ONE-DIMENSIONAL ANALYSIS TECHNIQUES FOR PULSED JET FLOW DISTRIBUTION SYSTEMS

By

KALYANASUNDARAM KRISHNAN

Bachelor of Technology

University of Kerala

Thiruvananthapuram, India

1997

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
December, 2004

ONE-DIMENSIONAL ANALYSIS TECHNIQUES FOR

PULSED JET FLOW DISTRIBUTION SYSTEMS

Thesis Approved:

Dr.Frank.W.Chambers

Thesis Advisor

Dr.David.G.Lilley

Dr.Eduardo.A.Misawa

Dr.A.Gordon Emslie

Dean of the Graduate College

ACKOWLEDGEMENTS

I express my sincere appreciation and gratitude to Dr. Frank.W.Chambers, my thesis and faculty advisor for his support, guidance and encouragement throughout this research. I would also like to thank my committee members, Dr. David .G. Lilley and Dr. Eduardo .A. Misawa for their support and guidance. I am also thankful to my colleague, Mr. Kapil Malhotra for his support towards the experimental component of this research.

I deeply appreciate the encouragement given by my grandmother, V.S. Saraswathy Ammal, my parents, Dr. K. Kalyanasundaram and Dr. E. K. Radha and all my friends who have helped me in this endeavor. I thank the department of Mechanical and Aerospace engineering (MAE) and the Oklahoma State University Advanced Technology Research Center (ATRC) for the computational and other facilities provided. I thank Oklahoma NASA EPSCOR and NASA Langley Research Center for the financial support extended for this research project.

Finally yet importantly, I thank the almighty for all the blessings I have received and for the mental strength that kept me going in successfully completing this endeavor.

TABLE OF CONTENTS

Chapter Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

NOMENCLATURE

a	=	Acoustic velocity of pressure wave, m/s
a_0	=	Acoustic velocity of pressure wave at reference conditions, m/s
A	=	Area, m^2
A_t	=	Area at the throat, m^2
c	=	Velocity of gas particle, m/s
c_t	=	Velocity of gas particle at throat, m/s
c_s	=	Superposition velocity of gas particle, m/s
C_p	=	Specific heat capacity at constant pressure, J/kg K
C_v	=	Specific heat capacity at constant volume, J/kg K
C_d	=	Coefficient of discharge
C_f	=	Skin friction coefficient
C_h	=	Convection heat transfer coefficient, W/m^2K
C_k	=	Coefficient of thermal conductivity, W/mK
D	=	Diameter, m
dp_f	=	Pressure loss due to friction, Pa
dQ	=	Heat generated, J
dt	=	Time step, s

f	=	Frequency of wave, Hz
G	=	Functions of ratio of specific heats (γ)
H	=	Enthalpy, J
L	=	length of computational mesh, m
m	=	Mass, kg
$\dot{m}$	=	Mass flow rate, kg/s
M	=	Mach number
M_s	=	Superposition Mach number
Nu	=	Nusselt number
p	=	Pressure, Pa
p_0	=	Reference/ambient pressure, Pa
p_r	=	Reflected pressure, Pa
p_s	=	Superposition pressure, Pa
P	=	Pressure ratio
r	=	Radius, m
R	=	Characteristic gas constant, J/kgK
Re	=	Reynolds number
t	=	Time, s
T	=	Temperature, K
u	=	Specific internal energy, J/kg
U	=	Internal energy, J
V	=	Volume, m^3

$\dot{V}$	=	Volume flow rate, m^3/s
x	=	length, m
X	=	Pressure amplitude ratio
X_t	=	Pressure amplitude ratio at throat
X_i	=	Pressure amplitude ratio (incident)
X_r	=	Pressure amplitude ratio (reflected)
X_s	=	Superposition pressure amplitude ratio

Greek Symbols

α	=	Velocity of pressure wave propagation, m/s
α_s	=	Velocity of pressure wave propagation at superposition, m/s
γ	=	Ratio of specific heats, C_p/C_v
ρ	=	Density of particle at any point on the wave, kg/m^3
ρ_0	=	Density at reference conditions, kg/m^3
ρ_s	=	Density at superposition conditions, kg/m^3
τ	=	Shear stress, N/m^2
μ	=	Coefficient of viscosity, kg/ms
Π	=	Purity of gas

CHAPTER I

INTRODUCTION

1.1 Background

Active flow control (AFC) is a very important area in aerodynamics today. The expression 'active' refers to the process of inputting small amounts of energy locally to achieve large performance gains throughout the flow field. The aim of AFC is typically delaying stall, increasing lift, reducing drag, enhancing combustion, and decreasing jet noise. This is achieved by delaying or advancing transition from laminar to turbulent flow, avoiding or delaying separation and suppressing or enhancing turbulence levels. Typical control loops used for AFC are open loop, reactive feedforward and reactive feedback open. In open loop control, the controlled variable is predetermined. No sensed information is fed forward in this case. Reactive control refers to a special class of active control where the input to the controller is continuously adjusted based on measurements of the controlled variable. In feedforward control, the measured variable and the controlled variable differ. For example, the pressure or velocity can be measured at an upstream location and the resulting signal is fed to an actuator, which sends appropriate control signals to influence the velocity at a downstream location. In feedback control, the controlled variable is measured, fed back and compared to a reference input. This device, which compares the feedback value and the reference value is called a Comparator. This device triggers an appropriate response from the actuator thereby controlling the input. Figures 1.1 and 1.2 summarize the engineering goals, their interrelation and corresponding flow changes.

Several methods are available for AFC. Figure 1.3 shows classification of flow control strategies. Gad-el-Hak [1] [2] describes the many methods, points to technologies currently under development and the modern tools used for soft computing. The term soft computing refers to several ingenious modes of computations that exploit tolerance for imprecision and uncertainty in complex systems to achieve tractability, robustness and low cost. It refers to a domain of computational intelligence that loosely lies between purely numerical computing and purely symbolic computations. The principal constituents of soft computing are neurocomputing, fuzzy logic and genetic algorithms. A detailed description of these methods can be found in [2].

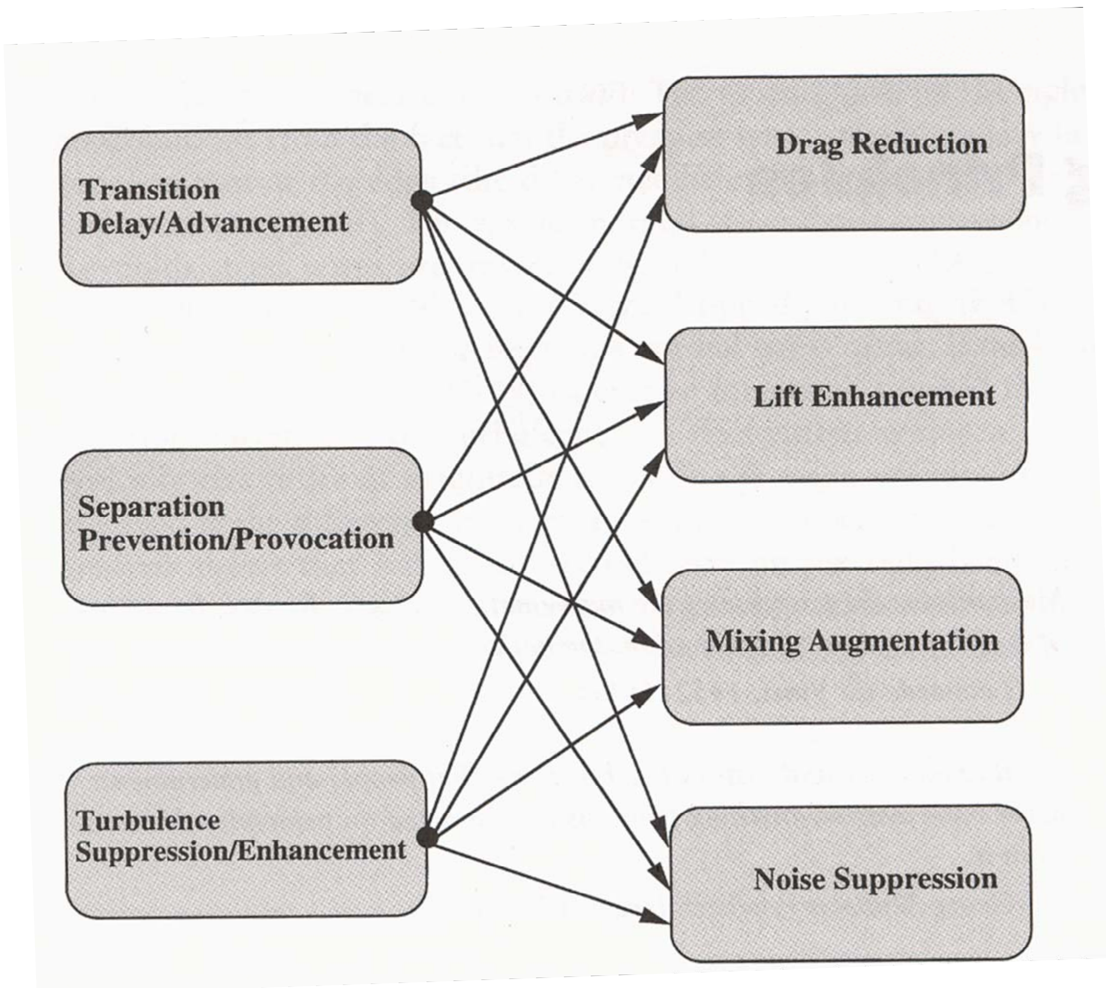


Figure 1.1 Engineering goals and corresponding flow changes [2]

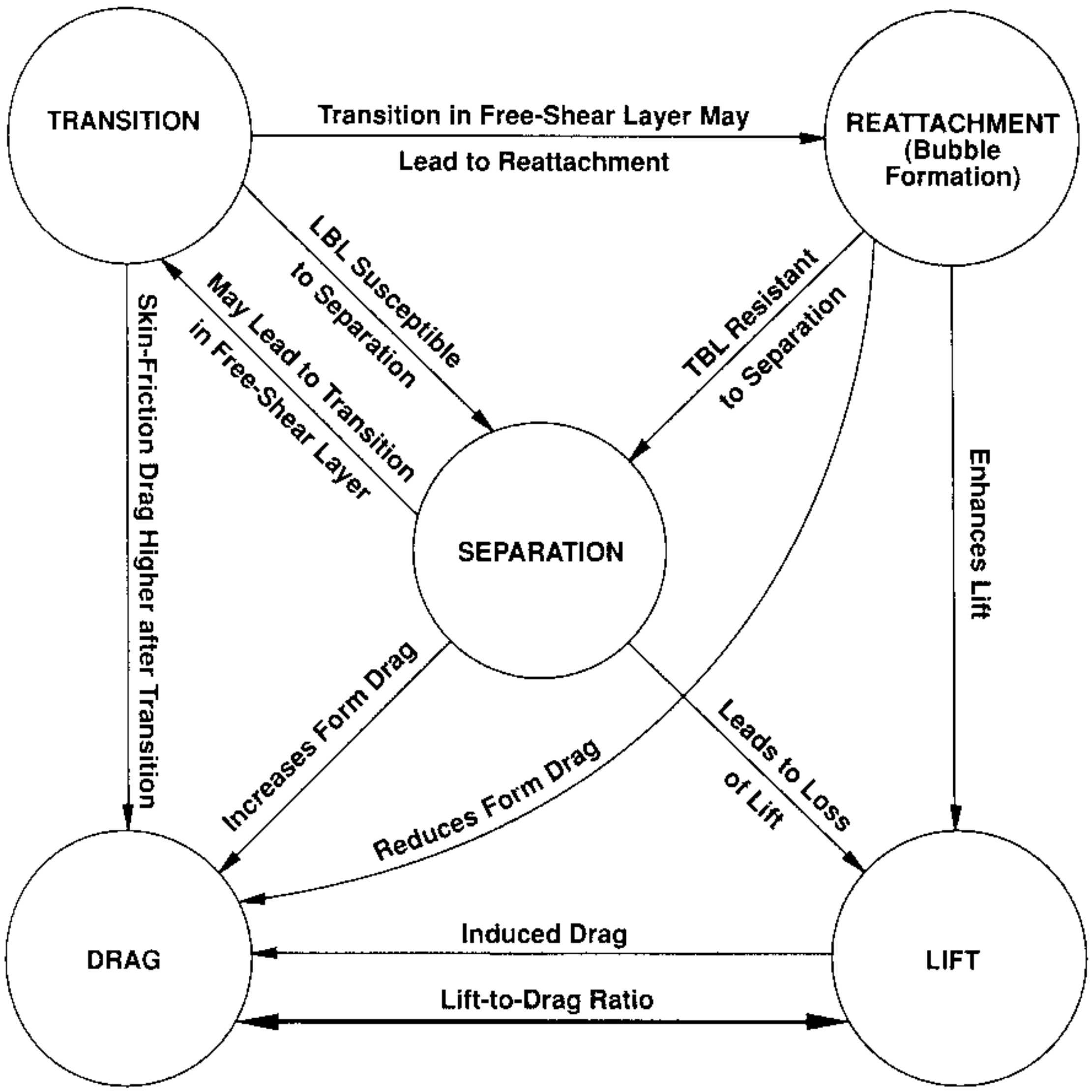


Figure 1.2 Interrelation between flow control goals [2]

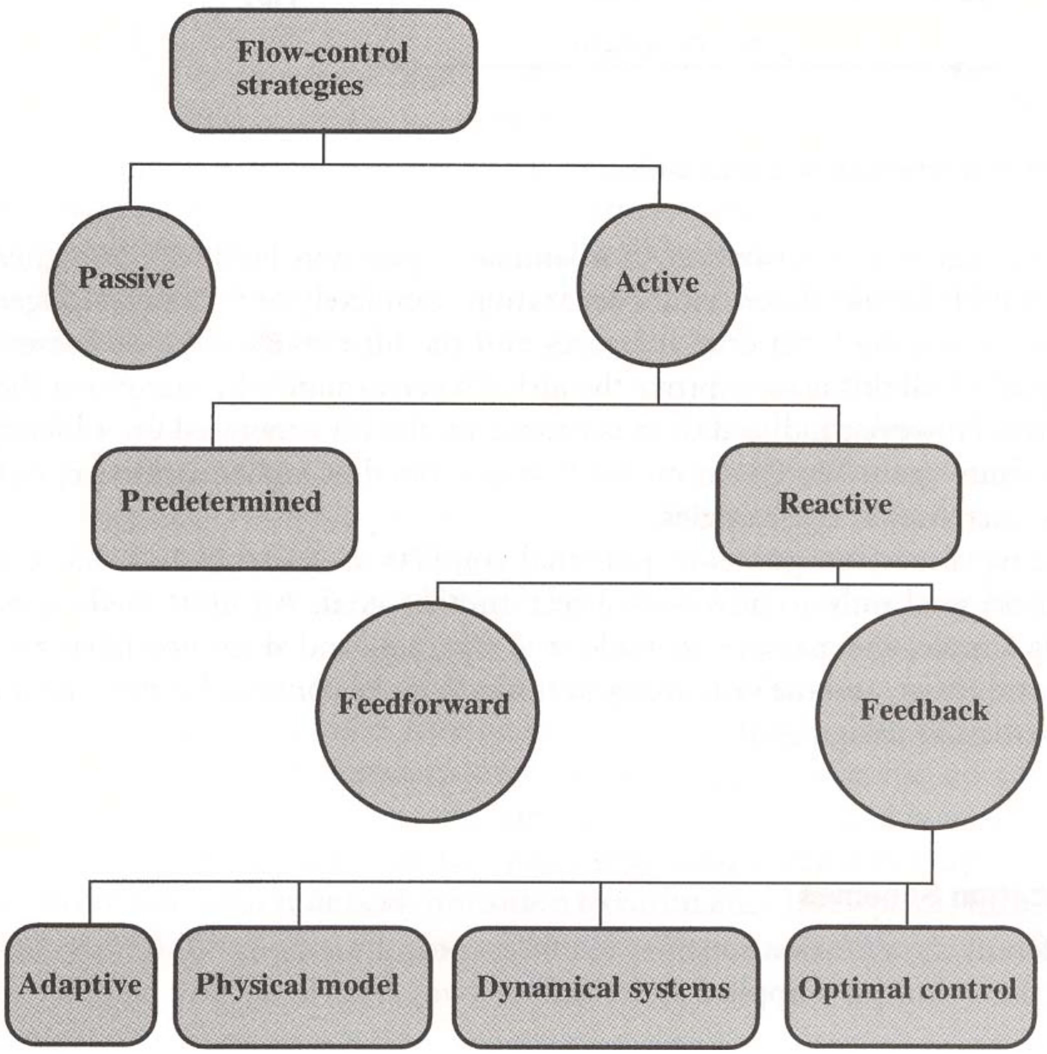


Figure 1.3 Classification of flow control strategies [2]

1.2 Circulation Control Wing Concept

Circulation control wings achieve high lift by turning the flow over the airfoil using the Coanda effect in place of trailing edge flaps. The turning is performed over a large radius trailing edge using blowing through narrow slots to enhance the Coanda effect for the flow over the airfoil. Use of steady jets even at very small mass flow rates can yield lift coefficients higher than conventional systems using flaps [6] [7]. Pulsed jets even at low duty cycles are able to accomplish the desired lift [7]. A 2-D supercritical airfoil model described by Jones et al. [6] is shown in figure 1.4.

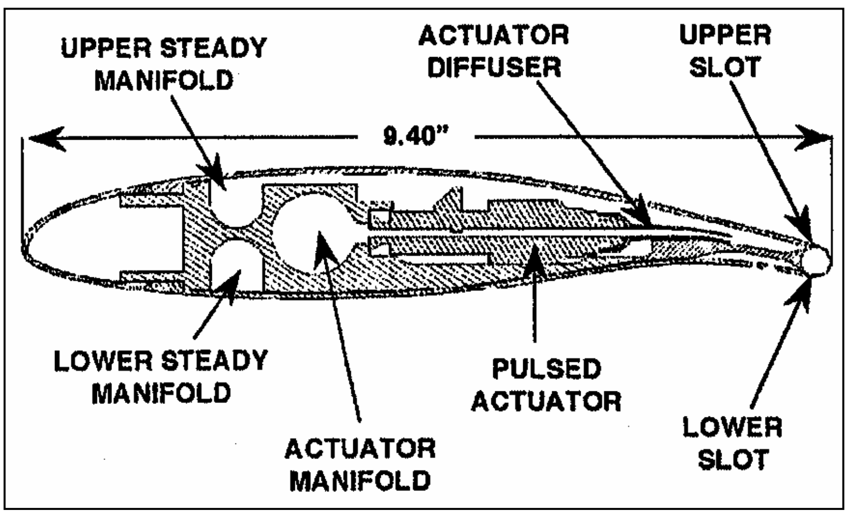


Figure 1.4 Circulation Controlled Airfoil Internal Passages [6]

In the setup shown in figure 1.4 two manifolds are shown. These manifolds supply air to an actuator. Slots are provided on the upper and lower surfaces of the wing. Moving from a high lift to a cruise configuration is dependent on the upper and blowing ratios and the free stream velocity. This is done by modifying the mass flow through the upper and lower slots. Each slot flow is independently controlled.

