAMIC RELATIONS

The derivations of such other thermodynamic relations for Noble-Abel gases can be found from the reference [10].

FLUENT also requires the specific heat at constant pressure as a function of density and temperature. Both the specific heats are defined as

$$C_p = \left(\frac{\partial h}{\partial T}\right)_p \qquad\qquad C_v = \left(\frac{\partial e}{\partial T}\right)_v. \tag{6}$$

where C_p, C_v are specific heat capacities at constant pressure and constant volume respectively. h is the specific enthalpy and e is the specific internal energy.

The gradient of C_p with respect to pressure at constant temperature can be related to equation of state [12] as

$$\left(\frac{\partial c_p}{\partial P}\right)_T = -T\left(\frac{\partial^2 \vartheta}{\partial T^2}\right)_P. \tag{7}$$

For a Noble-Abel gas, the specific heat function can be conveniently implemented in computer code as a function of temperature only and it can be written as follows

$$c_p(\rho, T) = c_p(T) \tag{8}$$

$$c_\vartheta(\rho, T) = c_\vartheta(T). \tag{9}$$

The specific heat curve fit values can be found from reference [15].

The Specific gas constant for a Noble-Abel gas is given by,

$$c_p - c_\vartheta = R \tag{10}$$

which is the same as that for ideal gases.

The entropy for a Noble-Abel gas may be derived starting from the thermodynamic relation and it is given as

$$s_2 - s_1 = c_v \ln\left(\frac{T_2}{T_1}\right) + R \ln\left|\frac{\vartheta_2 - b}{\vartheta_1 - b}\right|. \tag{11}$$

11

For an isentropic process in a calorically perfect, Nobel-Abel gas

$$P(\vartheta - b)^\gamma = constant. \tag{12}$$

This is similar to the familiar ideal-gas expression $P\vartheta^\gamma = constant.$

The speed of sound in a gas a, is defined as the speed of reversible pressure wave propagation and it can be written as

$$a = \left(\frac{\partial P}{\partial \rho}\right)^{\frac{1}{2}}. \tag{13}$$

The speed of sound in a calorically perfect Noble-Abel gas is given by

$$a = \frac{\vartheta}{\vartheta - b}\sqrt{\gamma RT}. \tag{14}$$

whereas the speed of sound in an ideal-gas is given by $\sqrt{\gamma RT}$. The speed of sound can be rewritten in the form as required by FLUENT,

$$a = \frac{1}{1 - b\rho}\sqrt{\gamma RT}. \tag{15}$$

The partial derivative of density with respect to pressure at constant temperature can be given as

$$\left(\frac{\partial \rho}{\partial P}\right)_T = \frac{(1 - \rho b)^2}{RT}. \tag{16}$$

Now the partial derivative of density with respect to temperature at constant pressure can be given as

$$\left(\frac{\partial \rho}{\partial T}\right)_P = \frac{b\rho^2 - \rho}{T}. \tag{17}$$

The partial derivative of enthalpy with respect to temperature at constant pressure is equal to c_p by definition. Also the partial derivative of enthalpy with respect to pressure at constant temperature is equal to the co-volume b:

$$\left(\frac{\partial h}{\partial T}\right)_P = c_p$$

$$\left(\frac{\partial h}{\partial P}\right)_T = b \,.$$

The Noble-Abel equation of state with its thermodynamic properties was built in to FLUENT using a User Defined Function (UDF) which can be seen in Appendix II.

4.0 NUMERICAL METHOD

4.1 GOVERNING EQUATIONS

The conservation equations in differential form for the two dimensional unsteady, compressible

flow, Navier Stokes equations are written as

$$\frac{\partial Q}{\partial t} + \frac{\partial E_i}{\partial x} + \frac{\partial F_i}{\partial y} = \frac{\partial E_v}{\partial x} + \frac{\partial F_v}{\partial y} + S \tag{18}$$

where Q is the state variable to be solved, E_i and F_i are the inviscid fluxes in two coordinate

directions, E_v and F_v are the viscid fluxes in two coordinate directions, and S is a source term.

All the variables are defined as follows

$$Q = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ e \end{bmatrix} \qquad\qquad S = \begin{bmatrix} \rho v \\ \rho u v - \tau_{yx} \\ \rho v^2 + 2\mu \left(\frac{v}{y} - \frac{\partial v}{\partial y}\right) \\ (e + P)v - \left(u\tau_{yx} + v\tau_{yy}\right) + q_y \end{bmatrix}$$

$$E_i = \begin{bmatrix} \rho u \\ \rho u^2 + P \\ \rho u v \\ (e + P)u \end{bmatrix} \qquad\qquad F_i = \begin{bmatrix} \rho v \\ \rho u v \\ \rho v^2 + P \\ (e + P)v \end{bmatrix}$$

$$E_v = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ u\tau_{xx} + v\tau_{xy} - q_x \end{bmatrix} \qquad\qquad F_v = \begin{bmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ u\tau_{yx} + v\tau_{yy} - q_y \end{bmatrix}$$

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

14

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

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

$$q_x = -k \frac{\partial T}{\partial x} \qquad\qquad q_y = -k \frac{\partial T}{\partial y}$$

where primitive variables in the unknown Q are density ρ, velocity components u and v, and total energy per unit volume E. Fluid pressure is denoted with p, which is related with the equation of state for the perfect gas given by

$$E = \frac{p}{\gamma-1} + \frac{\rho}{2}(u^2 + v^2). \tag{19}$$

For inviscid flows, the governing equations are obtained from the Navier-Stokes equations by dropping all second-order viscous diffusion terms. Equation (18) becomes

$$\frac{\partial Q}{\partial t} + \frac{\partial E_i}{\partial x} + \frac{\partial F_i}{\partial y} = S. \tag{20}$$

4.1.1 TURBULENCE MODEL

Turbulence effects are also considered in the simulation. The choice of turbulence model will depend on considerations such as the physics encompassed in the flow, the level of accuracy required, the available computational resources, and the amount of time available for the simulation. In terms of computation, the Spalart-Allmaras model is the least expensive turbulence model of the options provided in FLUENT, since only one turbulence transport equation is solved [16]. The Spalart-Allmaras turbulence model is highly suitable for the

simulation of gas flow with a wall boundary. Furthermore it is very effective for simulating

boundary layer flow fields for adverse pressure gradient problems [9].

The transport equation for turbulent energy of the Spalart-Allmaras model is written as follows

$$\frac{\partial\ (\rho\bar{\upsilon})}{\partial t} + \frac{\partial\ (\rho\bar{\upsilon}u_i)}{\partial x_i} = \ G_v + \frac{1}{\sigma_{\bar{\upsilon}}}\left[\frac{\partial}{\partial x_j}\left\{(\mu + \rho\bar{\upsilon})\frac{\partial\bar{\upsilon}}{\partial x_j}\right\} + C_{b_2}\rho\right] - Y_v \tag{21}$$

where ῡ is the transported variable in the model, G_v is the production of turbulent viscosity and

Y_v is the destruction of turbulent viscosity that occurs in the near-wall region due to wall

blocking and viscous damping. σ_k and C_{b_2} are constants.

4.2 FLUENT BACKGROUND

FLUENT [16] uses a control-volume based technique to convert a general scalar transport

equation to an algebraic equation that can be solved numerically. This control volume technique

consists of integrating the transport equation about each control volume, yielding a discrete

equation that expresses the conservation law on a control-volume basis. For an arbitrary control

volume V, the unsteady conservation equation for transport of a scalar quantity φ can be written

as

$$\int \frac{\partial\rho\emptyset}{\partial t}dV + \oint \rho\emptyset\vec{v}\cdot d\vec{A} = \ \oint \Gamma_{\emptyset}\ \nabla\emptyset\cdot d\vec{A} + \ \int S_{\emptyset}dV \tag{22}$$

where

ρ = density

$\vec{v}$ = velocity vector $(= u\hat{\imath} + \ v\hat{\jmath}$ in 2D$)$

$\vec{A}$ = surface area vector

16

$\Gamma_\emptyset$ = diffusion coefficient for φ

$\nabla\emptyset$ = gradient of φ

$S_\emptyset$ = source of φ per unit volume.

Equation (22) is applied to each control volume, or cell in the computational domain. Discretization of equation (22) on a given cell yields

$$\frac{\partial \rho \emptyset}{\partial t}V + \sum_f^{N_{faces}} \rho_f \vec{v}_f \emptyset_f \cdot \vec{A}_f = \sum_f^{N_{faces}} \Gamma_\emptyset \nabla\emptyset_f \cdot \vec{A}_f + S_\emptyset V \qquad (23)$$

where

N_{faces} = number of faces enclosing cell

$\emptyset_f$ = value of φ convected through face f

$\rho_f \vec{v}_f \cdot \vec{A}_f$ = mass flux through the face

$\vec{A}_f$ = area of face f

$\nabla\emptyset_f$ = gradient of φ at face f

V = cell volume.

FLUENT stores discrete values of the scalar φ at the cell centers. However, face value $\emptyset_f$ are required for the convection terms in equation (23) and must be interpolated from the cell center values. This is accomplished using an upwind scheme. Fluent allows users to choose from several upwind schemes. Third-order Monotone Upstream centered Scheme for Conservation Laws (MUSCL) was selected for this unsteady projectile aerodynamics simulation. The MUSCL

scheme was conceived by blending a central difference scheme and second-order upwind scheme. The third-order MUSCL scheme has a potential to improve spatial accuracy for all types of meshes by reducing numerical diffusion, most significantly for complex three-dimensional flows, and it is available for all transport equations.

4.2.1 OVERVIEW OF FLOW SOLVERS

The two types of solvers available in Fluent are

Pressure-based solver: This approach was developed for low speed incompressible flows. The velocity field is obtained from the momentum equations while the pressure field is extracted by solving a pressure or pressure correction equation which is obtained by manipulating continuity and momentum equations in pressure-based solver.

Density-based solver: The density-based approach was mainly used for high speed compressible flows. The velocity field is obtained from the momentum equations. In this approach, the continuity equation is used to obtain the density field while the pressure field is determined from the equation of state.

For the present CFD simulation, density-based solver was selected for its high speed compressibility.

4.2.2 CONVECTIVE FLUX TYPES

The three types of convective fluxes that are available in FLUENT, when using the density based solver, are:

1) Roe Flux-Difference Splitting (Roe-FDS)

2) Advection Upstream Splitting Method (AUSM)

3) Low diffusion Roe Flux-Difference Splitting (Low diffusion Roe-FDS)

The flux type selected for present simulation is the Advection Upstream Splitting Method (AUSM) due to its

- ❖ accurate capturing of shock and contact discontinuities

- ❖ entropy-satisfying solution

- ❖ uniform accuracy and convergence rate for all Mach numbers

4.2.3 DYNAMIC MESH

The dynamic mesh model in FLUENT can be used to model flows where the shape of the domain is changing with time due to motion on the domain boundaries. The motion can be either a prescribed motion or unprescribed motion. For a prescribed motion, we have to specify the linear and angular velocities about the center of gravity of a solid body with time. For an unprescribed motion, the subsequent motion is determined based on the solution at the current time (the linear and angular velocities are calculated from the force balance on a solid body). The update of the volume mesh is handled automatically by FLUENT at each time step based on the new positions of the boundaries. To use the dynamic mesh model, we need to provide a starting volume mesh and the description of the motion of any moving zones in the model. FLUENT allows users to describe the motion using either boundary profiles or User-Defined Functions (UDF). For the present simulation, the author uses a UDF to describe the motion of the projectile.

The integral form of the conservation equation for a general scalar, φ, on an arbitrary control volume V, whose boundary is moving, can be written as

$$\frac{d}{dt}\int \rho\varnothing dV + \int \rho\varnothing (\vec{u} - \vec{u}_g) \cdot d\vec{A} = \int \Gamma \nabla\varnothing \cdot d\vec{A} + \int S_\varnothing dV \qquad (24)$$

where

ρ = fluid density

$\vec{u}$ = flow velocity vector

$\vec{u_g}$ = grid velocity of the moving mesh

Γ = diffusion coefficient

$S_\varnothing$ = source term of φ.

The time derivative term in equation (24) can be written using a first order forward difference formula, as

$$\frac{d}{dt}\int \rho\varnothing dV = \frac{(\rho\varnothing V)^{n+1} - (\rho\varnothing V)^n}{\Delta t} \qquad (25)$$

where n and $n+1$ denote the current and next time level respectively. The next volume level V^{n+1} for the $(n+1)^{th}$ time level is computed from

$$V^{n+1} = V^n + \frac{dV}{dt}\Delta t \qquad (26)$$

where $\frac{dV}{dt}$ is the volume time derivative of the control volume. In order to satisfy the grid conservation law, the volume time derivative of the control volume is computed from

$$\frac{dV}{dt} = \int \vec{u}_g \cdot d\vec{A} = \sum_j^{n_f} \vec{u}_{g,j} \cdot \vec{A}_j \tag{27}$$

where n_f is the number of faces on the control volume and $\vec{A}_j$ is the j face area vector. The dot product $\vec{u}_{g,j} \cdot \vec{A}_j$ on each control volume face is calculated from

$$\vec{u}_{g,j} \cdot \vec{A}_j = \frac{\delta V_j}{\Delta t} \tag{28}$$

where δV_j is the volume swept out by the control volume face j over the time step Δt.

4.2.4 DYNAMIC MESH UPDATE METHODS

There are three mesh motion methods available in FLUENT to update the volume mesh in the deforming regions subject to the motion defined at the boundaries.

1) Spring-based smoothing

2) Dynamic layering method

3) Local remeshing method

Spring-based smoothing:

The spring-based smoothing method can be used for zones with a triangular or tetrahedral mesh to adjust the interior node locations based on the known displacements at the boundary nodes. The spring-based smoothing method updates the volume mesh without changing the mesh connectivity. In the spring-based smoothing method, the edges between any two mesh nodes are idealized as a network of interconnected springs. A displacement at a given boundary node will generate a force proportional to the displacement along all the springs connected to the node, and so the displacement of the boundary node is propagated through the volume mesh. We can use

the spring-based method to update any cell zone whose boundary is moving or deforming. However, for non-tetrahedral or non-triangular cell zones, the spring-based method is recommended when the following conditions are met:

- ❖ The boundary motion of the cell zone is predominantly in one direction (i.e., no excessive anisotropic stretching or compression of the cell zone)
- ❖ The motion is predominantly normal to the boundary zone.

If these conditions are not met, the resulting cells may have high skewness values, since not all possible combination of node pairs in non-tetrahedral or non-triangular cells are idealized as springs.

Dynamic layering method:

In prismatic (hexahedral/ wedge) mesh zones, dynamic layering can be used to add or remove layers of cells adjacent to a moving boundary based on the height of the layer adjacent to the moving surface. The dynamic mesh model in FLUENT allows an ideal layer height to be specified on each moving boundary.
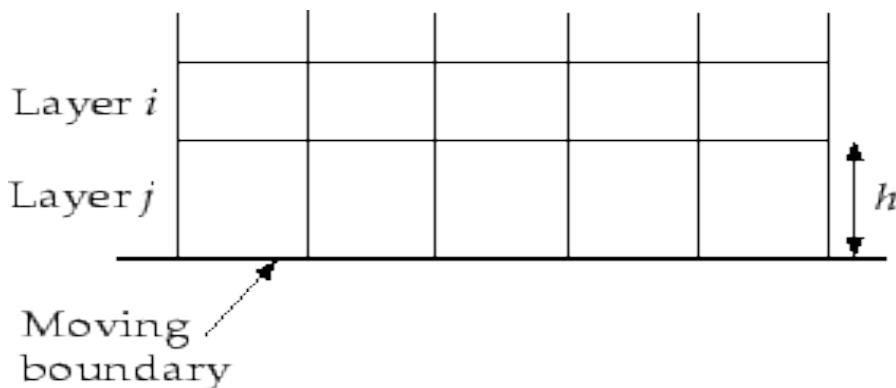
Figure 3. Schematic of dynamic layering mesh update method

The layer of cells adjacent to the moving boundary (layer j in figure 3) is subdivided or merged with the layer of cells next to it (layer i in figure 3) based on the height of the cells in layer j. If the cells in layer j are expanding, they are allowed to expand until

$$h > (1 + \alpha_h)h_{ideal}$$

where h_{ideal} is the ideal cell height and α_h is the global layer split/collapse factor. When this condition is met, the cells are split according to the predefined layer height (i.e. the cells in layer j are subdivided to create a layer of cells of height h_{ideal} and a layer of cells of height (h - h_{ideal}) If the cells in layer j are being compressed, they can be compressed until

$$h < \alpha_h h_{ideal}$$

when this condition is met, the compressed layer of cells is merged into the layer of cells above the compressed layer (i.e., the cells in layers i and j will be merged together)

Local remeshing method:

The spring-based smoothing method is normally used for triangular or tetrahedral mesh on cell zones. When the boundary displacement is large compared to the local cell size, the cell quality can deteriorate or the cells can become degenerate, and this will lead to convergence problems when the solution is updated to the next time step. To circumvent this problem, FLUENT agglomerates poor-quality cells and locally remeshes the agglomeration. FLUENT evaluates each cell and marks it for remeshing if it meets one or more of the following criteria:

❖ It is smaller than a specified minimum size.

❖ It is larger than a specified maximum size.

❖ It has a skewness that is greater than a specified maximum skewness.

For the present simulation, dynamic layering mesh method was used to update the volume mesh.

4.2.5 SIX-DOF SOLVER

FLUENT's Six Degree of Freedom (6DOF) solver computes external forces and moments such as aerodynamic and gravitational forces and moments on an object. These forces are computed by numerical integration of pressure and shear stress over the object's surface. The 6DOF solver in FLUENT uses the object's forces and moments in order to compute the translational and angular motion of the center of gravity of an object. We must use a UDF to define the six degrees of freedom parameters and the UDF is shown in the Appendix.

4.3 INITIAL AND BOUNDARY CONDITIONS

The dimension of the model used for the present simulation can be viewed from the following figure 4. All the dimensions are in millimeters (mm).
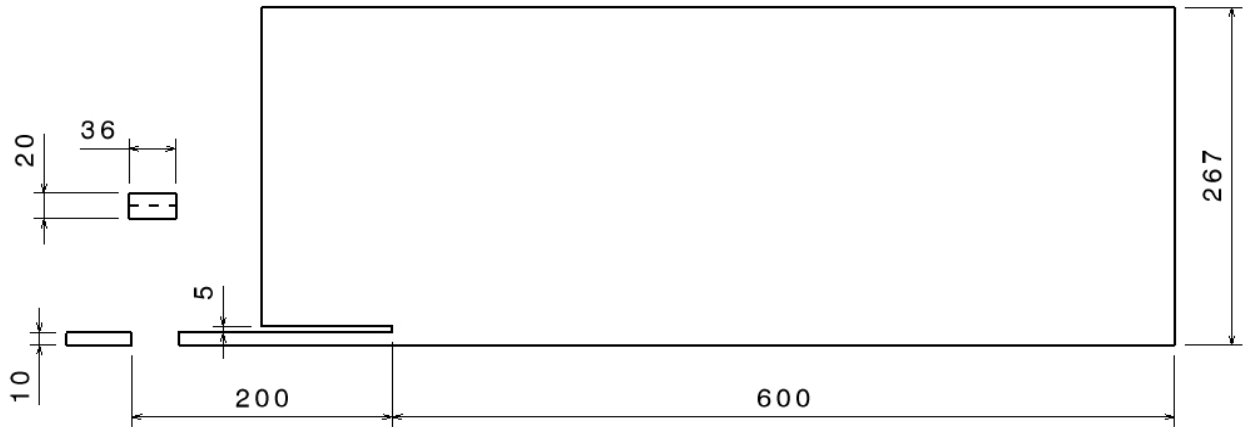
Figure 4. Computational domain of the validation cases [1]

The projectile is placed inside the launch tube where the initial conditions behind the projectile are pressure P_2, temperature T_2 and velocity U_2. The length of the launch tube is 200 mm and its diameter (D_p) is 20 mm. The still air which is outside the launch tube are at atmospheric conditions. The atmospheric domain extends to a length of 600 mm (3 times the barrel length) in the axial direction and 267 mm in the transverse direction which is away from the axis of launch tube. The length of the projectile is 36 mm and its width is 20 mm. The width of the projectile is same as that of the diameter of the launch tube. The projectile material is assumed to be aluminum. Due to the symmetric nature of the model, only the half of the projectile figure is modeled for the simulation so that the computing effort of the full domain can be reduced. In this thesis, the author follows exactly the same dimension as followed by Rajesh et al. [1] so that the computed results obtained can be compared against [1].

The boundary conditions applied on the present model can be seen from the following figure 5. Pressure boundary conditions are applied to the left, upper and right boundaries of the present computational domain and the wall boundary conditions are applied to the wall of the launch

tube, breech and the projectile surfaces of the present computational domain. The present simulation enables dynamic mesh, so the mesh interface has to be created between the two domains. Interface inner is created from the boundaries of the launch tube (moving domain) and the interface outer is created from the boundary of the atmospheric domain (static) Axi-symmetric conditions are employed to reduce the computing effort of the full domain.
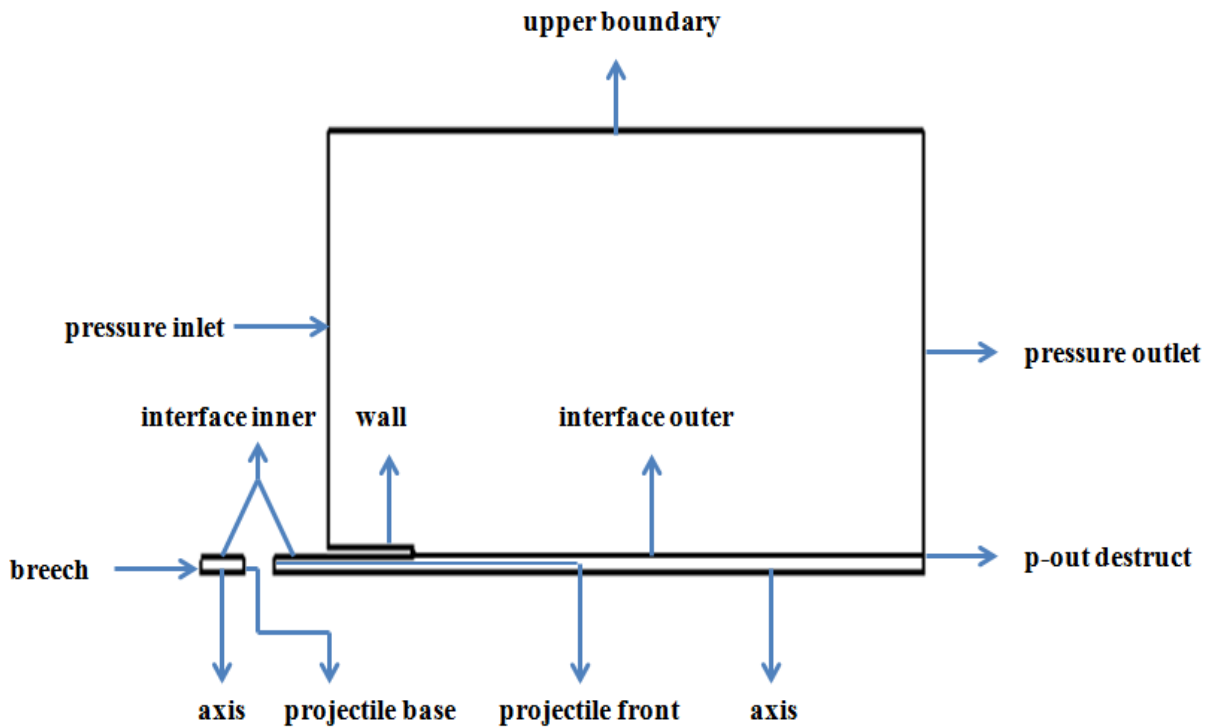

Figure 5. Boundary conditions [1]

In order to capture the pressure variations behind the blast wave, we need to measure the pressure at different positions along the length of the barrel. Figure 6 shows various pressure measuring locations in the flow field.
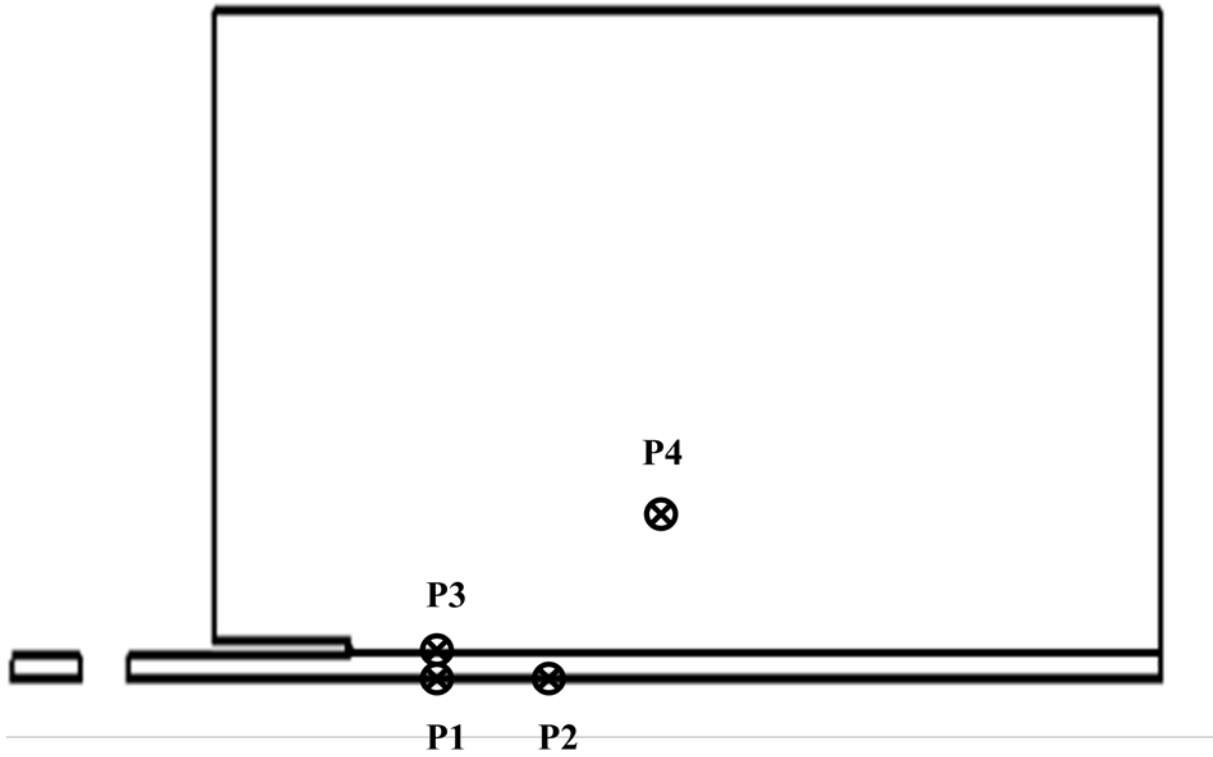
Figure 6. Pressure measuring locations for validation cases

Points	Location	
	X (m)	Y (m)
P1	0.31	0
P2	0.32	0
P3	0.31	0.01
P4	0.34	0.04

Table 1. Location of X and Y for pressure measurements

The co-ordinate system is fixed at the base of the breech. **P1** is located 10 mm from the exit of launch tube. (310 mm from the origin)

In this thesis, the goal of the author is to analyze the unsteady projectile aerodynamics for various initial conditions, viscosity state and for different projectile masses. Throughout all of the present computations, the still air outside the launch tube is kept at constant pressure P_1, temperature T_1 and velocity U_1 (0 m/s) while the pressure P_2 and temperature T_2 behind the projectile are varied for different cases. In order to study the influence of mass effects on the unsteady projectile aerodynamics, two cases with different masses were analysed. The following table represents the various initial conditions, viscosity state and different projectile masses used to carry out the present simulations.

Test Cases	P_1 (atm)	T_1 (K)	U_1 (m/s)	P_2 (atm)	T_2 (K)	U_2 (m/s)	M (kg)	Viscosity state
Case 1	1	300	0	2000	1000	0	0.00675	Inviscid
Case 2	1	300	0	2000	1000	0	0.0135	Inviscid
Case 3	1	300	0	2000	500	0	0.0135	Inviscid
Case 4	1	300	0	1000	1000	0	0.0135	Inviscid
Case 5	1	300	0	2000	1000	0	0.0135	Viscous
Case 6	1	300	0	2000	500	0	0.0135	Viscous
Case 7	1	300	0	3450	2500	0	0.09719	Viscous

Table 2. Computational conditions used for present simulation

Case 1 and Case 2 are used to study the effect of projectile mass for the same conditions of pressure and temperature for inviscid cases. Cases 2, 3 and 4 are used to study the effect of various initial conditions for the same mass. Cases 5 and 6 are used to study the viscous effects of projectile aerodynamics and then compared with the corresponding inviscid cases (Cases 2 and 3). Case 7 is the thesis objective of the author.