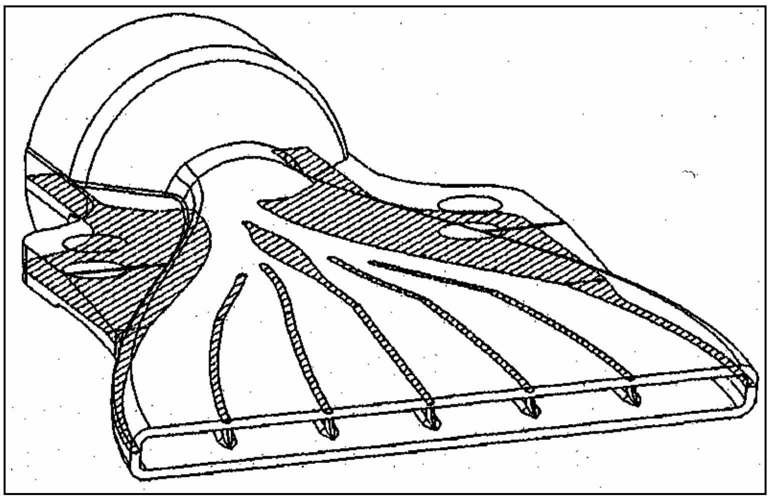


Figure 1.5 Individual Actuator Diffuser [6]

An individual actuator diffuser is shown in figure 1.5. The flow field out of the actuator is a small diameter circular high-speed jet. The objective of the diffuser is to transition from a circular, time-dependent high-speed jet to a low speed 2D, uniform jet. The effectiveness of the system depends on the actuator performance, diffuser performance and the response of the internal conduit prior to the jet exit. Even in the case in which an ideal pulsed flow can be created at the actuator, the effects of the flow passage from the actuator to the slot can distort the pulse so that the jet leaving the slot has a different, less effective, pulse shape. The pressure pulse that leaves the actuator is distorted in amplitude and shape by friction and reflections. For the design of pulsed blowing systems, a good model that considers these effects is needed to predict the characteristics of the pulse at the slot exit. A design tool is needed for this purpose. It is in this aspect that the present work becomes important. An appropriate model is required for evaluating the above effects. This is the objective of this research work.

Active flow control can be applied to aircraft design applications to achieve improvements in performance like increased lift, reduced drag, averted separation and decreased noise. Blowing is one method shown to be effective by many researchers. Continous blowing can require excessive amounts of engine bleed air in aircraft applications. Magill and McManus [3], Liu [4] and Kim [5] have demonstrated that pulsed blowing can achieve the same goals with lower air flow requirements. They attribute the effect to enhanced vorticity production for an impulsively started jet flow and reduced mass flow obtained with the low duty cycle. Liu [4] uses a square pulse with a 50% duty cycle and various frequencies of 40 Hz, 120 Hz and 400 Hz. He compares the performance of these three frequencies with a steady jet. The lift coefficient is found to be high for the pulsed jet with 400 Hz frequency. Further, a comparison of time averaged mass flow rate vs. momentum coefficient and efficiency vs. time averaged momentum coefficient reveals that high frequency jets are best suited for this application. Magill and McManus [3] conducted experiments to prove that square-pulsed jets at high frequency and low duty cycles of 25% provide the optimum increase in lift. These researchers are of the opinion that the most effective way in suppressing stall is to use an unsteady or pulsed jet with a duty cycle of 10-50% and high frequency. To summarize, the advantage of pulsed blowing is that minimum air is required from the compressor with high lift achieved even at low duty cycles. Pulses with a square profile with low duty cycles of 25% [3] [4] and high frequencies of approximately 400 Hz [4] have been found to be very effective.

1.3 Problem statement and presentation

Efficient systems for applications of pulsed blowing require careful design. The challenges are that the engine bleed air must be used as sparingly as possible and the air must be distributed to the multiple locations at which active control is to be implemented. This manifold distribution problem is complicated further by the unsteady pulsatile flow. The dynamics of the distribution system will play an important role in overall system effectiveness. Several studies [5] [6] [8] have concentrated on the actuators and their modeling but the studies on the transmission tubing have been limited. The distribution systems under consideration for pulsed blowing consist of plenums and multiple distribution tubes, with similarities to ventilation system ducting and the intake and exhaust systems of internal combustion engines. One-dimensional modeling approaches are well developed for the flow and acoustics of such systems. Munjal [9], Munjal and Doige [10] and Gupta et al. [11] describe these techniques for ducts and mufflers. Bulaty and Widenhorn [12] and Blair [13] are among those presenting similar techniques for the simulation of internal combustion engine system flow and acoustics. The unsteady pulsatile flow of internal combustion engine exhaust systems has many similarities to pulsed jet active flow distribution systems. Prediction techniques are needed for the dynamic performance of pulsed jet flow control systems.

1.4 Objectives of Research

The classic transmission line analysis of Brown [14] and the one-dimensional fluid-acoustic analysis techniques of Munjal [9] and of Blair [13] serve as guidelines for modeling the transmission system. Among these works, Blair's was selected because of simplicity, ease of modeling and accuracy of results.

The objectives of this research are:

(1) Development of a fluid – acoustic analysis scheme for predictions of pulsed jet flow distribution system performance. The parameters of interest are pressure, velocity and pulse frequency. The effect of input variables like inlet pressure and duty cycle and simulation parameters such as mesh size and time step need to be studied.

(2) Modeling of the transmission tubing using Blair's model.

(3) Implementation of the model in a computer program.

(4) Analyzing the effect of various input and simulation parameters.

(5) Code validation for select cases.

CHAPTER II

LITERATURE REVIEW

2.1 Studies on active flow control

The motivation for this research comes from several studies directly on flow control and other related work conducted by various researchers. Some of the studies have concentrated on the aerodynamic analysis and benefits of pulsed blown systems on aircraft. Some others have attempted to model the system itself. Both groups have highlighted the advantages of pulsed blowing as a means of active flow control (AFC). Other works have targeted flow control opportunities in non-aerodynamic areas.

Liu [4] has underlined the importance of pulsed blowing as an effective method of AFC. He suggests circulation control technology as a useful way of achieving very high lift. Two-dimensional blowing results are presented to prove that the pulsed jet at high frequencies is an effective way of obtaining high lift compared to a steady jet while requiring lower mass flow rates. Figure 2.1 shows the plot of lift coefficient vs. mass flow rate for steady and pulsed jets of different frequencies.

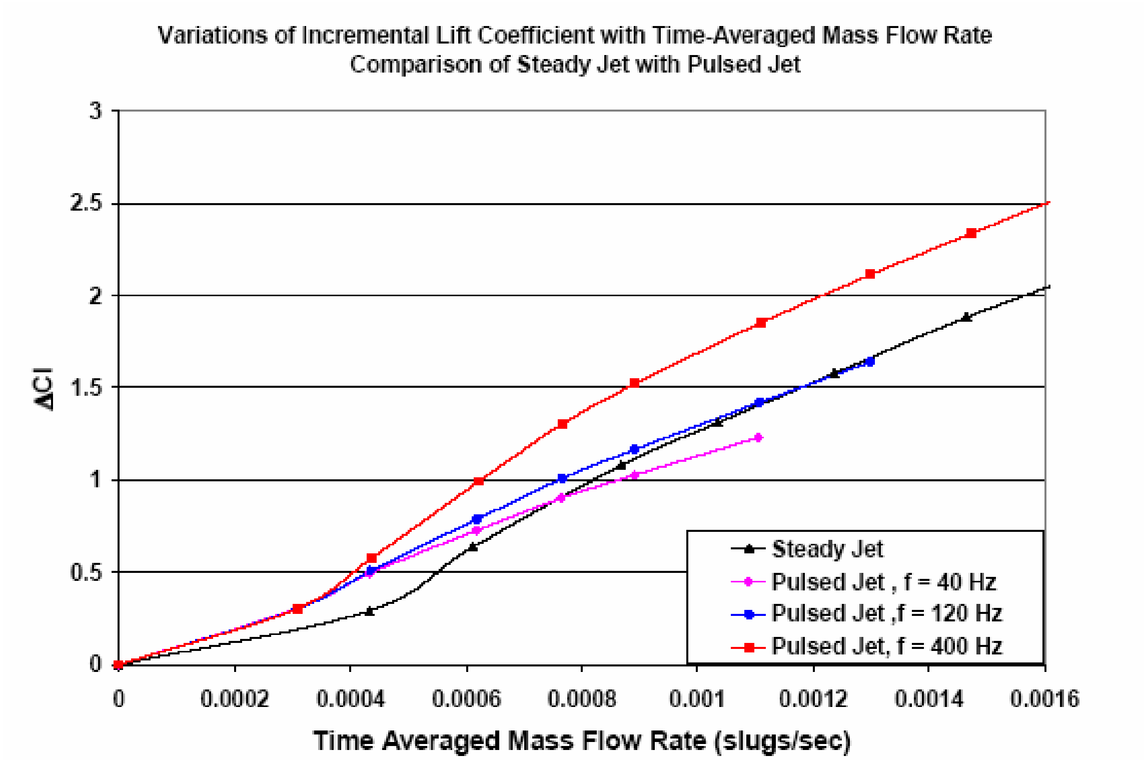


Figure 2.1 Variation of incremental lift coefficient with time-averaged mass flow rate [4]

Other works with similar findings are those by Magill and McManus [3] and Kim [5]. Magill and McManus [3] have shown that pulsed jets increase lift and Lift/Drag ratio affecting only small changes to drag. They also find that pulsed vortex generator jets are highly effective at high leading edge flap deflections.

Kim [5] has modeled a pulsed blowing system using a lumped element model for the actuator and a distributed model for the transmission tubing. In the lumped element analysis techniques developed in the 1960's for simulating fluid-acoustic phenomena, the entire actuator and tubing was modeled as a lumped mass. Kim [5] has improved upon this approach, noting that the length of the tubing connecting the actuator to the valve is not small compared to the acoustic wavelength. The transmission tubing distributed model in the above work is essentially based on the work by Brown [14] and Karam and Franke [15]. These works evaluate the characteristic impedance, amplitude frequency response and propagation factor in rigid uniform transmission lines with the effects of varying velocity profile and heat transfer included. For more basic theory, one may refer to the works of Nichols [16], Iberall [17], Tijdeman [18] and Hunt [19]. Kim [5] is of the opinion that Karam and Franke's study [15] is not realistic in engineering applications due to the use of a closed tube. However, he attributes this work to the basis of studies of more realistic systems where there are networks of tubes that have open ends. Experiments were conducted in this work involving different slot widths and heights. This work concludes that a mathematical model is required for guiding the system design.

Joslin et al. [20] focus on a strategy to develop tools for transitioning active flow control from the laboratory to applications. Pulsed pneumatic high lift technology and its potential for aircraft systems has been studied by Jones et al. [6]. They combine CFD and wind tunnel experiments to quantify flow parameters such as boundary layer separation, slot-velocity profiles, plenum pressures, lift, drag and pitching moment. These researchers underline the importance of time accurate measurements at the slot exit in understanding the flow physics of the pulsed circulation system. They point out the difficulties researchers face in making detailed and accurate measurements at the jet exit due to large perturbations in velocity. Figure 2.2 shows the comparison of lift coefficient for pulsed and steady circulation control from this study. It can be inferred from the figure that the lift coefficient is approximately 20% higher than for a steady jet retaining the same mass flow rate.

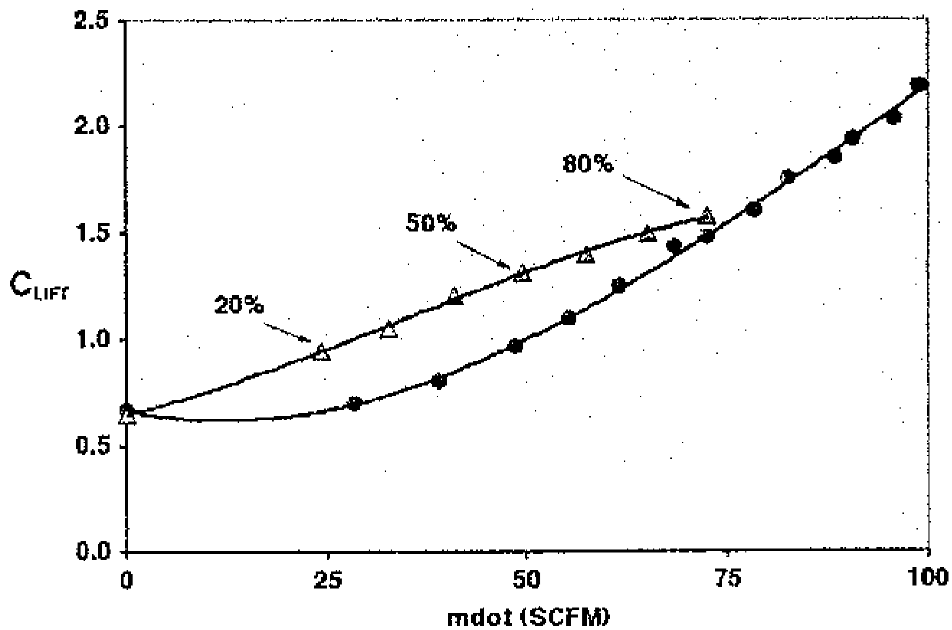


Figure 2.2 Comparison of pulsed and steady circulation control [6]
(Frequency 35 Hz and varying duty cycle)

Other methods like finite element methods have been used by workers like Perotti [21]. A discontinuous finite element method is used in this work. The author claims that this method is superior to finite difference schemes. The method is more efficient at discontinuities. Further, the interfaces between one-dimensional and three-dimensional meshes can be effectively modeled. The speed of execution of the finite element scheme is less than that of a finite difference scheme. Therefore, a speed up of the explicit discontinuous Galerkin finite element code through matrix inversion and accelerated time stepping is suggested.

Numerous researchers have conducted studies pertaining to modeling of flow in ducts. Gupta et al. [11] use a segmentation approach for analyzing ducts with mean flow. They use a transfer matrix for each segment and an overall transfer matrix is obtained by multiplying the individual transfer matrices. These matrices employ all state variables like acoustic pressure, acoustic mass velocity and interrelate these terms using gas dynamic equations. Brown [14] derives functional operators for the propagation factor and characteristic impedance in rigid uniform transmission lines. These factors are used to determine the response of the line to impulse and step excitation. The study by Karam and Franke [15] on frequency response of pneumatic lines focuses on the amplitude frequency response of lines used in fluidic systems. They use theory analogous to electric transmission theory to analyze response of fluid transmission lines. The volumetric flow rate is used rather than the mass flow rate to keep the theory analogous to electric circuit terminology. The volumetric flow rate is modeled as complex hyperbolic functions of impedance and propagation factor.

Bulaty and Widenhorn [12] generalize a three-branch model to an *n*-branch junction model and use it for the one-dimensional unsteady flow calculations in exhaust systems. They use an energy related pressure loss method.

2.2 Flow modeling in internal combustion engines

Unsteady flows that occur in pipe systems of internal combustion engines are similar to flows in our research problem. One-dimensional modeling approaches are well developed for such problems. Bulaty and Niessner [27] use linear finite difference methods with flux correction for such problems. Earlier methods used were the method of characteristics [28] and Lax-Wendroff methods [29].

The one-dimensional unsteady flow in a pipe is described by the governing equations – continuity, momentum and energy in differential form. In pipe systems of internal combustion engines, pressure waves travel back and forth causing reflections at discontinuities. These pressures are further modified due to friction and heat transfer. These phenomena are discussed in detail in chapter III. A comparison of algorithms used in unsteady flow calculations in inlet and exhaust systems of internal combustion engines is done by Vandevoorde et al. [30]. The method of characteristics, different Lax-Wendroff schemes, first order upwind schemes and the latest total variation diminishing (TVD) schemes are compared in this study. The method of characteristics does not discretize equations, but rearranges the initial equations to form non-dimensional equations using Riemann variables that are a combination of density, pressure and velocity and normalizing the equations using dimensionless variables. This method however is very time consuming. Lax-Wendroff schemes are finite volume schemes involving discretization of the conservative form of the governing equations. These schemes are centered in space. These methods work well for contact discontinuities and gave second order accuracy for the spatial and time derivative. For pressure waves, these provide a major improvement to the method of characteristics; however, it does not exactly represent the contact discontinuity as the method of characteristics. Discretization errors are common at section changes. In first-order upwind schemes, the information is obtained from the physically relevant directions unlike the Lax-Wendroff schemes that use the whole environment for the calculation of new value of the variable. The upwind schemes have first order accuracy. The TVD schemes use a cell vertex formulation. The fluxes are calculated on the boundaries of the control volumes in this scheme. The flux difference is distributed to the nodes inside the control volume in an upwind way, so that the node intercepts the flux in a direction that is relevant to the physics of the problem. This scheme is second order accurate in space and first order accurate in time. The authors conclude that no other schemes are suitable for the unsteady flow analysis. The TVD scheme exhibits acceptable computational time and accuracy. TVD property of flux corrected transport techniques is discussed by Gascon et al. [31].

Blair [13] uses a one-dimensional technique for the simulation of pressure wave motion in the intake and exhaust ducts of internal combustion engines. His model has the advantages of simplicity with acceptable accuracy. This method, called the GPB method, can simulate the fluid flow through a four-stroke engine with a user defined choice of fuel and other engine parameters like bore, stroke, air/fuel ratio, valve parameters and choice of inlet and exhaust pipe configuration. In this work, this method is adopted for the flow simulation. The inlet pressure and the piping configuration is user defined. From the standpoint of ease of application of the theory on a digital computer, this method has proven to be the motivating factor in this research. Kirkpatrick et al. [32] compares five methods namely the homentropic (isentropic) method of characteristics, the non-homentropic method of characteristics, the two-step Lax-Wendroff method with flux corrected transport, the Harten-Lax-Leer upstream difference method and the GPB method. This paper concludes that all these methods except homentropic method of characteristics are suitable for non-isentropic flow conditions. Blair et al. [33] discusses the experimental validation of the code for discontinuity of gas properties using carbon dioxide and air. The accuracy and the low execution time of the GPB modeling method are established through a series of experiments. Blair et al. [34] also present results for the experimental validation of the GPB modeling method for a pipe system containing area discontinuities. A sudden enlargement, sudden contraction, divergent taper, long and short megaphone and a convergent taper configuration are used for validation. Area ratios in the range of 2 to 18 are tested. The included angles for the taper configuration tested are in the range 2.9° - 28°. The capability of GPB method to accurately predict flow patterns where separation takes place – when the included angle is more than 10° is superior to other codes like Lax-Wendroff with flux corrected transport.

The GPB modeling method is the preferred choice for many reasons. The modeling is easier compared to other methods, has faster computation time, and provides acceptable accuracy of the solution.

CHAPTER III

APPROACH AND UNDERLYING THEORY

3.1 Theory behind the computational model

The model proposed in this work is a one-dimensional one used by Blair [13], [35] for analysis of internal combustion engines. This model can simulate straight pipe geometry and a straight pipe with one sudden expansion and one sudden contraction. The theory behind the computational model is explained below. The numerical aspects will be covered in the next chapter.