4.4 COMPUTATIONAL SETUP IN FLUENT

The geometrical model was constructed using grid generation software, Pointwise V16.04 [13] The grid was made structured in order to capture the shock waves more effectively. The structured grid looks like as below in Figure 7
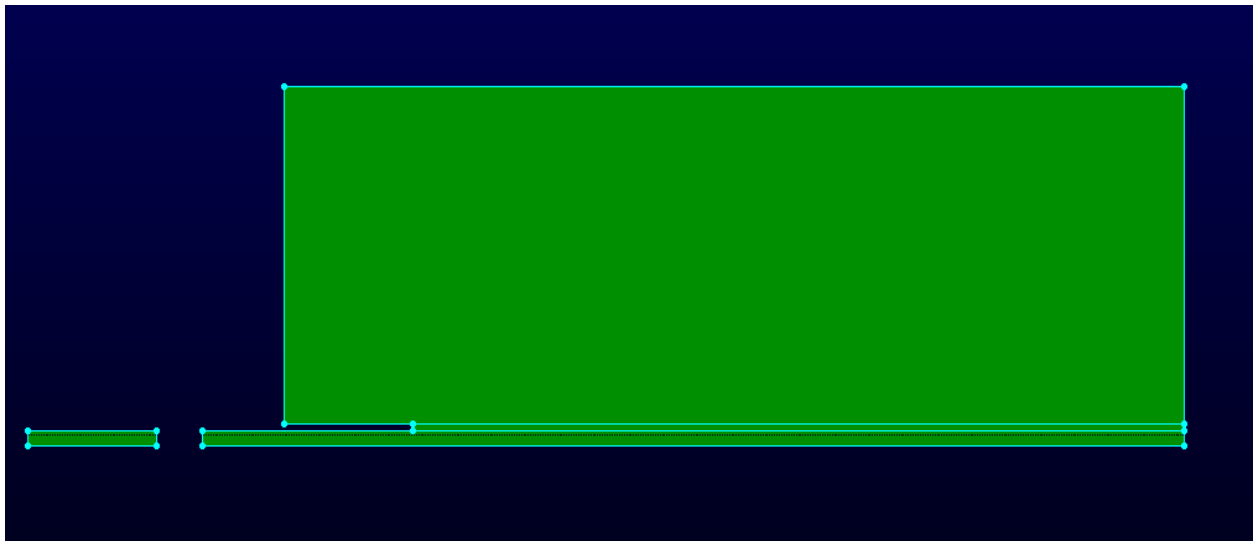
Figure 7. Computational structured grid made in Pointwise

 The author chose ANSYS FLUENT for this analysis of unsteady projectile aerodynamics. The 2D or 3D model can be selected under dimension. Before exporting the model to FLUENT, boundary conditions have to be specified on every connector. In order to specify the boundary

conditions, first select the corresponding connector and then chose the appropriate boundary conditions from the list of available conditions. Make sure that every connector had a condition. The volume conditions (Fluid or Solid) should also be specified on each domain. This particular problem had 4 different domains, out of which 2 domains are stationary and the other 2 are moving domains.
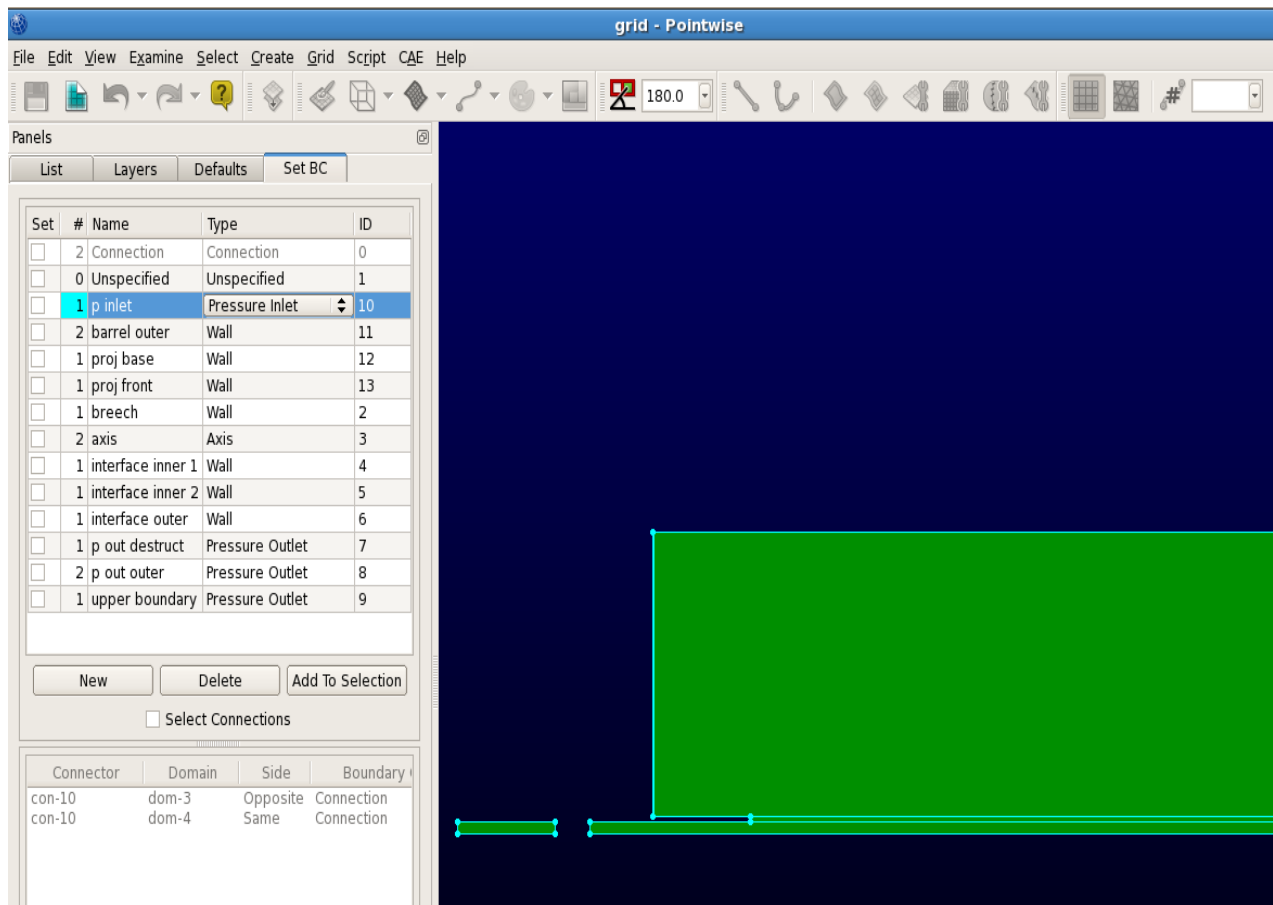
Figure 8. Computational grid with boundary conditions in Pointwise

Now export the model as

File ⟶ Export ⟶ CAE

Make sure to select all the domains while exporting. Different solvers have different file types to save the grid. In order to analyze the problem in ANSYS FLUENT, the grid should be saved as .cas file

The author initially tried using ANSYS CFX to solve this particular problem. CFX is a pressure based solver. The choice of selecting the numerical schemes like Roe or AUSM is not available in CFX. The current problem for analysis is highly compressible and transient with very high pressure and temperature. Because of these limitations in CFX, the author tried to use a different CFD solver, ANSYS FLUENT.

ANSYS FLUENT has the following advantages over CFX:

➢ Choice of selecting pressure based or density based solvers (For high speed transient, compressible flows, density based solver can be used to capture the shock waves effectively)

➢ Choice of available numerical schemes such as Roe and AUSM

➢ Higher order numerical schemes like third order MUSCL are available.

➢ Dynamic mesh features and different types of mesh update methods

➢ Six DOF (Degrees of Freedom) solver options to calculate forces and moments.

➢ Availability of real gas models. If a particular problem needs a different real gas model, a UDF (User Defined Function) can be written for the problem requirements and loaded in to FLUENT.

➢ The real gas law works only with density based solver.

The following steps were carried out in FLUENT for the analysis of unsteady projectile aerodynamics.

Before starting ANSYS FLUENT, make sure that the UDF and the case file (.cas) are in the same folder. Open FLUENT through the folder where all the files are located. Now read the mesh file (.cas) in to FLUENT.

Step 1: General options
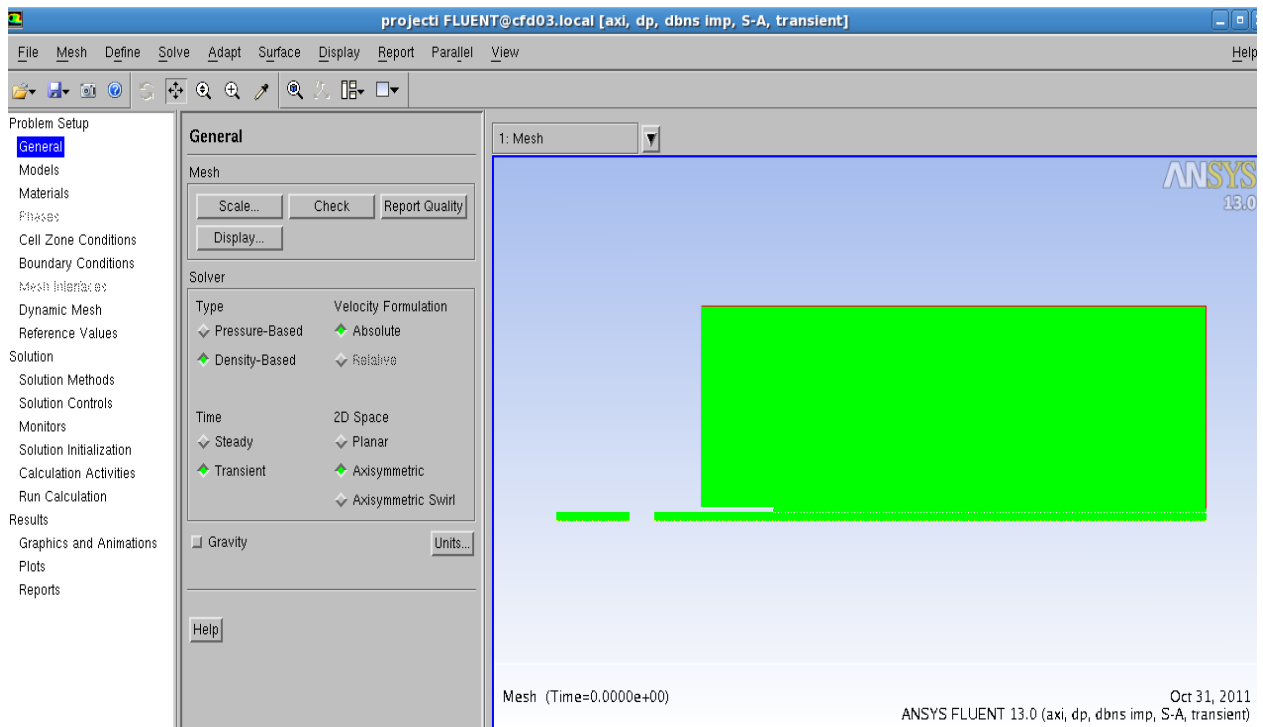
Figure 9.Solver setting option

Figure 9 explains the general setting options available in Fluent.

a) Scale: The units option is not available in Pointwise. The proper units and the dimension of the grid were set in Fluent.

b) Check: It is very important to check the mesh before starting the problem. It will report any negative volumes if they exist in the grid. If negative volume is reported, go back to Pointwise and redefine the mesh.

c) Select density based in the Type list (Real gas law works only with density based solver).

d) Select Transient under Time list.

e) Select Axisymmetric in the 2D space list.

f) Set the units of pressure to atm.

g) Set the operating pressure to 0 atm (This setting is usually used with the density-based solver, to avoid confusion between absolute and gauge pressures. With the operating pressure set to zero, the gauge and absolute pressures will be equivalent).

h) Merge Zones: There are four zones created for this grid. Out of which 2 are moving and 2 are stationary zones. Merge the 2 moving zones into one zone and the 2 stationary zones into one. Figure 10 shows the settings of merge zones.
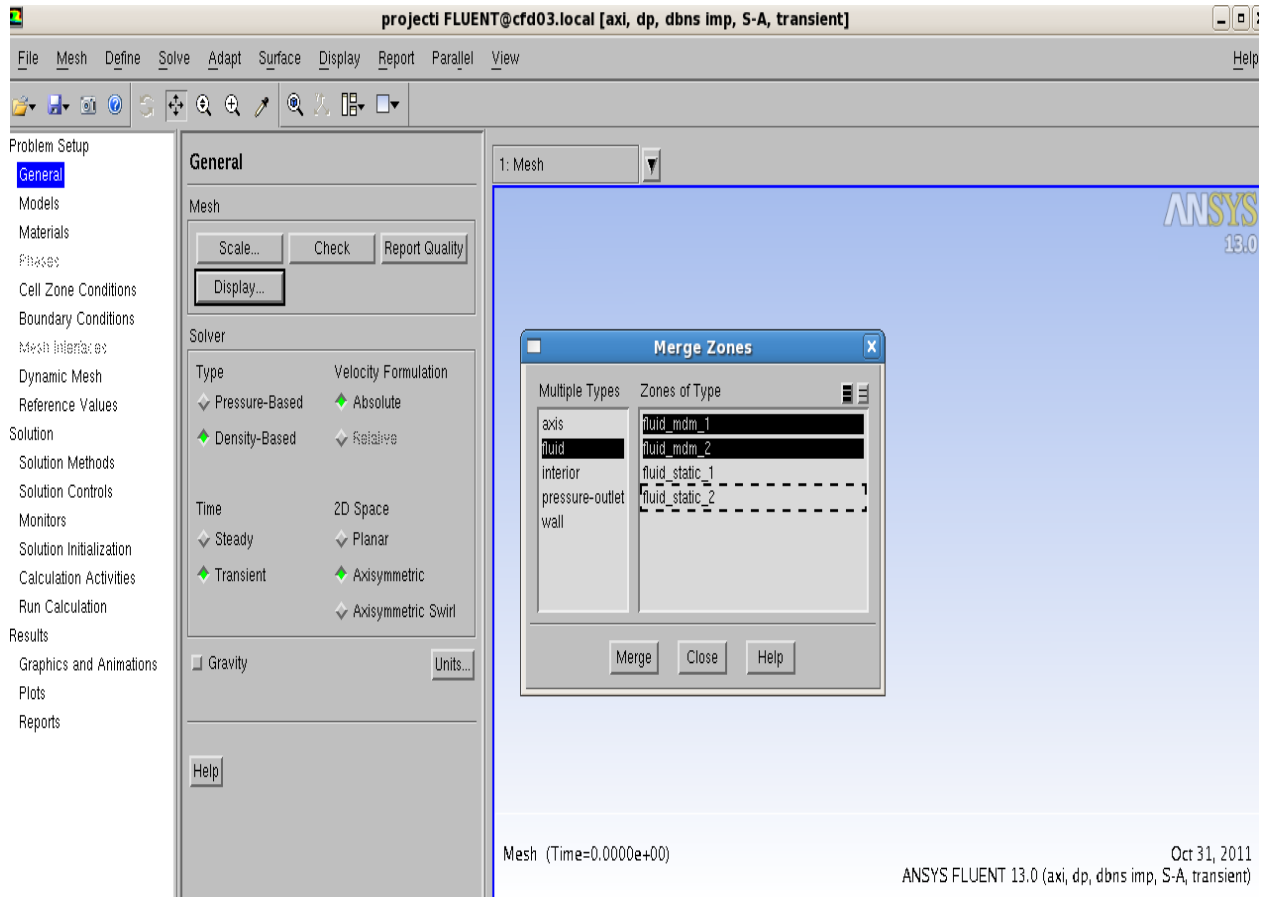
Figure 10.Settings of merge zone

Step 2: Models

a) Enable the Energy equation.

b) Set the turbulence model to Spalart-Allmaras (1 eqn).

Step 3: User-Defined Functions (UDF)

a) Compile the UDF.

Define ⟶ User-Defined ⟶ Functions ⟶ Compiled

34
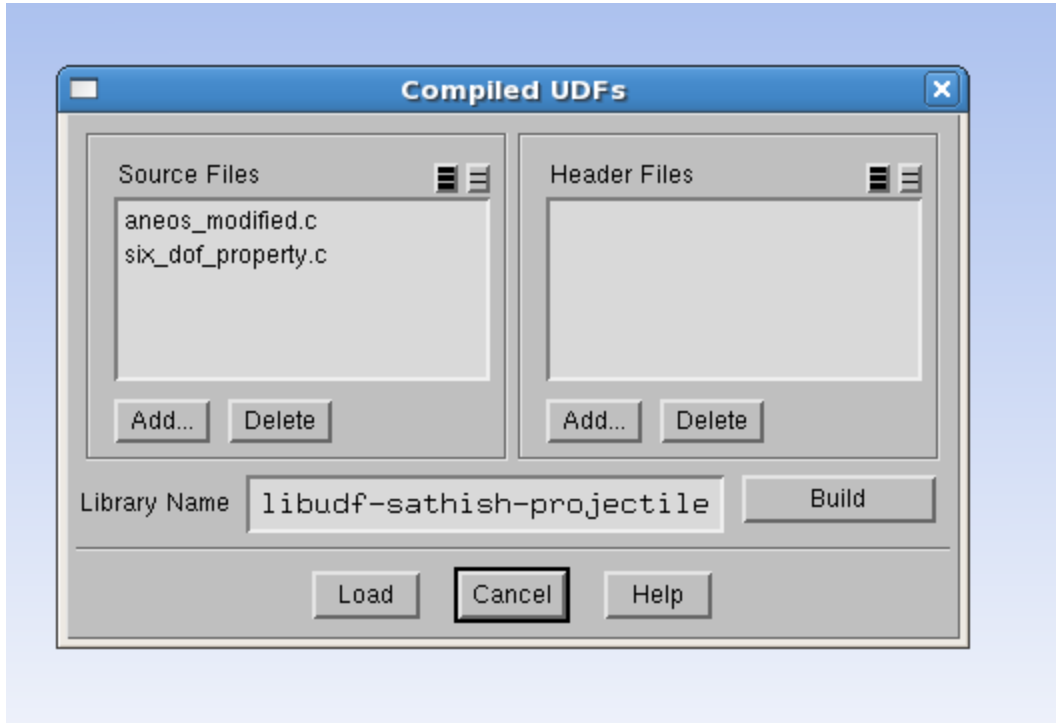
Figure 11.Compilation of source files

b) Click add and select the source files to be added. Figure 11 shows the compilation of source files.

c) Click build to build the library (Do not Load the files before enabling the real gas model).

d) Enable the real gas model using the TUI (Text User Interface) commands. Figure 12 shows the Text User Interface (TUI) commands to load the UDF in to Fluent.

```
> /define/user-defined/real-gas-models/user-defined-real-gas-model
use user-defined real gas? [no] yes
User-Defined Realgas Library Name [""] libudf-sathish-projectile

Opening user-defined realgas library "libudf-sathish-projectile"...
Library "libudf-sathish-projectile/lnx86/2ddp/libudf.so" opened

Setting material "real-gas-fluid" to a real-gas...

Loading Real-Ideal Library:
```

Figure 12.Text User Interface to load the UDF

e) Now load the UDF and close the dialog box.

Step 4: Boundary Conditions

The settings of boundary conditions can be seen from figure 13.

a) Set the boundary condition for P_inlet.
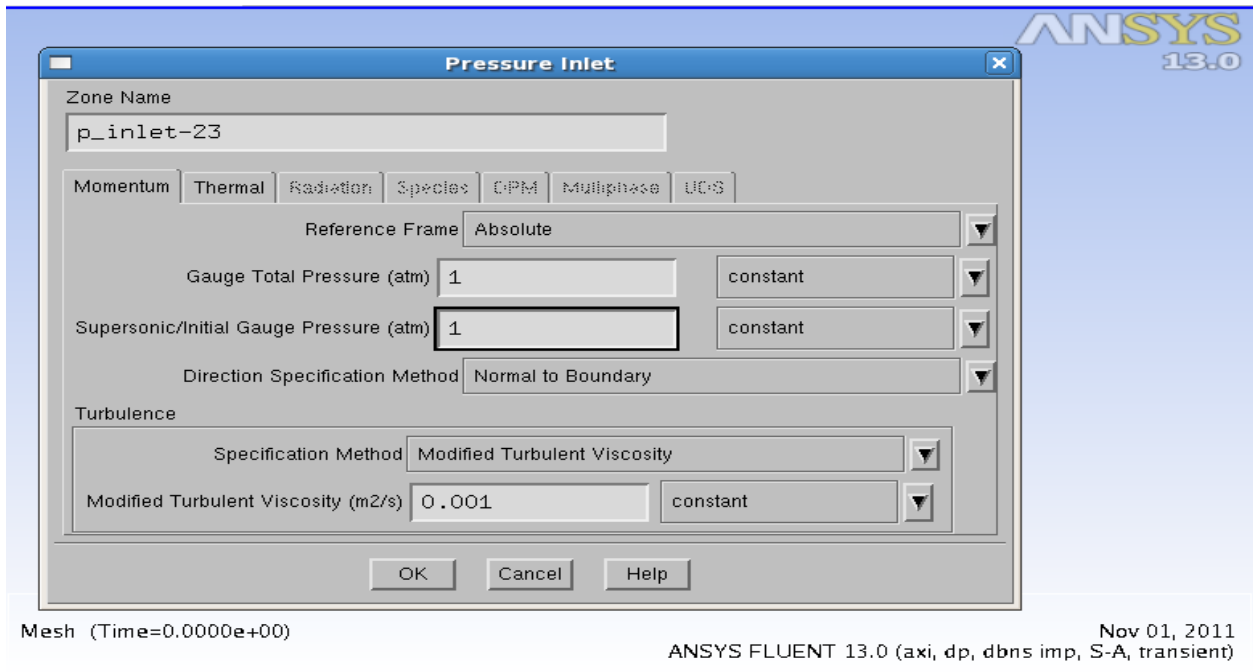
Figure 13.Settings of boundary conditions

Gauge Total Pressure and Supersonic/Initial Gauge Pressure are set to 1 atm. The total temperature is set to 300 K in the Thermal tab.

b) The same conditions of pressure and temperature are maintained for P_out_destruct, P_out_outer and upper_boundary.

c) Mesh Interface

Enter the name of the interface under mesh interface and select the corresponding interfaces in each zone as seen from figure 14.
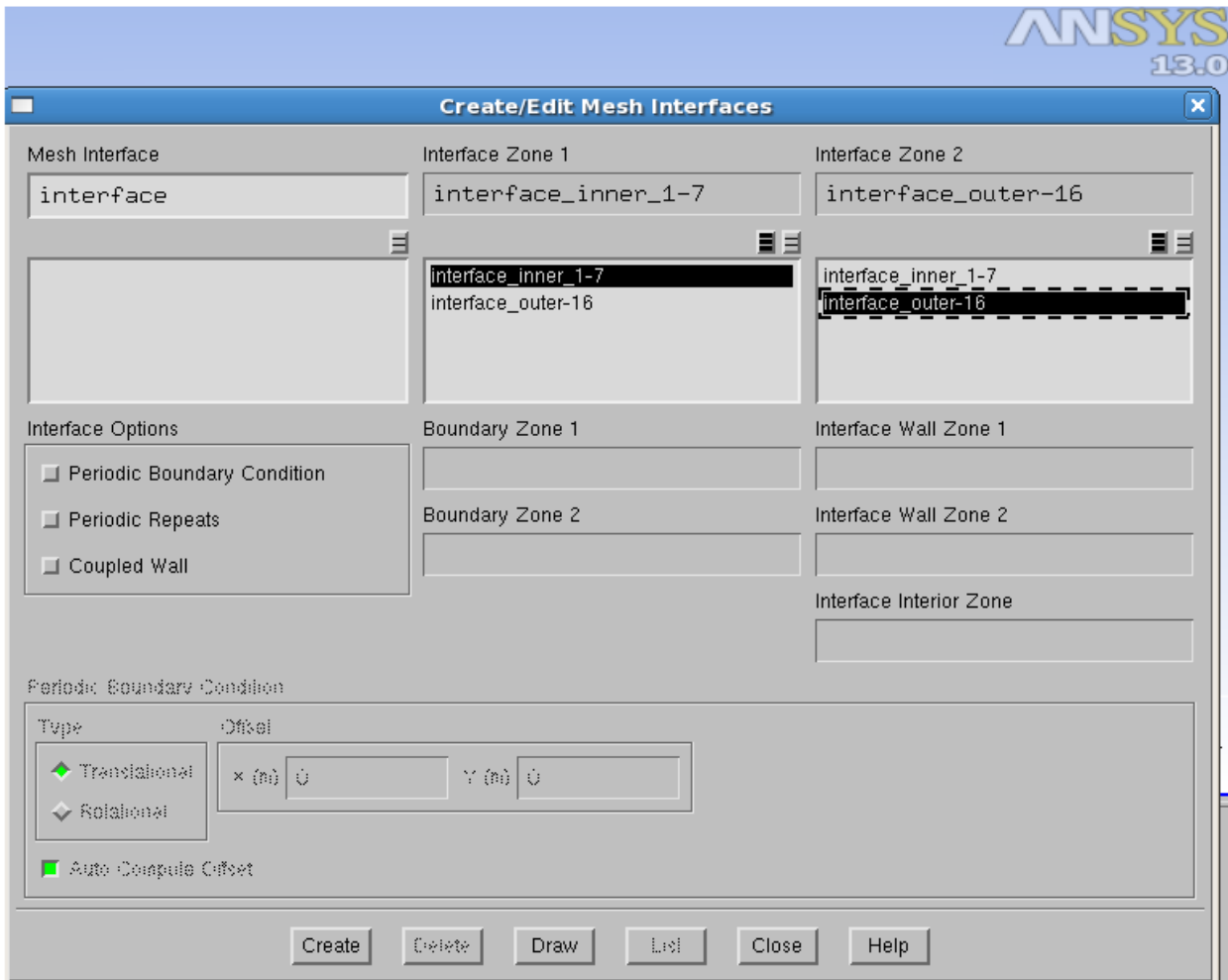
Figure 14.Mesh interface

Step 5: Dynamic Mesh Setup

Figure 15 explains the settings of dynamic mesh.

a) Enable Dynamic mesh and Six-DOF solver options.

b) Select the layering mesh method (Disable smoothing) and set the split factor and collapse factor to 0.4.

c) Specify the rigid body motion for the projectile and the surrounding fluid.

Fluid_mdm_1 is selected from the Zone name. Rigid body is selected under Type. Ensure that On is enabled in the Six DOF options and the UDF file is selected under Six DOF UDF.

d) Specify conditions for surfaces of projectile.

The surfaces of projectile are selected from the Zone names and rigid body motion is selected under Type. Set the appropriate cell height under meshing options.

e) Specify the stationary zone where new cell layers are created.

Select breech from the zone names and stationary from the Type. Set the appropriate cell height for meshing options.

f) Specify the stationary zone where old cell layers are destroyed.

Select P_out_destruct from the zone name and stationary from the Type. Keep the same cell height for meshing options.
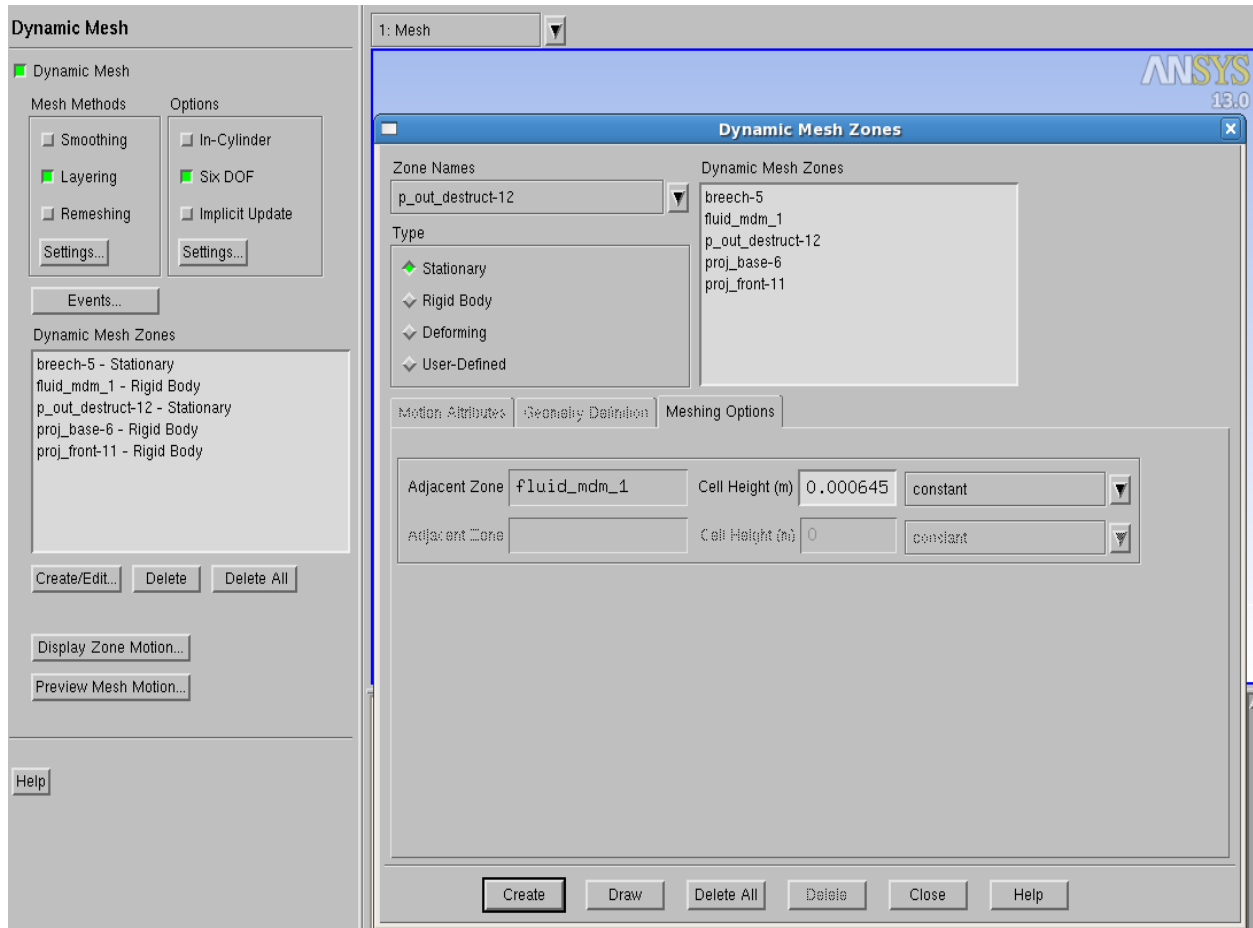
Figure 15.Settings of Dynamic mesh

Step 6: Solution methods

a) Implicit Formulation is chosen for this simulation.

b) Advection Upstream Splitting Method (AUSM) scheme is used for flux type.

c) Third order MUSCL scheme is used for the flow type.

d) First Order Implicit Transient Formulation is used.

Step 7: Solution Controls

40

a) Set the upper limits for pressure and temperature as seen from figure 16.
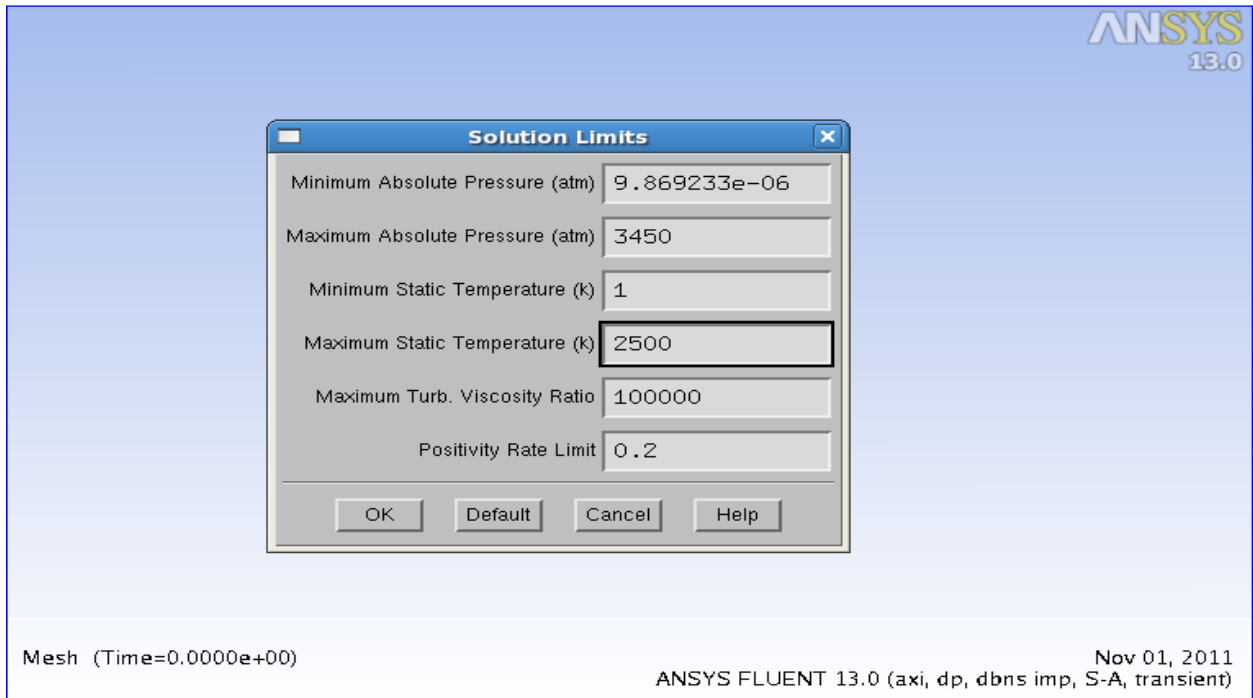
Figure 16.Settings of solution limits

b) Initialize the solution from P_inlet.

c) Define an adaption register. Figure 17 explains about adapting a region
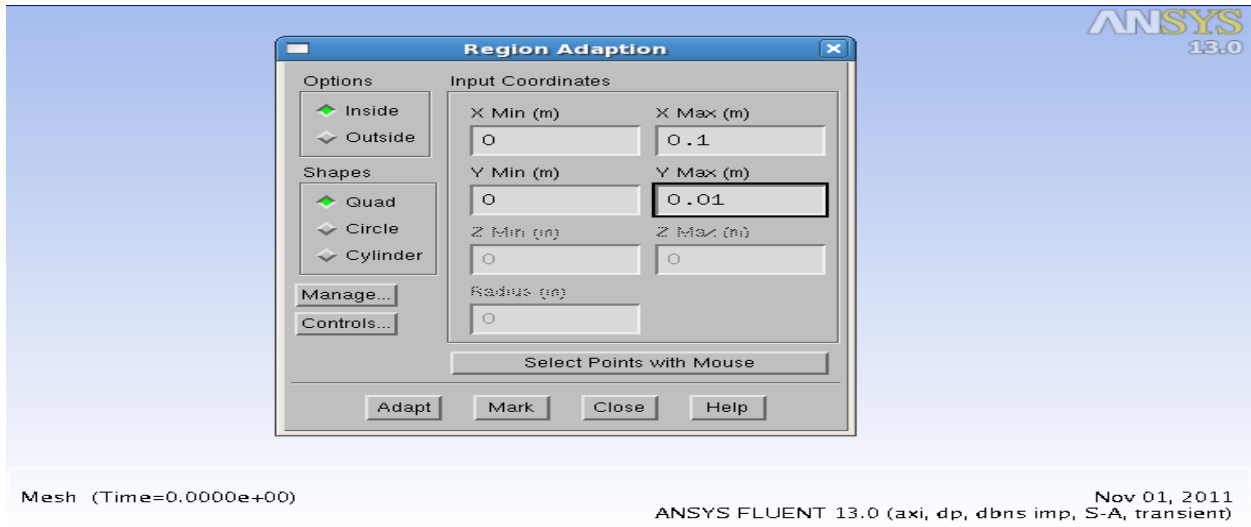
Figure 17.Settings of Region adaption

Enter the values of x and y for the region to be adapted. Click mark that will report the number of cells marked for refinement.

d) Click Manage to open the Manage Adaption Registers dialog box and click Display. Figure 18 shows the display of adapted region and marked as hexahedron-r0.