3.2 Governing equations

The control surface for one-dimensional compressible flow with heat transfer in a duct is shown in figure 3.1. The governing equations [32] for this flow in differential form are

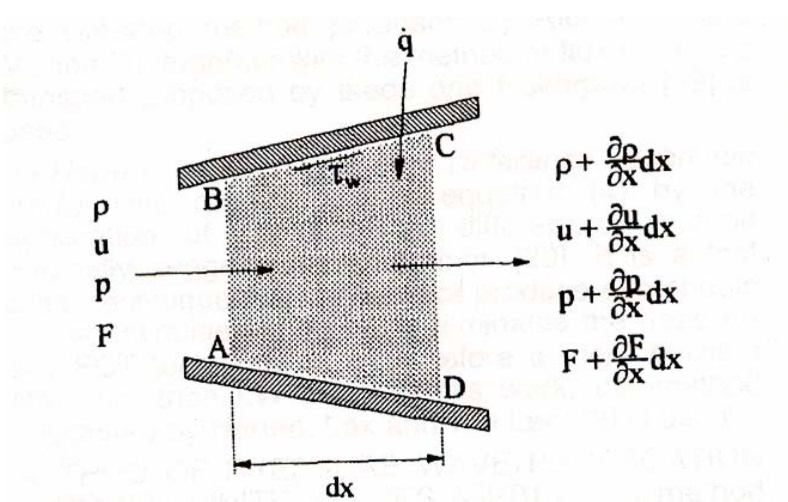


Figure 3.1 Control surface for one-dimensional unsteady compressible flow [32]

Continuity

$$\frac{\partial\rho}{\partial t}+\rho\frac{\partial u}{\partial x}+u\frac{\partial\rho}{\partial x}+\frac{\rho u}{F}\frac{dF}{dx}=0 \qquad (3\text{-}1)$$

Momentum

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+\frac{1}{\rho}\frac{\partial\rho}{\partial x}+g=0 \qquad (3\text{-}2)$$

First law of thermodynamics (Energy equation)

$$\dot{q}\,\rho F dx=\frac{\partial}{\partial t}\left\{(\rho F dx)\left(C_v T+\frac{u^2}{2}\right)\right\}+\frac{\partial}{\partial x}\left\{(\rho u F)\left(C_v T+\frac{u^2}{2}+\frac{p}{\rho}\right)\right\}dx \qquad (3\text{-}3)$$

Where

C_v = specific heat at constant volume, $J/kg\,K$

F = Cross sectional area, m^2

p = Pressure, N/m^2

$\dot{q}$ = Heat transfer rate per unit mass per unit time, $J/kg\,s$

t = time, s

T = Temperature, K

u = particle velocity, m/s

x = Distance, m

ρ = Density, kg/m^3

These are the differential form of the equations and are solved in conventional computational fluid dynamics using finite difference, finite volume and other techniques. Several methods have been used to solve these equations [30] [32]. Some of them are method of characteristics - homentropic and non-homentropic (homentropic methods assume constant entropy and non-homentropic methods assume non-isentropic flow conditions); finite difference methods - Lax-Wendroff with flux corrected transport and Harten-Lax-Leer upstream difference technique. Modern methods include TVD cell vertex schemes. The method used in this work is the method of pressure wave propagation through finite spaces (GPB). This method uses the integral formulation of the governing equations. The flow is assumed quasi-steady. The governing equations in integral form for a control volume [36] are

Continuity

$$\frac{\partial}{\partial t}\int_V \rho \, dV \quad + \quad \int_A \rho(\vec{V} \bullet \hat{n})\, dA \quad = 0 \tag{3-4}$$

Momentum

$$\sum \vec{F} = \frac{\partial}{\partial t}\int_V \rho \vec{V} \, dV \quad + \quad \int_A \rho \vec{V}(\vec{V} \bullet \hat{n})\, dA \tag{3-5}$$

First law of thermodynamics (Energy equation)

$$\dot{Q}_S - \dot{W}_S = \frac{\partial}{\partial t}\int_V \rho(u + \frac{V^2}{2} + g\, z)\, dV + \int_A \rho(h - \frac{p}{\rho} + \frac{V^2}{2} + g\, z)(\vec{V} \bullet \hat{n})\ dA \tag{3-6}$$

Where

dA = infinitesimal area in a control surface, m^2

$\vec{V}$ = velocity vector, m/s

$\vec{F}$ = external force vector acting on inertial system of fluid particles, N

u = internal energy per unit mass, J/kg

g = acceleration due to gravity, m/s^2

z = elevation, m

h = specific enthalpy, J/kg

$\dot{Q}_S$ = Heat transfer per unit time, W

$\dot{W}_S$ = Work transfer per unit time, W

Analysis of internal engine exhaust systems uses a quasi-steady approach and hence the time derivatives are set to zero. So the integral form of the governing equations reduce to

<u>Continuity</u>

$$\dot{m} = \rho AV = \text{Constant} \qquad (3\text{-}7)$$

<u>Momentum</u>

$$-F_f + p_1 A_1 - p_2 A_2 = \rho_2 A_2 V_2^2 - \rho_1 A_1 V_1^2 \qquad (3\text{-}8)$$

<u>First law of thermodynamics (Energy equation)</u>

$$\delta Q_{system} + \Delta m_1 \left(h_1 + \frac{c_1^2}{2} \right) = dE_{system} + \Delta m_2 \left(h_2 + \frac{c_2^2}{2} \right) + \delta W_{system} \qquad (3\text{-}9)$$

δQ_{system} = heat transfer, J

m = mass, kg

h = specific enthalpy, J/kg

c = particle velocity, m/s

δW_{system} = Work transfer, J

dE_{system} = internal energy, J

F_f = frictional force acting on the side walls of the control volume, N

A_1 = Cross-sectional area of pipe at section 1, m^2

A_2 = Cross-sectional area of pipe at section 2, m^2

These equations are modified for the sudden expansion and sudden contraction cases. For the straight pipe, the continuity and energy equations are solved. The derivation of equations in the form used for solution is discussed in Appendix A. The theory of pressure wave motion and all the required theory are discussed in the rest of this chapter. The governing equations in the form used for the solution use the notation described in section 3.3.

3.3 Unsteady flow Analysis by pressure wave theory

What follows is a brief summary of relevant concepts that are used in the computational model to analyze pressure wave motion in a duct. A detailed discussion can be found in [13].

3.3.1 Motion of pressure waves in a pipe

Motion of pressure waves of small amplitude is familiar to us through the theories of acoustic wave motion (sound). Pressure waves are of two types-compression and expansion waves. The compression wave is shown in figure 3.2. An expansion wave is shown in figure 3.3.

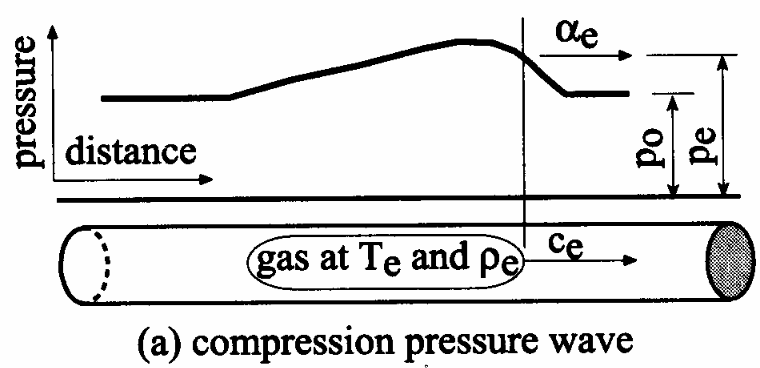


Figure 3.2 Compression wave [10]

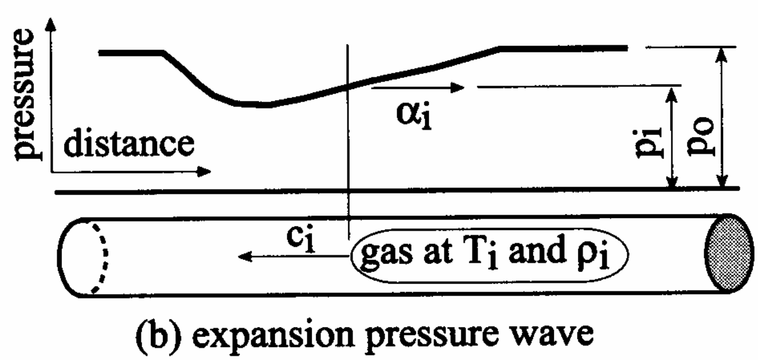


Figure 3.3 Expansion wave [10]

A compression wave increases the particle velocity and decreases the pressure in the direction of travel and an expansion wave decreases the particle velocity and increases the pressure in the direction of travel.

Motion of pressure waves back and forth takes place continuously in the exhaust pipes of automobile engines. When two waves approach each other, they undergo a superposition process. This process is further explained in section 3.3.3. After this process the waves split into leftward and rightward moving waves. A pressure transducer kept in the plane of superposition measures the superposition wave pressure. Since this process takes place continuously, an analytical method needs to be defined to assess the unsteady pressure and velocity at specific locations in the duct. This theory is true for any duct that has pressure waves traveling continuously inside it and hence this theory is adapted for modeling the tubing for the pulsed blowing system.

Some of the relevant parameters used to define and model the flow inside the transmission tubing is detailed below.

3.3.2 Definitions

Pressure ratio (P)

The pressure ratio "P" for any pressure wave is defined as the pressure p at any point of the wave under consideration divided by the undisturbed pressure (ambient) also called the reference pressure.

$$\therefore \quad P = \frac{p}{p_0} \tag{3-10}$$

Characteristic gas constant (R)

This is the gas constant for the particular gas through which the wave propagates. For air the value of R is 287 J/kgK.

Specific heat at constant pressure and volume

These are denoted by C_p and C_v.

Ratio of specific heats (γ)

This is the ratio of specific heat at constant pressure to that at constant volume. This is denoted by γ.

$$\gamma = {C_p}/{C_v} \tag{3-11}$$

Functions of ratio of specific heats

For convenience, various functions of specific heats are defined below, as in Blair [13]. The subscripts are obtained from values of these ratios for air. For example, for air,

$\gamma = 1.4,\ G_3 = 3$

$$G_3 = \frac{4-2\gamma}{\gamma-1} \tag{3-12}$$

$$G_4 = \frac{3-\gamma}{\gamma-1} \tag{3-13}$$

$$G_5 = \frac{2}{\gamma-1} \tag{3-14}$$

$$G_6 = \frac{\gamma+1}{\gamma-1} \tag{3-15}$$

$$G_7 = \frac{2\gamma}{\gamma-1} \tag{3-16}$$

$$G_{17} = \frac{\gamma-1}{2\gamma} \tag{3-17}$$

$$G_{35} = \frac{\gamma}{\gamma-1} \tag{3-18}$$

$$G_{67} = \frac{\gamma+1}{2\gamma} \tag{3-19}$$

Acoustic velocity (a_0)

Acoustic velocity is the velocity of sound in air.

$$a_0 = \sqrt{\gamma R T_0} = \sqrt{\frac{\gamma p_0}{\rho_0}} \tag{3-20}$$

Pressure amplitude ratio (X)

This is defined as

$$X = \left(\frac{p}{p_0}\right)^{\frac{\gamma-1}{2\gamma}} = P^{\frac{\gamma-1}{2\gamma}} \tag{3-21}$$

Gas particle velocity (c)

This is the velocity of a particle of the fluid medium at a particular point on the wave. The gas particle velocity is the speed at which the particle is moving in response to the pressure wave driving it. It is a function of the pressure amplitude ratio of the driving wave. It represents the characteristic velocity in a pipe system and is the parameter measured by a transducer located in the piping system.

$$c = \frac{2}{\gamma-1} a_0 \left[\left(\frac{p}{p_0}\right)^{\frac{\gamma-1}{2\gamma}} - 1\right] = \frac{2}{\gamma-1} a_0 (X-1) = G_5 a_0 (X-1) \tag{3-22}$$

Propagation velocity (α)

When an acoustic wave travels under conditions where pressure is 'p' and the temperature is 'T,' the wave travels at acoustic velocity on top of gas particles that are moving at particle velocity c. The absolute propagation velocity of any point on a wave is the sum of the local acoustic velocity and the gas particle velocity.

$$\therefore \quad \alpha = a + c \tag{3-23}$$

where ' a ' is the local acoustic velocity at pressure ' p ' and temperature ' T '.

From the above definition, we get the following expression for α .

$$\alpha = a_0 X + \frac{2}{\gamma - 1} a_0 (X - 1) = a_0 \left[\frac{\gamma + 1}{\gamma - 1} \left(\frac{p}{p_0} \right)^{\frac{\gamma - 1}{2\gamma}} - \frac{2}{\gamma - 1} \right] = a_0 [G_6 X - G_5] \tag{3-24}$$

Density (ρ)

The density at any point on a wave of pressure p can be written as

$$\rho = \rho_0 X^{G_5} \tag{3-25}$$

This results from the isentropic theory of a perfect gas

$$\frac{\rho}{\rho_0} = \left(\frac{p}{p_0} \right)^{\frac{1}{\gamma}} = X^{\frac{2}{\gamma - 1}} = X^{G_5} \tag{3-26}$$

3.3.3 Superposition of Pressure Waves in a pipe

Figure 3.4 shows two pressure waves superposing in a duct. The state variables can be calculated for this superposed condition.

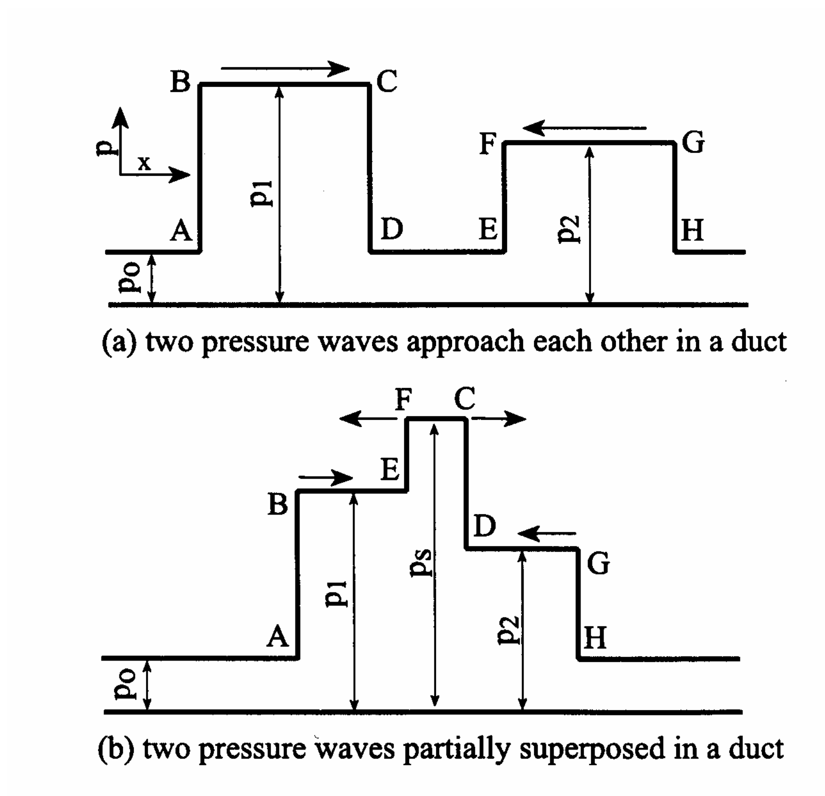


Figure 3.4 Superposition of pressure waves [13]

<u>Superposition velocity</u>

The two waves in the duct superpose and create a superposition wave pressure p_s. Assuming the rightward direction as positive and leftward as negative; the particle and propagation velocities on the wave front BC as c_1 and α_1.

They can be defined with respect to earlier definitions as

$$c_1 = G_5 a_0 (X_1 - 1); \quad \alpha_1 = a_0 (G_6 X_1 - G_5) \tag{3-27}$$

For wave top FG, similarly

$$c_2 = -G_5 a_0 (X_2 - 1); \quad \alpha_2 = -a_0 (G_6 X_2 - G_5) \tag{3-28}$$

Now the superposition velocities are found by summing up the particle velocity of F with respect to BE with c_1.

Therefore the expressions for pressure ratio and particle velocity at superposition become

$$X_s = X_1 + X_2 - 1 \tag{3-29}$$

and

$$c_s = G_5 a_0 (2X_1 - X_s - 1) = -G_5 a_0 (2X_2 - X_s - 1) = G_5 a_0 (X_1 - X_2) \tag{3-30}$$

Similarly the acoustic velocity during superposition is

$$a_s = a_0 X_s \tag{3-31}$$

Therefore the sum of local acoustic and particle velocities give the superposition propagation velocity.

The expression for the same is

$$\alpha_{s\,rightward} = a_s + c_s = a_0 (G_6 X_1 - G_4 X_2 - 1) \tag{3-32}$$

$$\alpha_{s\,leftward} = -a_s + c_s = -a_0 (G_6 X_2 - G_4 X_1 - 1) \tag{3-33}$$

Mass flow rate

The mass flow rate can be obtained from the relation

Mass flow rate = density x area x velocity = $\rho_s A c_s$ (3-34)

and hence

$$\dot{m} = G_5 a_0 \rho_0 A (X_1 + X_2 - 1)^{G_5} (X_1 - X_2) \tag{3-35}$$

Mach number

The superposition Mach number can be expressed as

$$M_s = \frac{c_s}{a_s} = \left| \frac{G_5 a_0 (X_1 - X_2)}{a_0 X_s} \right| \tag{3-36}$$

3.3.4 Friction Pressure loss and heating during wave propagation

Figure 3.5 shows a section of the mesh where the two oppositely moving waves undergo a superposition process along with friction and heat transfer.

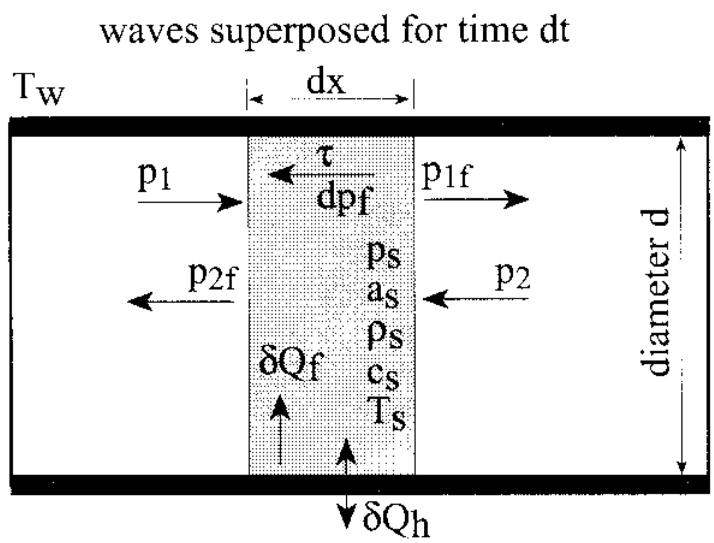


Figure 3.5 Friction loss and heat transfer [13]

The particle flow in the pipe produces two effects:

(1) Pressure loss to the wave in a direction opposite to particle motion.

(2) Work expended acts as internal heating.

Pressure loss

The shear stress at the wall can be expressed as

$$\tau = C_f \frac{\rho_s c_s^2}{2} \tag{3-37}$$

If flow is turbulent (as in most cases)

$$C_f = \frac{0.0791}{\mathrm{Re}^{0.25}} \text{ for } \mathrm{Re} \geq 4000 \text{ where } \mathrm{Re} = \frac{\rho_s d c_s}{\mu_{Ts}} \text{ [37]} \tag{3-38}$$

Coefficient of viscosity

$$\mu = 7.457 \times 10^{-6} + 4.1547 \times 10^{-8} T - 7.4793 \times 10^{-12} T^2 \ \ kg/ms \tag{3-39}$$

The variation of coefficient of viscosity of air with temperature in the range 0 to 2000 K is shown in Figure 3.6, as given by Blair [13].

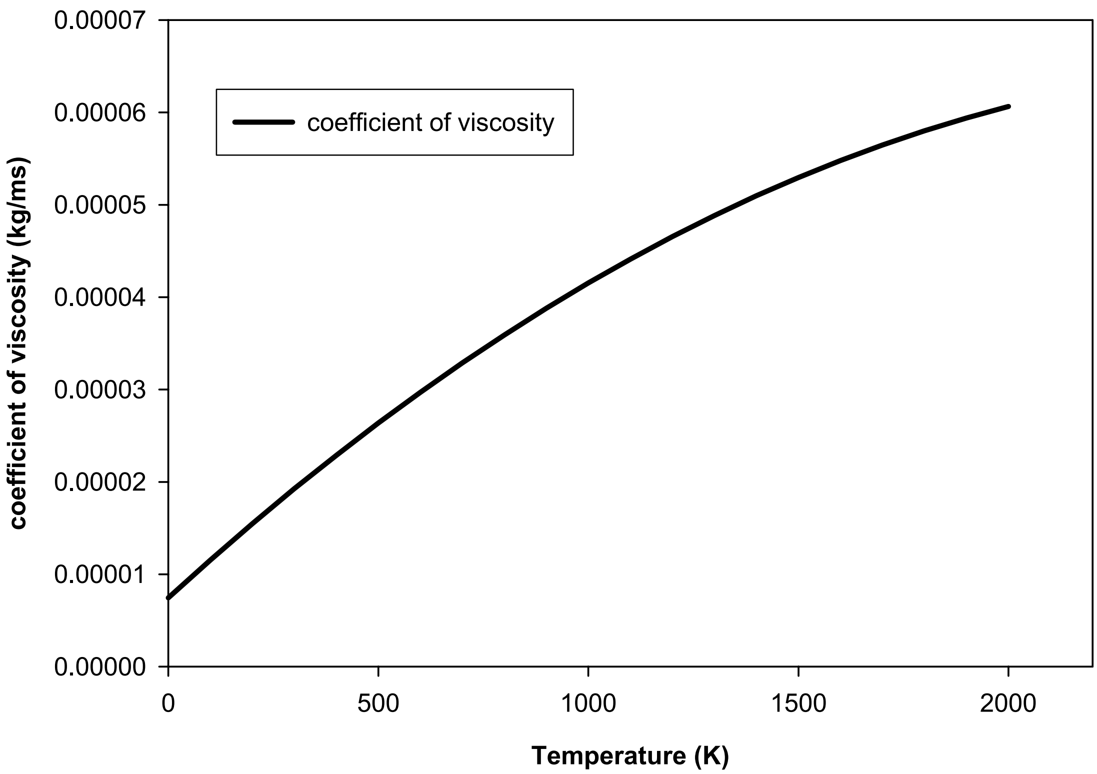


Figure 3.6 Variation of coefficient of viscosity of air with temperature [13]

For laminar flow, assuming $C_f = 0.01$ [37]

The pressure loss is given by the equation

$$dp_f = \frac{2C_f \rho_s c_s^3 dt}{d} \tag{3-40}$$

where dt is the time step of travel of the pressure wave.

The new superposition pressure of the wave after time step dt will be

$$p_{sf} = p_s \pm dp_f \tag{3-41}$$

The pressure ratios for the ongoing pressure waves can be written as

$$X_{1f} = \frac{1}{2}\left(1 + X_{sf} + \frac{c_{sf}}{G_5 a_0}\right) \text{ and } X_{2f} = 1 + X_{sf} - X_{1f} \tag{3-42}$$

The pressures then are

$$p_{1f} = p_0 X_{1f}^{G7} \text{ and } p_{2f} = p_0 X_{2f}^{G7} \tag{3-43}$$

The internal heat generated due to the shear forces can be expressed as

$$\delta Q_f = \frac{\pi d C_f \rho_s c_s^4 \, dt^2}{2} \tag{3-44}$$

External heat transfer

This section covers any external heat transfer to the pipe across the pipe wall. Convection is the main mode of heat transfer. By Reynolds analogy between friction and heat transfer, the Nusselt number can be defined.

$$Nu = \frac{C_f \operatorname{Re}}{2} \tag{3-45}$$

By definition of Nusselt number,

$$Nu = \frac{C_h d}{C_k} \qquad \therefore \quad C_h = \frac{C_k Nu}{d} = \frac{C_k C_f \operatorname{Re}}{2d} \tag{3-46}$$

The correlation of C_k with temperature for the temperature range of 300-2000 K [13] can be expressed as

$$C_k = 6.1944\times10^{-3} + 7.3814\times10^{-5} T - 1.2491\times10^{-8} T^2 \ W/mK \tag{3-47}$$

Variation of thermal conductivity of air with temperature in the range 0 to 2000 K is shown in Figure 3.7, as given by Blair [13].

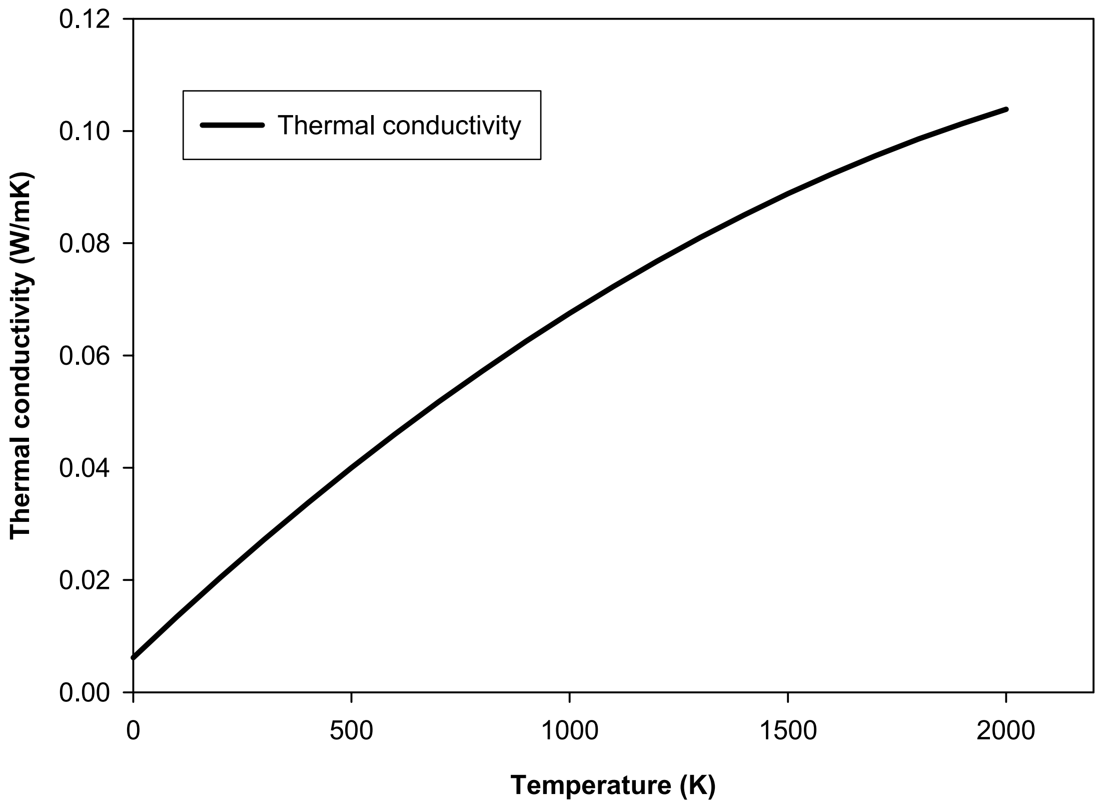


Figure 3.7 Variation of thermal conductivity of air with temperature [13]

Therefore, the external heat transfer can be expressed as

$$\delta Q_h = \pi d C_h \, dx \left(T_w - T_s\right) dt \tag{3-48}$$

The expression for total heat transfer is $\delta Q_{fh} = \delta Q_f + \delta Q_h$ (3-49)

3.3.5 Reflection of pressure waves

Reflection of pressure waves occurs as a superposition process of oppositely moving waves. Reflections occur at boundaries or interface where there is a discontinuity in gas properties, change in area, etc. This reflection causes a pressure wave to propagate in a direction opposite to the incident wave. Some of the cases where it can occur are at the outflow boundary from a cylinder, open end of a pipe, a sudden expansion, sudden contraction etc. These cases are relevant to our analysis; so are discussed in detail here. The objective of this analysis is to evaluate the parameters associated with the reflected and transmitted waves after the reflection process. This theory is incorporated into the simulation later.

3.3.5.1 Reflection at outflow from a cylinder

The thermodynamic conditions and properties at the outflow boundary of a cylinder are shown schematically below in Figure 3.8. The properties under consideration are pressure, temperature, density and particle velocity.

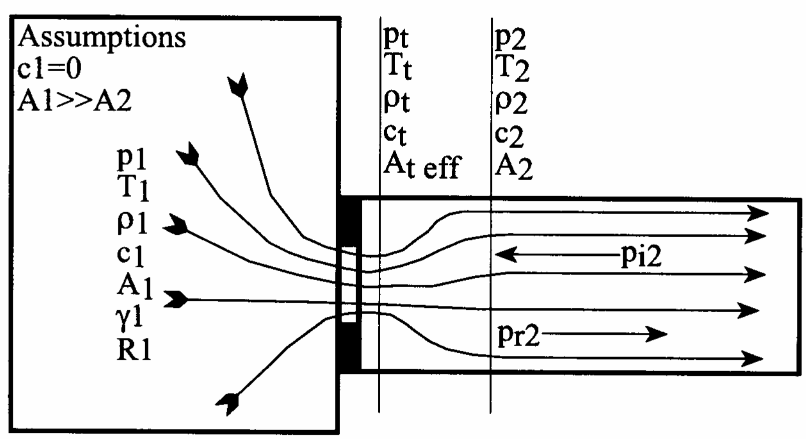
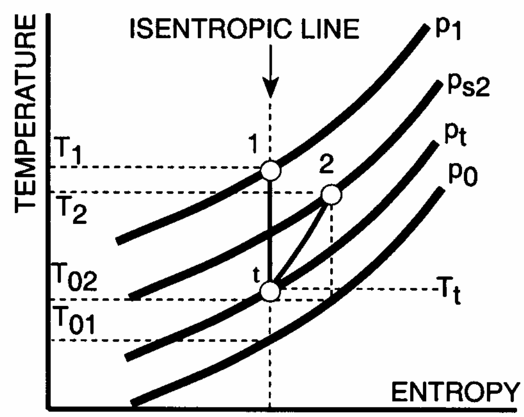
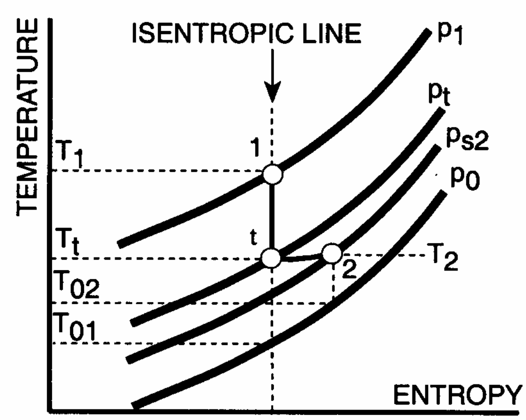


Figure 3.8 Outflow from a cylinder [13]



temperature-entropy characteristics for subsonic outflow.



temperature-entropy characteristics for sonic outflow.

Figure 3.9 Temperature-Entropy characteristics for outflow from a cylinder [13]

Figure 3.9 shows the temperature-entropy characteristics for sonic and subsonic outflow conditions. Subsonic outflow is non-isentropic and the sonic case is isentropic from the throat to the superposition station. The equations to be solved [13] are the continuity, momentum and the energy equations. These equations are presented in differential and integral form in section 3.2. The governing equations solved are summarized below. Their derivation can be found in Appendix A.

Continuity

$$\rho_{01} X_t^{G5} C_d A_t c_t - \rho_{02} \left(X_{i2} + X_{r2} - 1 \right)^{G5} A_2 G_5 a_{02} \left(X_{i2} - X_{r2} \right) = 0 \tag{3-50}$$

Since $a_{02} = \sqrt{\gamma R T_{02}} = \sqrt{\dfrac{\gamma P_0}{\rho_{02}}}$ the above equation can be modified to accommodate known variables

$$a_{02} \rho_{01} X_t^{G5} C_d A_t c_t - \gamma p_0 \left(X_{i2} + X_{r2} - 1 \right)^{G5} A_2 G_5 \left(X_{i2} - X_{r2} \right) = 0 \tag{3-51}$$

Energy equation/ First law of thermodynamics

For the flow from cylinder to superposition station 2 (1-2)

$$G_5 \left(a_{01} X_1 \right)^2 - \left[\left(G_5 a_{02} \left(X_{i2} - X_{r2} \right) \right)^2 + G_5 a_{02}^2 \left(X_{i2} + X_{r2} - 1 \right)^2 \right] = 0 \tag{3-52}$$

For the flow from cylinder to throat (1-t)

$$G_5 \left[\left(a_{01} X_1 \right)^2 - \left(a_{01} X_t \right)^2 \right] - c_t^2 = 0 \tag{3-53}$$

Momentum

$$p_0\left[X_t^{G7}-\left(X_{i2}+X_{r2}-1\right)^{G7}\right]+\left[\rho_{02}\left(X_{i2}+X_{r2}-1\right)^{G5}\times G_5 a_{02}\left(X_{i2}-X_{r2}\right)\right]\times\left[c_t-G_5 a_{02}\left(X_{i2}-X_{r2}\right)\right]=0$$

(3-54)

The solution is done by Newton-Raphson method.

The unknowns are X_{r2}, X_t, a_{02} and c_t.

3.3.5.2 Reflection at discontinuities in gas properties

Consider the general case of common gas composition i.e. the gas (air) composition is assumed invariant. The discontinuity is of infinitesimal thickness that the effect of friction is ignored. The notations used are defined in figure 3.10.

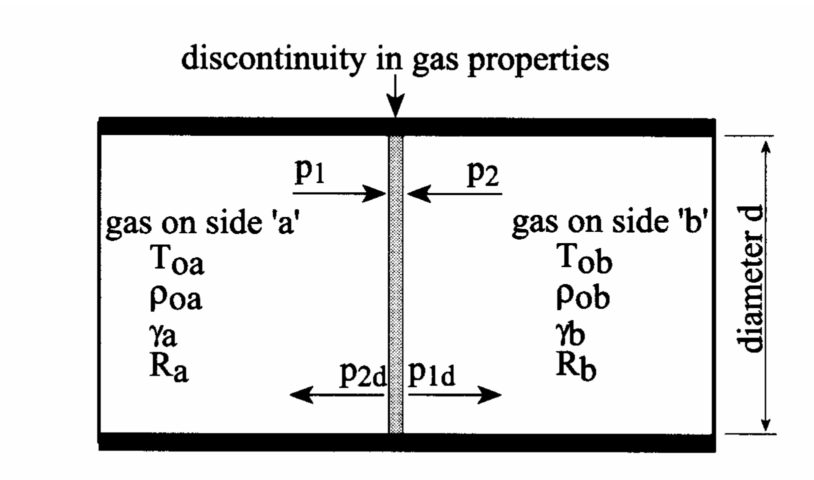


Figure 3.10 Wave reflections at a property discontinuity [13]

Applying the continuity equation across the discontinuity

$$\dot{m}_{side\ a} = \dot{m}_{side\ b} \tag{3-55}$$

The momentum equation gives

$$A\left(p_{s\ side\ a} - p_{s\ side\ b}\right) = \dot{m}_{side\ a}\, c_{s\ side\ a} - \dot{m}_{side\ b}\, c_{s\ side\ b} \tag{3-56}$$

A is the cross-sectional area of the pipe.

Combining the above two equations

$$p_{s\ side\ a} = p_{s\ side\ b} \tag{3-57}$$

$$c_{s\ side\ a} = c_{s\ side\ b} \tag{3-58}$$

Therefore the governing equations are

$$G_{5a} a_{0a} \left(X_1 - X_{2d} \right) = G_{5b} a_{0b} \left(X_{1d} - X_2 \right) \tag{3-59}$$

which reduces to

$$\frac{G_{5a} a_{0a}}{G_{5b} a_{0b}} \left(X_1 - X_{2d} \right) = \left(X_{1d} - X_2 \right) \tag{3-60}$$

and

$$\left(X_1 + X_{2d} - 1 \right)^{G7_a} = \left(X_{1d} + X_2 - 1 \right)^{G7_b} \tag{3-61}$$

The solution for this case is

$$X_{2d} = \frac{2X_2 - X_1 \left(1 - \frac{a_{0a} G_{5a}}{a_{0b} G_{5b}} \right)}{1 + \frac{a_{0a} G_{5a}}{a_{0b} G_{5b}}} \tag{3-62}$$

$$X_{1d} = X_1 + X_{2d} - X_2 \tag{3-63}$$

The reflected wave pressures are

$$p_{1d} = p_0 X_{1d}^{G7b} \tag{3-64}$$

$$p_{2d} = p_0 X_{2d}^{G7a} \tag{3-65}$$

3.3.5.3 Wave reflection at open end of a pipe

The out flow at the open end is shown in figure 3.11. In the plane of superposition at the exit, the pressure is atmospheric.

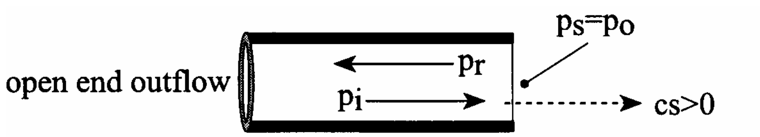


Figure 3.11 Wave reflection at open end of a pipe [13]

Here the superposition pressure is the atmospheric pressure.

$$\therefore X_s = X_i + X_r - 1 = 1 \; ; \quad X_r = 2 - X_i \tag{3-66}$$

$$p_r = p_0 (2 - X_i)^{G7} \tag{3-67}$$

$$c_s = G_5 a_0 (X_i - 1) - G_5 a_0 (X_r - 1) = 2c_i \tag{3-68}$$

3.3.5.4 Wave reflection at sudden area changes

Consider the sudden area changes in a pipe. The two possibilities are sudden expansion and sudden contraction. The summary of notations can be seen in figure 3.12.

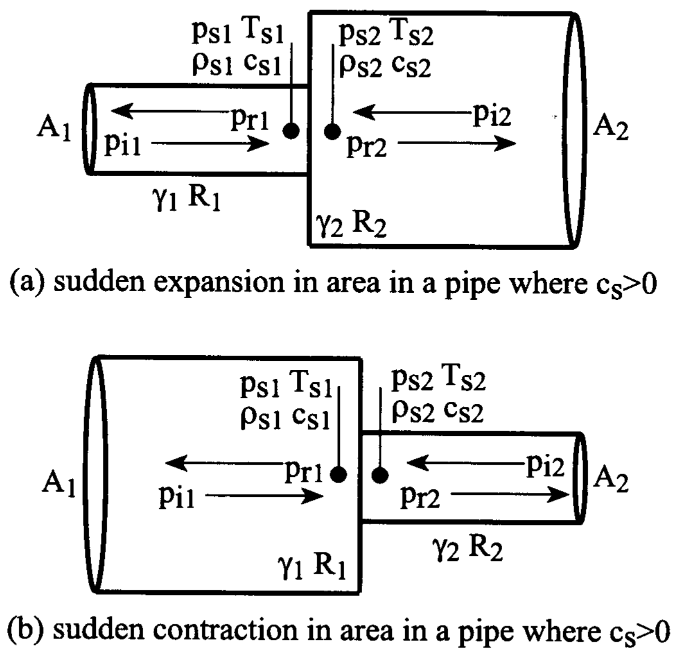


Figure 3.12 Reflections at sudden area changes [13]

The temperature-entropy characteristics and particle flow patterns can be seen in figures 3.13 and 3.14.

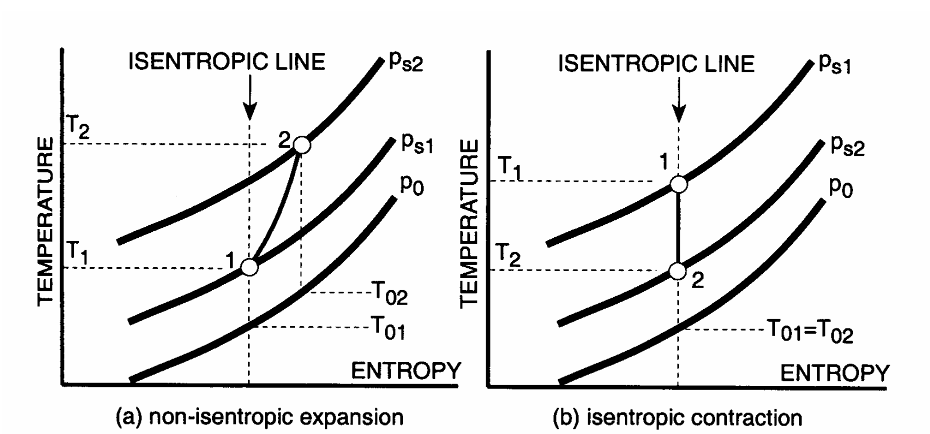


Figure 3.13 Temperature-Entropy characteristics [13]

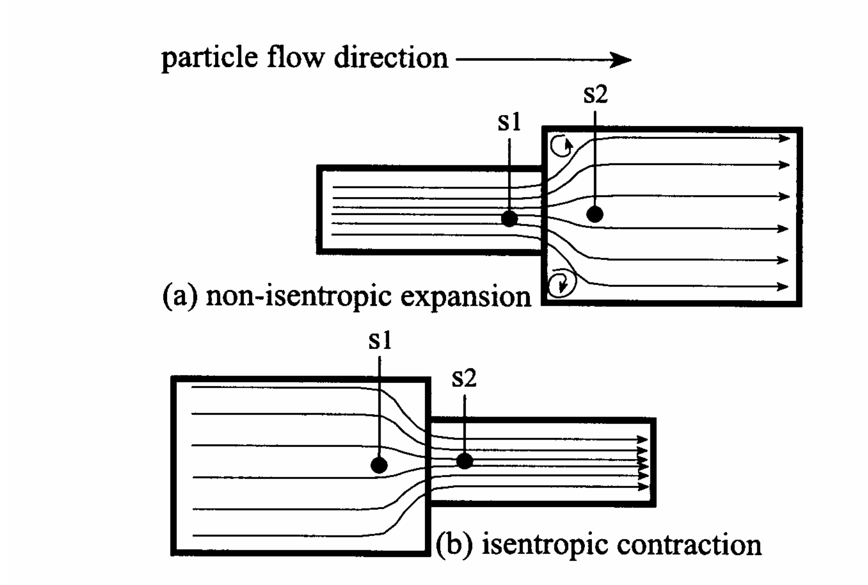


Figure 3.14 Particle flow [13]

Sudden expansion

The equations to be solved here are the continuity, momentum and the energy equations. The derivation of the governing equations in this form can be found in Appendix A.