Adaption Markings (hexahedron-r0) (Time=0.0000e+00)   Nov 01, 2011
                    ANSYS FLUENT 13.0 (axi, dp, dbns imp, dynamesh, S-A, transient)

Figure 18.Adaption markings of high pressure region

The chamber now appears in red.

e) Patch the chamber. Figure 19 shows the settings for patching the red color chamber.
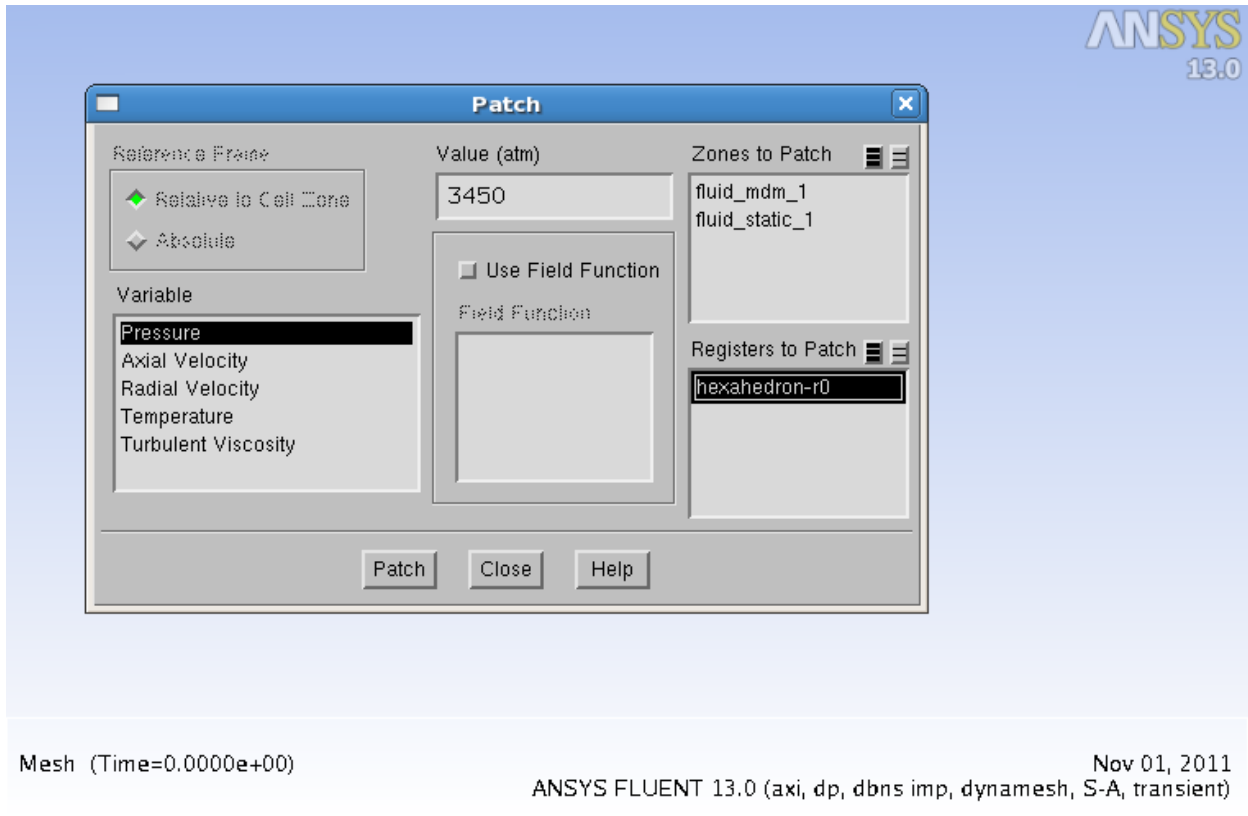
Figure 19.Patch settings for pressure and temperature

Select the variables to be patched and enter the values. Ensure that hexahedron-r0 is selected under Registers to Patch. For this simulation pressure and temperature are patched at very high values of 3450 atm and 2500 K respectively.

Step 8: Calculation Activities

a) Save the solution file for every 50 time steps. This can be done using the Autosave options.

b) Set the time step size to 1e-07 s.

c) Enter the Number of time steps. (Flow time = Time step size * No. of time steps)

d) Set the Max Iterations/Time step to 20.

44

e) Calculate the solution.

It took 4-5 days to run a simulation. Once the solution converges, post processing of results was done by using TECPLOT.

5.0 COMPARISON

The present computations are validated using other available CFD results [1]. In this section, the effect of projectile mass and various initial conditions were compared against the reference [1]. The author also studies the viscous effect of unsteady projectile aerodynamics and compared with the inviscid cases.

5.1 EFFECTS OF PROJECTILE MASS

Case 1 and 2 were used to study the effects of projectile mass. The following figure shows the present computed iso-pressure contours for Case 2 which are compared with the Reference [1]. The left side shows the present computations while the right side shows the result of Reference [1].
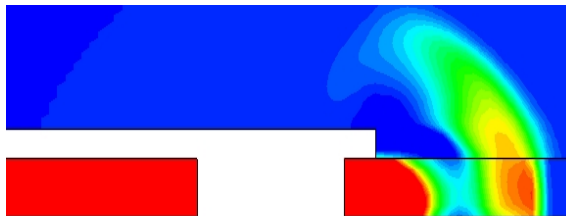
(a) $t = 0.364$ ms
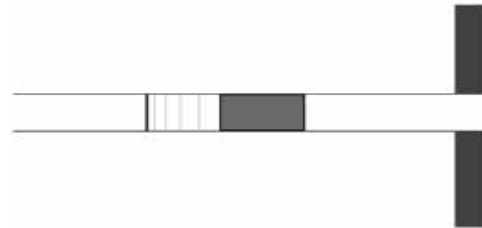
(a) $t= 0.364$ (ms)
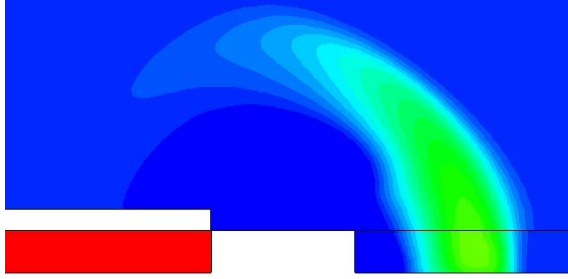
(b) *t*= 0.428 ms
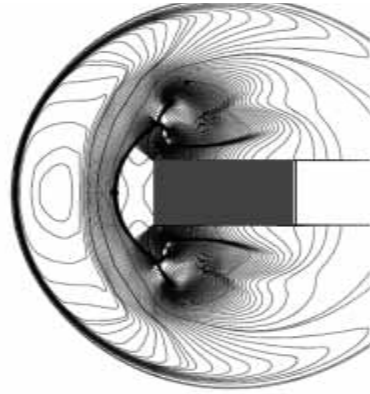
(b) *t*= 0.428 (ms)
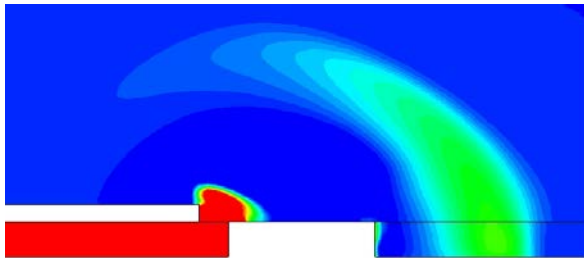
(c) *t*=0.435 ms
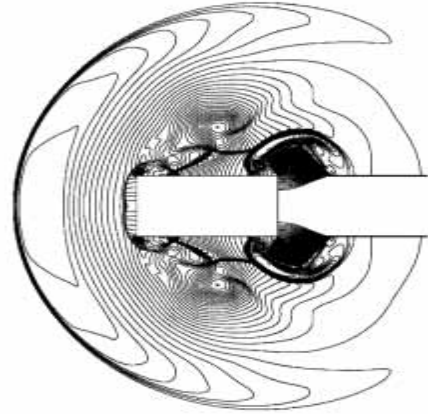
(c) *t*= 0.435 (ms)
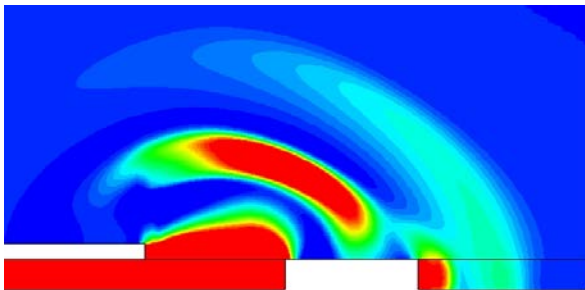
(d) *t* = 0.483 ms
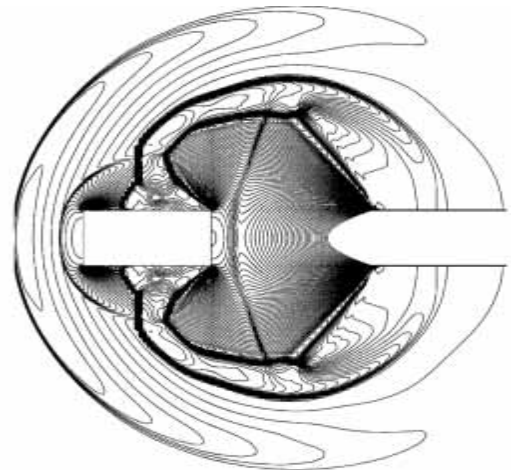
(d) *t*= 0.483 (ms)
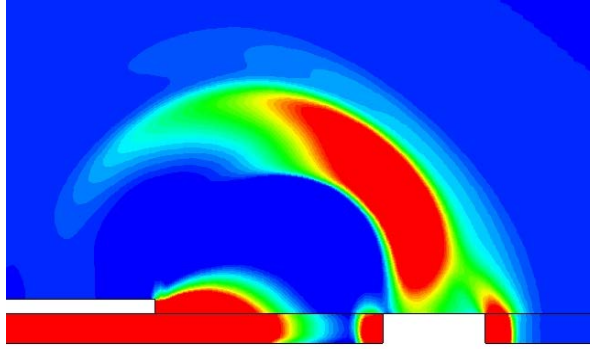
(e) $t = 0.541$ ms
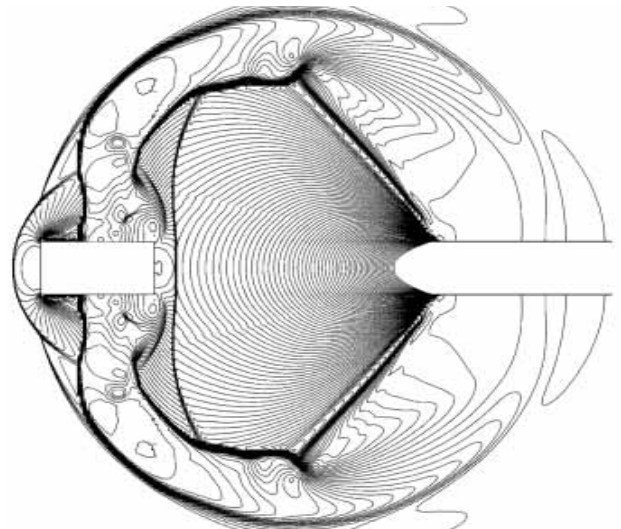
(e) $t = 0.541$ (ms)
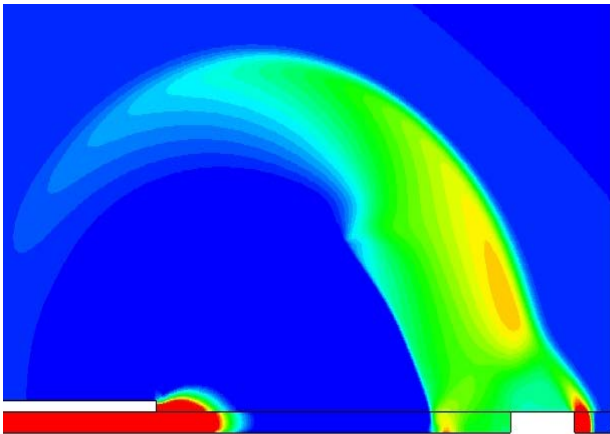
(f) $t = 0.716$ ms
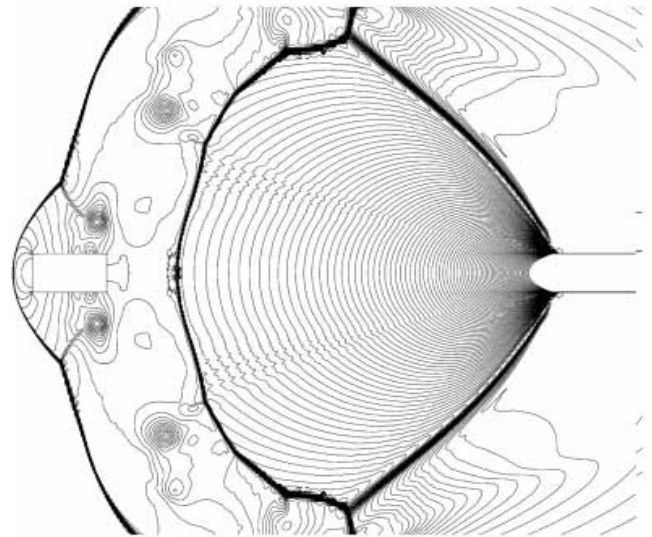
(f) $t = 0.716$ (ms)

Figure 20. Computed pressure contours for Case 2 to study the mass effect [1]

With the beginning of computations, the projectile moves from its starting position to the exit of launch tube. At $t= 0.364$ ms, compression waves formed in front of the projectile, are seen to be coalescing into a shock wave, while the projectile is moving still inside the launch tube. The present computation shows, at $t= 0.428$ ms, the projectile is on the verge of discharging from the launch tube which is in good agreement with the reference [1] at $t=0.428$ ms. The jet flow in front of the projectile is completely developed at the open end of the launch tube. A contact surface is formed which separates the gas in the supersonic jet and the background gas which has early been perturbed by the blast wave. The secondary shock wave is also formed in between the starting vortices. It grows in strength as the projectile moves out of the launch tube.

At $t= 0.435$ ms, the projectile is shortly after discharging from the exit of the launch tube, and the first blast wave is spreading around. When the projectile is discharged from the launch tube, the high pressure, high temperature gas behind the projectile rushes out as a highly under-expanded jet, forming a second blast wave with another contact discontinuity just downstream of the blast wave. Unlike the first blast wave, the second blast wave develops in the radial direction as the axial component is deteriorated due to its strong interaction with the supersonic jet around the projectile and the shear layer. However, the wave system behind second blast wave will be stronger as it moves in to relatively slower background gas and the background flow becomes non-uniform too. The present computation shows the same flow features as Reference [1].

At $t=0.483$ ms, the projectile after overtaking the secondary shock wave behind the first blast wave, runs in to a low speed gas. The projectile Mach number relative to this low speed gas becomes supersonic under the present computational conditions, leading to the generation of the bow shock in front of the projectile. At $t=0.541$ ms, the secondary shock wave is diffracting at the rear end of the projectile and the bow shock wave is interacting with the first blast wave. At

t=0.716 ms, the bow shock wave becomes stronger after it is merged with the first blast wave, and the projectile moves through still air. It is interesting to note that the projectile overtakes the blast wave completely within a very short distance from the exit of the launch tube. The present computation is in good agreement with the reference [1] and the slight differences between the flow fields can be attributed to the pressure contour levels chosen and the time differences between the compared figures.

In order to study the effects of projectile mass on the unsteady projectile aerodynamics, the time histories of projectile velocity and acceleration graphs were plotted and compared with [1] for comparison. Case 1 (light mass) and Case 2 (heavy mass) were used to study the mass effect.
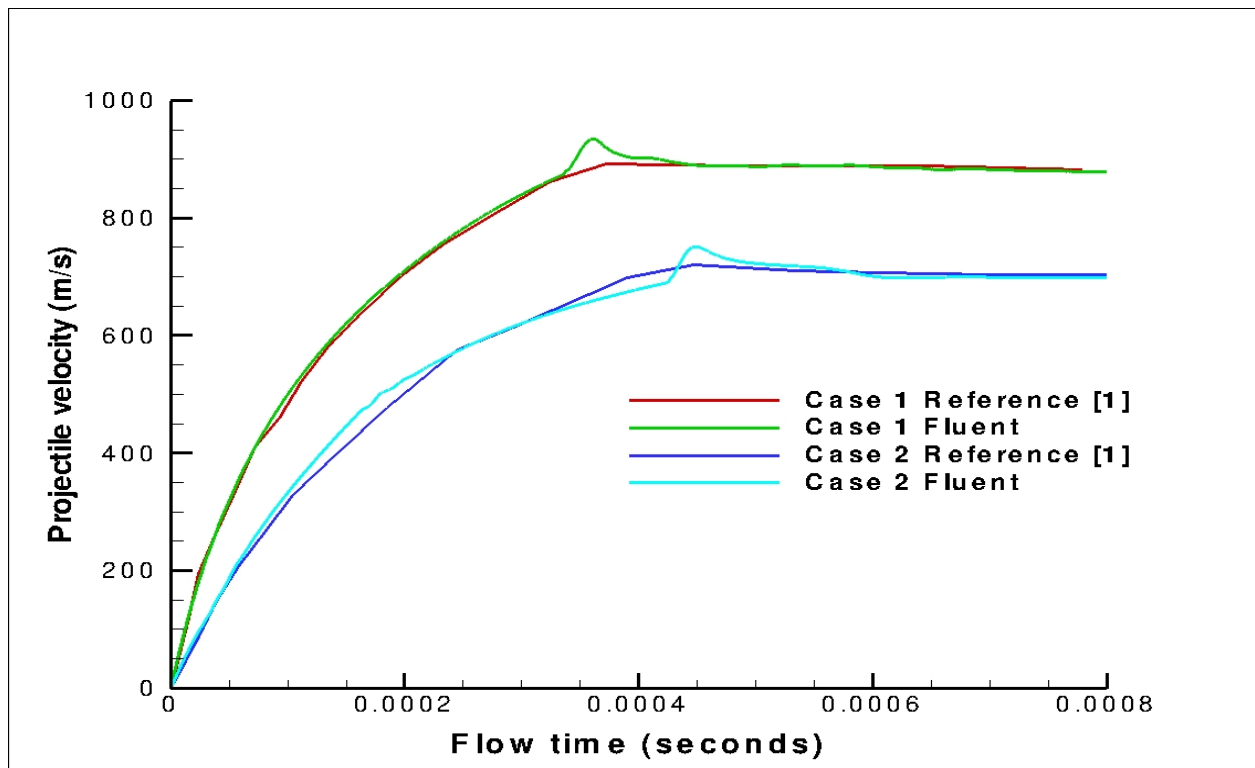
Figure 21.Time histories of projectile velocity for Cases 1 & 2

From the above graph, it is evident that the lighter mass has higher projectile velocity when compared to heavy mass. The present computed graph is in good agreement with the Reference [1] except for the hike in velocity at the exit of the launch tube. The lighter mass projectile comes out of the launch tube quicker than the heavy mass. When the projectile is at the exit of the launch tube, there is a further increase in velocity, corresponding to a decrease in pressure for an inviscid case. Most researchers take the muzzle velocity as the maximum projectile velocity. In reality, after the projectile exits the barrel, the projectile velocity will increase further due to the influence of high pressure gas ejected from the gun bore.

When the computation begins, the projectile accelerates suddenly inside the launch tube, and this instantaneous acceleration persists over a very short distance due to the inertia effect even after the projectile has been discharged from the launch tube. This duration is strongly dependant on the projectile mass. Later, the projectile is decelerated due to aerodynamic drag effects.

Figure 22 shows how the projectile acceleration varies for each case. The impulsive acceleration of the projectile at the beginning of the computation suddenly decreases due to the wave drag from the compression waves in front of it. A sharp reduction in the acceleration can also be seen later when the projectile is discharged from the launch tube. The acceleration of the projectile is strongly affected by the mass. From the graph, it is seen that for a lighter projectile, the acceleration is more. It is noted that when the mass of the projectile doubles, the acceleration of the projectile becomes halved. Thus the present computation proves to be in good agreement with the reference [1]. Fluent does not plot acceleration directly. However, it reports the Force values for every time step. Once the magnitude of force for a particular time step is known, acceleration can be calculated using Newton's second law. The mass of the projectile is constant.
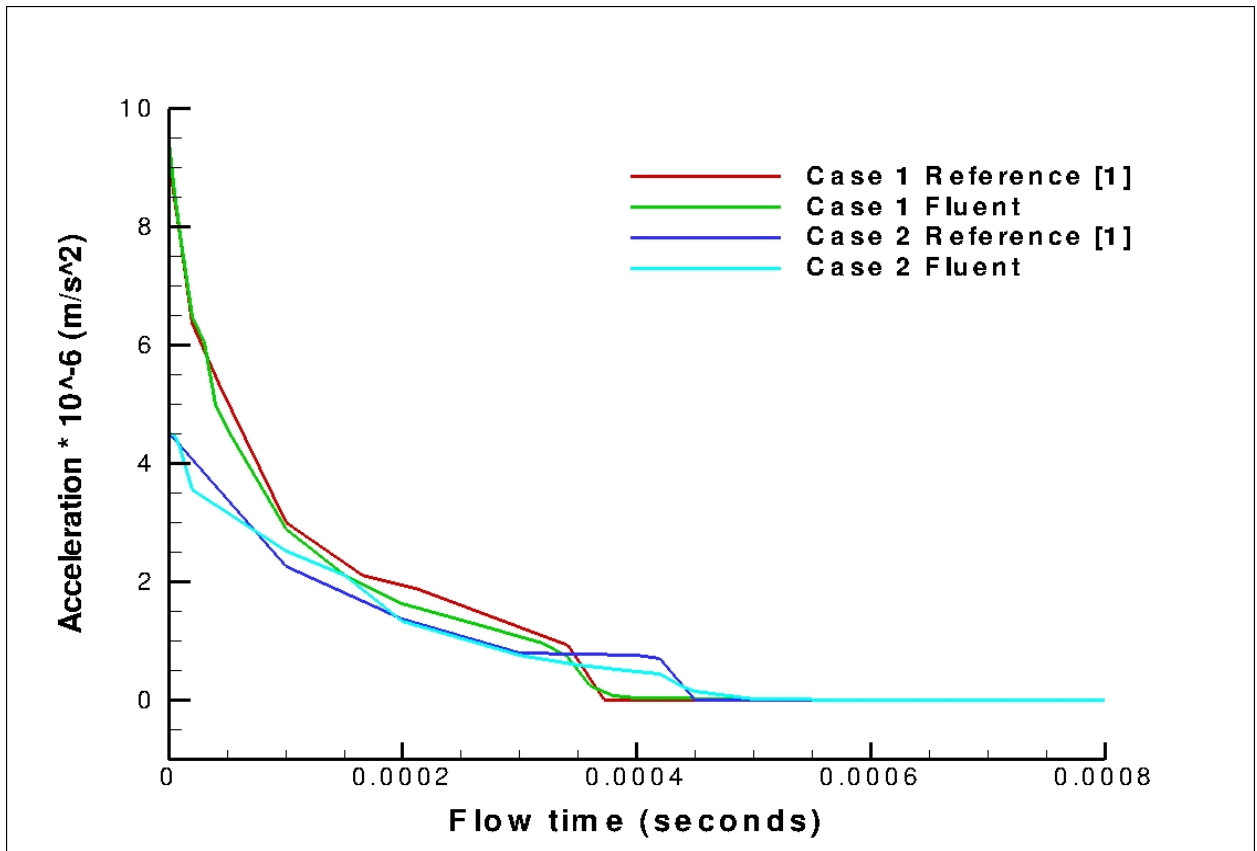
51

Figure 22.Time histories of projectile acceleration for Cases 1 & 2

5.2 EFFECT OF INITIAL CONDITIONS BEHIND THE PROJECTILE

Cases 2, 3 and 4 were used to study the effect of initial conditions behind the projectile. The still air outside the launch tube is kept at P= 1 atm and T =300K. The mass of the projectile is kept constant(M=0.0135 kg) for all the three cases. The reduced temperature effect is studied by Case 3 and the reduced pressure effect is studied by Case 4.

Figure 23 shows the time histories of projectile velocity for various initial conditions and they are compared with the Reference [1].
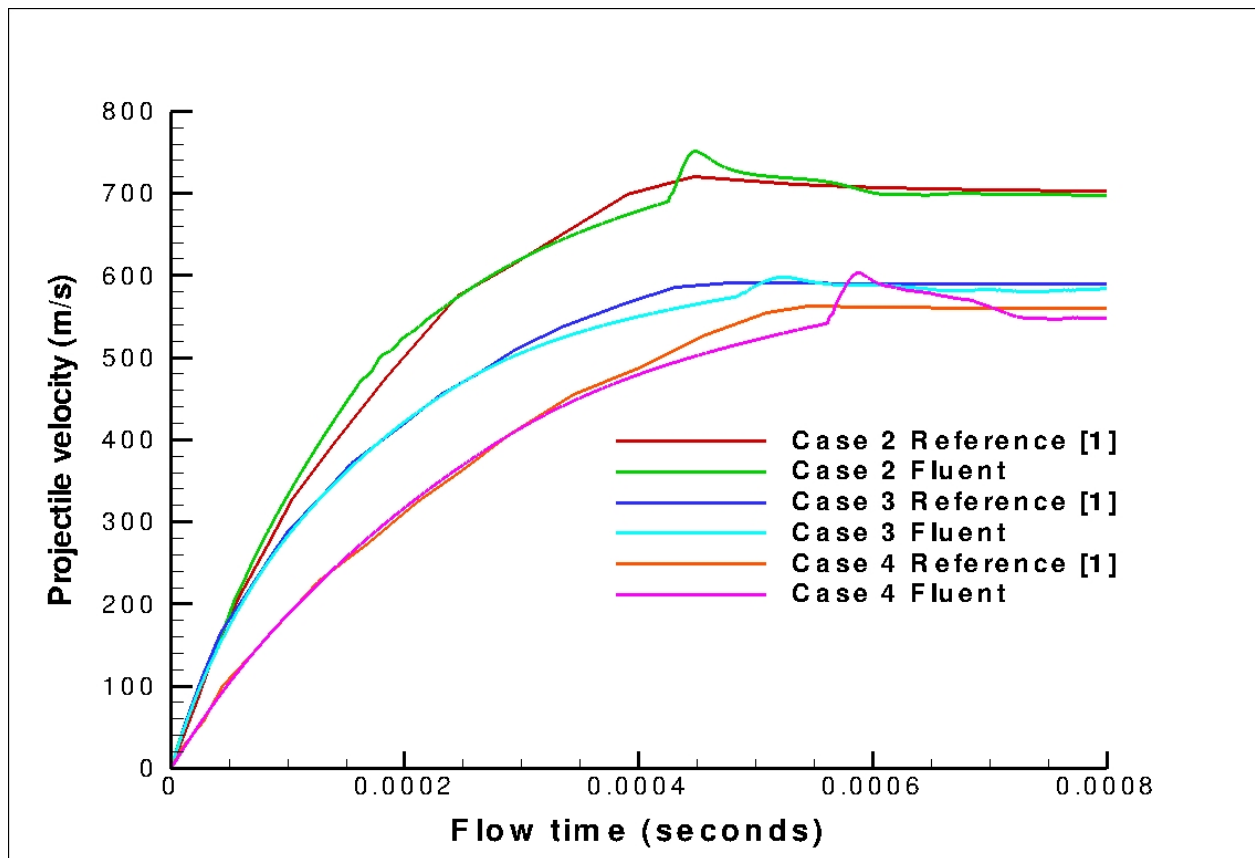
Figure 23.Time histories of projectile velocity for various initial conditions

As expected, the projectile attains the maximum velocity in the case of highest initial pressure and temperature behind it. The projectile velocity reduces significantly when the initial pressure is reduced (Case 4) compared to the other two cases, due to a much weaker gas expansion behind the projectile. From the graph it is observed that the high pressure gas projectile comes out of the launch tube at t=0.43 ms, whereas the low pressure gas projectile comes out at t=0.56ms. The present computation results are mostly in good agreement with the reference [1] except for the hike in velocity at the exit of the launch tube. Once the projectile comes out of the launch tube, it moves with constant velocity. There is a sudden decrease in pressure at the exit of launch tube, which corresponds to a hike in velocity at the exit. From the above graph, it is interesting to note that for Case 3(reduced temperature) the intensity of velocity hike is reduced when compared to other two cases. It is because the decrease in pressure at the exit of launch tube is less, when compared to Cases 2 and 4 where the pressure decreases suddenly. This can be seen from the figure 23.
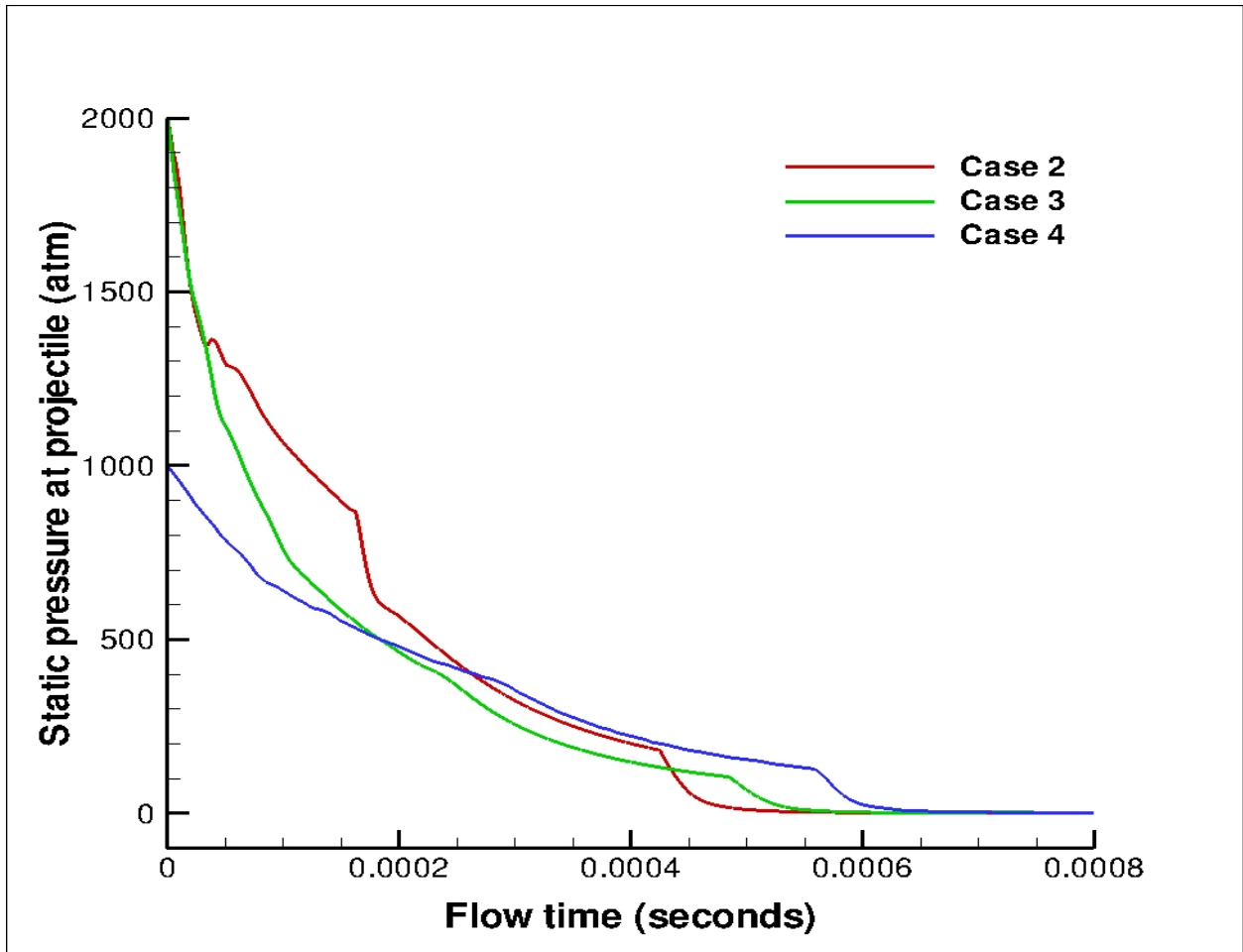
Figure 24. Static pressure at base of the projectile for various initial conditions

Most researchers take the muzzle velocity as the maximum projectile velocity. In reality, after the projectile exits the muzzle, the projectile velocity will increase further due to the influence of high pressure gas ejected from the barrel [9].

In order to investigate the effects of the initial conditions on the near-field of the launch tube, pressure were measured at four different locations for Cases 2, 3, and 4. The present computed pressures at these locations were compared with the bench mark case of reference [1] for comparison. The present CFD results were almost matched with the bench mark case.

The following graphs compare the pressure fluctuations at different locations between the present computation and Referance [1] for Cases 2, 3 and 4.
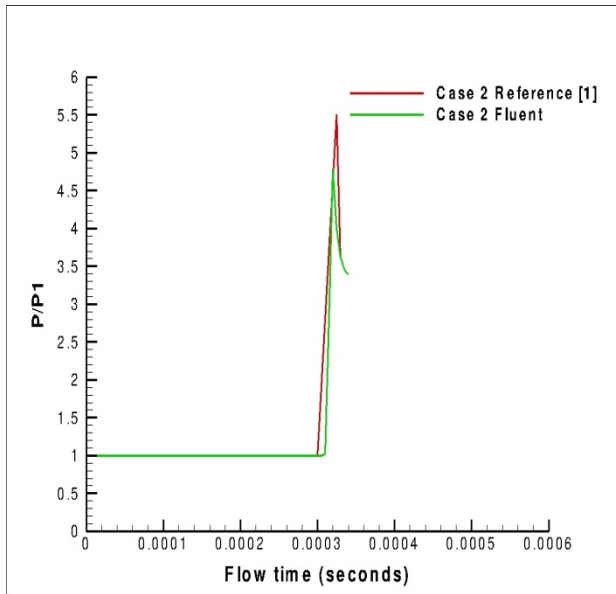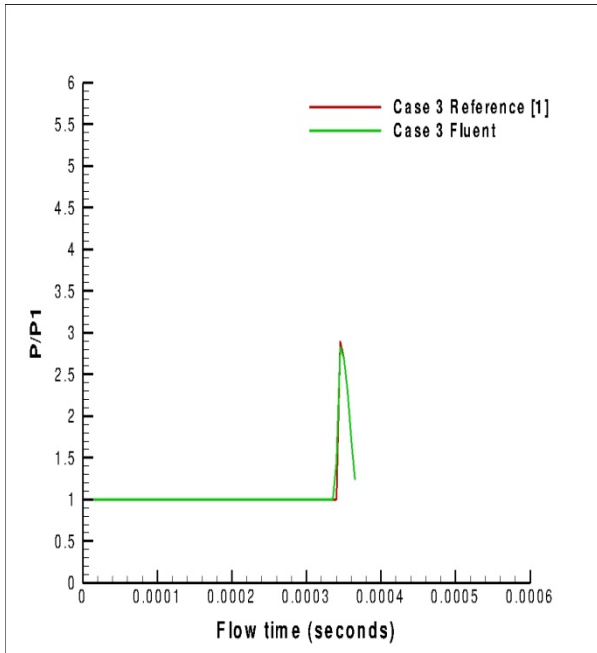

Point 1
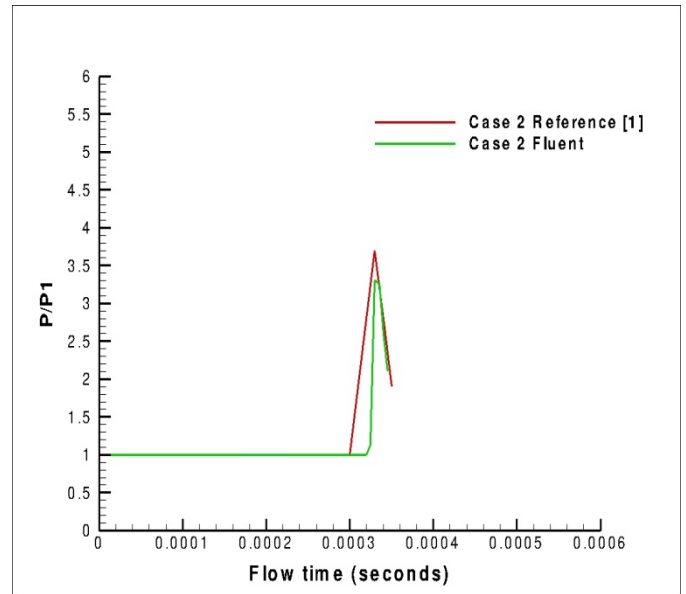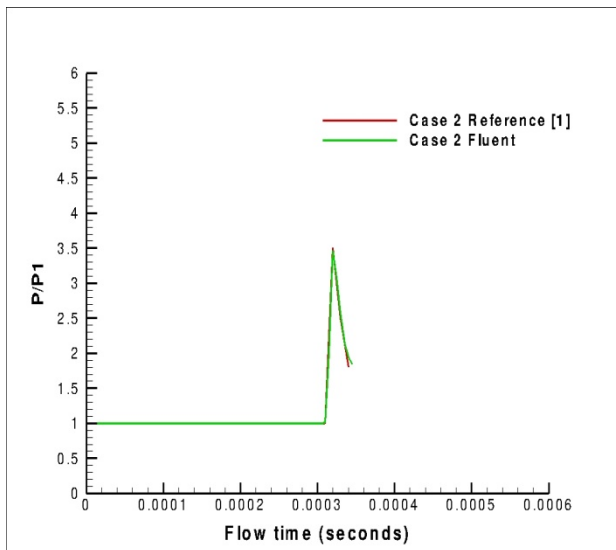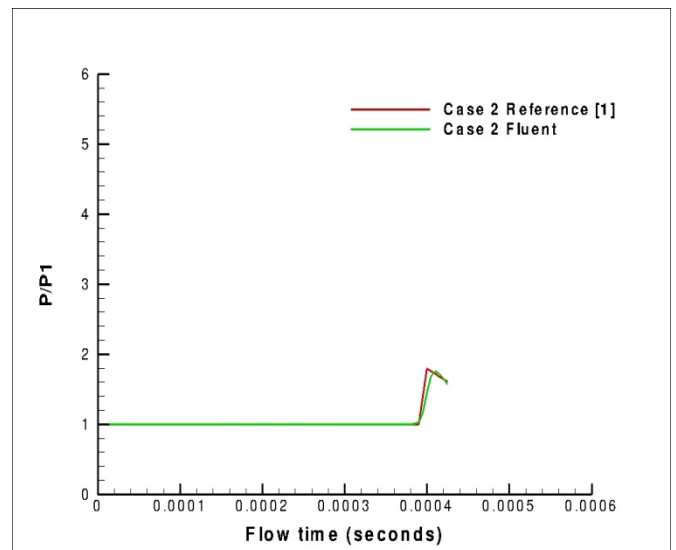
Point 2

Point 3

Point 4

Figure 25. Pressure variations at various measuring locations for Case 2
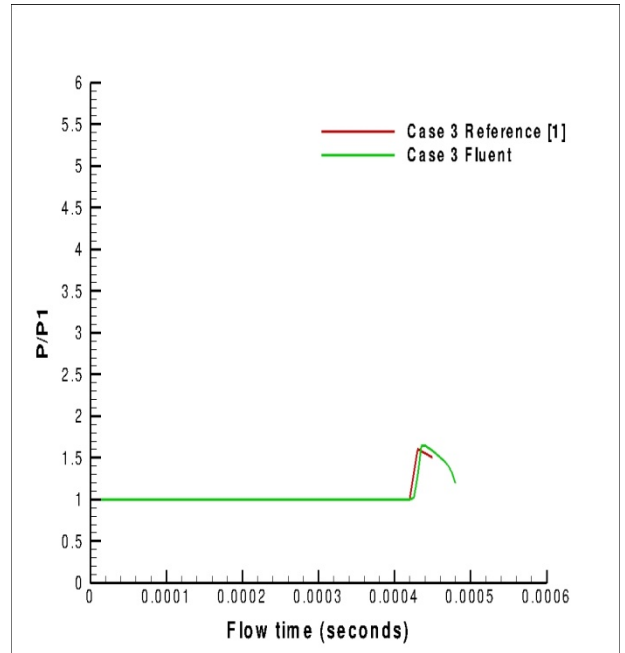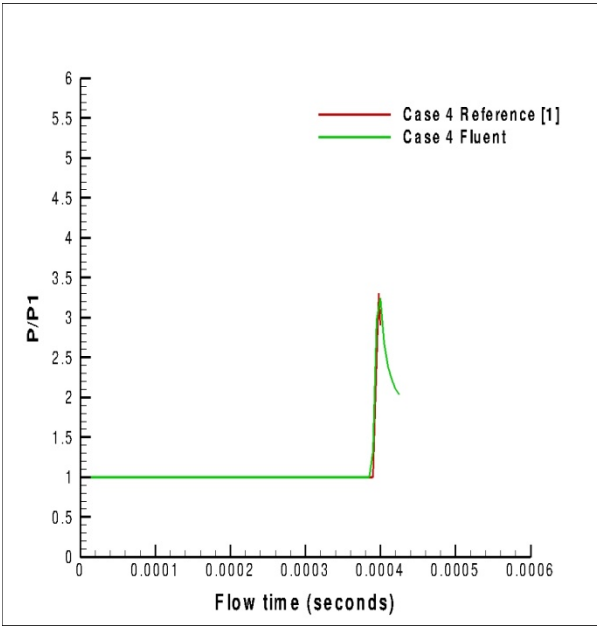
56
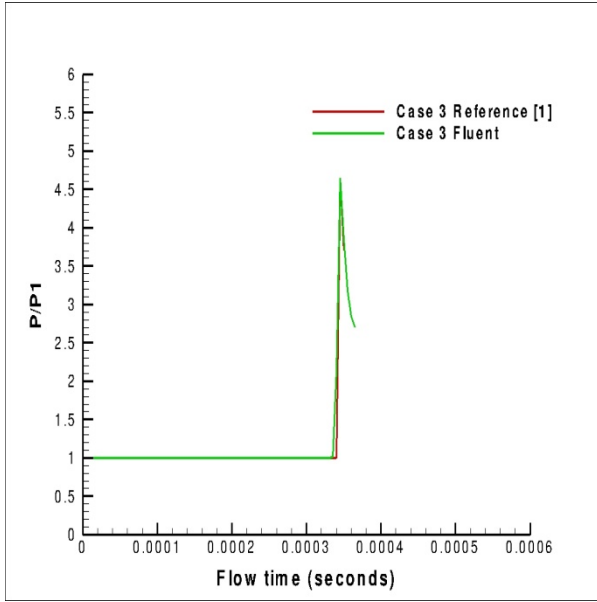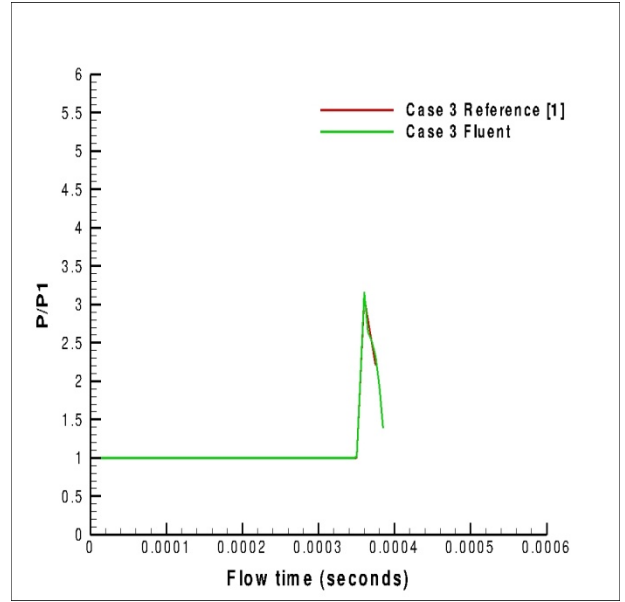
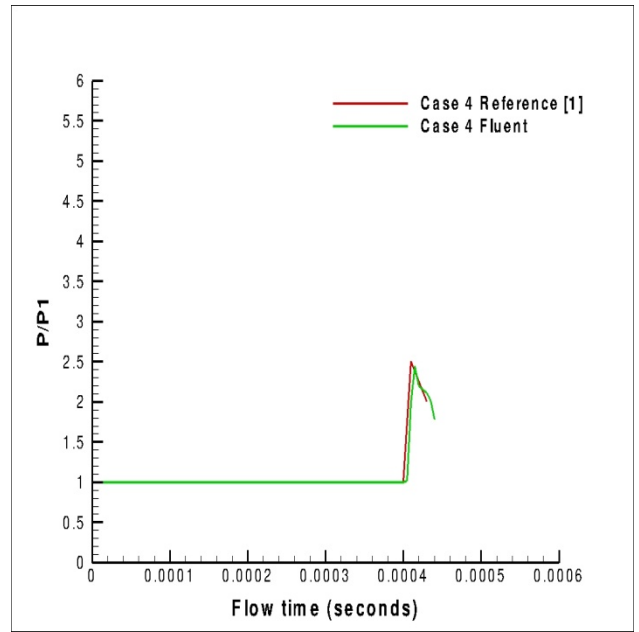Point 1

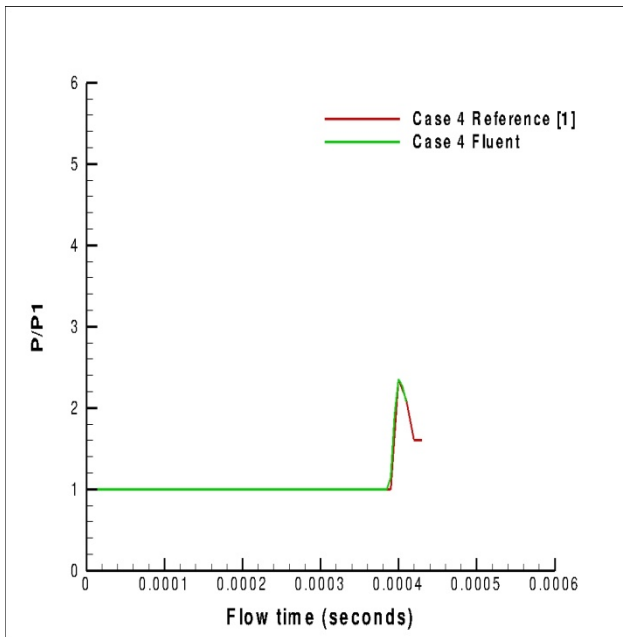Point 2

Point 3

Point 4

Figure 26.Pressure variations at various measuring locations for Case 3
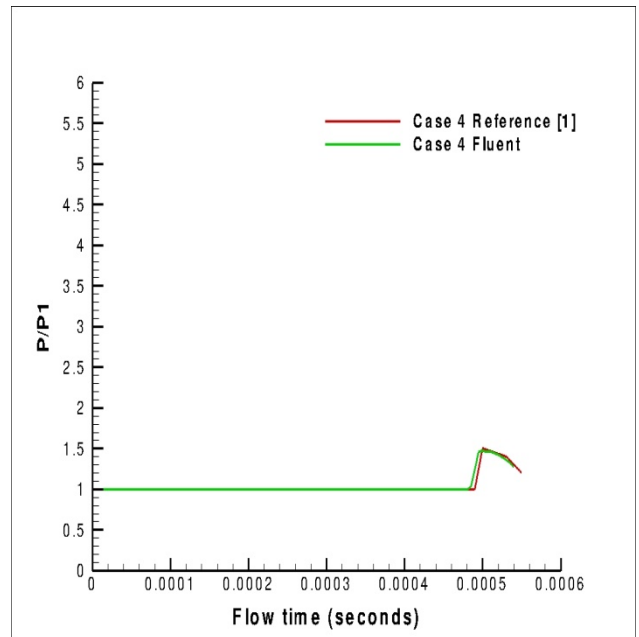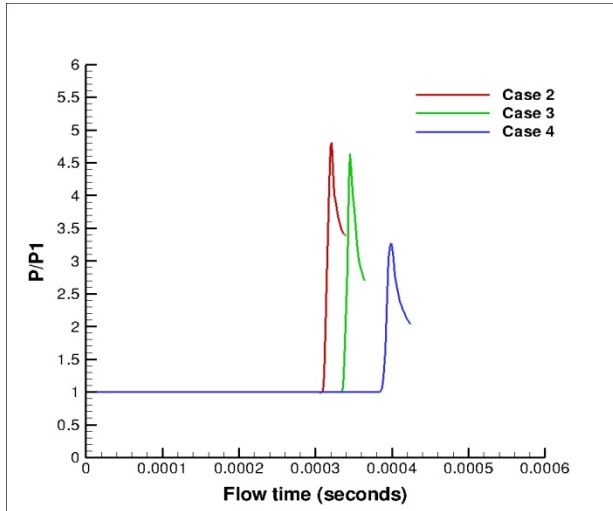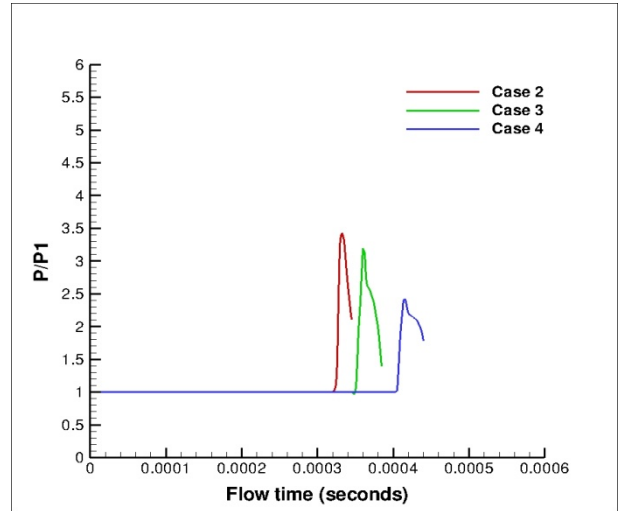
Point 1

Point 2

Point 3

Point 4

Figure 27.Pressure variations at various measuring locations for Case 4
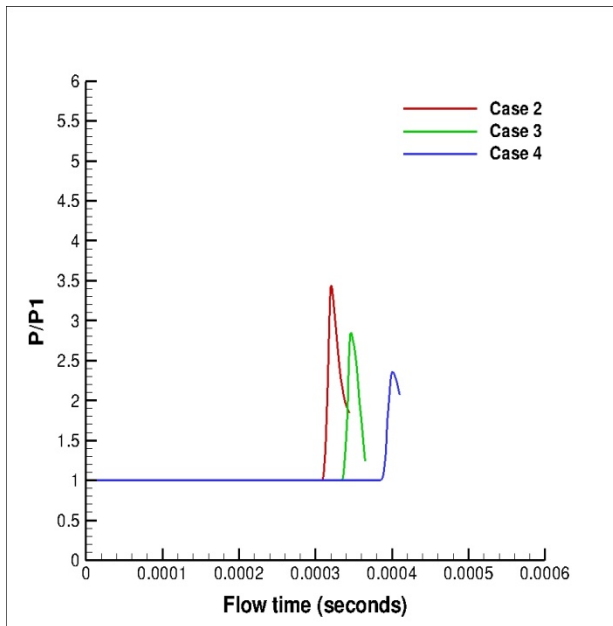
Figure 28 shows the present computed pressure history at different locations (P1, P2, P3, and P4) for Cases 2, 3 and 4.
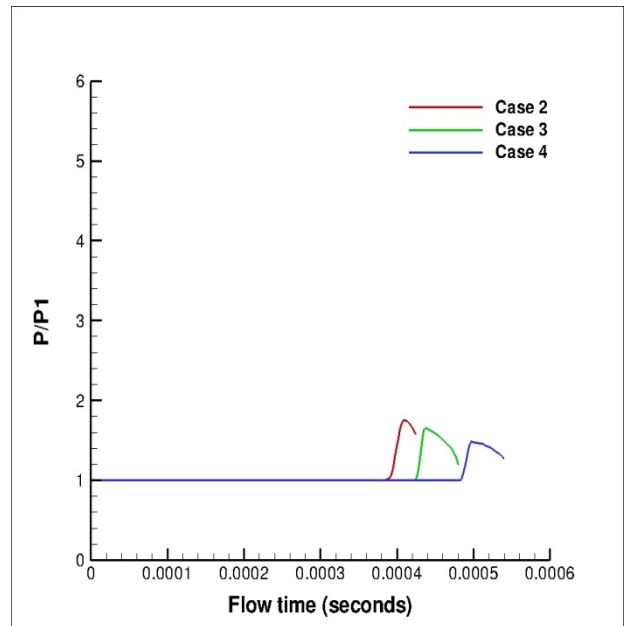

Point 1

Point 2

Point 3

Point 4

Figure 28.Computed pressure histories at various locations for Cases 2, 3 and 4

The still air outside the launch tube is at atmospheric conditions. Point 1 (P1) which is close to the launch tube (10 mm from the exit) has the high intensity of pressure values. The intensity of pressure value for Case 4 (reduced pressure) is less compared to that of Case 2 possibly due to a weaker expansion of gas behind the projectile. From the graph, it is evident that the high pressure gas projectile comes out quickly from the launch tube, whereas low pressure gas projectile takes more time. The amplitude of pressure value decreases if the point is further downstream. The reduction in the initial pressure behind the projectile has a stronger influence than that in initial temperature on the development of the blast wave and flow field [1].

The magnitude of the pressure variations in point 2 is less compared with that of point 1 due to the fact that the blast wave is expanding both temporally and spatially. This can clearly be understood by observing the pressure rise due to the supersonic jet in front of the projectile. The supersonic jet is expanding as it progresses along with the blast wave and hence the pressure rise at point 2 due to the supersonic jet will be not as high as point 1. The qualitative features of the pressure history at various locations in Case 3 are the same as that of Case 2 [1]. The amplitudes of the pressure variations at locations 3 and 4 are not as high as that of points 1 and 2 due to the expanding blast wave. The pressure fluctuations at locations 3 and 4 in Cases 3 and 4 are qualitatively the same as that of Case 2. Even though the general features of the flows in Case 2 and Case 3 are the same, there are big differences in the amplitudes of the pressure fluctuations due to the weaker expansion of the gas behind the projectile due to a reduced initial temperature. A significant reduction in the strength of the pressure variations at points 3 and 4 can be observed in Case 4 compared with Case 2, due to the reduced initial pressure and hence a weaker gas expansion behind the projectile.

5.3 VISCOUS EFFECTS

Case 5 and Case 6 were used to study the effects of viscous terms on the unsteady projectile aerodynamics. Various numerical solutions have been reported in the past decade for different muzzle-blast cases for Eulerian equations; therefore, they do not include the effects of viscosity [14]. A viscous flow is one where the transport phenomena of friction, thermal conduction and/or mass diffusion are included. These transport phenomena are dissipative - they always increase the entropy of the flow [17]. The following figures show the comparison between Case 5 and Case 6 (reduced temperature effect) for projectile velocity, static pressure at breech and static pressure at base of the projectile.
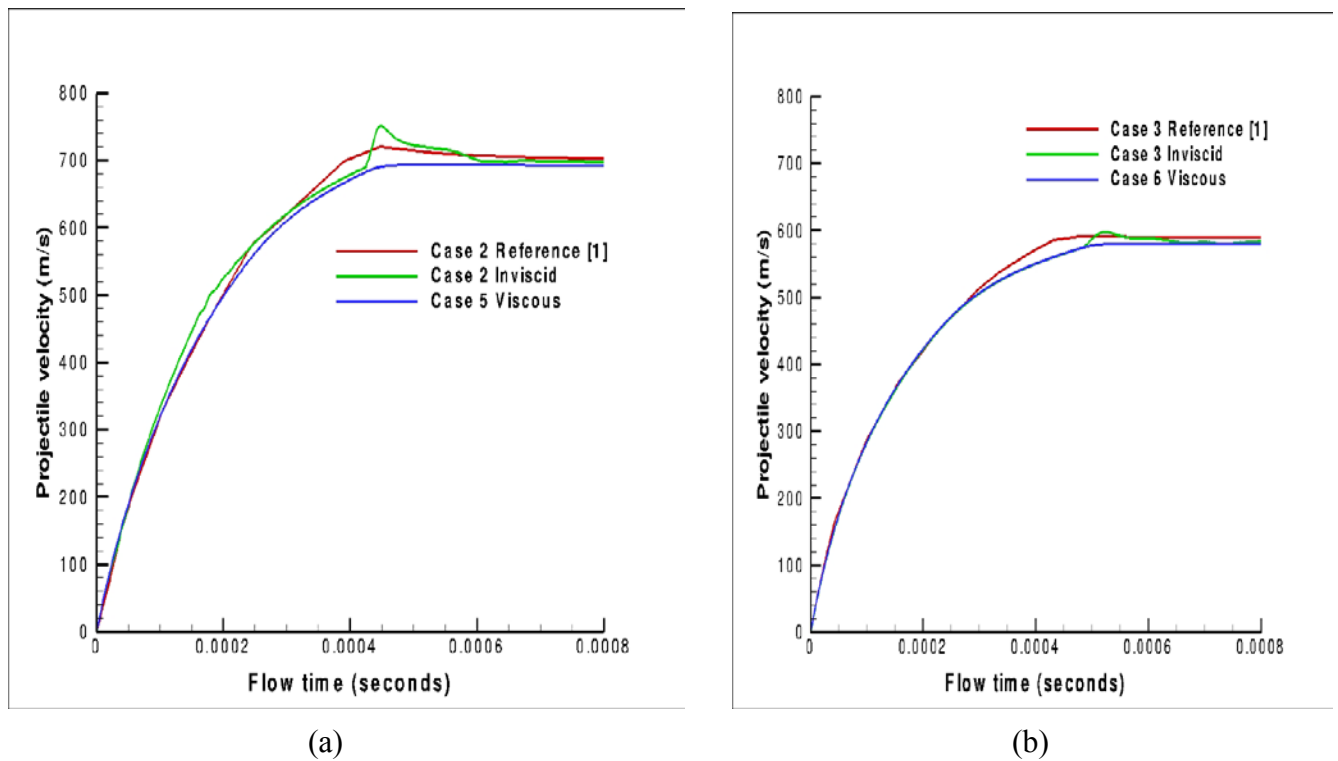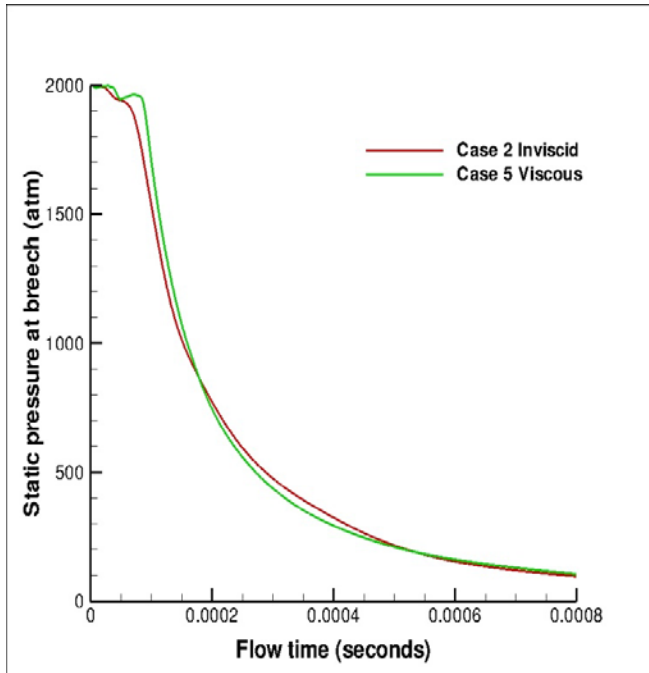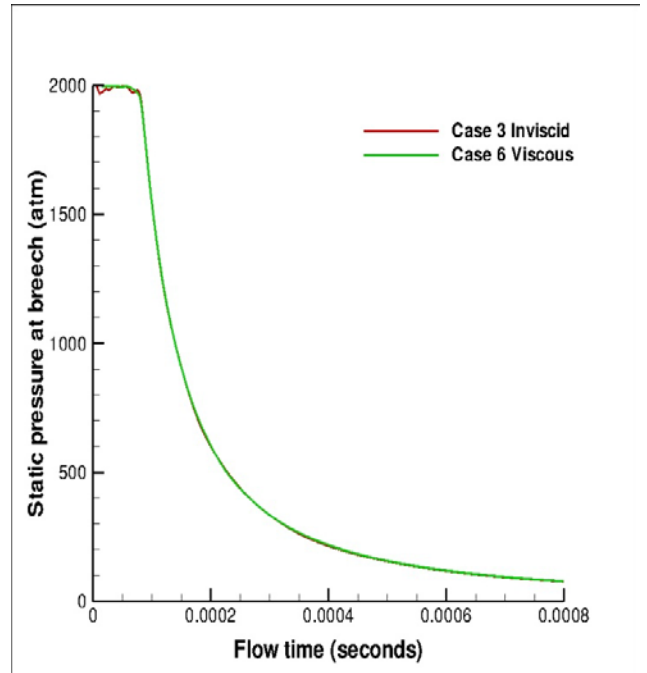

(a)   (b)

Figure 29.Comparison of time histories of projectile velocity between viscous and inviscid for Cases 5 & 6

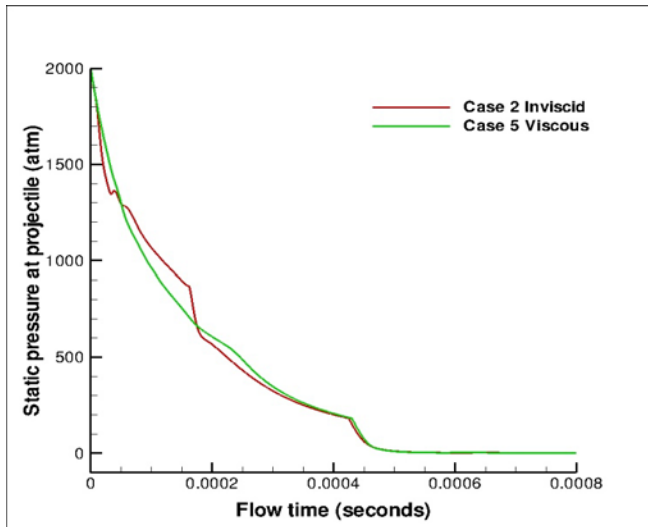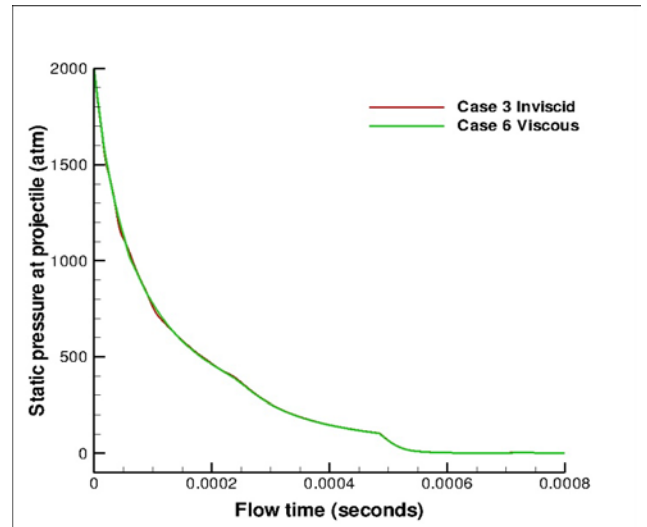(a)   (b)

Figure 30.Comparison of time histories of static pressure at breech between viscous and inviscid case

(a)   (b)

Figure 31.Comparison of time histories of static pressure at base of the projectile between viscous and inviscid case

Figure 29 (a) and (b) shows the comparison between projectile velocity for Cases 5 and 6 (right side) respectively. From the graph, it is observed that the velocity of the projectile decreases for reduced temperature for both inviscid and viscous cases. In Fig. 29 (a), an instantaneous increase in velocity is observed for inviscid case when the projectile exits the barrel, whereas when viscous effects are considered, there is no increase in velocity at the exit because of viscous dissipation. For an inviscid case, the effects of viscosity are not considered which means the effect of friction and thermal conduction are not taken in to consideration. So when the projectile discharged from the launch tube, there is a sudden increase in velocity at the exit in which most researchers does not account for that velocity. For viscous flows, the transport phenomena of friction, thermal conductivity are considered. Viscous effects are dissipative and dispersive; not only the attenuation, but also the propagation patterns are altered due to viscosity. Although its

63

effects are strongest at the regions close to the solid boundaries, they are not confined to those regions. The viscous forces acting near body surfaces reduce the momentum in the boundary-layer regions so that, when acted upon by the initial forces, the boundary layer may separate from the body [14]. Frictional term reduces the further increase in velocity at the exit of the launch tube for viscous cases.