Continuity

$$\rho_{01}\left(X_{i1}+X_{r1}-1\right)^{G5} A_1 G_5 a_{01}\left(X_{i1}-X_{r1}\right)+\rho_{02}\left(X_{i2}+X_{r2}-1\right)^{G5} A_2 G_5 a_{02}\left(X_{i2}-X_{r2}\right)=0$$

(3-69)

Momentum

$$p_0 A_2\left[\left(X_{i1}+X_{r1}-1\right)^{G7}-\left(X_{i2}+X_{r2}-1\right)^{G7}\right]+$$
$$\left[\rho_{01}\left(X_{i1}+X_{r1}-1\right)^{G5} A_1 G_5 a_{01}\left(X_{i1}-X_{r1}\right)\right]\times\left[G_5 a_{01}\left(X_{i1}-X_{r1}\right)+G_5 a_{02}\left(X_{i2}-X_{r2}\right)\right]=0$$

(3-70)

Energy (first law of Thermodynamics)

$$\left[\left(G_5 a_{01}\left(X_{i1}-X_{r1}\right)\right)^2+G_5 a_{01}^2\left(X_{i1}+X_{r1}-1\right)^2\right]-\left[\left(G_5 a_{02}\left(X_{i2}-X_{r2}\right)\right)^2+G_5 a_{02}^2\left(X_{i2}+X_{r2}-1\right)^2\right]=0$$

(3-71)

The unknowns are X_{r1}, X_{r2} and a_{02}.

These equations are solved by a Newton-Raphson and Gauss elimination method.

Benson's Approach for initial guesses [28]

The assumption for Benson's guess is that the superposition pressure at the plane of junction is the same in both pipes at the instant of superposition. This assumes an isentropic process. Nevertheless, this has proved to be a good initial guess especially where the area ratios are in the ratio

$$\frac{1}{6}<A_r<6 \qquad (3\text{-}72)$$

where $A_r=\dfrac{A_2}{A_1}$ (3-73)

This gives

$$X_{r1}=\frac{\left(1-A_r\right)X_{i1}+2X_{i2}A_r}{1+A_r} \qquad (3\text{-}74)$$

$$X_{r2}=\frac{2X_{i1}-X_{i2}\left(1-A_r\right)}{1+A_r} \qquad (3\text{-}75)$$

The superposition Mach number has to be evaluated at each time step and should not be allowed to exceed the value of unity.

$$M_{s1} = \frac{c_{s1}}{a_{s1}} = \frac{G_5 a_{01}\left(X_{i1} - X_{r1}\right)}{X_{i1} + X_{r1} - 1} \qquad (3\text{-}76)$$

$$\therefore if \; M_{s1} \geq 1 \quad M_{s1} = 1$$

Hence $X_{r1} = \dfrac{M_{s1} + X_{i1}\left(G_5 - M_{s1}\right)}{M_{s1} + G_5} = \dfrac{1 + G_4 X_{i1}}{G_6}$ (3-77)

This reduces one variable in the solution during the particular iteration.

Sudden contraction

Observing the particle flow profile in figure 3.14, the contracting flow is seen to flow smoothly from the larger to the smaller cross-section. The streamlines do not give rise to flow separation and so the flow is considered isentropic. Since there is no entropy gain, one of the unknowns – the acoustic velocity disappears from the equation and hence the number of unknowns reduces to two. The unknowns are X_{r1} and X_{r2}. The solution of two unknowns requires only two equations. So the momentum equation is ignored.

Continuity

$$\left(X_{i1}+X_{r1}-1\right)^{G5} A_1\left(X_{i1}-X_{r1}\right)+\left(X_{i2}+X_{r2}-1\right)^{G5} A_2\left(X_{i2}-X_{r2}\right)=0 \tag{3-78}$$

Energy (first law of Thermodynamics)

$$\left[\left(G_5 a_{01}\left(X_{i1}-X_{r1}\right)\right)^2+G_5 a_{01}^2\left(X_{i1}+X_{r1}-1\right)^2\right]-\left[\left(G_5 a_{02}\left(X_{i2}-X_{r2}\right)\right)^2+G_5 a_{02}^2\left(X_{i2}+X_{r2}-1\right)^2\right]=0 \tag{3-79}$$

Here $a_{01}=a_{02}$ (3-80)

The unknowns are X_{r1} and X_{r2}.

Therefore, the above equation reduces to

$$\left[G_5\left(X_{i1}-X_{r1}\right)^2+\left(X_{i1}+X_{r1}-1\right)^2\right]-\left[G_5\left(X_{i2}-X_{r2}\right)^2+\left(X_{i2}+X_{r2}-1\right)^2\right]=0 \tag{3-81}$$

These equations are solved by a Newton-Raphson and Gauss elimination method.

Initial guesses are done using a Benson's simple solution [28] approach.

The superposition Mach number has to be evaluated at each time step and cannot be allowed to exceed the value of unity.

$$M_{s2} = \frac{c_{s2}}{a_{s2}} = \frac{G_5 a_{02}\left(X_{i2} - X_{r2}\right)}{a_{02} X_{s2}} = \frac{G_5 a_{02}\left(X_{i2} - X_{r2}\right)}{X_{i2} + X_{r2} - 1} \tag{3-82}$$

$$\therefore if\ M_{s2} \geq 1 \quad M_{s2} = 1$$

Hence $X_{r2} = \dfrac{M_{s2} + X_{i2}\left(G_5 - M_{s2}\right)}{M_{s2} + G_5} = \dfrac{1 + G_4 X_{i2}}{G_6}$ (3-83)

This reduces one variable in the solution during the particular iteration.

3.4 The Computational Model

3.4.1 Assumptions and limitations of the model

Assumptions

(1) The flow is assumed to be one-dimensional i.e. the flow properties vary in only one direction - in the direction of flow.

(2) The rate of change of the cross-sectional area along the duct axis is small. Wherever a sudden change in cross-sectional area is found, the governing equations are solved using relevant numerical techniques. The governing equations are solved for a sudden expansion and contraction as discussed in section 3.3.5.4.

(3) The radius of curvature of the duct axis is very large compared to the diameter of the duct.

(4) Velocity and temperature profiles across a cross-section remain unchanged from one section to the other along the duct.

(5) Uniform properties exist across any cross section.

(6) The average pressure amplitude ratio throughout the mesh is considered the mean of the superposition pressures at the ends of the mesh. This ratio is used for calculation of particle velocity, density and all other flow parameters for the mesh. The inherent assumption is that the superposition conditions are representative of the state conditions in the mesh. The average of the superposition conditions at the left and right boundaries of the mesh is the state condition that prevails throughout the mesh.

(7) The wall temperature of the tube is assumed constant. The skin temperature of the pipe wall is assumed to be at ambient temperature. This is justified in a quasi-steady analysis where the wall loses heat to the external surroundings thereby remaining at constant temperature. This considerably simplifies the analysis at each time step.

Limitations

(1) The successful working of the simulation is dependent upon the boundary conditions (as in any CFD analysis of nozzle non-isentropic flow) and the mesh sizing. In the computational fluid dynamic analysis of compressible nozzle flows, the pressure ratio (p/p0) at the inlet and exit, mesh size and the geometry of the nozzle (the function used to describe the nozzle shape) affect the outcome of the solution [29].

(2) The expansion and contraction ratios need to be limited to maximum value of 1/6 to 6. This condition is required because the initial assumption of unknown parameters for analysis of sudden expansion and contraction using the criterion of Benson [28] are based on these limits of area ratios. This criterion is discussed in section 3.3.5.4.

(3) Testing of the code is done by comparison to test cases found in the literature [13], [34]. These test cases are analyzed and results discussed in the chapter on results and discussion. Other applications of the technique are experimentally validated as reported by Kirkpatrick et al. [32], Blair et al. [33] [34]. The mesh size independence criterion is also discussed.

Now that all the underlying theory has been explained, the actual computational model and numerical solution methodology may be discussed. This is the purpose of the next chapter. Organization of the computer program, explanation of the model and the simulation procedure will also be discussed in chapter 4.

CHAPTER IV

NUMERICAL SOLUTION TO THE GOVERNING EQUATIONS

4.1 The Computational Model

4.1.1 Simulation Procedure

The Simulation procedure consists of the following steps

(1) Setting up the mesh

(2) The selection of a time step 'dt' for the simulation.

(3) Analysis of wave transmission through mesh space J.

(4) Mass and energy transport through mesh J.

(5) Effect of friction and area change.

(6) Effect of discontinuity in gas properties.

(7) Effect of geometrical discontinuities.

4.1.2 Steps in simulation

The steps involved in the procedure for the simulation are explained below.

4.1.2.1 Setting up the mesh and meshing details

A typical meshing structure is depicted in figure 4.1. The pressure amplitude ratios at the left end of the mesh is denoted by the subscript 'R' and 'L' and at the right end by 'R1' and 'L1'. These ratios are modified during a particular time step by friction, heat transfer, and reflections due to area change and difference in gas properties. All other parameters are derived from the representative pressure amplitude ratio for the mesh.

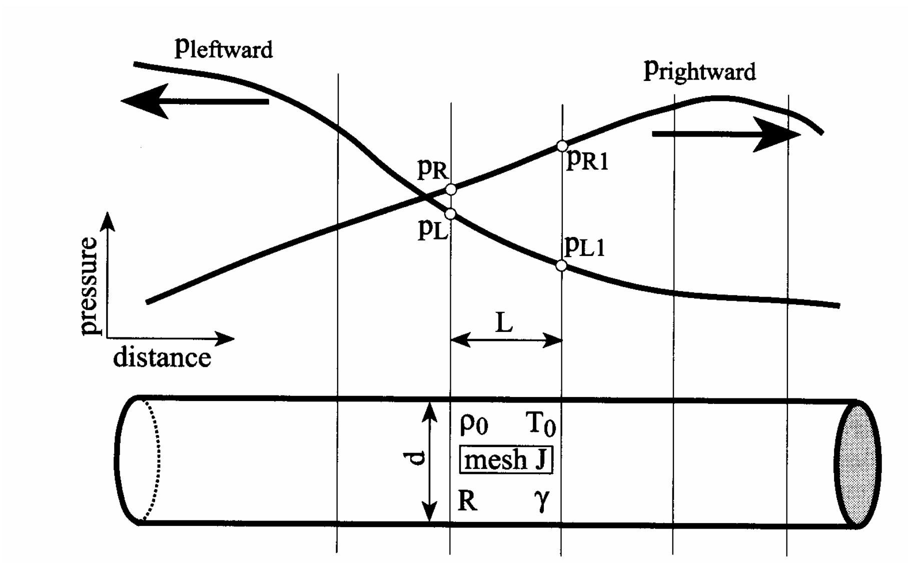


Figure 4.1 Meshing details for duct [13]

<u>Assumption</u>

The average pressure throughout the mesh is considered to be the mean of the superposition pressures at the ends of the mesh. Therefore the pressure amplitude ratio can be written as

$$X_J = \frac{\left(X_R + X_L - 1\right) + \left(X_{R1} + X_{L1} - 1\right)}{2} \tag{4-1}$$

The average pressure, density, temperature, acoustic velocity and mass in the mesh can be found from the following formulae. These equations follow from the isentropic relations for a perfect gas.

$$X = \left(\frac{p}{p_0}\right)^{\frac{\gamma-1}{2\gamma}} = \left(\frac{p}{p_0}\right)^{G_{17}} \therefore \left(\frac{p}{p_0}\right) = X^{G_7} \tag{4-2}$$

$$p_J = p_0 X_J^{G7} \tag{4-3}$$

$$\left(\frac{\rho}{\rho_0}\right) = \left(\frac{p}{p_0}\right)^{\frac{1}{\gamma}} = X^{\frac{2}{\gamma-1}} = X^{G_5} \tag{4-4}$$

$$\rho_J = \rho_0 X_J^{G5} \tag{4-5}$$

$$\left(\frac{T}{T_0}\right) = \left(\frac{p}{p_0}\right)^{\frac{\gamma-1}{\gamma}} = X^2 \tag{4-6}$$

$$T_J = T_0 X_J^2 \tag{4-7}$$

$$\frac{a}{a_0} = \sqrt{\frac{T}{T_0}} = \left(\frac{p}{p_0}\right)^{\frac{\gamma-1}{2\gamma}} = X \tag{4-8}$$

$$a_J = a_0 X_J \tag{4-9}$$

$$m_J = \rho_J V_J \tag{4-10}$$

where

$$V_J = \frac{\pi}{4} d^2 L \tag{4-11}$$

The main task of the simulation is to find the values of X_R, X_L, X_{R1}, X_{L1} at each time step for all meshes. This depends on the unsteady flow analysis, wave reflections and the thermodynamics of gas flow in each mesh during the time step ‘dt’.

All other parameters can be expressed in terms of the pressure amplitude ratio X_J.

4.1.2.2 Selection of time step

The time step dt is calculated by sweeping across each mesh space and finding the fastest propagation velocity in the system. It is assumed that there are linear variations of pressure and velocity within the mesh length L.

$$dt = \frac{L}{\alpha_{sL}} \quad or \quad dt = \frac{L}{\alpha_{sL1}} \quad or \quad dt = \frac{L}{\alpha_{sR}} \quad or \quad dt = \frac{L}{\alpha_{sR1}} \tag{4-12}$$

Sweeping all meshes

$$dt_{\min imum} = 0.99 \times \left| \frac{L}{\alpha_{s\ fastest\ in\ J}} \right|_{J=1}^{J=total} \tag{4-13}$$

The factor 0.99 is an additional arithmetic insurance to avoid violating numerical stability thus satisfying the Courant, Friedrich and Lewy stability criterion [38] that ensures that all subsequent iterative procedures for all mesh spaces are by interpolation and not by extrapolation.

4.1.2.3 Wave transmission during time step

Figure 4.2 summarizes the wave propagation during time step dt. For the time step dt calculated from the previous section, it is obvious that not all waves will traverse the mesh length L in the time dt. This section attempts to add a correction factor to accurately predict the pressure amplitude ratios at either end of the mesh after time dt.

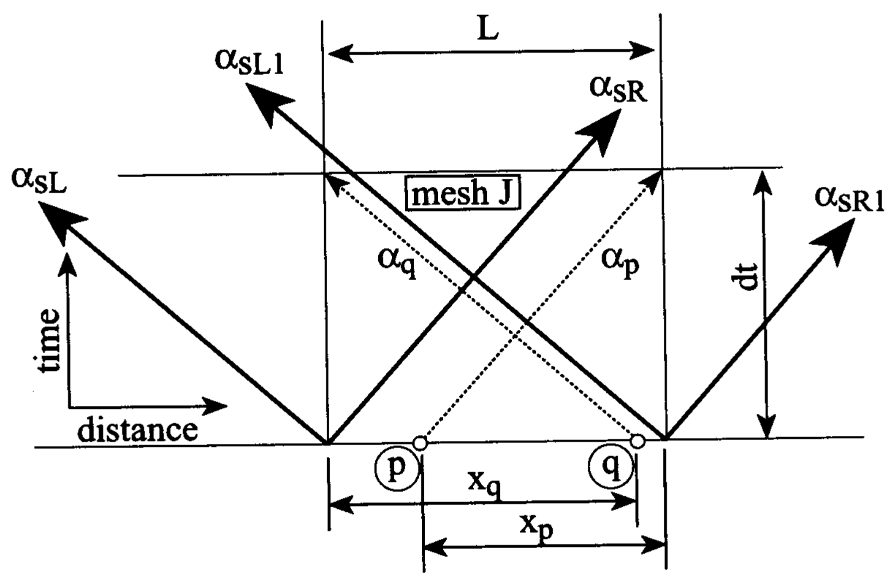


Figure 4.2 Propagation of pressure waves in Mesh J [13]

The pressure amplitude ratio calculation is solved by employing the continuity equation. However the method of solution assumes a linear interpolation approach. The assumption is that between any two meshes there is a linear variation of wave pressure, wave superposition pressure and superposition propagation velocity.

Consider the mesh of length L as shown in figure 4.2. The calculation of time step in section 4.1.4.1 was based on the fastest propagation velocity. Therefore, for all other meshes the wave is not fast enough to reach the end of the mesh in the duration of the time step. Consider such a mesh. A wave traveling towards the right having a pressure amplitude ratio of X_p will just reach the right end of the mesh in time dt. This value of X_p is linearly related to its physical position in the mesh and is a linear function of X_R and X_{R1}. This is the value of X that will be able to reach the right end of the mesh. Location of this wave at start of time step is p. Similarly, for a leftward moving wave, the location is q and the pressure amplitude ratio is X_q. X_q is a linear function of X_L and X_{L1}. At the end of the time step values of X_p and X_q will represent the values of the rightward and leftward pressure wave amplitude ratios at end of time step dt. At the start of the time step the values of X_L, X_{L1}, X_R and X_{R1} are required to be known.

The expressions for X_p and X_q are listed below. Derivation of these equations can be found in Appendix A.

$$X_p = \frac{1 + D + F_L + F_L C}{G_4 \left(F_R F_L - 1 \right)} \tag{4-14}$$

$$X_q = \frac{1 + C + F_R + F_R D}{G_4 \left(F_R F_L - 1 \right)} \tag{4-15}$$

where

$$F_R = \frac{G_6 + \frac{1}{A}}{G_4} \tag{4-16}$$

$$F_L = \frac{G_6 + \frac{1}{B}}{G_4} \tag{4-17}$$

$$A = E\left(X_{R1} - X_R\right) \tag{4-18}$$

$$B = E\left(X_L - X_{L1}\right) \tag{4-19}$$

$$C = \frac{X_{R1}}{A} \tag{4-20}$$

$$D = \frac{X_L}{B} \tag{4-21}$$

$$E = \frac{a_0\, dt}{L} \tag{4-22}$$

The new values of X_{R1} and X_L are given by

$$X_{R1new} = X_p + \{\pm friction\, effects \left(\pm\right) area\, change\, effects\} \tag{4-23}$$

$$X_{Lnew} = X_q \; + \{\pm friction\, effects \left(\pm\right) area\, change\, effects\} \tag{4-24}$$

The mean of superposition conditions at either end of the mesh are assumed to be characteristic of the mesh space. These values are the parameters measured by a transducer and constitute the representative variables in the mesh space. These superposition values are for evaluating the properties of the gas in the mesh in the time step. These values can be found from the following formulae

Pressure amplitude ratio $X_s = X_p + X_q - 1$ (4-25)

Pressure $p_s = p_0 X_s^{G7}$ (4-26)

Density $\rho_s = \rho_0 X_s^{G5}$ (4-27)

Temperature $T_s = T_0 X_s^2$ (4-28)

Particle velocity $c_s = G_5 a_0 \left(X_p - X_q \right)$ (4-29)

Singularities during simulation

(1) If $X_R = X_{R1}$ then

$$X_p = X_{R1} \tag{4-30}$$

$$X_q = \frac{1 + D + G_4 X_p}{G_6 + \frac{1}{B}} \tag{4-31}$$

The above condition is a possibility during a start up situation for all meshes except the first mesh.