Figure 30 (a) and (b) show the static pressure at breech for Cases 5 and 6. The breech is initially under a high pressure of 2000 atm. As the projectile moves inside the launch tube, the pressure decreases rapidly as the gas expands. For low temperatures, the decrease in pressure follows the same trend for both viscous and inviscid cases. Whereas for higher temperatures, the decrease in pressure follows the same trend but deviates. This leaves us a conclusion that for higher temperatures, viscous effects should be considered.

Figure 31 (a) and (b) shows the static pressure at base of the projectile for Cases 5 and 6 respectively. For low temperatures, the viscous effects are negligible. Whereas for higher temperatures, the viscous effects are important to be considered. The decrease in pressure for viscous case is smoother and uniform but for an inviscid case, pressure decrease is not uniform and it deviates from the viscous case. The reason for the deviation for an inviscid case is that the thermal conduction is not taken in to account. Therefore it is very important to consider the viscous effects for high temperature flow fields.

It is conclusive that for all practical purposes, viscosity is relatively small, but its effects essentially control the basic properties. Therefore for a more realistic mathematical model of the muzzle blast, the inclusion of the viscous terms is perceived to be essential.
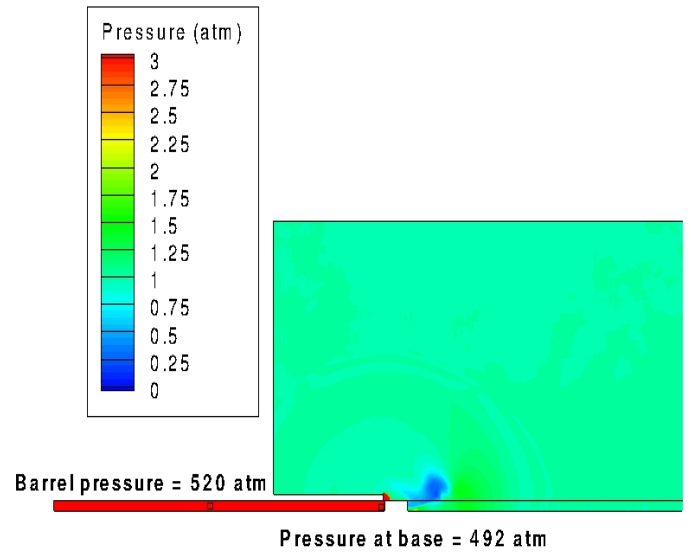
6.0 RESULTS

6.1 MUZZLE FLOW FORMATION AND DEVELOPMENT PROCESS

In the gun firing process, the projectile is pushed forward by the high pressure propellant gas which is behind the projectile. Air in front of the projectile is continually compressed by the moving projectile, resulting in the formation of a series of compression waves. The speed of each subsequent compression wave is faster than the preceding one because of the projectile acceleration. The volume of the precursor flow field increases continually due to the expansion of shock waves from the barrel. The formation of precursor muzzle flow and its development process are clearly shown in figure 32. Contour levels are scaled linearly between zero and the maximum value and the number of levels are 50. The actual pressure behind the projectile is 3450 atm and it decreases as the projectile moves downstream of the barrel. Though the color of the contour is red inside the barrel, the pressure actually decreases inside which can be clearly seen from the labeled barrel pressure values.

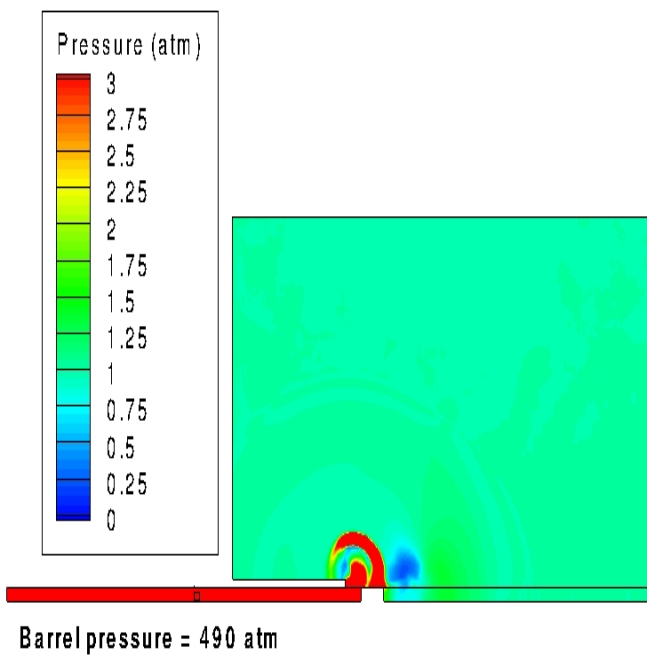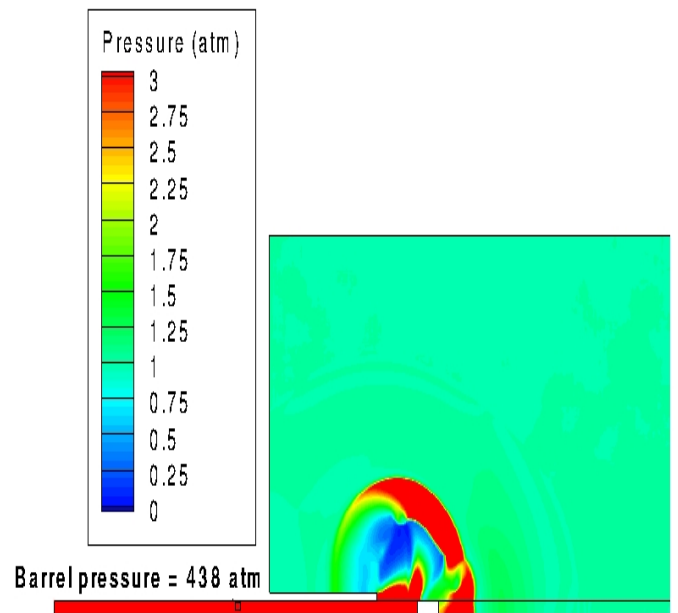(a) t= 0.3 ms   (b) t= 0.5 ms
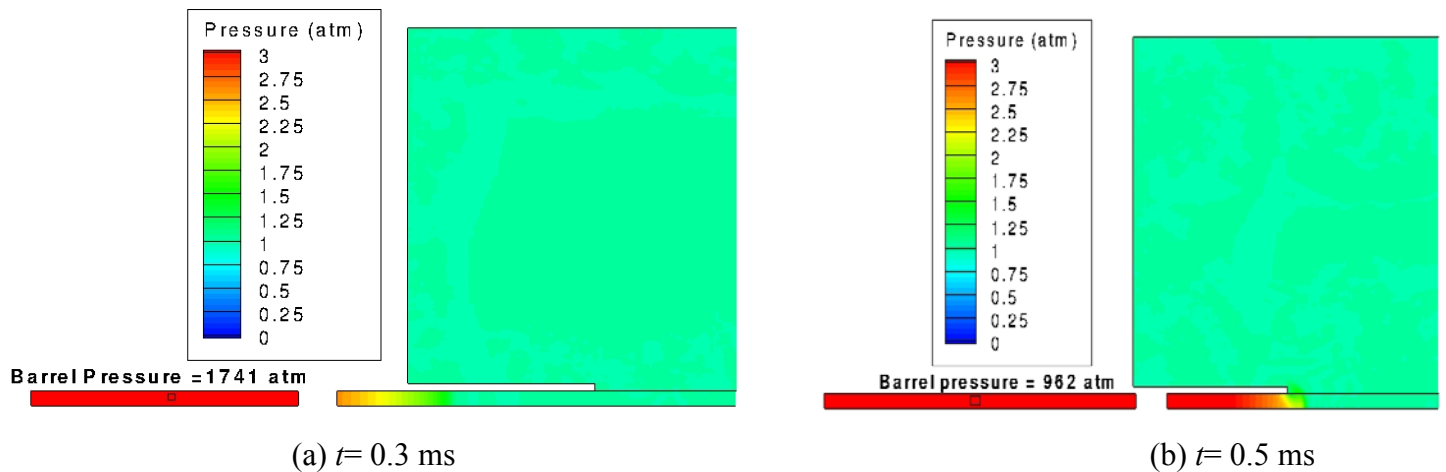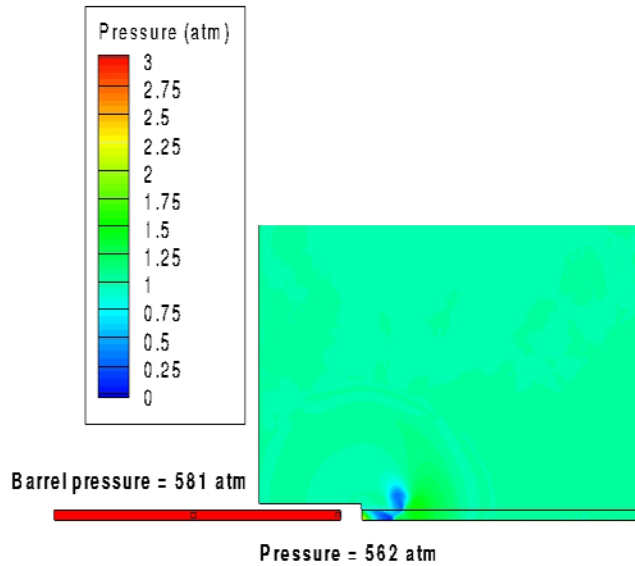
(c) *t*= 0.715 ms

(d) *t*= 0.77 ms

(e) *t*= 0.8 ms

(f) *t*= 0.86 ms

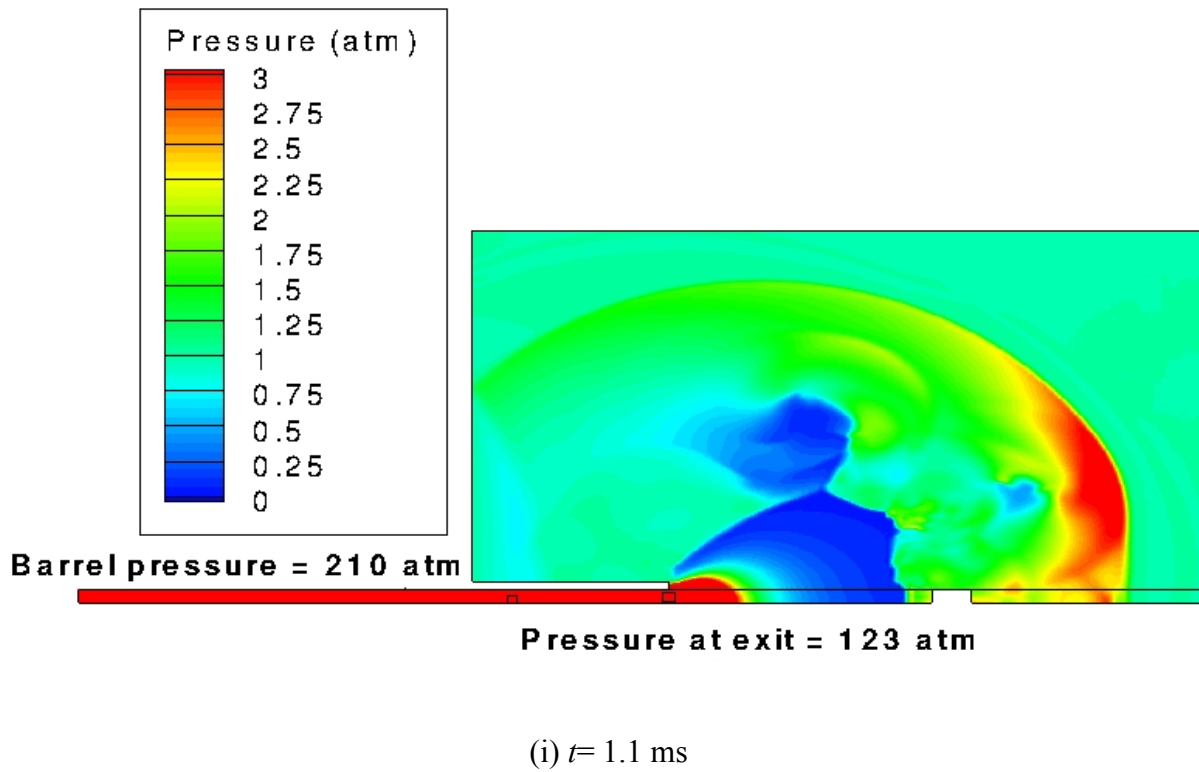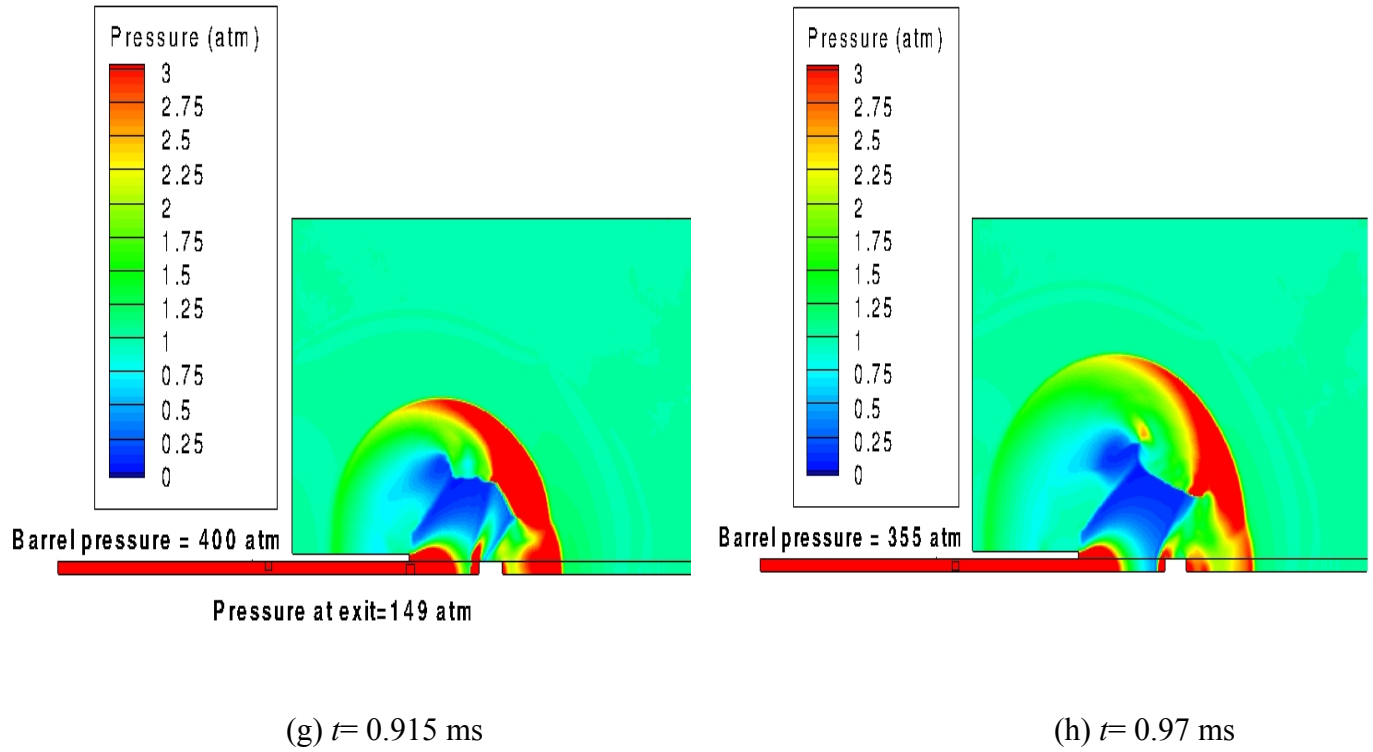(g) *t*= 0.915 ms

(h) *t*= 0.97 ms

(i) *t*= 1.1 ms

Figure 32.Time development of the muzzle flow (pressure contours)

67

Figures 32(a)-(d) show the development of the precursor flow field before the projectile completely shoots out of the launch tube. The precursor shock wave originated ahead of the projectile is being compressed by the moving projectile which is driven by the high pressure gas from behind and the wave becomes increasingly stronger as shown in Fig. 32(a). The precursor shock wave later propagates outwards and diffracts at the tube exit as shown in Fig. 32(b), prior to the projectile reaching the exit. As the high speed compressed gas jets out, the under-expanded jet pattern appears. In Fig. 32(c), the upward-facing shock wave with the shape of a spherical cap is shown, and its rim is sucked in to the core of the primary vortex ring to adjust the flow between its wave front zone (i.e., the supersonic expansion zone towards the tube exit) and the wave back (i.e., behind the precursor shock wave at the back of the tube exit). At the intersection of the projectile head and tube exit, a weak cylindrical shock wave is generated around the projectile due to rising pressure caused by the sudden emergence of the projectile and the effect of the primary vortex ring. In Fig. 32(d), the projectile is about to completely discharged from the launch tube at $t = 0.77$ ms.

In Fig. 32(e), the high pressure gas behind the projectile expands out of the launch tube and the second blast wave develops. The second blast consists of a leading shock wave, a contact surface and an expansion fan. The muzzle blast wave propagates outward mainly in the radial direction with an annulus-like shape, but also along the side wall of the projectile. The second blast wave moves faster to catch up with the projectile. There is nothing that is observable in front of the projectile, because it moves in the region where the particle velocity is approximately equal to the projectile velocity. The second blast overtakes the projectile at $t = 0.86$ ms that can be seen from Fig. 32(f). The projectile has penetrated through the upward-facing secondary shock wave as shown in Fig. 32(g), thereby it is moving in a region where the projectile speed is much higher

68

than the particle velocity in front of it. This results in development of a bow shock wave ahead of the projectile because its speed is supersonic with respect to the gas ahead of it. The bow shock wave occurs first, in the corner of the projectile as shown in Fig. 32(g) at t= 0.915 ms and then develops into shape in Fig. 32(h) at t= 0.97 ms. Actually, the bow shock wave is much stronger than the second blast because the leading shock wave damps rapidly with the distance increases from the muzzle, but the bow shock wave is driven by the projectile at almost constant speed. It is important to point out that the bow shock wave is generated in the background flow created by the first blast and also disturbed by the second blast. Fig. 32(i) shows the bow shock wave catches up with the leading shock wave of the second blast. It is conclusive that the high pressure gas behind the projectile produces a stronger second blast that leads to more intensive wave interaction, but the primary wave processes are not affected significantly.

6.2 PRESSURE DISTRIBUTION ALONG DIFFERENT POSITION

In order to study the effects of unsteady projectile aerodynamics on the near-field of the launch tube, the pressure histories were plotted for different locations.
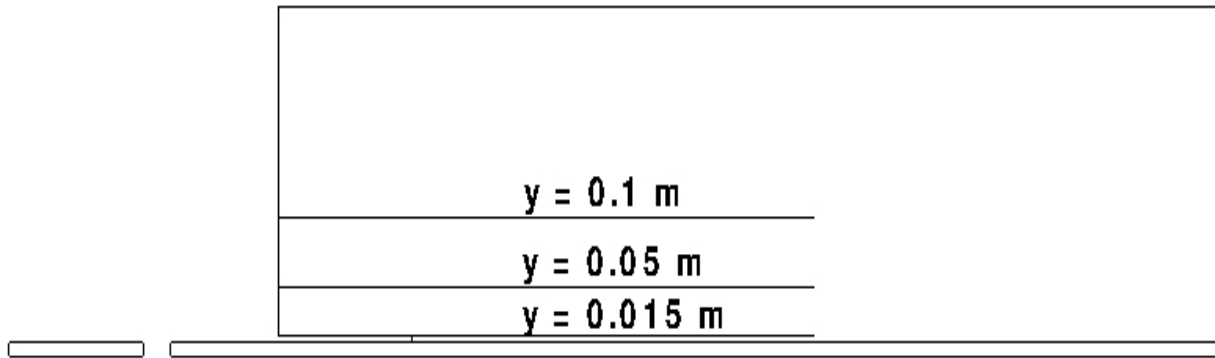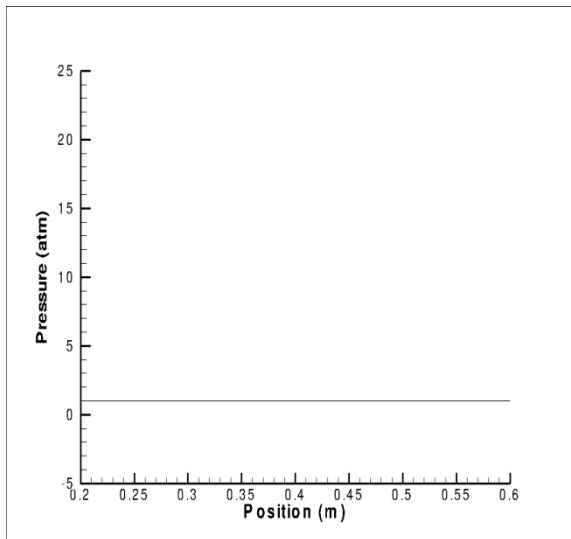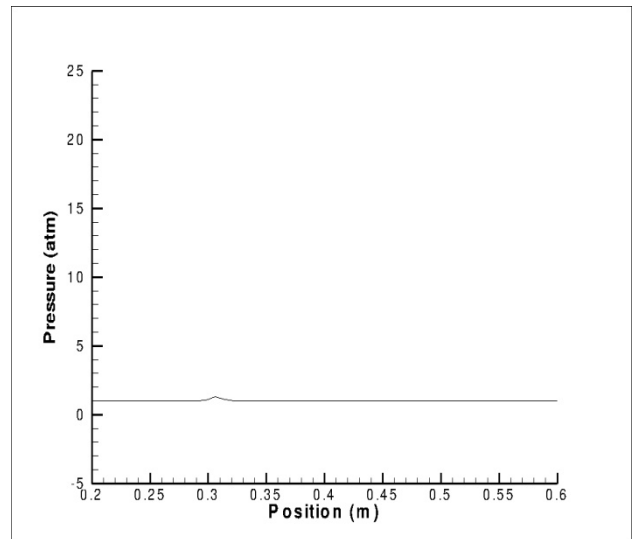
Figure 33.Locations for pressure distribution along three different lines
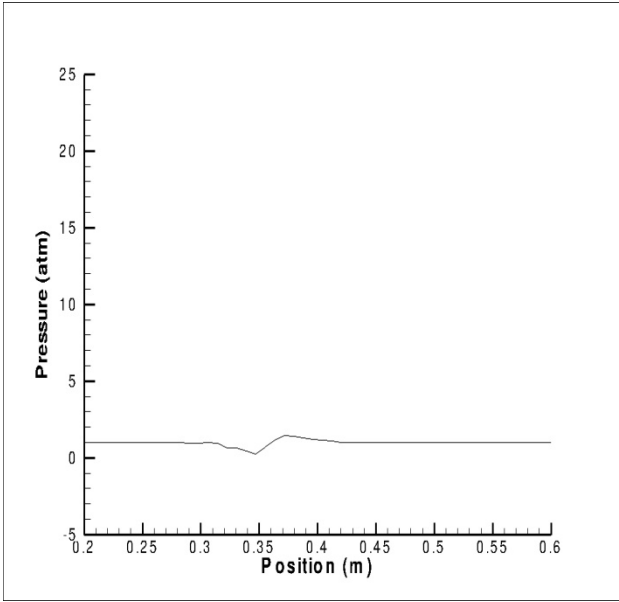
Fig.33 shows the location along which the pressures were plotted to study the near-field aerodynamics of the launch tube. Fig. 34 shows the pressure distribution for different time values along a line which is located 15 mm away from the gun axis (y= 0.015 m)
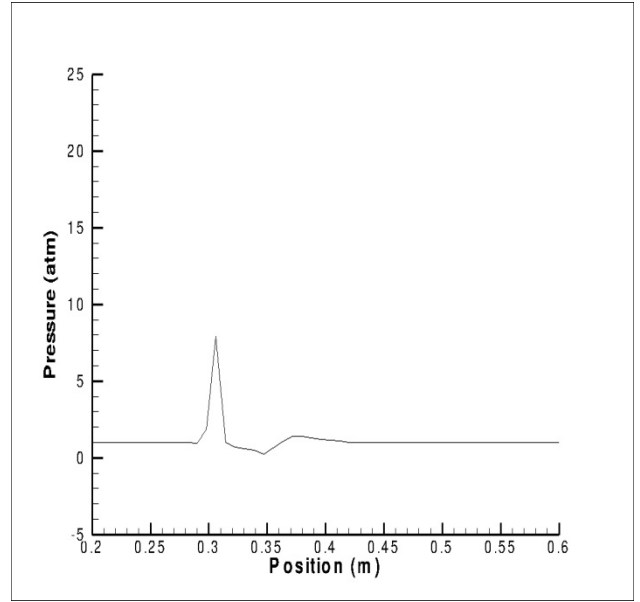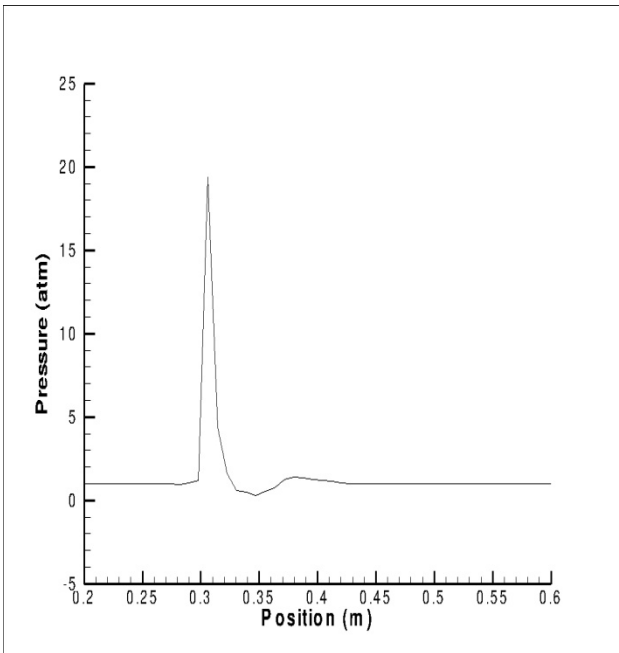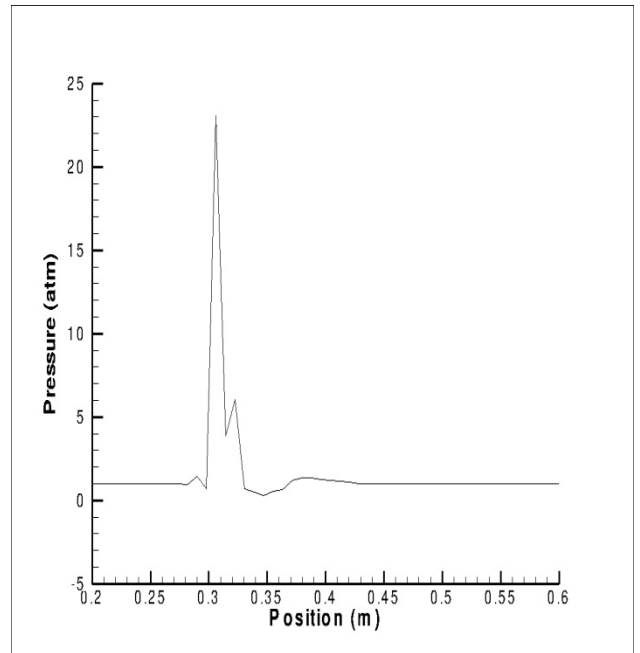

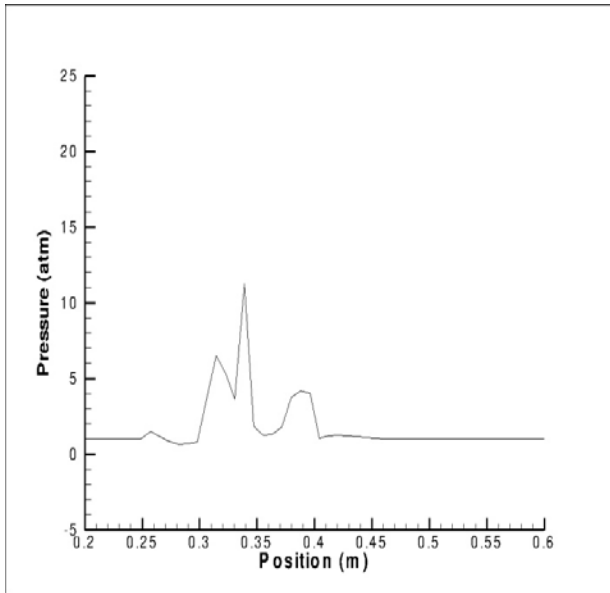(a) t= 0 ms

(b) t= 0.51 ms

(c) *t* = 0.77 ms

(d) *t* = 0.775 ms

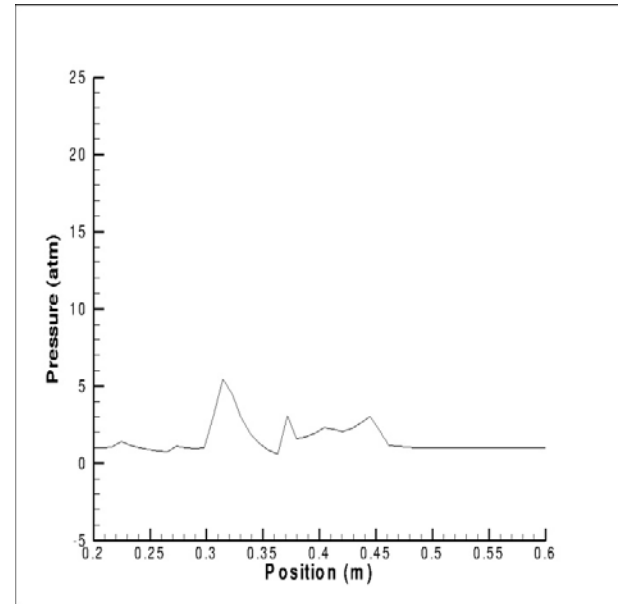(e) *t* = 0.785 ms

(f) *t* = 0.79 ms

(g) *t*= 0.87 ms　　　　　　　　　　　　(h) *t*=0.95 ms

Figure 34.Pressure distribution for various times at y= 0.015 m

The pressure behind the projectile is very high while the still air outside the launch tube is maintained at atmospheric conditions (P= 1 atm) when the projectile moves inside the launch tube and the pressure outside is still ambient which can be seen from Fig. 34(a). At *t*= 0.51 ms, the pressure starts to increase because of the first blast wave and the projectile is still inside the launch tube. The pressure rise due to the first blast wave develops which can be seen from Fig. 34 (c). When the projectile is completely out of the launch tube, the high pressure gas behind the projectile expands out of the launch tube and the second blast wave develops. There is a steep increase in pressure because of the second blast waves which can be seen in Fig. 34(d) at *t*= 0.775 ms. Further increase in pressure can be seen from Fig. 34(e)-(f) to a maximum of 23 atm. The effect of the second blast wave is more intense than the first blast wave because the high pressure gas behind the projectile expands out and develops an under expanded jet flow. The

72

amplitude of the pressure waves increases within a short interval of time. In Fig. 34(g), pressure increases due to the first blast wave, and then reduces sharply due to the presence of the low pressure background or expanded post-shock flow. The pressure then steeply increases again due to the shock wave of the secondary blast developing behind the projectile and then it drops down to the atmospheric pressure.