(2) If $X_L = X_{L1}$ then

$$X_q = X_{L1} \tag{4-32}$$

$$X_q = \frac{1 + C + G_4 X_q}{G_6 + \frac{1}{A}} \tag{4-33}$$

The above condition is a possibility during a start up situation for all meshes.

(3) If $X_R = X_{R1}$ and $X_L = X_{L1}$ then

$$X_q = X_{L1} \tag{4-34}$$

$$X_p = X_{R1} \tag{4-35}$$

The above condition is true during a start up situation for all meshes except the first mesh.

4.1.2.4 Mass and Energy transport along the duct during a time step

A rough guideline for the mesh size is in the range of 10-25 mm, which by calculation [13] is found to be feasible. This value is deduced from the assumptions used in design of internal combustion engines. The assumption is that the time step dt should be equivalent to a crank angle of 1 - 2°.

$$L = 1000 \times \alpha \times \frac{d\theta}{360} \times \frac{60}{N} = 1000 \frac{\alpha \times d\theta}{6N} \tag{4-36}$$

For an engine running at 3000 – 5000 rpm and using γ = 1.3 and R = 300 J/kgK this turns out to be in the range 10 – 25 mm. For our research problem however this is a rough guideline. For large geometries involving long lengths, larger mesh sizes may be used. The test case 2 comparisons in chapter 5 use a mesh length of 35 - 50 mm. The analysis of mass and energy transport through the duct employs a first law of thermodynamics analysis. Here the heat transfer occurring in the mesh to the gas by means of friction heating will be considered.

In our analysis only friction heating is present as the air bleed is compressed air from the engine at reference temperature T_0. There are four 'events' likely to occur in the mesh during the time step dt. They are summarized in figure 4.3 and figure 4.4.

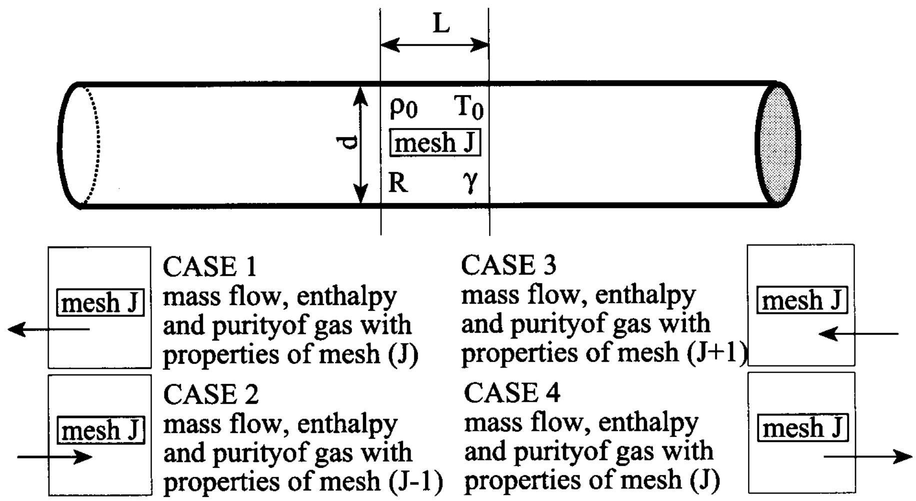


Figure 4.3 Mass and energy transport at mesh J during time step dt [13]

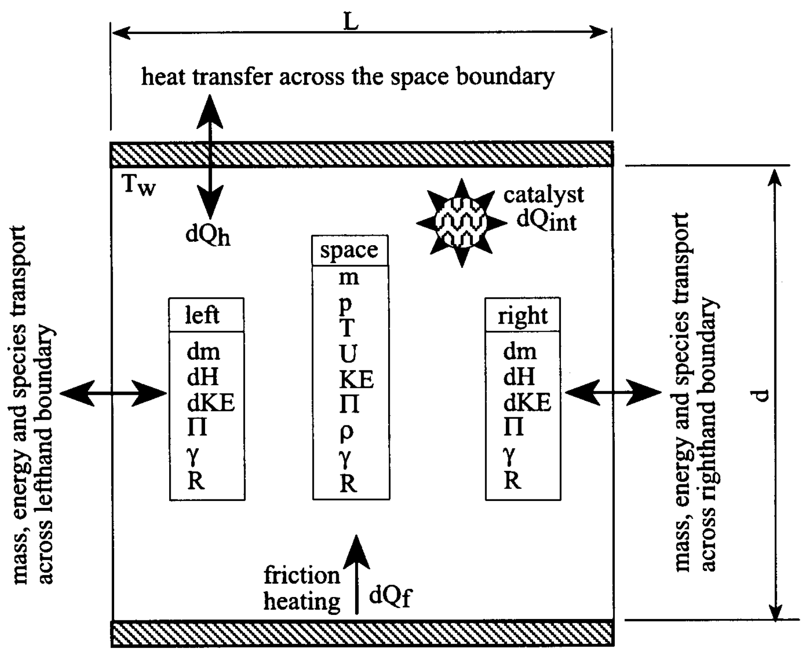


Figure 4.4 Heat and mass transfer across Mesh J [13]

The parameters that are important in our analysis for each case are summarized below. The subscript 'a' refers to the value after time step and 'b' for the value before time step. The left side of the mesh is referred to as the 'in' side and the right end as the 'out' side.

Case 1

Pressure $${}_J X_{in} = {}_J X_R + {}_J X_L - 1 \quad (4\text{-}37)$$

Particle velocity $${}_J c_{in} = {}_J G_5 \, {}_J a_0 \left({}_J X_R - {}_J X_L \right) \quad (4\text{-}38)$$

Density $${}_J \rho_{in} = {}_J \rho_0 \, {}_J X_{in}^{{}_J G_5} \quad (4\text{-}39)$$

Specific Enthalpy $${}_J dh_{in} = {}_J C_P \, {}_b T_J + \frac{{}_J c_{in}^2}{2} \quad (4\text{-}40)$$

Mass flow increment $${}_J dm_{in} = {}_J \rho_{in} \, A_J \, {}_J c_{in} \, dt \quad (4\text{-}41)$$

Enthalpy increment $${}_J dH_{in} = {}_J dh_{in} \, {}_J dm_{in} \quad (4\text{-}42)$$

Air flow increment $${}_J d\prod_{in} = {}_J \prod \, {}_J dm_{in} \quad (4\text{-}43)$$

Case 2

Pressure $${}_J X_{in} = {}_J X_R + {}_J X_L - 1 \quad (4\text{-}44)$$

Particle velocity $${}_J c_{in} = {}_{J-1} G_5 \, {}_{J-1} a_0 \left({}_J X_R - {}_J X_L \right) \quad (4\text{-}45)$$

Density $${}_J \rho_{in} = {}_{J-1} \rho_0 \, {}_J X_{in}^{{}_{J-1} G_5} \quad (4\text{-}46)$$

Specific Enthalpy $${}_J dh_{in} = {}_{J-1} C_P \, {}_b T_{J-1} + \frac{{}_J c_{in}^2}{2} \quad (4\text{-}47)$$

Mass flow increment $${}_J dm_{in} = {}_J \rho_{in} \, A_J \, {}_J c_{in} \, dt \quad (4\text{-}48)$$

Enthalpy increment $${}_J dH_{in} = {}_J dh_{in} \, {}_J dm_{in} \quad (4\text{-}49)$$

Air flow increment $${}_J d\prod_{in} = {}_{J-1} \prod \, {}_J dm_{in} \quad (4\text{-}50)$$

Case 3

Pressure	$_J X_{out} = {}_J X_{R1} + {}_J X_{L1} - 1$	(4-51)
Particle velocity	$_J c_{out} = {}_{J+1} G_5 \; {}_{J+1} a_0 \left({}_J X_{R1} - {}_J X_{L1} \right)$	(4-52)
Density	$_J \rho_{out} = {}_{J+1} \rho_0 \; {}_J X_{out}^{\;{}_{J+1} G_5}$	(4-53)
Specific Enthalpy	$_J dh_{out} = {}_{J+1} C_P \; {}_b T_{J+1} + \frac{{}_J c_{out}^2}{2}$	(4-54)
Mass flow increment	$_J dm_{out} = {}_J \rho_{out} \; A_J \; {}_J c_{out} \; dt$	(4-55)
Enthalpy increment	$_J dH_{out} = {}_J dh_{out} \; {}_J dm_{out}$	(4-56)
Air flow increment	$_J d \prod_{out} = {}_{J+1} \prod \; {}_J dm_{out}$	(4-57)

Case 4

Pressure	$_J X_{out} = {}_J X_{R1} + {}_J X_{L1} - 1$	(4-58)
Particle velocity	$_J c_{out} = {}_J G_5 \; {}_J a_0 \left({}_J X_{R1} - {}_J X_{L1} \right)$	(4-59)
Density	$_J \rho_{out} = {}_J \rho_0 \; {}_J X_{out}^{\;{}_J G_5}$	(4-60)
Specific Enthalpy	$_J dh_{out} = {}_J C_P \; {}_b T_J + \frac{{}_J c_{out}^2}{2}$	(4-61)
Mass flow increment	$_J dm_{out} = {}_J \rho_{out} \; A_J \; {}_J c_{out} \; dt$	(4-62)
Enthalpy increment	$_J dH_{out} = {}_J dh_{out} \; {}_J dm_{out}$	(4-63)
Air flow increment	$_J d \prod_{out} = {}_J \prod \; {}_J dm_{out}$	(4-64)

For the meshes at the end of the pipe, the required information for the above equations is deduced from the boundary conditions of the flow, which have been applied at the right or left end of a mesh space. For the left end the pressure boundary condition is the applied pulse peak pressure and for the right end it is the ambient pressure. As far as the temperatures are concerned, the left end temperature is calculated from the pressure ratio and for the right end it is once again the ambient condition.

The thermodynamics of the mesh space during the time step involves applying the continuity and energy equations.

<u>Continuity</u>

The new mass in the mesh space ${}_J m_a$ after the time step can be derived from the continuity equation

$$ {}_J m_a = {}_J m_b + {}_J dm_{in} - {}_J dm_{out} \tag{4-65} $$

<u>Energy equation</u>

The first law of thermodynamics can be applied to the mesh space

Heat transfer + energy in = change of system state + energy out + work done

$$ \left(dQ_{\text{int}} + dQ_f + dQ_h\right)_J + {}_J dH_{in} = dU_J + {}_J dH_{out} + P_J dV_J \tag{4-66} $$

where

$$ dQ_f = \left| \frac{C_f A_{sJ}\, \rho_s\, c_s^3\, dt}{2} \right| \tag{4-67} $$

$$ dQ_h = C_h A_{sJ} \left(T_w - T_s\right) dt \tag{4-68} $$

and $A_{sJ} = \pi dL$ (4-69)

The work term $P_J dV_J$ is zero. The internal generation term δQ_{int} is known. This term arises due to the use of a catalyst used that can cause exothermic reactions releasing heat energy in the exhaust systems of engines. For our case there is no heat generation inside the pipe system. Therefore, this term is zero.

The internal energy term can be expressed as

$$dU_J = {}_a\left[{}_a m_J \left({}_a u_J + \frac{{}_a c_J^2}{2} \right) \right]_J - {}_b\left[{}_b m_J \left({}_b u_J + \frac{{}_b c_J^2}{2} \right) \right]_J \tag{4-70}$$

The system particle velocity is assumed to be the mean of that at either end of the mesh.

$${}_a c_J = \frac{1}{2}\left(\frac{{}_J c_{in}^2}{2} + \frac{{}_J c_{out}^2}{2} \right) \tag{4-71}$$

The energy equation becomes

$$dU_J = {}_J C_v \left({}_a m_J \, {}_a T_J - {}_b m_J \, {}_b T_J \right) + \frac{1}{2}\left({}_a m_J \, {}_a c_J^2 - {}_b m_J \, {}_b c_J^2 \right) \tag{4-72}$$

where ${}_J C_v$ is the specific heat at constant volume of the gas in mesh J.

The above equation can be solved for the system temperature ${}_a T_J$ after the time step.

The new reference temperature ${}_a T_0$ can be found for the mesh space as

$${}_a T_0 = \frac{{}_a T_J}{{}_a X_J^2} \tag{4-73}$$

The other reference conditions after a time step are

$${}_a a_0 = \sqrt{{}_a \gamma_J \, {}_a R_J \, {}_a T_0} \qquad {}_a \rho_0 = \frac{p_0}{{}_a R_J \, {}_a T_0} \tag{4-74}$$

4.1.2.5 Reflection of waves at discontinuities after a time step

In this section only the sudden expansion and contraction conditions will be considered as only it is relevant to the configuration of the system as discussed later in the results section. See figure 4.5 for the mesh set up. The equations to be solved and the solution methodology have already been discussed in the "Wave reflection at sudden area changes" section. These have to be performed after the solution of the pressure amplitude ratios have been obtained at either side of the discontinuity (X_{p1} and X_{q2} in figure 4.5) for the time step.

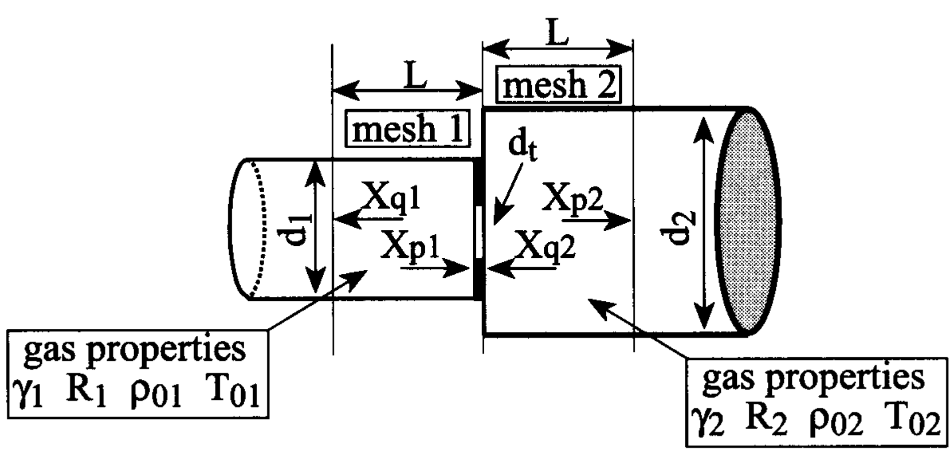


Figure 4.5 Two adjacent meshes in a restricted pipe [13]

4.1.2.6 Reflection of waves at ends of the pipe after a time step

At the end of the time step, for the end meshes the pressure amplitude ratios are calculated after calculating the reflections at the ends. This depends on the connection at the ends. In the model the left end is connected to a cylinder or a duct and the right end is connected to the atmosphere. Therefore at the left end the theory for the outflow from a cylinder applies and at the right end the theory for the open end to the atmosphere applies.

At the end of each time step the modeling method must calculate the values of ${}_1X_R$ and ${}_{end}X_{L1}$ from the above calculation, where the subscripts '1' and 'end' refer to the left end mesh and right end mesh respectively.

4.2 A Computer program for performing the simulation

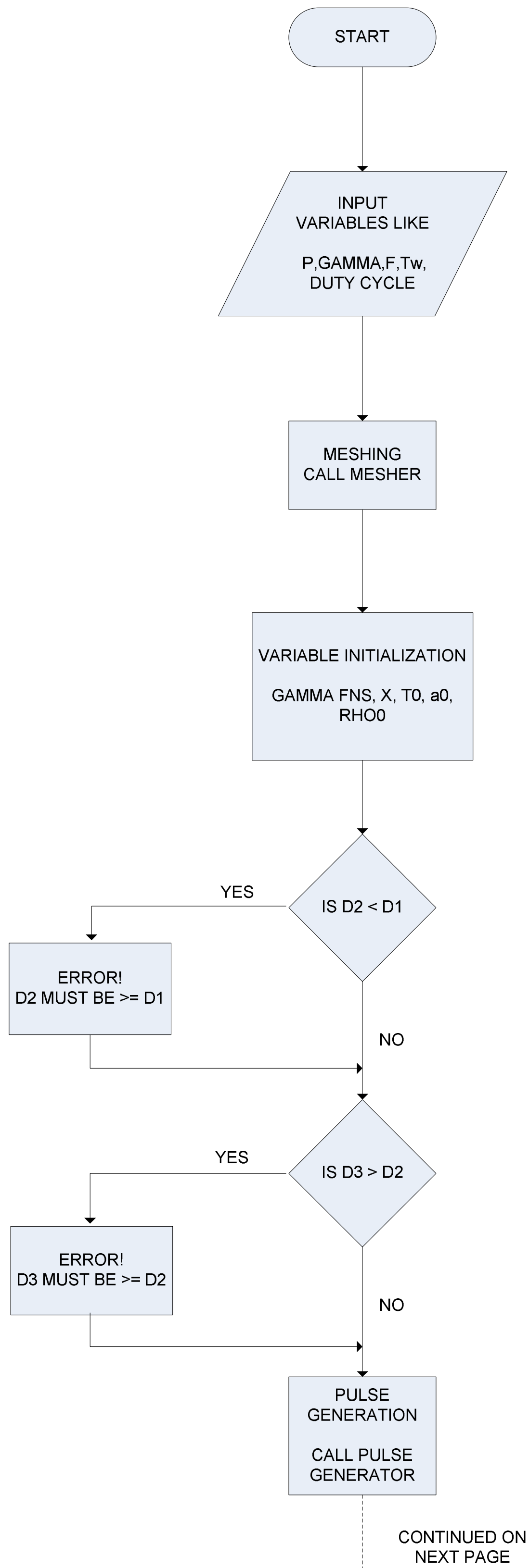
Now that the theory has been presented, the next step is to look at the development of a program to execute the theory on the digital computer. This is what is discussed in this section.

Evaluation of the theory for checking the suitability of the application is the first step in the development process. A commercially available software – Virtual four stroke SAE edition® software used for simulation of four stroke engines was employed for this application. The objective was to evaluate the effectiveness of this simulation for application in prediction of system dynamic parameters. The results and configuration modeled is described in the results and discussion chapter.

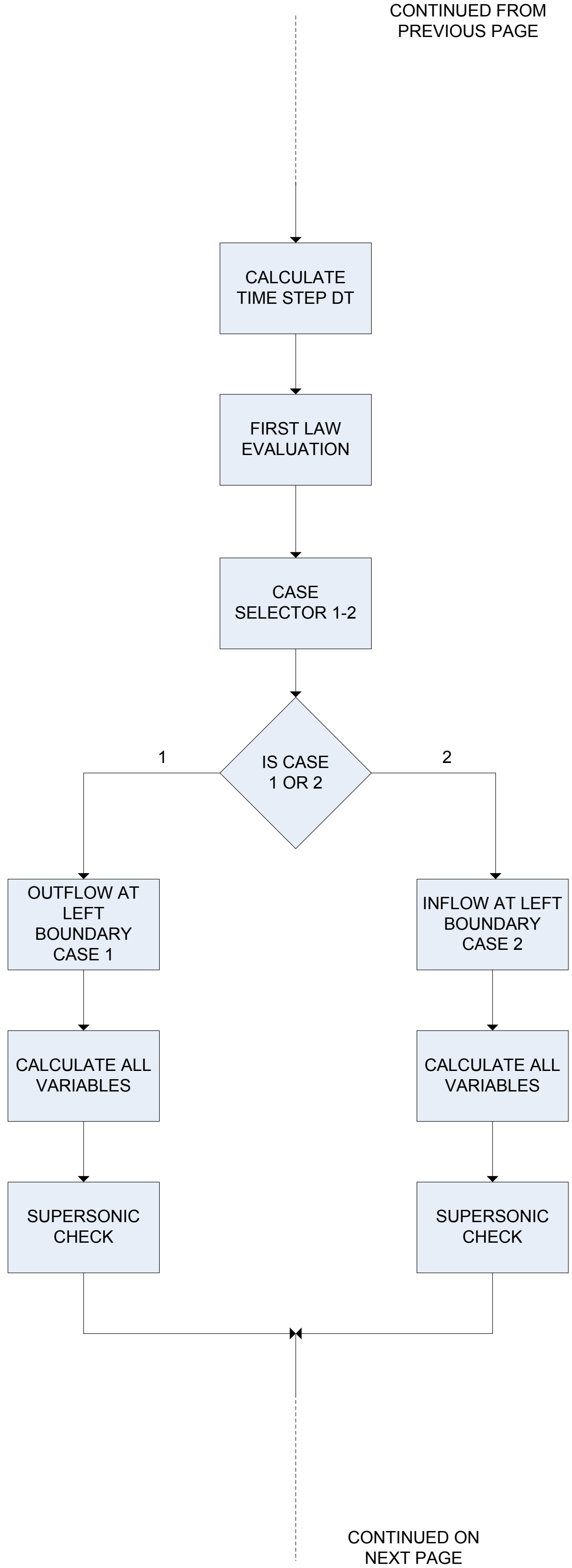
4.2.1 Flow chart for Simulation

An algorithm is developed and a flow chart for proceeding to the coding is shown in figure 4.6. The program statement flow can be seen from the flow chart. This helps in defining the structure and constitution of the program. The program is written in modular fashion involving several subroutines. This helps in making testing of the program easier. Details of the program and explanation of the subroutines is described in the next section.

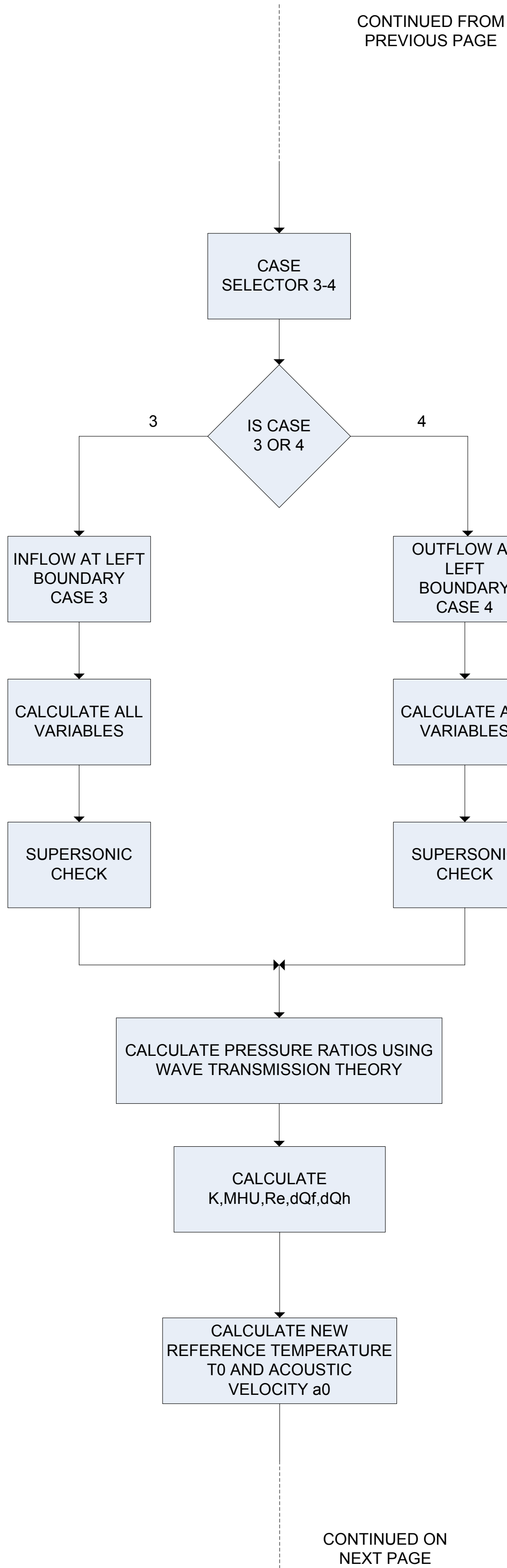
FLOW CHART FOR THE PROGRAM

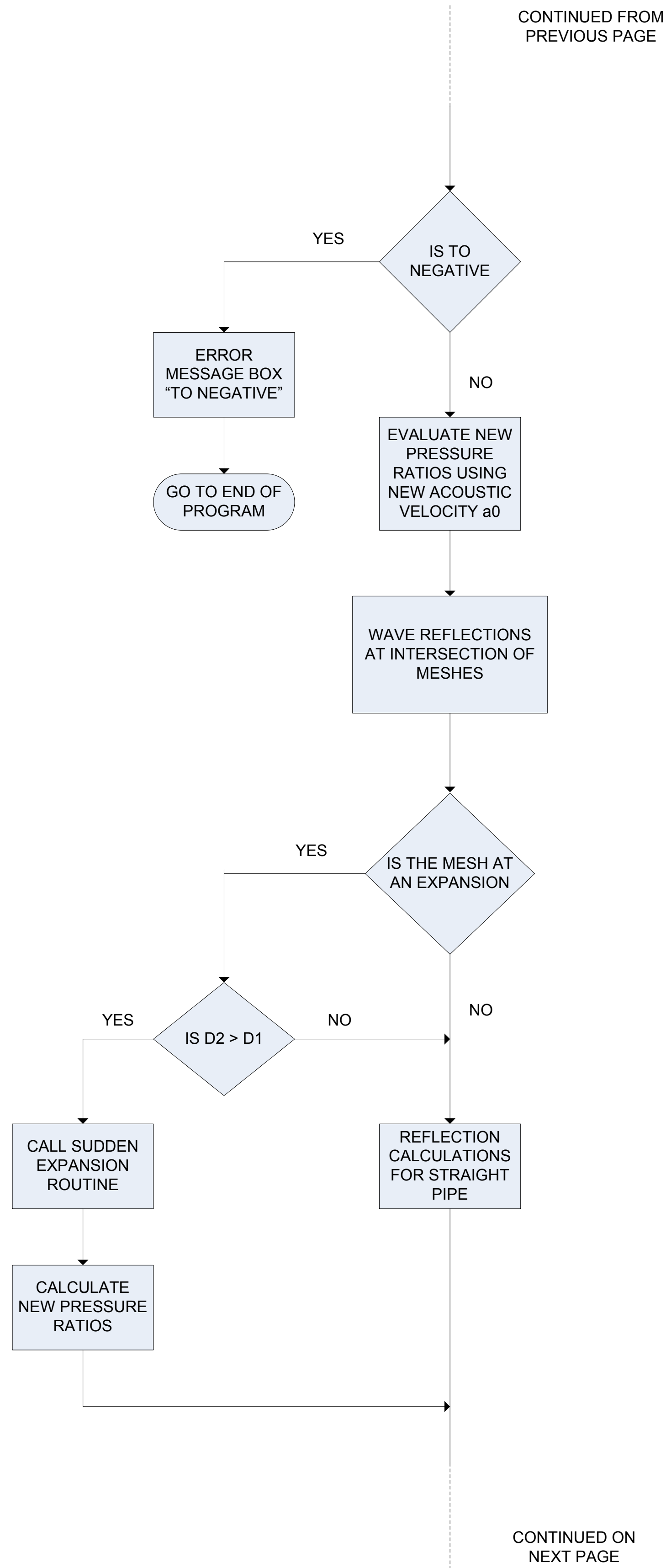


CONTINUED FROM
PREVIOUS PAGE



CONTINUED ON
NEXT PAGE


CONTINUED FROM PREVIOUS PAGE
CASE SELECTOR 3-4
IS CASE 3 OR 4
3
4
INFLOW AT LEFT BOUNDARY CASE 3
OUTFLOW AT LEFT BOUNDARY CASE 4
CALCULATE ALL VARIABLES
CALCULATE ALL VARIABLES
SUPERSONIC CHECK
SUPERSONIC CHECK
CALCULATE PRESSURE RATIOS USING WAVE TRANSMISSION THEORY
CALCULATE K,MHU,Re,dQf,dQh
CALCULATE NEW REFERENCE TEMPERATURE T0 AND ACOUSTIC VELOCITY a0
CONTINUED ON NEXT PAGE


CONTINUED FROM PREVIOUS PAGE
IS TO NEGATIVE
YES
ERROR MESSAGE BOX "TO NEGATIVE"
GO TO END OF PROGRAM
NO
EVALUATE NEW PRESSURE RATIOS USING NEW ACOUSTIC VELOCITY a0
WAVE REFLECTIONS AT INTERSECTION OF MESHES
IS THE MESH AT AN EXPANSION
YES
NO
IS D2 > D1
YES
NO
CALL SUDDEN EXPANSION ROUTINE
CALCULATE NEW PRESSURE RATIOS
REFLECTION CALCULATIONS FOR STRAIGHT PIPE
CONTINUED ON NEXT PAGE

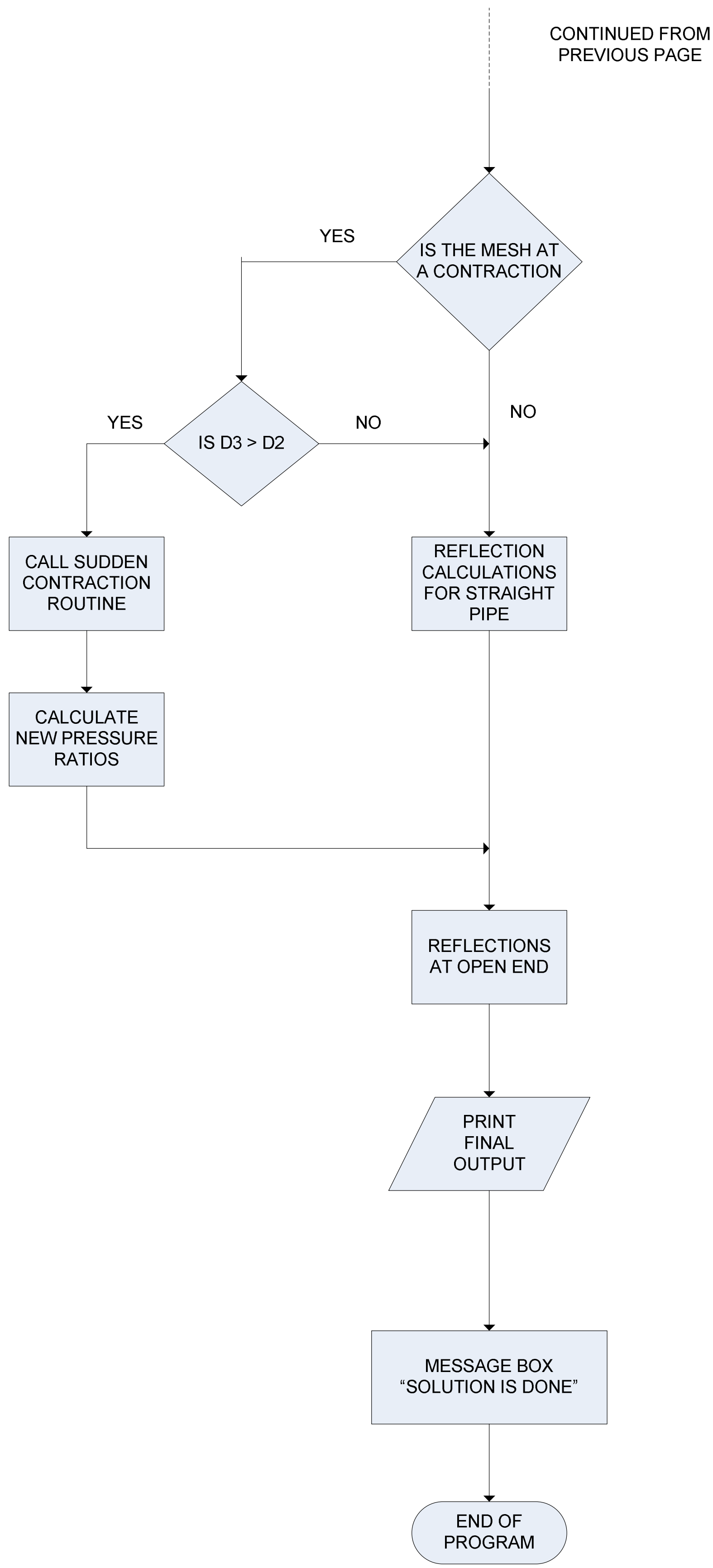


Figure 4.6 Flow chart for simulation

4.2.2 Development of flow simulation code

The program consists of several subroutines. The full listing of the code can be found in Appendix B. A brief explanation follows. The subroutines are

(1) input_read

(2) simulate

(3) mesher

(4) pulse_generator

(5) supersonic_check

(6) case_selector_1_2

(7) case_selector_3_4

(8) sudden_expansion

(9) sudden_contraction

(10) gauss_expn

(11) pivot_expn

(12) par_expn

(13) fun_expn

(14) der_expn

(15) gauss_contrn

(16) pivot_contrn

(17) par_contrn

(18) fun_contrn

(19) der_contrn

(20) clearcells

input_read

This subroutine reads all input provided by the user like inlet pressure, gas constant, wall temperature of pipe etc. These values are stored in variables for further processing by other subroutines. This subroutine also initializes several variables like reference pressure and temperature, specific heats etc. The functions of gamma are also evaluated here.

simulate

This routine is the heart of the simulation and is the driver routine. The whole simulation procedure is done here. The read routine is first called. Then the pulse generator is called which assigns the inlet boundary condition according to the duty cycle. Calculation of time step follows and then the first law of thermodynamics analysis is done. In the first law section the following subroutines are called: supersonic_check, case_selector_1_2 and case_selector_3_4. The descriptions of these routines follow later. After the first law calculations are done, the new reference temperature is evaluated and the wave transmission is redone to give the final pressure ratios at the end of the particular time step. The sudden expansion or sudden contraction routines are called if the model contains these.

mesher

This routine creates the mesh structure from the given inputs of segment lengths. Here the objective is to get a uniform mesh length through the pipe or keep the mesh length values as equal as possible while maintaining the general-purpose nature of the routine.

pulse_generator

This sub routine defines the inlet boundary condition depending on the application time of the pulse (T_p) calculated from the duty cycle and the frequency. If the aggregate time of simulation exceeds the T_p value then the inlet pressure ratio is set to 1. Otherwise it is retained at the value of X_{inlet}.

supersonic_check

This routine calculates the particle Mach number and checks for supersonic condition. If found to be supersonic, it modifies the opposite moving pressure wave amplitudes for a shock condition and outputs the new amplitudes of pressure ratios.

case_selector_1_2

This subroutine selects the case of inflow or outflow at the left boundary of all meshes depending on the magnitude of difference in left and right pressure ratios.

case_selector_3_4

This subroutine selects the case of inflow or outflow at the right boundary of all meshes depending on the magnitude of difference in left and right pressure ratios.

sudden_expansion

This subroutine solves the non-isentropic sudden expansion problem and outputs the reflected pressure ratios and also the acoustic velocity of the first mesh at the expansion.

This is achieved by solving the continuity, momentum and the first law equations. A Newton-Raphson methodology is used to solve the equations.

<u>sudden_contraction</u>

This subroutine solves the isentropic sudden contraction problem and outputs the reflected pressure ratios. This is achieved by solving the continuity, momentum and the first law equations. A Newton-Raphson methodology is used to solve the equations.

<u>gauss_expn</u>, <u>pivot_expn</u>, <u>par_expn</u>, <u>fun_expn</u>, <u>der_expn</u>

These routines are used in the Newton-Raphson solver used in the sudden expansion routine. These perform the Gaussian elimination, pivot checking, partial differential evaluation, function evaluation and derivative evaluation respectively.

<u>gauss_contrn</u>, <u>pivot_contrn</u>, <u>par_contrn</u>, <u>fun_contrn</u>, <u>der_contrn</u>

These routines are used in the Newton-Raphson solver used in the sudden contraction routine. These perform the Gaussian elimination, pivot checking, partial differential evaluation, function evaluation and derivative evaluation respectively.

<u>clearcells</u>

This routine does the task of clearing the output data. This subroutine makes it easy for the user to delete all output before running a new simulation case. The user can click on “clear cells” button to delete all output data.

4.2.3 Working of the program

The simulate subroutine acts as the driver routine. The input read routine is first called to read in all input. Then the mesher routine is called to create the mesh. The boundary pressure ratios for each mesh are defined. Variable initializations are done here. At the start of the time loop, the pulse generator routine is called which assigns the pulse to the XR(1) variable according to the aggregate time value (aggtime). The time step value "dt" is evaluated next.

The simulation proceeds to the first law of thermodynamics evaluation where the case selector routines are called to decide on outflow or inflow condition at each mesh boundary. Every time a particle velocity is evaluated, a supersonic check is done using the supersonic_check routine. The thermal conductivity, viscosity coefficient, flow Reynolds number and heat transfer are evaluated. The new reference temperature is obtained at the end of first law evaluation. The wave transmission calculations are redone to calculate the modified pressure ratios at the end of the time step. The sudden expansion or sudden contraction routines are called at the appropriate meshes if needed. At the end of the time step, the flow parameters are evaluated. The simulation then proceeds to the next time step. Once the aggregate time equals the total time, the time loop exits and the program ends.

CHAPTER V

RESULTS AND DISCUSSION

5.1 Overview

The objective of this chapter is to discuss the results from the simulation. Results from a test case employed to determine the feasibility of the code for the intended problem are presented. This involves simulation using an academic version of a commercial code for design and simulation of four stroke engines. This is called the Virtual 4 Stroke® software. This software was developed by Dr. G.P.Blair at the Queens University of Belfast and marketed by SAE (Society of Automotive Engineers, PA). After reviewing the results from this simulation, the actual flow simulation code was written adapting the one-dimensional theory of internal combustion engine design to pulsed blowing systems. Two test cases found in the literature associated with simulation codes for the design of internal combustion engines and ventilation duct systems are discussed. The various simulation parameters, the working, and the physics of the simulation are also discussed. The significance of the simulation input parameters with respect to stability of the simulation is also outlined. Further, a test case is analyzed with different mesh sizes to judge the independence of mesh size on the solution.

5.2 Results from Virtual 4 Stroke® software

The preliminary approach adopted to checking the feasibility of obtaining the solution was to solve the problem using commercially available software. This was done using the Virtual 4 Stroke® software used for simulation of internal combustion engines. This software was developed by Dr. G.P. Blair at the Queens University of Belfast and marketed by SAE (Society of Automotive Engineers, PA). The code used was an academic version, hence has inherent limitations, for example the wall temperature and initial temperature of the gas could not be changed.

The attempt was to simulate an engine to achieve the required flow conditions at the exhaust port. This is achieved by varying the engine parameters like speed, valve overlap period etc. A few important parameters are engine cylinder parameters, intake, exhaust valve seat diameter and valve open and close crank angles. Some of the important design parameters for the engine can be found in table 5-1.

Engine details

1	ClosedCycle.CombustionEfficiency	0.85
2	ClosedCycle.IgnitionDelay	13
3	ClosedCycle.IgnitionDuration	44
4	ClosedCycle.IgnitionTiming	-27
5	ClosedCycle.TrappedAirFuelRatio	12
6	ClosedCycle.WiebeA	6.02
7	ClosedCycle.WiebeM	1.64
8	ClosedCycle.IsBurnByUser	0
9	ClosedCycle.IsDebug	0
10	ClosedCycle.IsSynch	0
11	ConnectingRod.Diameter	0
12	ConnectingRod.Length	0.148
13	Piston.CompressionHeight	0.04
14	Piston.Height	0.08
15	Piston.InitialTemperature	300
16	Bore	0.088
17	Stroke	0.082
18	FrictionFactor	350
19	FrictionConstant	100000
20	SquishClearance	0.00125
21	ClearanceVolume	0.00003788
22	HeadSurfaceFactor	1.5
23	InitialGasPresFactor	4
24	InitialGasTemp	927
25	WallTemp	150
26	HeadTemp	300
27	HeadType	4 Stroke 2 Valve

Table 5-1. Engine design parameters for the software test case

Design parameters for the intake and exhaust valves are shown in tables 5-2 and 5-3 respectively. A detailed description of all input parameters can be found in Appendix C.