Figure 35 shows the locations of points in which pressure is plotted against flow time along different lines.
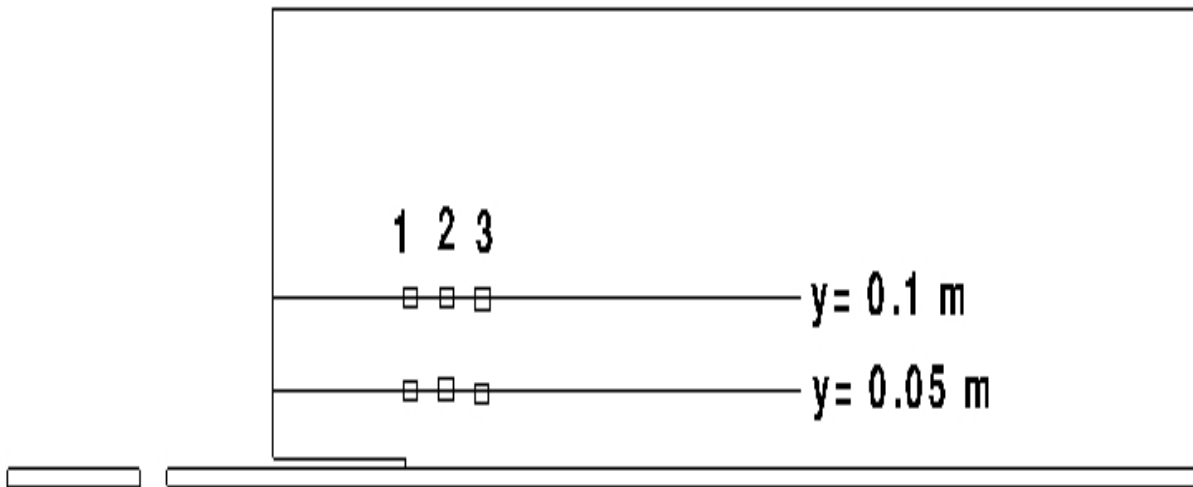
Figure 35.Locations for pressure distribution along the same line for different positions

Point 1 is located at 0.30 m (at the exit of barrel) and Point 2 is located at 0.31 m (10 mm from the exit of barrel) and Point 3 is located at 0.32 m (20mm from the exit of barrel)

Figure 36 shows the pressure distribution along the line y= 0.05 m for different positions. Figure 37 shows the pressure distribution along the line y= 0.1 m for different positions. The intensity

of pressure is more near the gun axis. From Figure 36 and 37 it is clear that away from the gun axis, the pressure decreases.
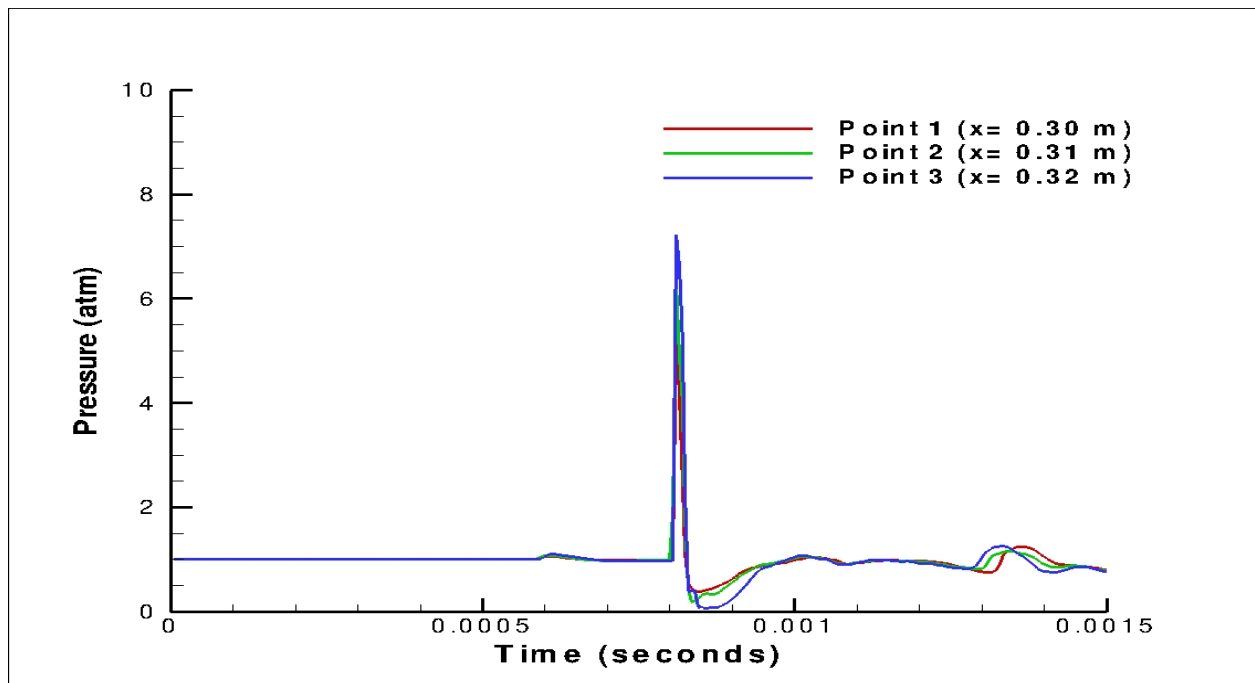
Figure 36.Pressure distribution along the line y= 0.05 m for different positions
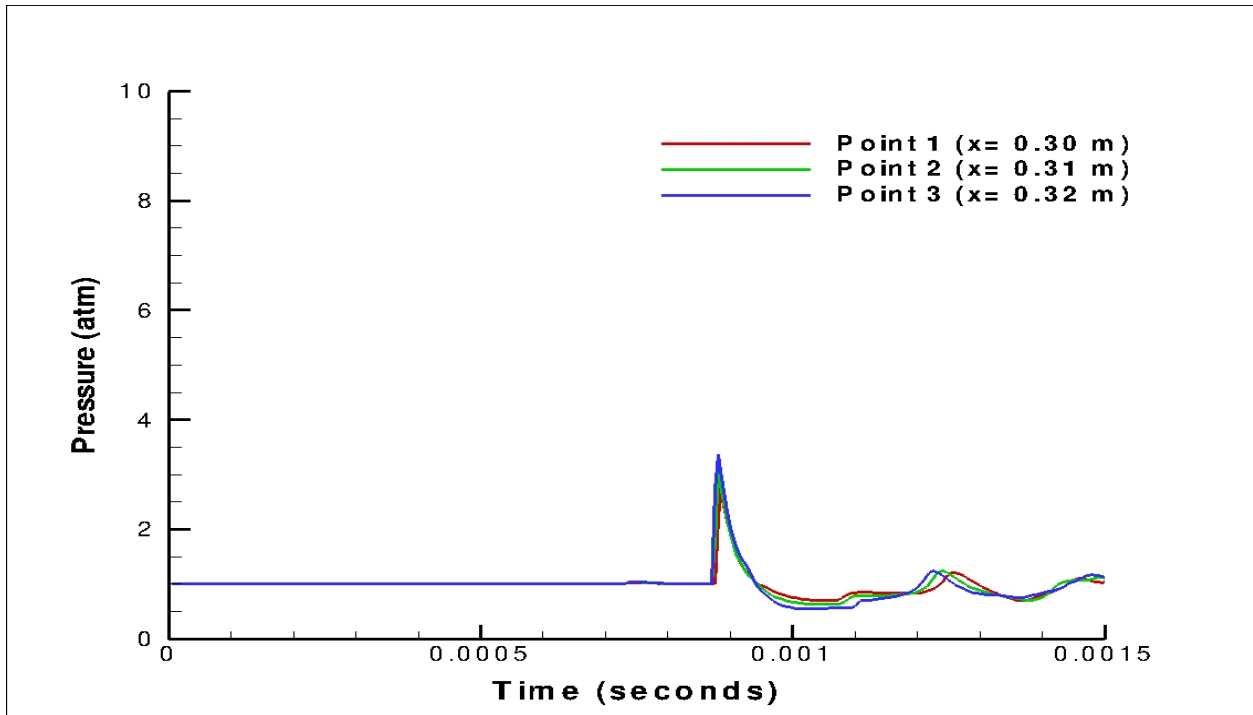
Figure 37.Pressure distribution along the line y= 0.1 m for different positions

6.3 DISTRIBUTION OF SOUND PRESSURE LEVEL ALONG DIFFERENT POSITION

The acoustic environment produced by gunfire is of the impulse type with the muzzle being the primary source. The two major effects of the induced gunfire noise environment that must be considered are human tolerance and aircraft structural fatigue. Sound Pressure Level (SPL) is a measure of mean square level of pressure fluctuation on a logarithmic scale. Equation (1) is used to calculate the SPL. The lowest pressure that can be heard by a person with normal hearing is 20 μPa. The loudest sound pressure that most humans can tolerate is about 200 Pascals, which is 10 million times greater than the lowest sound pressure that can be heard [18]. The general range of human hearing is from 0-120 dB. A typical muzzle blast generates an impulse sound wave with

75

the Sound Pressure Level that ranges from 120-160 dB. At 194 dB, actual sound waves begin to distort and now are called as shock waves [19]. Sound Pressure Levels were plotted along different positions ($y = 0.015$ m, $y = 0.05$ m, $y = 0.1$ m) for Case 7. Fig. 38(a)-(h) shows the SPL plots at $y= 0.015$ m. The SPL plots for $y= 0.05$ m & $y= 0.1$ m can be found in Appendix.
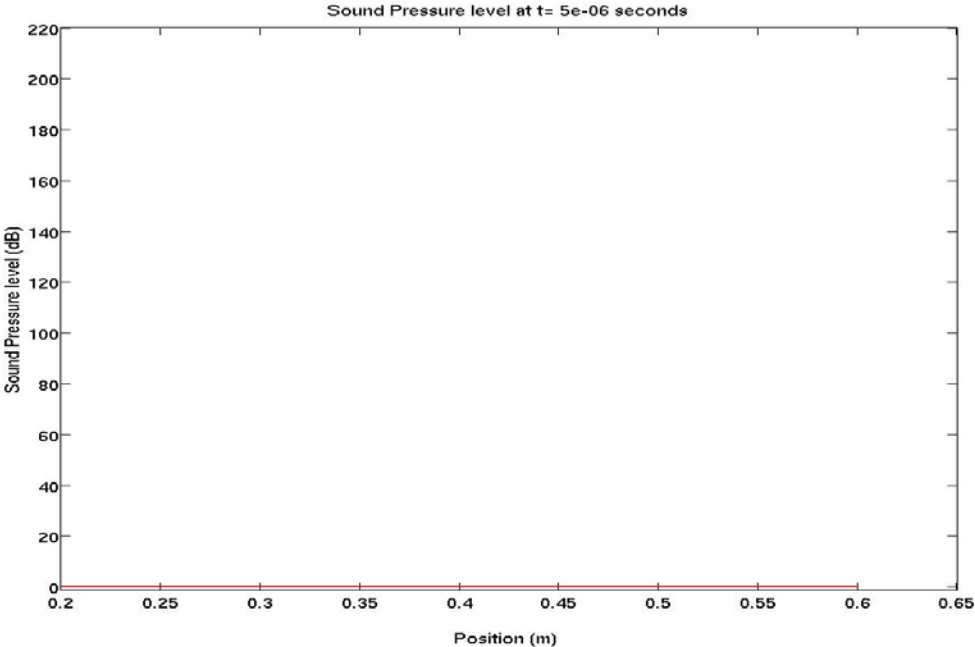
(a) $t= 0$ ms
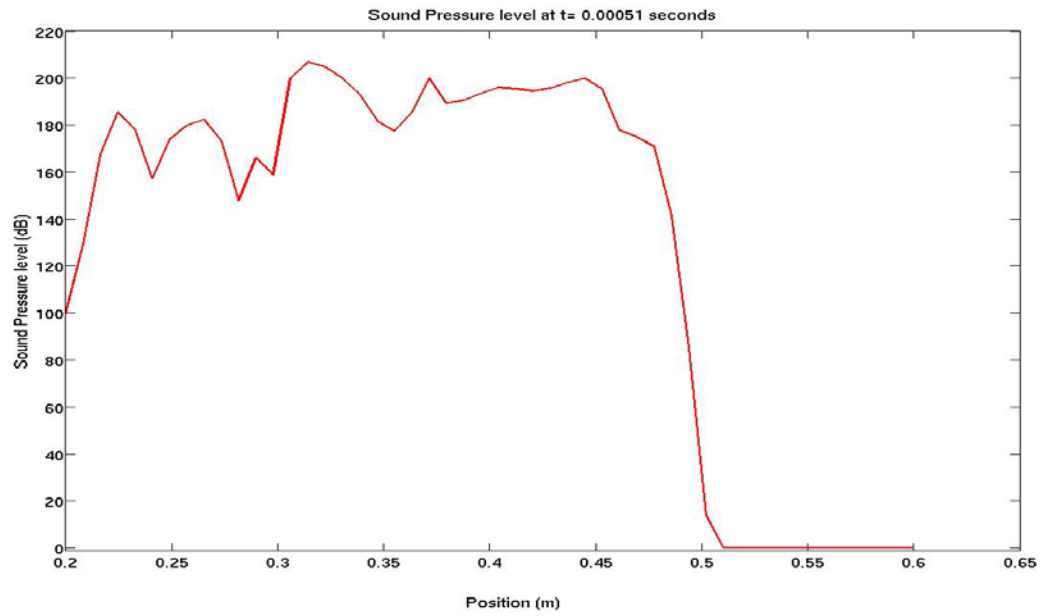
(b) *t*= 0.51 ms
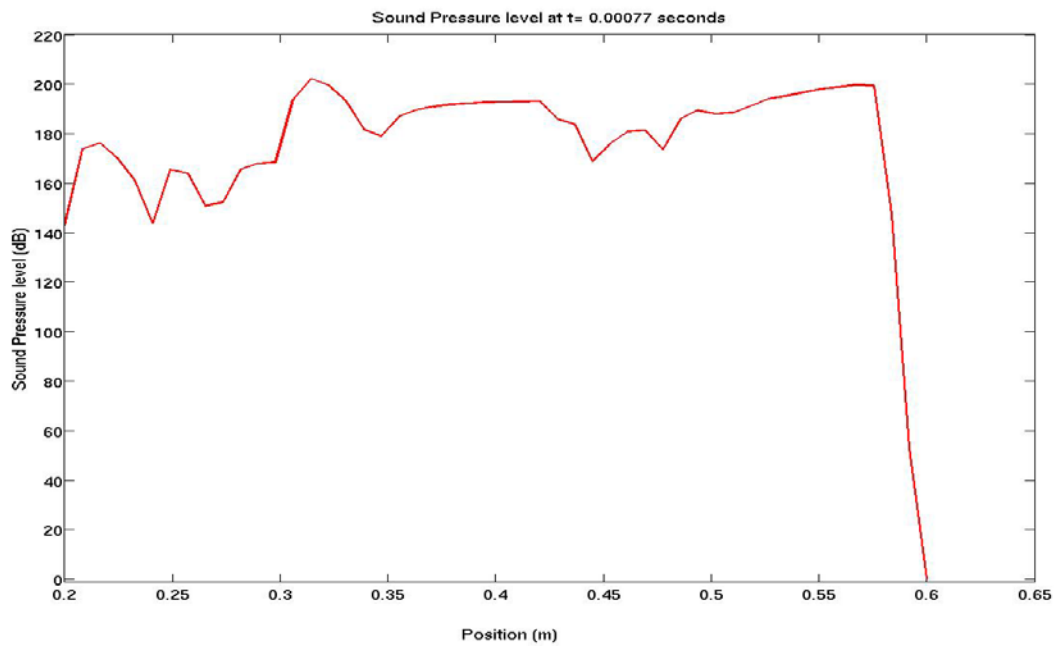
(c) *t*= 0.77 ms
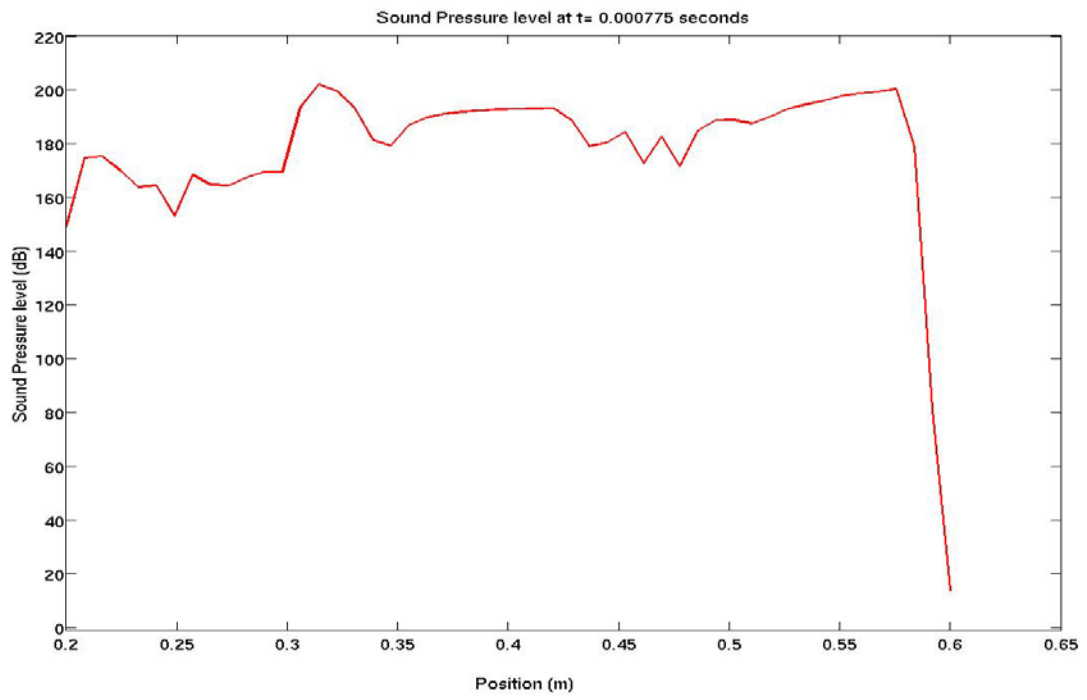
(d) *t*= 0.775 ms
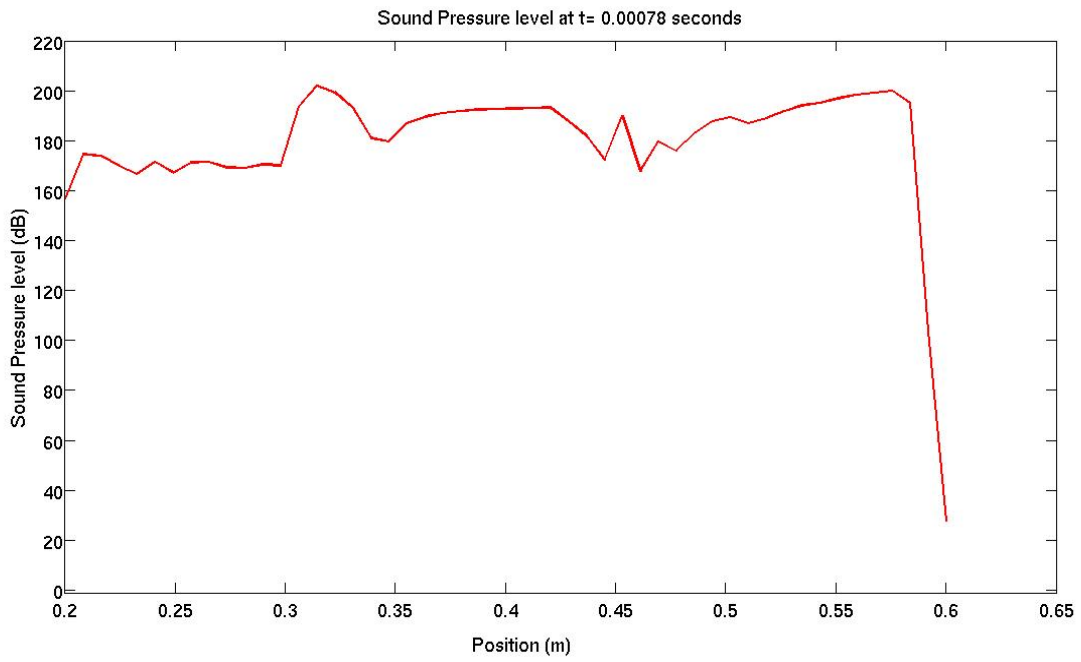
(e) *t*= 0.78 ms
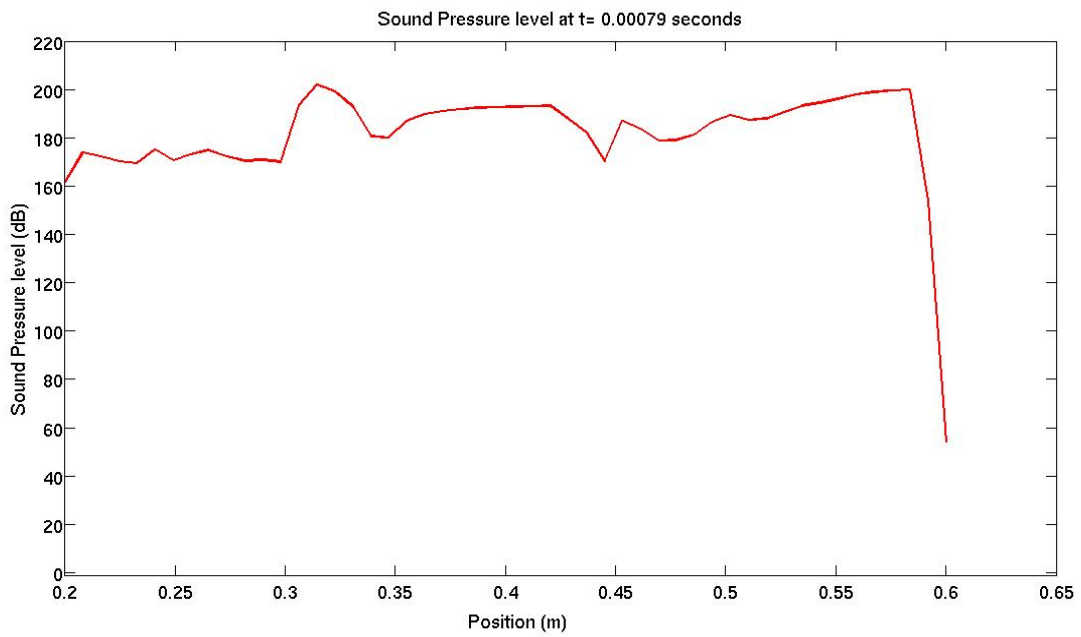
(f) *t*= 0.79 ms
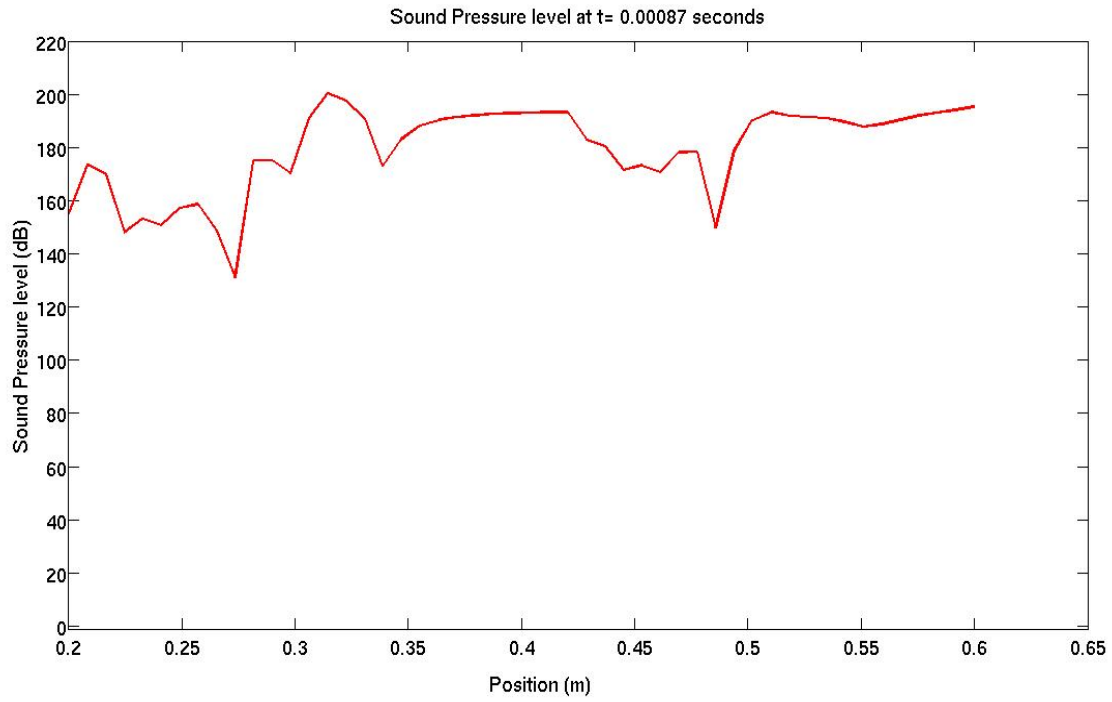
(g) *t*= 0.87 ms
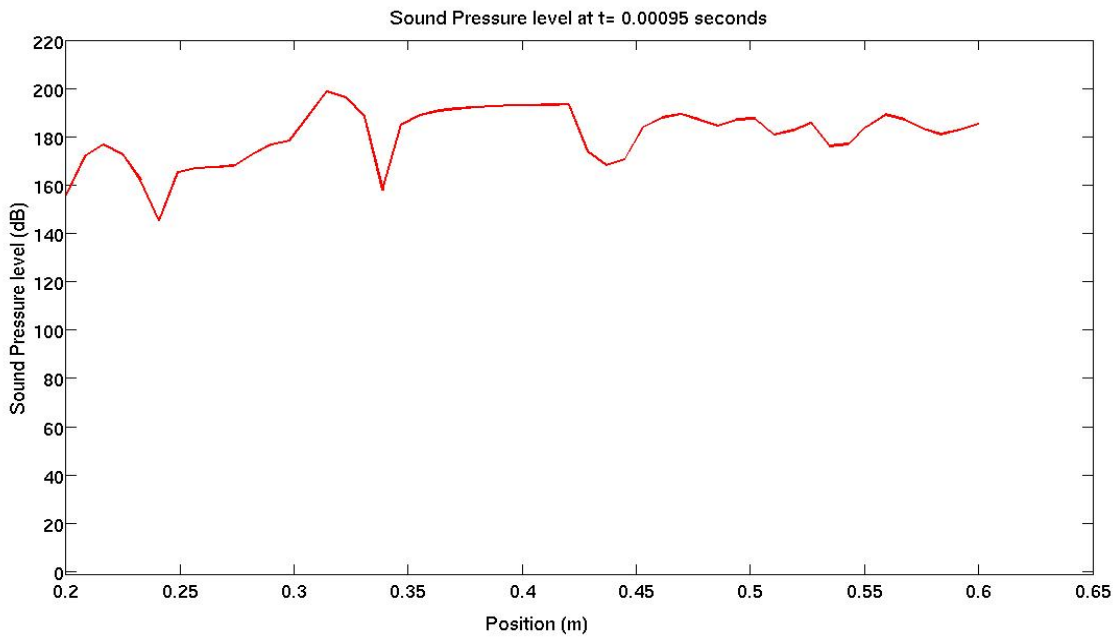
(h) *t*= 0.95 ms

Figure 38.Distribution of sound pressure level for various times at y= 0.015 m

SPL was calculated using equation (1). The perturbation pressure values were extracted from the regions of x= 0.2 m to x= 0.6 m through time. These pressure values were used to plot the SPL using MATLAB. In Fig. 38(b) at t= 0.51 ms, a huge drop in dB can be seen at 0.475 m. This is because the projectile has travelled only 0.475 m at t= 0.51 ms, but data are being extracted till 0.6m. There are no available data after 0.475 m and hence it drops to zero dB. Fig. 38(c) – (f) the drop in dB can be seen because the projectile has not travelled the required distance. Fig. 38(g) at t= 0.87 ms, the projectile has crossed 0.6 m and the SPL plot clearly shows the dB variation. A steep increase in dB can be seen at x= 0.3 m because the projectile leaves the launch tube at that position and the secondary blast wave develops because of the expansion of high pressure gas behind the projectile. Since the Sound Pressure level is greater than 194 dB, the sound wave distorts to form a shock wave. From the results, it is concluded that further away from the gun axis the amplitude and maximum intensity of SPL decreases.

Fig.39 shows the location (x=0.3m) for measuring Sound Pressure Level (dB)
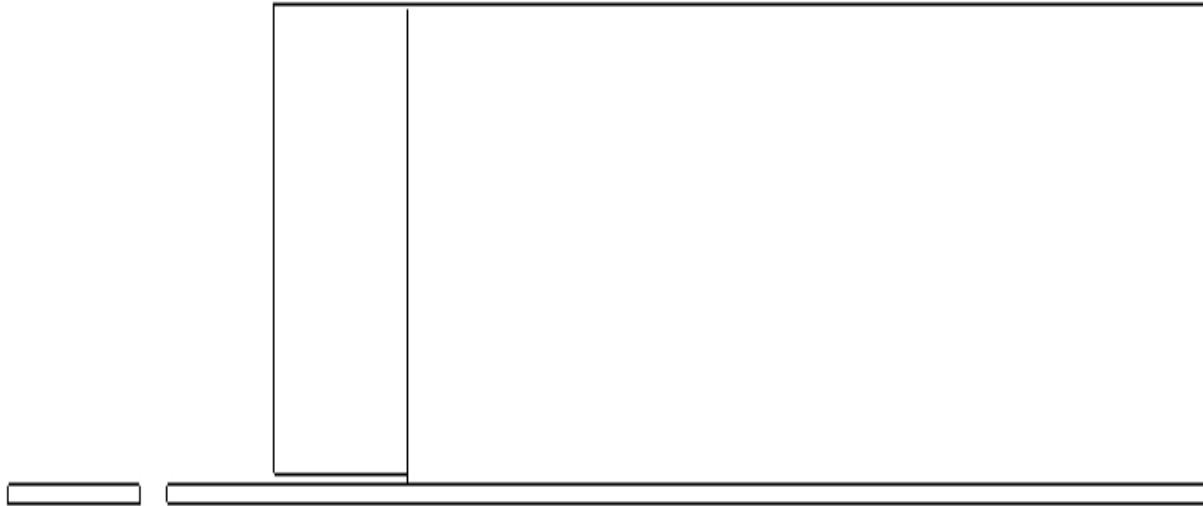
Figure 39.Location for Sound Pressure Level (dB) along the line x=0.3 m

The Sound Pressure Level in dB were plotted along the line x=0.3m and the plots for different instant of time is shown below.

Fig.40 (a) shows the SPL plot for $t = 0.51$ ms. As expected, the intensity is more close to the barrel. Since the intensity of SPL level is more than 194 dB close to the barrel, the acoustic waves are called as shock waves. Further away from the barrel towards the fuselage of the aircraft, the intensity of SPL values decreases and goes to 0 after x=0.2 m. This means that there is no pressure fluctuations beyond x=0.2 m.

Sound Pressure level at t= 0.00051 seconds

(a) *t*= 0.51 ms

Sound Pressure level at t= 0.000775 seconds

(b) *t*= 0.775 ms
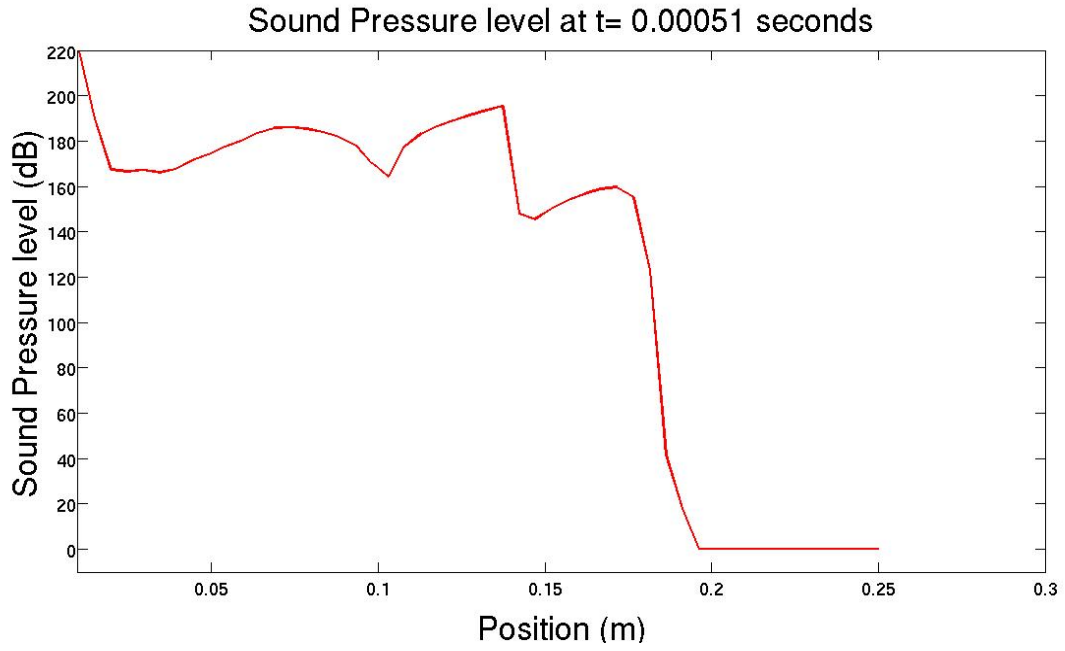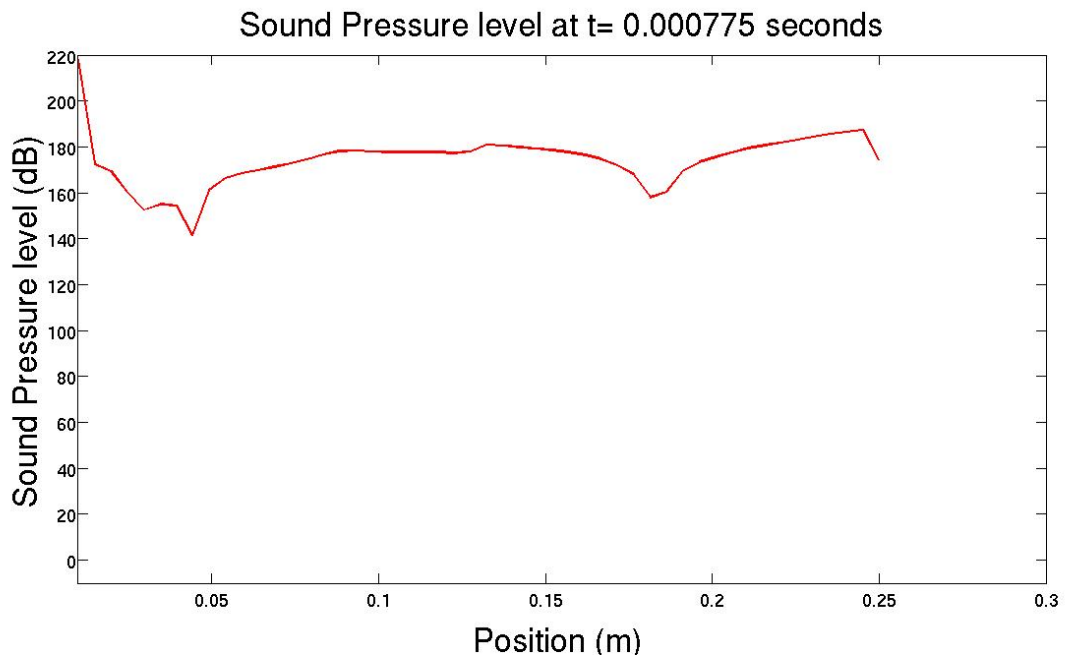
Sound Pressure level at t= 0.00087 seconds
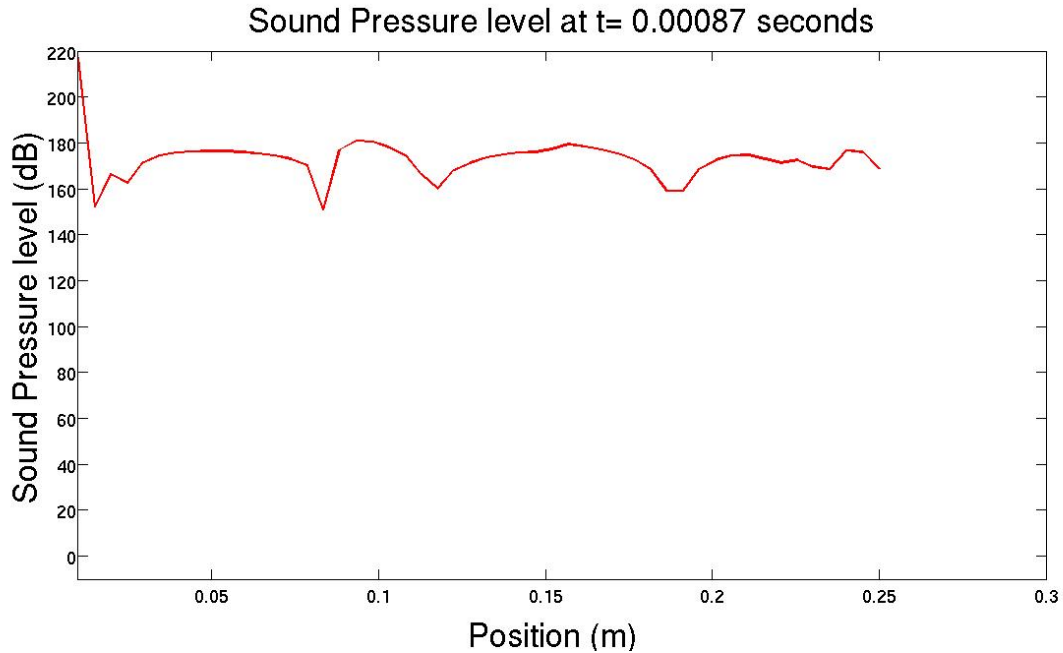
(c) *t*= 0.87 ms

Figure 40.Distribution of Sound Pressure Level (dB) for various times at *x*=0.3 m

Fig.40 (b) and (c) show the SPL plots for *t*= 0.775 ms and *t*= 0.87 ms respectively. As the projectile further moves down the barrel, the region beyond *x*=0.2 m are also affected by pressure disturbance. Because of the fluctuation in pressure, the Sound Pressure Level also varies and it does not drop down to zero whereas in Fig. 40 (a) it drops to zero.

7.0 CONCLUSIONS

CFD software ANSYS FLUENT, has been applied to simulate the pressure blast of the projectile which is launched from a barrel. The unsteady, compressible, axi-symmetric Euler equations (Cases 1, 2, 3 &4) and Navier-Stokes equations (Cases 5, 6 &7) were numerically solved using the Advection Upstream Splitting Method (AUSM) scheme; with third-order Monotone Upstream centered Scheme for Conservation Laws (MUSCL) approach. The projectile mass, initial conditions behind the projectile and viscous effects were changed to investigate their effects on the flow structures which affect the unsteady aerodynamics of the projectile in the near-field. The present computations were validated with the available CFD results [1]. It is known that the computed results reasonably capture the major flow features such as shock waves, blast waves, shear layer, etc., which are generated in launching a projectile up to supersonic speed.

Cases 1 & 2 were used to study the mass effect and it is observed that the projectile mass has a strong influence on the acceleration of the projectile inside the launch tube and deceleration of the projectile outside the launch tube. The lighter mass (Case 1) has more acceleration when compared to heavier mass (Case 2). Cases 2, 3 & 4 were used to study the effect of initial conditions behind the projectile. Reduced initial temperature and pressure conditions behind the projectile significantly affect the flow field as these conditions reduce the strength of the blast wave and the rate at which it is expanding. As expected, the projectile attains the maximum velocity in the case of highest initial pressure and temperature (Case 2) behind it. The projectile velocity reduces significantly when the initial pressure is reduced (Case 4) compared to the other two cases, due to a much weaker gas expansion behind the projectile. Cases 5 & 6 were used to study the viscous effects of unsteady projectile aerodynamics. Based on comparison with

inviscid cases, it is concluded that for all practical purposes, viscosity is relatively small, but its effects essentially control the basic properties. Therefore, for a more realistic mathematical model of the muzzle blast, the inclusion of the viscous terms is perceived to be essential especially for high temperature flow fields.

Case 7 which is the thesis objective of the author is a realistic example of an analysis of unsteady projectile aerodynamics of a 20 mm gun that fires a 1500 grain (97.197 gram) projectile at a breech pressure and temperature of 3450 atm and 2500 K respectively for a 200 mm barrel. The computed results reasonably captures the major flow field features as expected. The near-field aerodynamics of the projectile was studied by plotting the pressure distribution along different position ($y=0.015$m, $y=0.05$ m, $y=0.1$ m) away from the gun axis. Pressure plots were shown at these positions for different time values. Sound Pressure Level (SPL) was also calculated from the perturbation pressure and it is plotted along different positions. Snap shots of SPL plots were shown for different time intervals.

REFERENCES

1. Rajesh, G., Kim, H.D., Matsuo, S., and Setoguchi, T., "A Study of the Unsteady Projectile Aerodynamics using a moving coordinate method," Proceedings of the Institution of Mechanical Engineers, Journal of Aerospace Engineering, Vol.221, No.5, 2007, pp. 691-706. DOI: 10.1243/09544100JAERO227.

2. Cler, D.L., Chevaugeon, N., Shephard, M.S., and Remacle, J.F., 2003, "CFD Application to Gun Muzzle Blast a Validation Case Study," 41st AIAA Aerospace Science Meeting and Exhibit, Reno, NV.

3. Jiang, Z., "Wave dynamic processes induced by a supersonic projectile discharging from a shock tube," Physics Fluids, 2003, pp. 1665-1675.

4. Glass, I.I., "Shock waves and man," 1974 (The University of Toronto Press, Canada).

5. Schmidt, E., and Shear, D., " Optical Measurements of Muzzle blasts," AIAA Journal., 1975, 13, pp. 1086-1096.

6. Erdos, J.I., and Del Guidice, P.D., "Calculation of Muzzle blast flow fields," AIAA Journal., 1975, 13, pp.1048-1057.

7. Merlen, A., and Dyment, A., "Similarity and Asymptotic analysis for gun firing aerodynamics," Journal of Fluid Mechanics., 1991, 225, pp. 497-502.

8. Jiang, Z., Takayama, K., and Skews, B.W., "Numerical study on blast flow fields induced by supersonic projectiles discharged from shock tubes," Phys. Fluids, 1998, pp. 277-288.

9. Yu, W., Zhang, X., "Aerodynamic Analysis of Projectile in Gun System Firing Process," Journal of Applied Mechanics, 2010, Vol. 77/ 051406-1. DOI: 10.1115/1.4001559.

10. Ian A. Johnston., "The Noble-Abel Equation of State: Thermodynamic Derivations for Ballistics Modeling," Weapons Systems Division, Defense Science and Technology Organisation, DSTO-TN-0670.

11. Krier, H., Summerfield, M., "Interior Ballistics of Guns," Vol. 66 of Progress in Astronautics and Aeronautics, AIAA, New York, United States, pp. 325-348.

12. Van Wylen, G.J., and Sonntag, R.E., "Fundamentals of Classical Thermodynamics," 3rd edn, 1985, John Wiley & Sons, Chapter 10.

13. Pointwise, Inc. "Pointwise User Manual," 2009.

14. Baysal, O., "Navier-Stokes analysis of Muzzle blast type waves," AIAA Journal, Vol. 24, No.5, 1985.

15. Maurice, J.Z., Joe, D.H., "Gas Dynamics," Volume I, John Wiley & Sons, Chapter 1.

16. Ansys Fluent, Inc. 'Fluent 6.3 User's Guide".

17. Anderson, J.D., "Computational Fluid Dynamics – The Basics with Applications," International Editions 1995, McGraw-Hill, Inc.

18. Eric, L.C., "The Decibel (dB) & Sound Measurement," www.elcaudio.com/decibel.htm.

19. "Noise Level Chart," www.noisehelp.com/noise-level-chart.html.

20. "Guide to Aircraft/Stores Compatibility," MIL-HDBK-244A, April 1990.

APPENDIX I
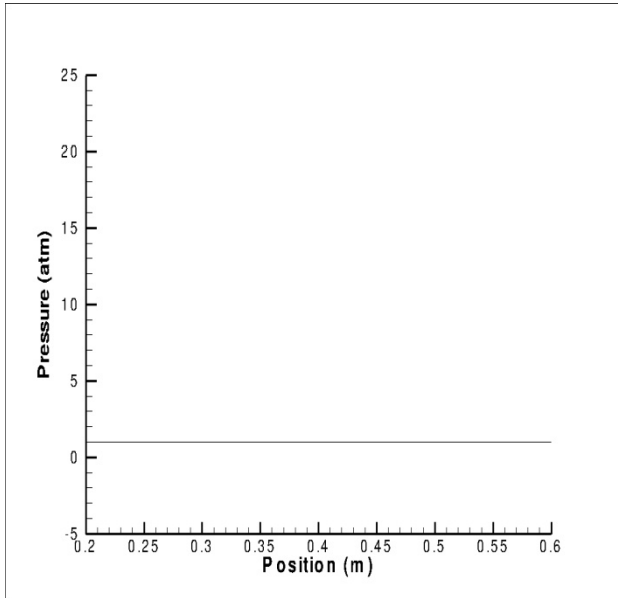
(a) *t* = 0 ms
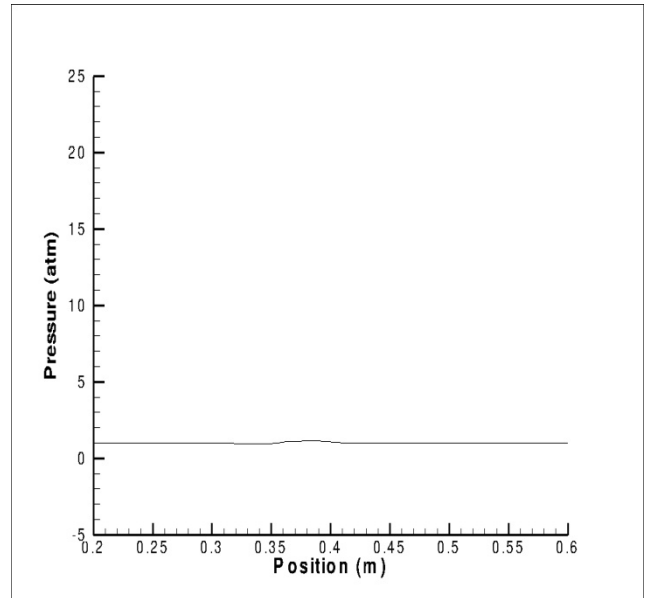
(b) *t* = 0.77 ms
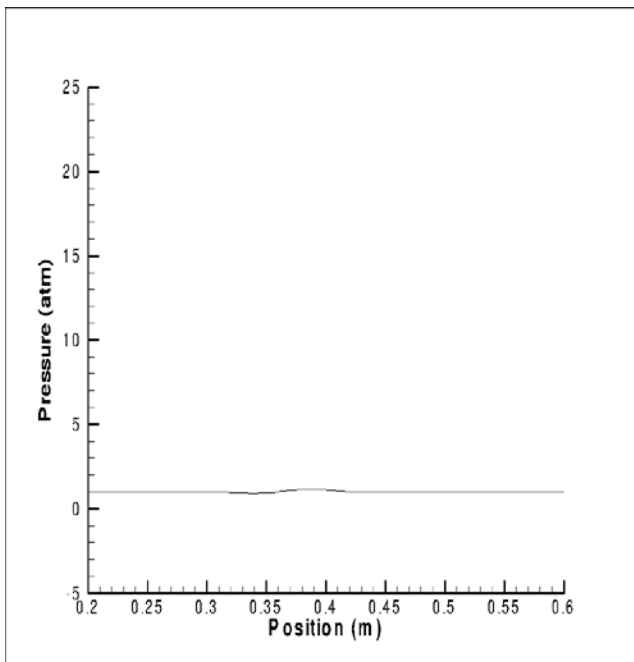
(c) *t* = 0.79 ms
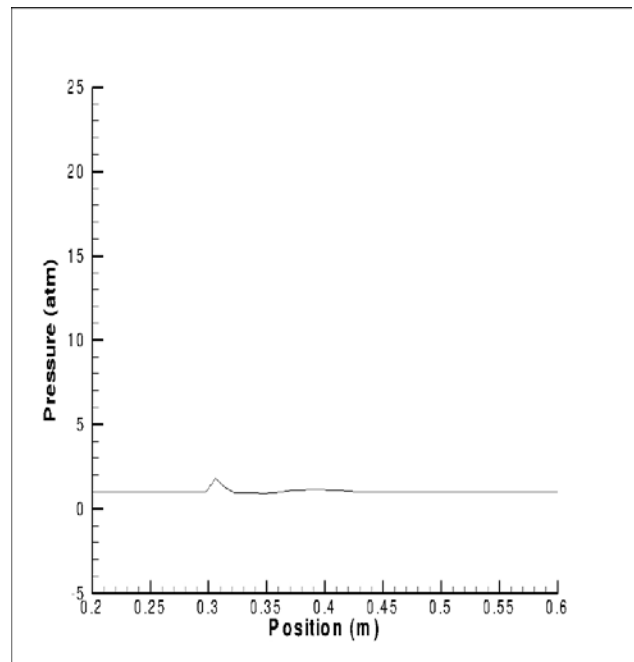
(d) *t* = 0.8 ms
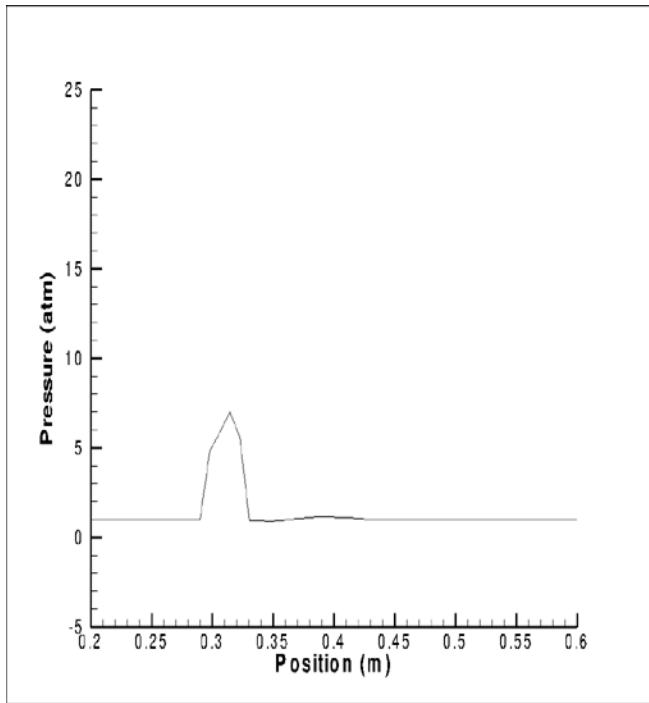
(e) *t*= 0.81 ms
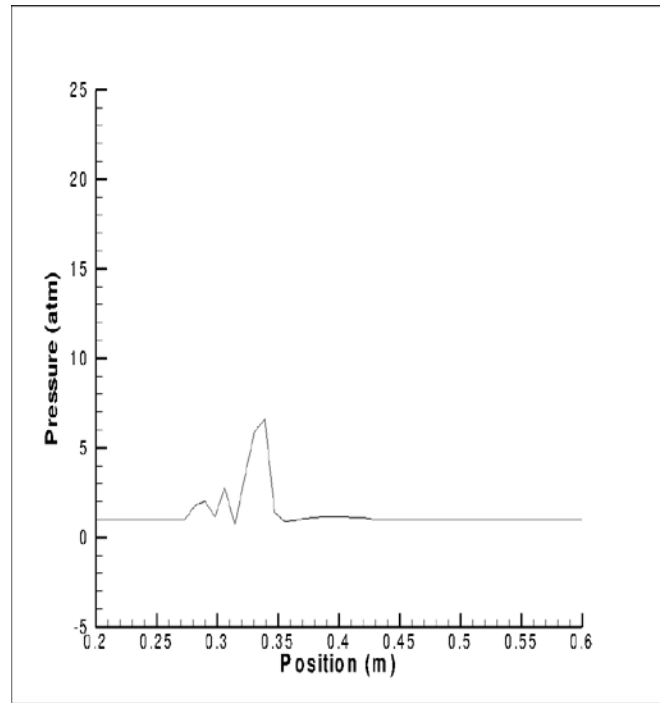
(f) *t*= 0.825 ms
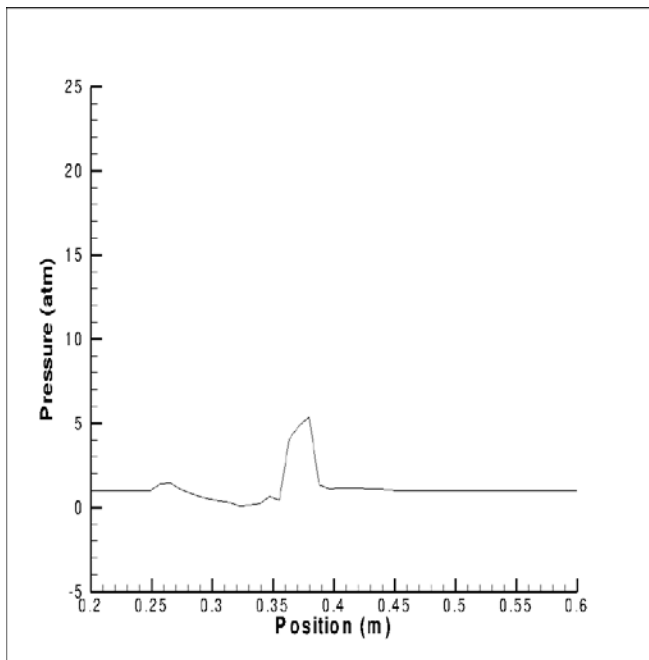
(g) *t*= 0.87 ms
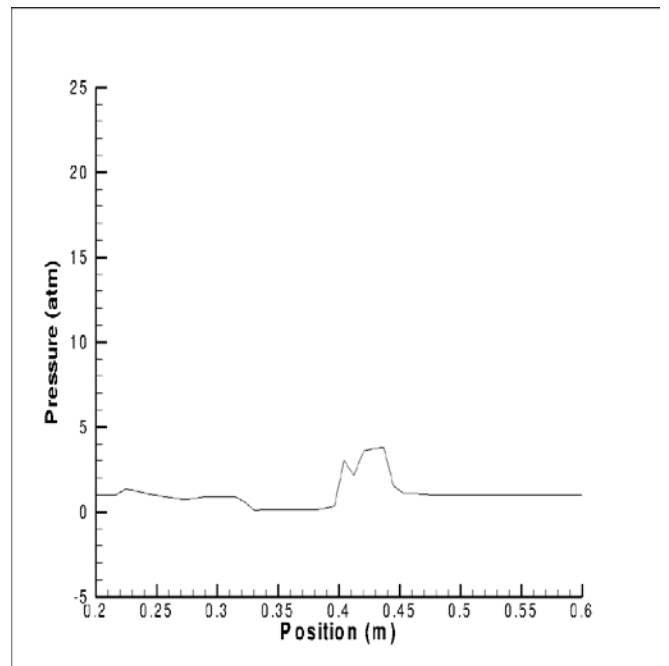
(h) *t*= 0.95 ms

Figure 41.Pressure distribution for various times at *y*= 0.05 m
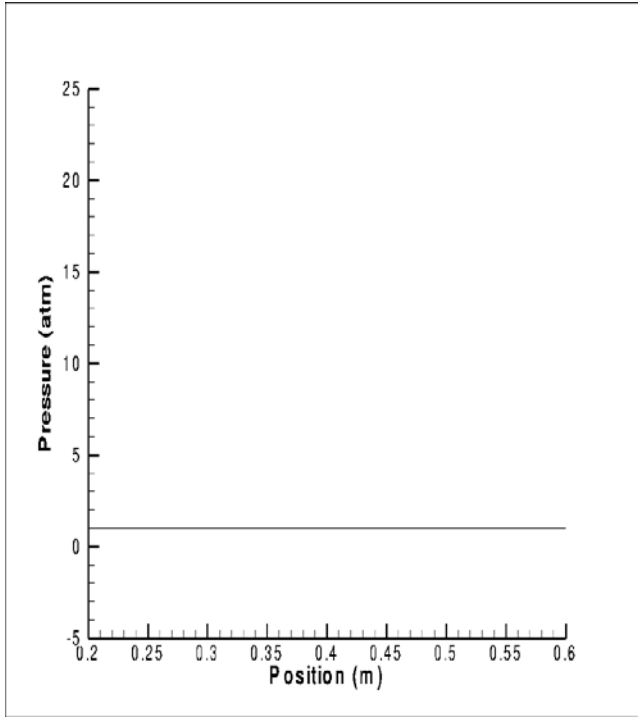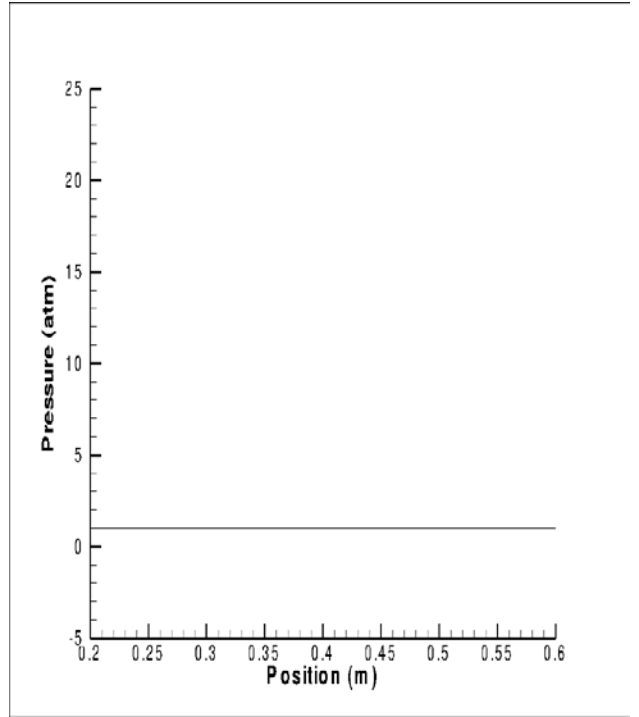
(a) *t*= 0 ms

(b) *t*= 0.51 ms
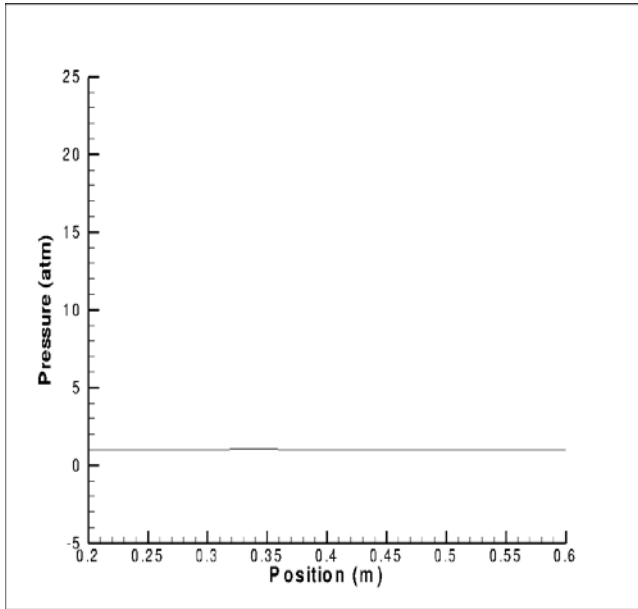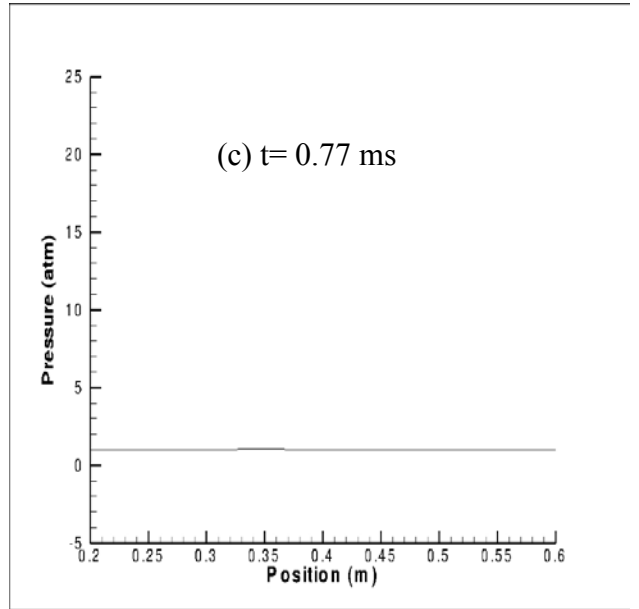
(c) t= 0.77 ms

(c) *t*= 0.77 ms

(d) *t*= 0.785 ms
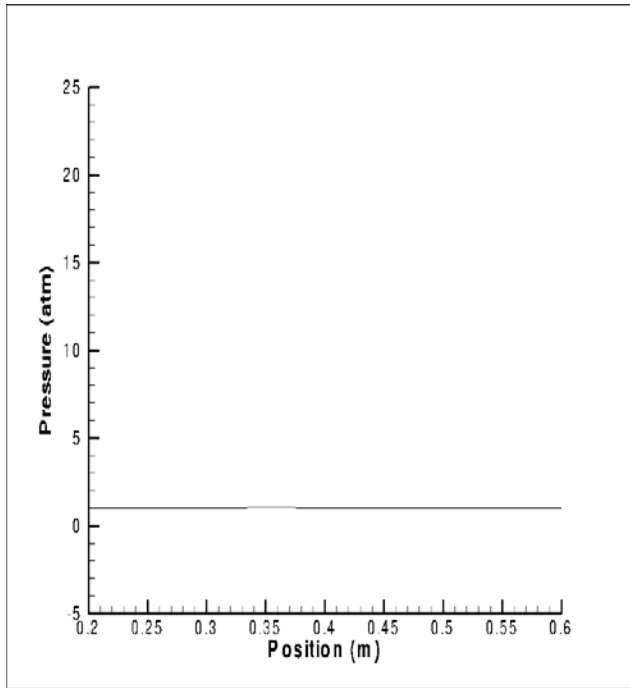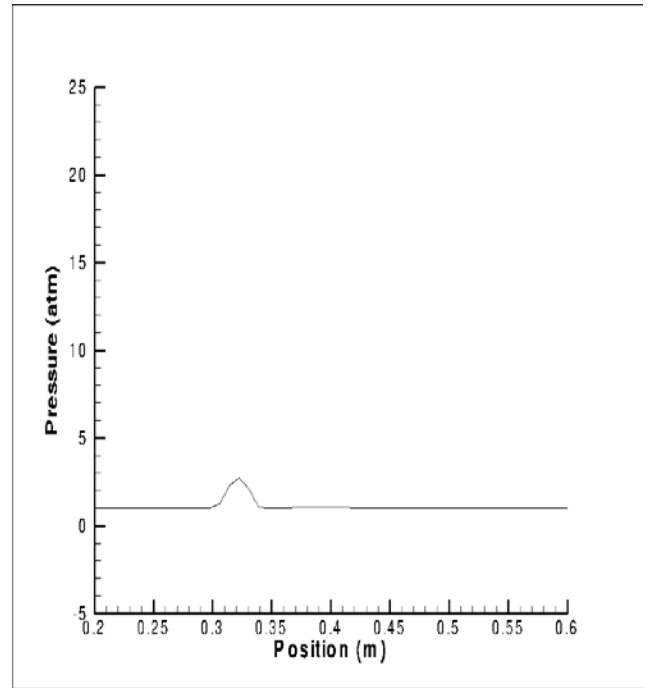
(e) *t*= 0.79 ms

(f) *t*= 0.875 ms
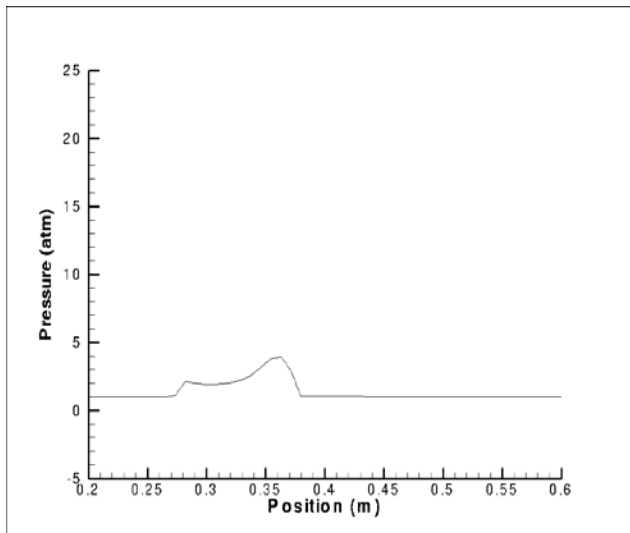
(g) *t*= 0.9 ms
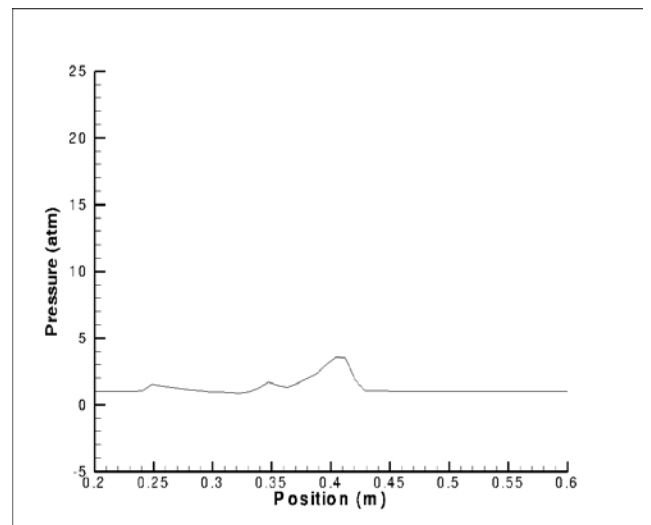
(h) *t*= 0.95 ms

Figure 42.Pressure distribution for various times at *y*= 0.1 m

(a) $t = 0$ ms

(b) $t = 0.77$ ms

(c) $t = 0.79$ ms

(d) $t = 0.8$ ms

93
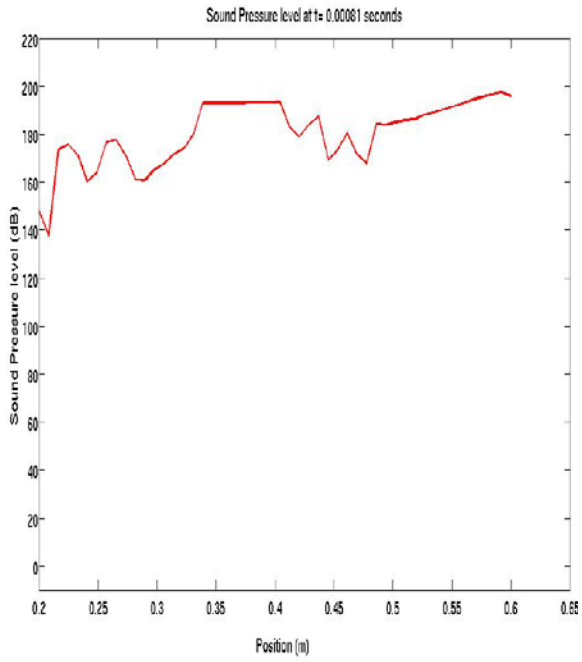
(e) *t*= 0.81 ms
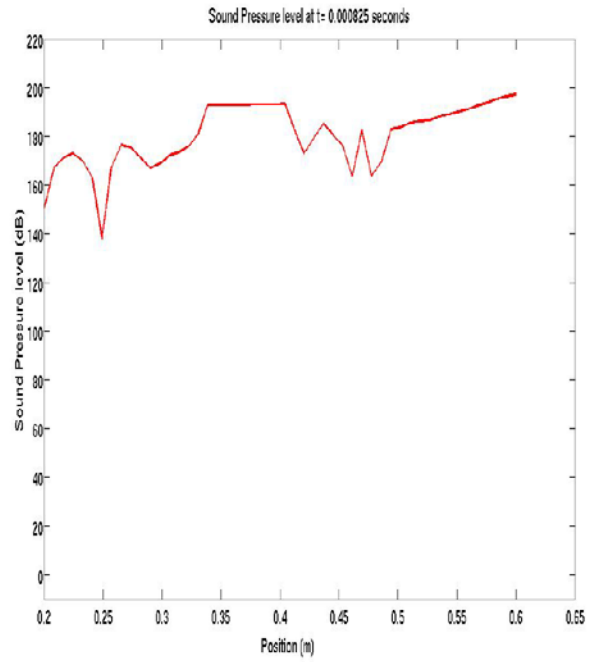
(f) *t*= 0.825 ms
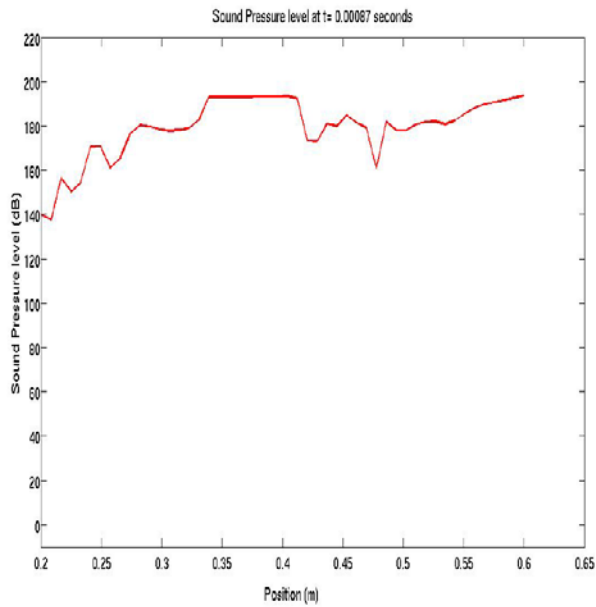
(g) *t*= 0.87 ms
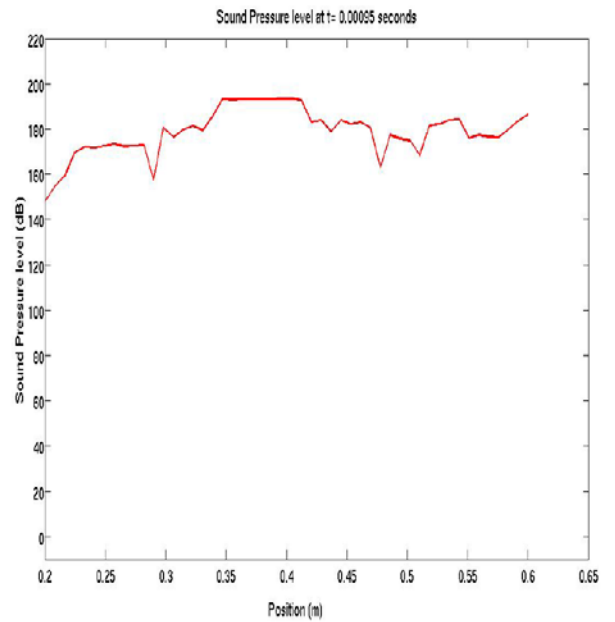
(h) *t*= 0.95 ms

Figure 43.Distribution of sound pressure level for various times at *y*= 0.05 m
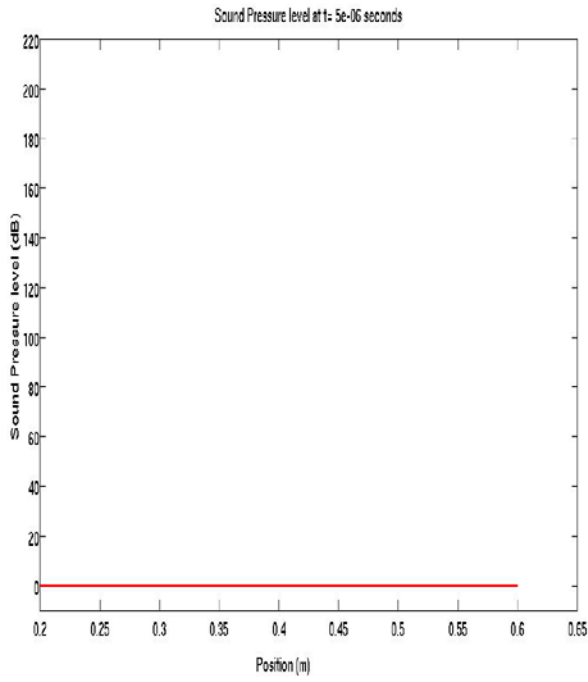
(a) *t*= 0 ms
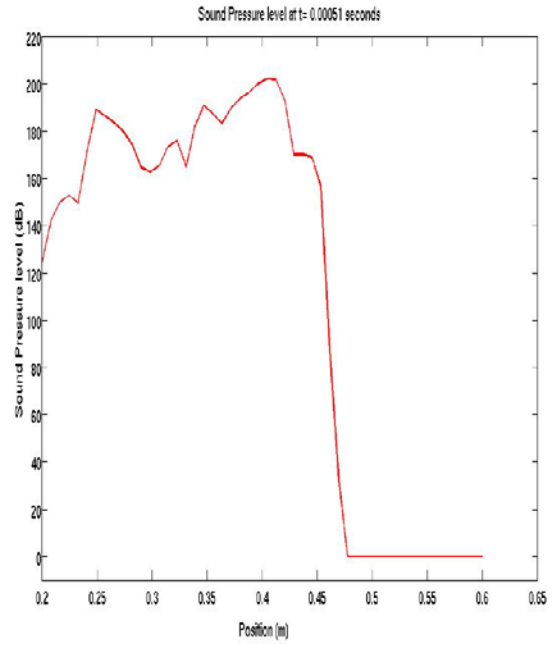
(b) *t*= 0.51 ms
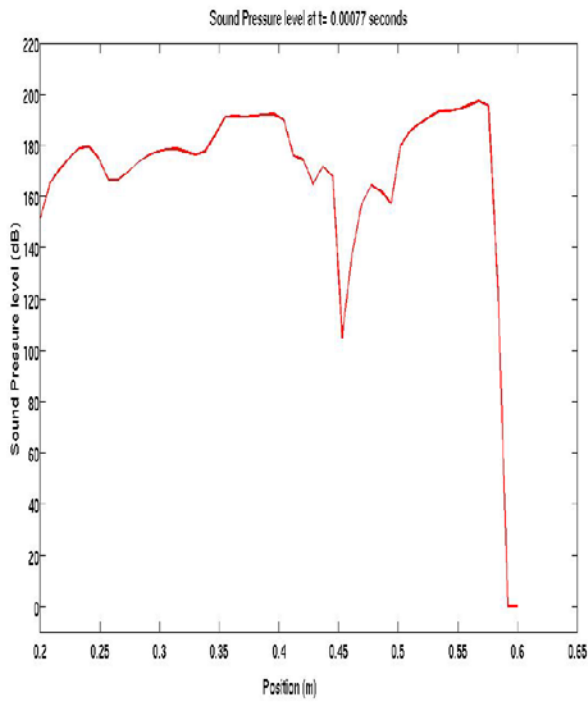
(c) *t*= 0.77 ms
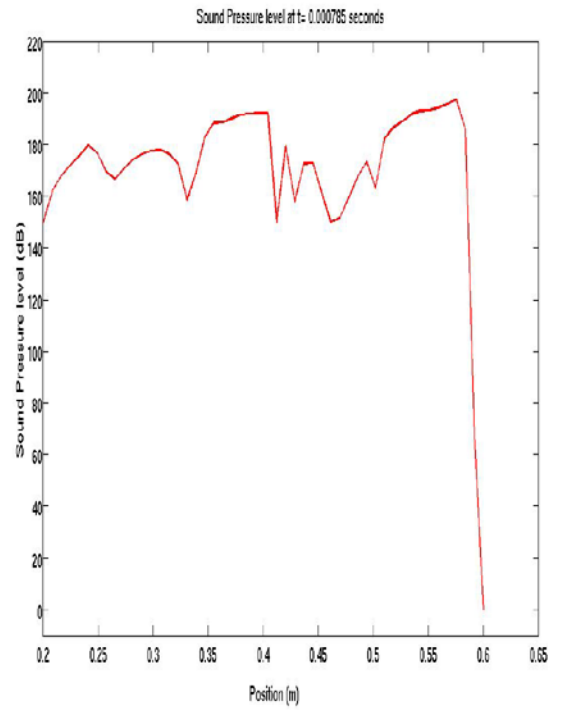
(d) *t*= 0.785 ms
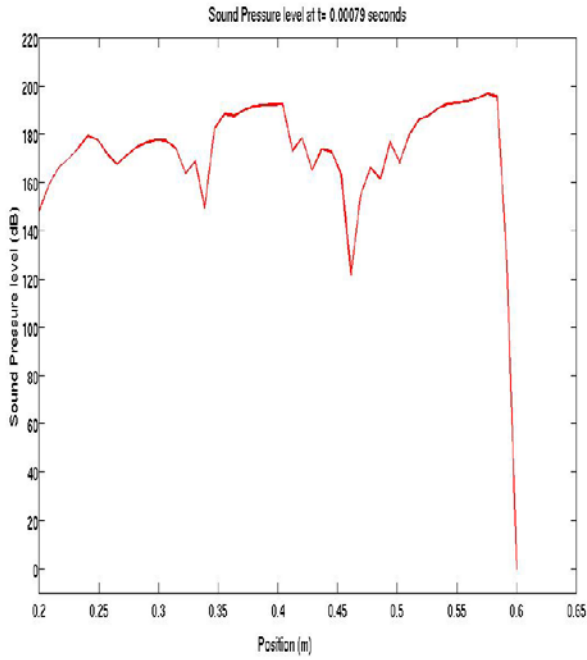
(e) *t*= 0.79 ms   (f) *t*= 0.875 ms
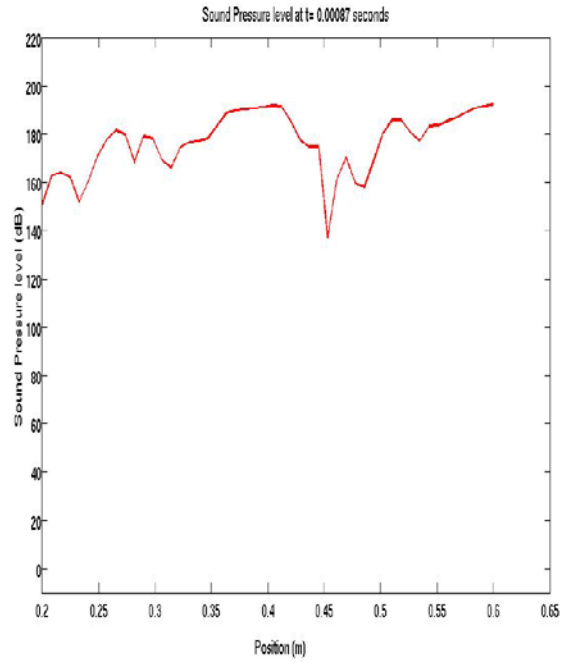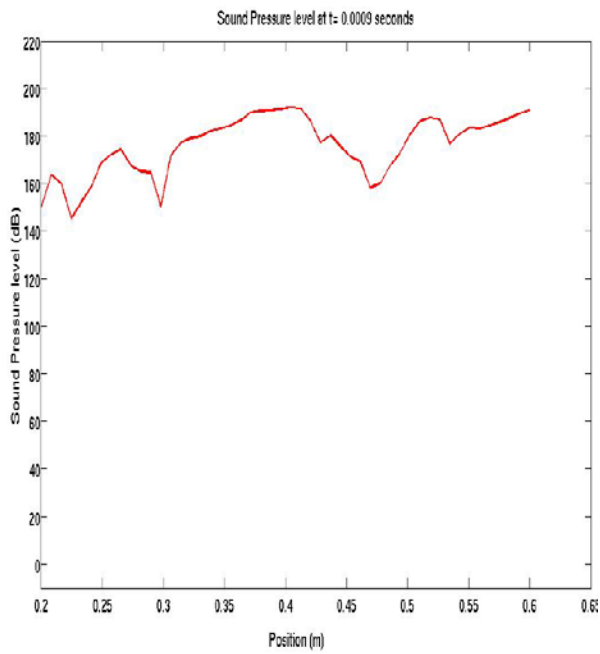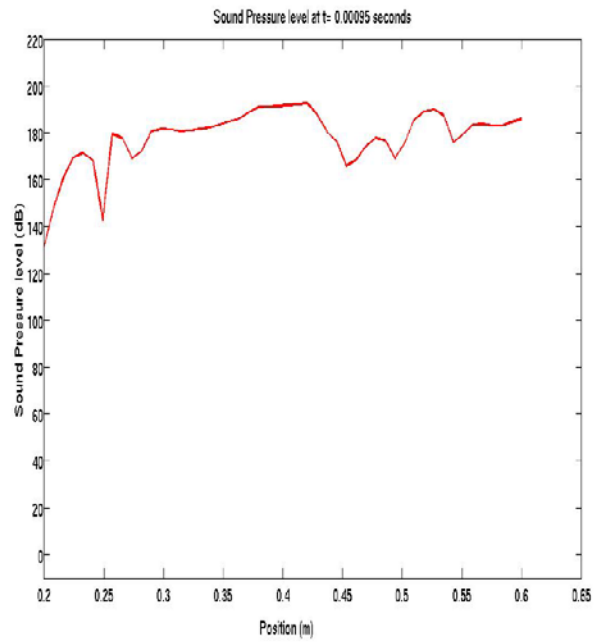
(g) *t*= 0.9 ms   (h) *t*= 0.95 ms

Figure 44.Distribution of sound pressure level for various times at *y*= 0.1 m

96

APPENDIX II

1. USER DEFINED FUNCTION (UDF) FOR NOBLE-ABEL EQUATION OF STATE

```
/*****************************************************************/


/* UDF purpose:                              */
/*                              */
/*   To allow Fluent to use the real gas model in the form of the  */
/*     Abel_Nobel Equation Of State (ANEOS):                */
/*                              */
/*          press * (v - b) = R * T                */
/*                              */
/* where b is the covolume term.                */
/*   The thermodynamic properties given are for air.           */
/*   All units are SI.                    */
/*                              */
/* Christoph Hiemcke, Fluent, Inc.                */
/*                              */
/* 7 August 2003                       */
/*                              */
/* Procedure for use:                     */
/*                              */
/* 1. User changes the UDF: input the correct values for:       */
/*     Thermodynamic data        MW, b, CC*, P_REF, T_REF     */
/* 2. Launch Fluent, read the mesh                */
/* 3. Compile the UDF, but do not yet load the library        */
/* 4. Use the TUI to launch the real gas law:              */
/*     def/ud/real-gas/udrgm                    */
```

```
/* 5. Go back to Define/UserDefined/Functions/Compiled, and now load */

/*    the library, so that the MDM function becomes available;      */

/*                                              */

/****************************************************************/


#include "udf.h"

#include "stdio.h"

#include "ctype.h"

#include "stdarg.h"


#define  RGASU (UNIVERSAL_GAS_CONSTANT)

#define  MW 28.966      /* molec. wt. for single gas (kg/kmol)      */

#define  b  0.001265    /* covolume term for single gas (m^3/kg)   */

                        /* b = R * T_crit / ( 8 * p_crit )         */

/* Ideal gas specific heat curve fit; the "cold" range is from 300   */

/*   to 1000 K, whereas the hot range is from 1000 to 5000 K; the    */

/*   values are for air, from [Zucrow & Hoffman, 1976].            */


#define CC1_cold  1048.71

#define CC2_cold -0.383871

#define CC3_cold  0.945559e-3

#define CC4_cold -0.548647e-6

#define CC5_cold  0.0790677e-9


#define CC1_hot  873.949

#define CC2_hot  0.384069

#define CC3_hot -0.140147e-3

#define CC4_hot  0.0245553e-6

#define CC5_hot -1.63649e-12
```

```c
/* Reference State:                              */

#define P_REF 101325.0

#define T_REF 288.15


static int  (*usersMessage)(char *,...);

static void (*usersError)(char *,...);

static double rgas, cp_int_ref;


/******************************************************************/

/* Thermodynamic functions:                         */

/******************************************************************/


/*---------------------------------------------------------------*/

/* FUNCTION: ANEOS_error                             */

/*---------------------------------------------------------------*/

DEFINE_ON_DEMAND(I_do_nothing)

{

/* this is a dummy function to allow us */

/* to use the compiled UDFs utility */

}

void ANEOS_error(int err, char *f, char *msg)

{

 if (err)

   usersError("ANEOS_error (%d) from function: %s\n%s\n",err,f,msg);

}

/*---------------------------------------------------------------*/

/* FUNCTION: ANEOS_Setup                             */

/*        Compute some constants                     */
```

```c
/*-----------------------------------------------------------------*/

void ANEOS_setup(Domain *domain, cxboolean vapor_phase, char *filename,

            int  (*messagefunc)(char *format, ...),

            void (*errorfunc)(char *format, ...))

{

  rgas = RGASU/MW;

  cp_int_ref = CC1_cold*log(T_REF) + T_REF*(CC2_cold

              + T_REF*(0.5*CC3_cold+ T_REF*(0.333333*CC4_cold +

               0.25*CC5_cold*T_REF)));


  usersMessage = messagefunc;

  usersError   = errorfunc;

  usersMessage("\nLoading Real-Ideal Library: %s\n", filename);

}
/*-----------------------------------------------------------------*/
/* FUNCTION: ANEOS_density                          */
/*        Returns density given T and press               */
/*-----------------------------------------------------------------*/


double ANEOS_density(double Temp, double press, double yi[])

{

  double rho = 1.0 / ( (rgas*Temp/press) + b );

  return rho;            /* (kg/m^3)                */

}
/*-----------------------------------------------------------------*/
/* FUNCTION: ANEOS_H_ideal_gas                       */
/*        Returns ideal gas specific enthalpy for a given T      */
/*-----------------------------------------------------------------*/
double ANEOS_H_ideal_gas(double Temp)
```

```
{
  double Ho = 0.0;


  if (Temp<=1000)

          Ho = Temp*(CC1_cold + Temp*(0.5*CC2_cold

              + Temp*(0.333333*CC3_cold+ Temp*(0.25*CC4_cold

              + Temp*0.2*CC5_cold))));

  if (Temp>1000)

    Ho = Temp*(CC1_hot  + Temp*(0.5*CC2_hot+ Temp*(0.333333*CC3_hot

        + Temp*(0.25*CC4_hot + Temp*0.2*CC5_hot))));

  return Ho;

}
/*------------------------------------------------------------------*/
/* FUNCTION: ANEOS_cp_ideal                      */
/*        Returns specific heat at a given T, assuming ideal gas   */
/*------------------------------------------------------------------*/
double ANEOS_cp_ideal(double Temp)

{
  double cpo = 0.0;


  if (Temp<=1000)

          cpo = CC1_cold + Temp*(CC2_cold + Temp*(CC3_cold

              + Temp*(CC4_cold + Temp * CC5_cold)));

  if (Temp>1000)

        cpo = CC1_hot  + Temp*(CC2_hot + Temp*(CC3_hot

              + Temp*(CC4_hot + Temp * CC5_hot)));


  return cpo;             /* (J/kg-K)                 */

}
```

```
/*----------------------------------------------------------------*/
/* FUNCTION: ANEOS_specific_heat                  */
/*        Returns specific heat at a given T        */
/*----------------------------------------------------------------*/


double ANEOS_specific_heat(double Temp, double rho, double P, double yi[])
{
   double cp;
   cp     = ANEOS_cp_ideal(Temp);


    return cp;              /* (J/Kg-K)              */
}
/*----------------------------------------------------------------*/
/* FUNCTION: ANEOS_enthalpy                       */
/*        Returns specific enthalpy given T and rho       */
/*----------------------------------------------------------------*/
double ANEOS_enthalpy(double Temp, double rho, double P, double yi[])
{
   double delta_h,press,v,h;
   v      = 1.0 / rho;
   press   = rgas*Temp / (v - b);
   delta_h = press*b;
   h      = ANEOS_H_ideal_gas(Temp)+delta_h;
    return h;             /* (J/Kg)              */
}
/*----------------------------------------------------------------*/
/* FUNCTION: ANEOS_entropy                        */
/*        Returns entropy given T and rho          */
/*----------------------------------------------------------------*/
```

```c
double ANEOS_entropy(double Temp, double rho, double P, double yi[])
{
  double delta_s,v,vo,s;
  double cp_integral = 0.0;


  if (Temp<=1000)
   cp_integral = CC1_cold*log(Temp) + Temp*(CC2_cold

               + Temp*(0.5*CC3_cold+ Temp*(0.333333*CC4_cold

               + 0.25*CC5_cold*Temp)))- cp_int_ref;
  if (Temp>1000)
   cp_integral = CC1_hot*log(Temp)  + Temp*(CC2_hot

               + Temp*(0.5*CC3_hot+ Temp*(0.333333*CC4_hot

               + 0.25*CC5_hot*Temp)))- cp_int_ref;
   v     = 1.0 / rho;
   vo     = rgas*Temp/P_REF;
   delta_s = rgas*log((v-b)/vo);
   s     = delta_s + cp_integral;     /* (J/kg)            */
   return s;
}
/*----------------------------------------------------------------*/
/* FUNCTION: ANEOS_mw                              */
/*        Returns molecular weight                */
/*----------------------------------------------------------------*/


double ANEOS_mw(double yi[])
{
 return MW;            /* (Kg/Kmol)              */
}
```

103

```c
/*----------------------------------------------------------------*/
/* FUNCTION: ANEOS_speed_of_sound                    */
/*        Returns s.o.s given T and rho              */
/*----------------------------------------------------------------*/
double ANEOS_speed_of_sound(double Temp, double rho, double P, double yi[])
{
    double cp = ANEOS_specific_heat(Temp, rho, P, yi);

    double v = 1./rho;

    double c = (v/(v-b))*sqrt(cp*rgas*Temp/(cp-rgas));


    return c;              /*   (m/s)                */
}
/*----------------------------------------------------------------*/
/* FUNCTION: ANEOS_viscosity                       */
/*        Viscosity via 3-coefficient Sutherland Law       */
/*----------------------------------------------------------------*/
double ANEOS_viscosity(double Temp, double rho, double P, double yi[])
{
    double mu = 0.0;
    double mu_Sutherland = 1.7894e-05;
    double T_Sutherland  = 273.11;
    double S_Sutherland  = 110.56;


    mu = mu_Sutherland * (  pow( (Temp/T_Sutherland),1.5)  )
            * (T_Sutherland + S_Sutherland) / (Temp + S_Sutherland);
    return mu;             /* (Kg/m-s)                */
}


/*----------------------------------------------------------------*/
```

```c
/* FUNCTION: ANEOS_thermal_conductivity               */
/*          Euckun formula                            */
/*----------------------------------------------------*/

double ANEOS_thermal_conductivity(double Temp, double rho, double P, double yi[])
{
  double cp,mu,k;
  cp = ANEOS_specific_heat(Temp, rho, P, yi);
  mu = ANEOS_viscosity(Temp, rho, P, yi);
  k  = (cp+1.25*rgas)*mu;
  return k;              /* (W/m-K)                   */
}
/*----------------------------------------------------*/
/* FUNCTION: ANEOS_rho_t                              */
/*          Returns derivative of rho wrt. T at constant p  */
/*----------------------------------------------------*/
double ANEOS_rho_t(double Temp, double rho, double P, double yi[])
{
  double press,v,rho_t;
  v    = 1.0 / rho;
  press = rgas*Temp/(v - b);
  rho_t = -rgas/(press*v*v);
  return rho_t;          /* (Kg/m^3-K)                */
}
/*----------------------------------------------------*/
/* FUNCTION: ANEOS_rho_p                              */
/*          Returns derivative of rho wrt. p at constant T  */
/*----------------------------------------------------*/
```

```c
double ANEOS_rho_p(double Temp, double rho, double P, double yi[])

{

  double press,v,rho_p;

  v    = 1.0 / rho;

  press = rgas*Temp/(v - b);

  rho_p = rgas*Temp/(press*press*v*v);

  return rho_p;            /* (Kg/m^3-Pa)                */

}

/*----------------------------------------------------------------*/

/* FUNCTION: ANEOS_enthalpy_t                            */

/*        Returns derivative of h wrt. T at constant p, which     */

/*        equals cp                            */

/*----------------------------------------------------------------*/

double ANEOS_enthalpy_t(double Temp, double rho, double P, double yi[])

{

  return ANEOS_specific_heat(Temp, rho, P, yi);

}

/*----------------------------------------------------------------*/

/* FUNCTION: ANEOS_enthalpy_p                            */

/*        Returns derivative of h wrt. p  at constant T;        */

/*----------------------------------------------------------------*/

double ANEOS_enthalpy_p(double Temp, double rho, double P, double yi[])

{

  return b ;

}

/******************************************************************/

/* Actual UDF hook (macro) into the User Defined Real Gas Model:     */

/******************************************************************/
```

```
/*---------------------------------------------------------------*/
/* Export real gas functions to the solver:                */
/*---------------------------------------------------------------*/

UDF_EXPORT RGAS_Functions RealGasFunctionList =
{
  ANEOS_setup,              /* initialize              */
  ANEOS_density,            /* density                 */
  ANEOS_enthalpy,           /* enthalpy                */
  ANEOS_entropy,            /* entropy                 */
  ANEOS_specific_heat,      /* specific_heat           */
  ANEOS_mw,                 /* molecular_weight        */
  ANEOS_speed_of_sound,     /* speed_of_sound          */
  ANEOS_viscosity,          /* viscosity               */
  ANEOS_thermal_conductivity,  /* thermal_conductivity    */
  ANEOS_rho_t,              /* drho/dT |const p        */
  ANEOS_rho_p,              /* drho/dp |const T        */
  ANEOS_enthalpy_t,         /* dh/dT |const p          */
  ANEOS_enthalpy_p          /* dh/dp |const T          */
};
/*****************************************************************/
/* End of the User Defined Real Gas Model UDF (Abel-Nobel EOS)      */
```

2. UDF TO DEFINE SIX-DEGREES OF FREEDOM (DOF) PARAMETERS

```
/*****************************************************

SDOF property compiled UDF with external forces/moments

*****************************************************/

#include "udf.h"

DEFINE_SDOF_PROPERTIES(projectile, prop, dt, time, dtime)

{

  prop[SDOF_MASS]     = 0.097198;

  prop[SDOF_IXX]      = 0.0;

  prop[SDOF_IYY]      = 0.0;

  prop[SDOF_IZZ]      = 0.0;

  printf ("\nprojectile: updated 6DOF properties");

}
```