Intake valve details

1	OuterSeatDiameter	0.0502
2	InnerSeatDiameter	0.0482
3	SeatAngle	45
4	StemDiameter	0.0079
5	PortDiameter	0.0472
6	ManifoldDiameter	0.0381
7	ValveOpen	305
8	ValveClose	616
9	RampUpPeriod	40
10	RampUpRatio	0.2
11	RampDownPeriod	40
12	RampDownRatio	0.2
13	MaxLift	0.012

Table 5-2. Intake valve design parameters for the software test case

Exhaust valve details

1	OuterSeatDiameter	0.0413
2	InnerSeatDiameter	0.0393
3	SeatAngle	45
4	StemDiameter	0.0111
5	PortDiameter	0.0385
6	ManifoldDiameter	0.0413
7	ValveOpen	195
8	ValveClose	355
9	RampUpPeriod	40
10	RampUpRatio	0.2
11	RampDownPeriod	40
12	RampDownRatio	0.2
13	MaxLift	0.01
14	Count	1
15	InterValveClearance	0.005

Table 5-3. Exhaust valve design parameters for the software test case

The plots shown are the pressure and velocity results for particular locations for a particular exhaust piping configuration. The configuration modeled for the software test case is shown in figure 5.1.

Figure 5.1 Pipe configuration for software test case

The pressure boundary condition at the left end is achieved by assigning relevant engine parameters. Some of these parameters are RPM, ignition delay, duration and timing; valve overlap period in terms of crank angle and seat diameters etc. A trial and error approach is used to get matching results. The results were encouraging enough to use the method in a flow simulation code with provisions for using required pressures, geometry and initial and boundary conditions.

The test runs were conducted for several different RPM corresponding to the pulse frequencies. The assumption is that the pulse is ejected for a period of time during which the exhaust valve is open. This assumption is proven valid from the results.

The time period and rpm of the engine can be related by the following equation,

$$dt = \frac{d\theta}{360} \times \frac{60}{N} \tag{5-1}$$

where

$d\theta$ is the crank angle corresponding to time dt and N is the rpm of the engine.

For the test case, the exhaust valve opens at 195° and closes at 355°. So the angle for which the exhaust valve is open is 160°.

Substituting $d\theta = 160$,

$$dt = \frac{26.6}{N} \text{ seconds} \tag{5-2}$$

The frequency of the pulse is the inverse of the time period, therefore

$$f = \frac{N}{26.6} \tag{5-3}$$

For $N = 3000$, $f = 112\,Hz$

The results above are shown for 3000 RPM, which correspond to a pulse frequency of 112 Hz. Results for other frequencies can be found in Appendix C. The pressure time history for the above case is shown in figure 5.2.

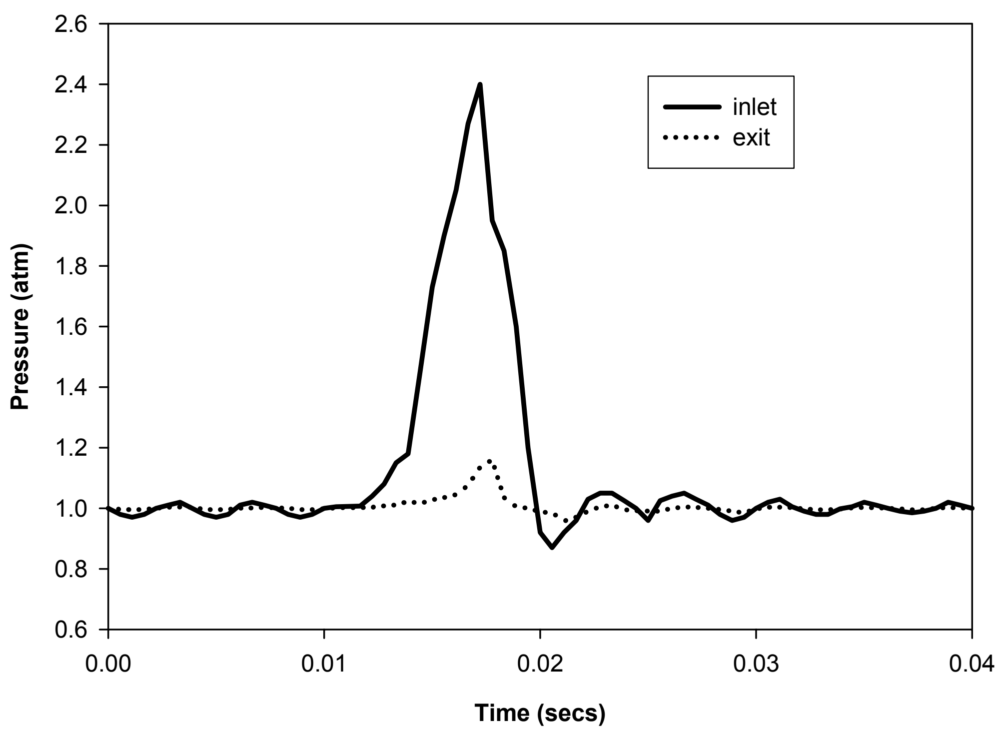


Figure 5.2 Pressure plot for software test case

Figure 5.2 represents the pressure fluctuation with respect to time for a full 720° crank rotation. The plot is shown for two locations – the inlet and the exit of the pipe section. These are the pressure values recorded by a transducer at the above locations. The simulations of the unsteady flow are done for several engine cycles. A periodic steady solution is obtained after several cycles. Figure 5.2 is the plot of such a solution. The peak pressure corresponds to the time period when the exhaust valve is open. The exhaust valve opens at approximately 0.01 seconds and closes nearly at 0.02 seconds. This is equivalent to 160° of crank rotation. The experimental measurement of this pressure history with a transducer is difficult as the high velocity of the ensuing pulse damages the filament (wire) of the transducer as reported by Jones et al. [6]. The plot at the exit shows that the pressure pulse has been attenuated by approximately 50%. This is attributed to the friction and heat transfer effects as the wave travels through the pipe.

The velocity time history plot for the above case is shown in figure 5.3. The plot at the inlet of the pipe section shows a peak between 0.01 and 0.02 seconds indicating that the exhaust valve is open during this period, discharging a pressure pulse of high velocity. The velocities are high due to the high temperatures in the engine cylinder. The velocity increases towards the outlet of the pipe section and is higher at the exit than the inlet. The velocity is increased by nearly 40%. This is in line with compressible flow theory in which the pressure decreases and velocity increases for a subsonic flow.

It can be seen that there is a pressure attenuation and velocity amplification in the unsteady compressible flow through the pipe section. The simulation provides an accurate way of predicting the unsteady pressure and velocity at any particular location in the pipe system. This measurement is difficult to conduct due to the difficulties mentioned above.

These are the principal output variables for our problem. This time history of pressure and velocity can be used as input to the aerodynamic analysis for lift, drag and other computations.

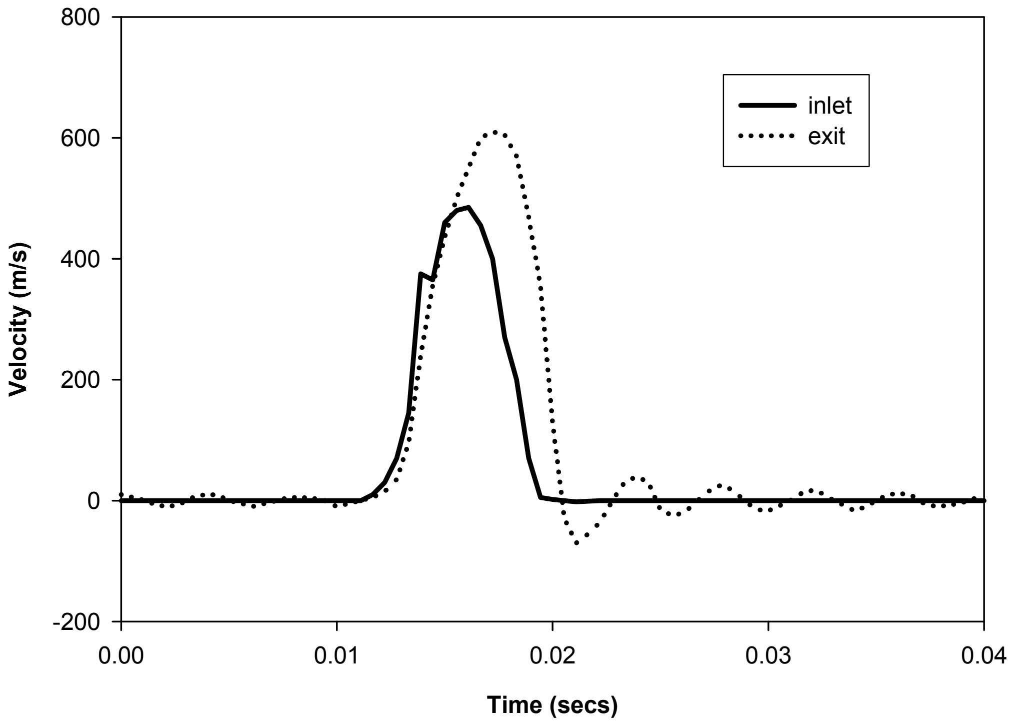


Figure 5.3 Velocity plot for software test case

5.3 Simulation using Flow Simulation code

After reviewing the results from the simulation using the Virtual 4 Stroke® software, a general-purpose flow simulation code was written. Results derived from this code tailored to the needs of the research problem are discussed in this section. This code can simulate a straight pipe and a pipe with one expansion and one contraction. However due to the nature of the problem in hand, the stability of the solution and accuracy of results are strongly dependent on input parameters, the mesh size and the configuration itself. For example, very high pressure at the left end of the pipe can cause very strong expansion waves to be induced in the direction opposite to flow direction thereby causing negative reference temperatures during the course of the simulation. This gives an erroneous result. This occurs due to the instability of the simulation for the particular time step and propagates through the solution domain. An analysis of stability problems that could be encountered is explained in section 5.4. The stability and accuracy of simulation of isentropic nozzle flows is described in [29]. The relevant details are summarized in section 5.4. The notations used for input to the program are shown in figure 5.4.

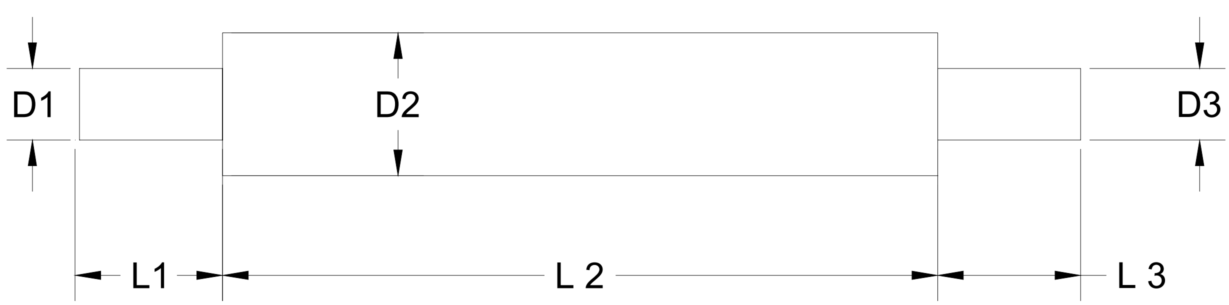


Figure 5.4 Notations used for input to the program

Two test case configurations found in the literature [34] used to validate the GPB modeling method are also used for comparisons to this flow simulation code.

5.3.1 Test case 1

A representative configuration modeled which also is the test case 1 [13] is shown in figure 5.4. This test case is used for validation of the GPB modeling method in the literature [34]. The entire methodology is experimentally validated as described in [32], [33] and [34]. The input parameters used for this test case in the literature are shown in table 5-4 and for the flow simulation code in table 5-5.

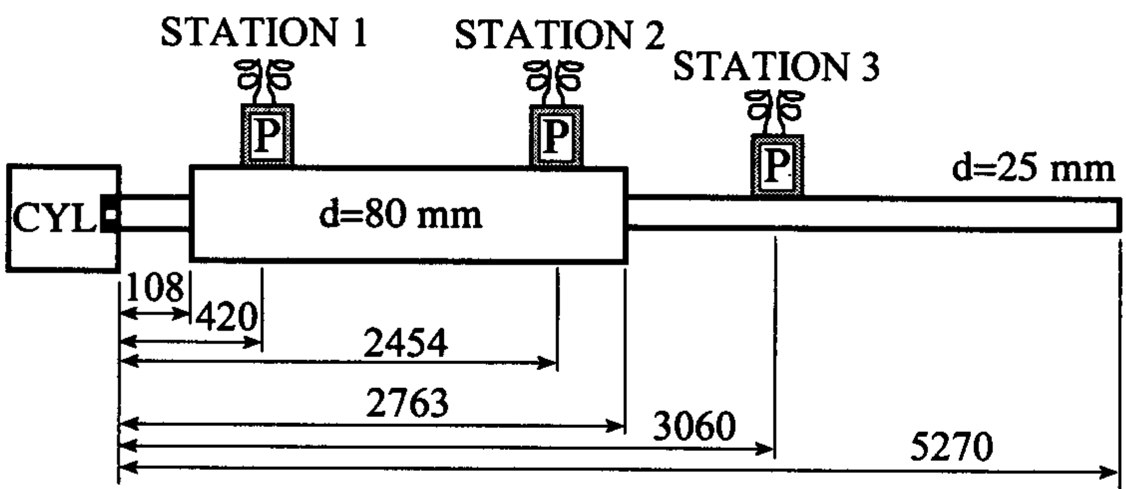


Figure 5.5 Configuration for test case 1 [13]

Simulation parameters in literature for test case

PARAMETER	VALUE	UNIT	DESCRIPTION
Pinlet	1.50	bar	pressure in cylinder
medium	air	-------	-------
speed	3000.00	RPM	Engine RPM
Reference temperature	293.00	K	Reference temperature
Port opening	0.008	secs	time during which exhaust port is open during a cycle

Table 5-4. Input parameters for test case 1 in literature [13] [34]

Simulation parameters for flow simulation code

PARAMETER	VALUE	UNIT	DESCRIPTION
Pinlet	150000	Pa	pressure at inlet of duct
γ	1.4	-------	ratio of specific heat capacities
R	287	J/kgK	Characterestic gas constant
frequency - f	100	Hz	frequency (Hz)
wall temperature - Tw	300	K	Temperature of the duct wall
Duty cycle	0.25	-------	fraction of T when Xp is present
Ambient reference temperature	293	K	Initial (undisturbed) temperature
Ambient reference pressure	101325	Pa	Initial (undisturbed) pressure
Exit pressure	101325	Pa	pressure at exit of pipe

Geometric details for flow simulation code

PARAMETER	VALUE	UNIT	DESCRIPTION
L1	108.00	mm	length of segment 1
D1	25.000	mm	diameter of segment 1
L2	2655.0000	mm	length of segment 2
D2	80.000	mm	diameter of segment 2
L3	2507.00	mm	length of segment 3
D3	25.000	mm	diameter of segment 3

Table 5-5. Input parameters for test case 1 for flow simulation code

Discussion of results

Figure 5.9 shows the pressure amplitude plot for the transducer locations.

The pulsed pressure wave introduced at the left boundary has an initial pressure amplitude ratio (X) of 1.057. The short length of 108 mm makes the pressure wave encounter a sudden expansion and this reduces the pressure amplitude ratio to a peak value of 1.01 about 0.002-0.003 seconds. The pressure wave leaves station 1 at 0.0048 seconds and the pressure amplitude ratio value drops to 1. The wave travels to the right to station 2 and reaches the peak at 0.009 seconds. The peak pressure here is higher than station 1, because the sudden contraction sends a reflected compression wave to the left, which superposes with the rightward moving compression wave and results in a superposed condition at station 2 resulting in a higher pressure amplitude ratio. At station 3, the rise in pressure is attributed to the sudden contraction to a smaller pipe from 80 mm to 25mm. For the time period of the simulation corresponding to the input frequency, the pressure wave has not reached the end of the pipe, so undisturbed conditions exist in the end mesh. It can be seen from the results that the modeling method provides a sufficient explanation of the physics of the transmission of the wave through the pipe.

Figures 5.9 - 5.11 show plots of various parameters with respect to time at various transducer locations in the geometry. The results showed the same trend as the results in [13]. Figures 5.6 – 5.8 show results from the simulation in [13]. In the engine test case, the results are averaged over the entire engine cycle after reaching a periodic steady condition. In our study, the simulation is performed only during the period corresponding to the frequency of the pulse emitted. Hence, the time period is less than for the engine simulation case. The entry pressure boundary condition into the pipe is fixed as the input to the pipe comes from the output of the actuator generating the pulse. The calculations associated with the thermodynamics of the cylinder in the engine test case are absent in our test case.

Comparing the results shown in figures 5.6 - 5.8 and 5.9, it is seen that for the time period of 0.01 seconds, the peak pressures at station 2 and 3 are higher than the peak value at station 1. The ratio of peak pressures to one another is approximately equal for both engine and flow simulation cases.

This code has inherent limitations and does not solve the problem for all geometry and initial and boundary conditions. The peculiarity of similar problems applied to nozzle flows is discussed in detail in [29]. The solution to such problems is dependent on boundary conditions, nozzle geometry and other factors. This is further discussed in section 5.4. The user may perform numerical experiments by varying the mesh size, boundary conditions and input parameters etc to achieve an optimal solution.

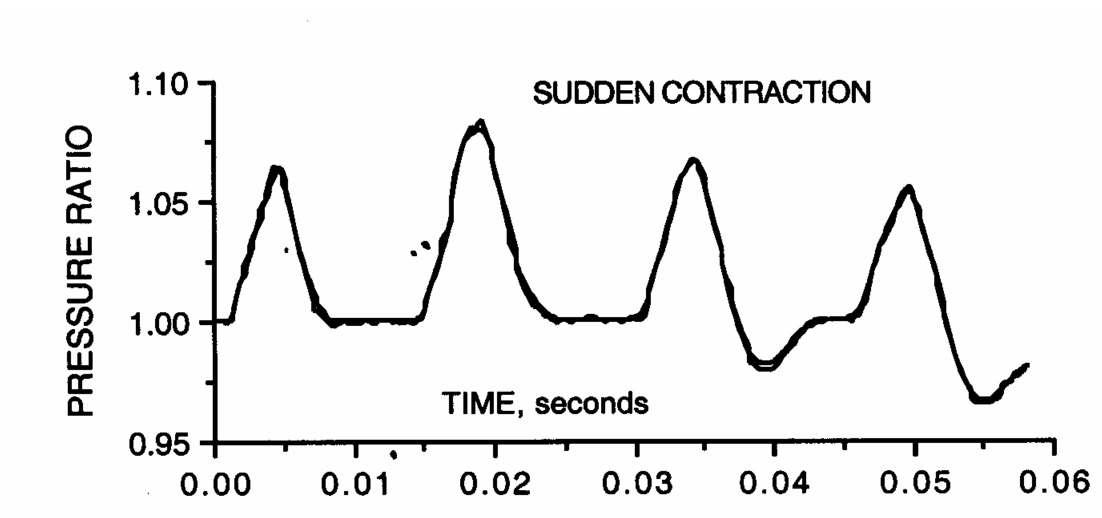


Figure 5.6 Pressure amplitude ratio at station 1 from engine simulation test case 1 in literature [13]

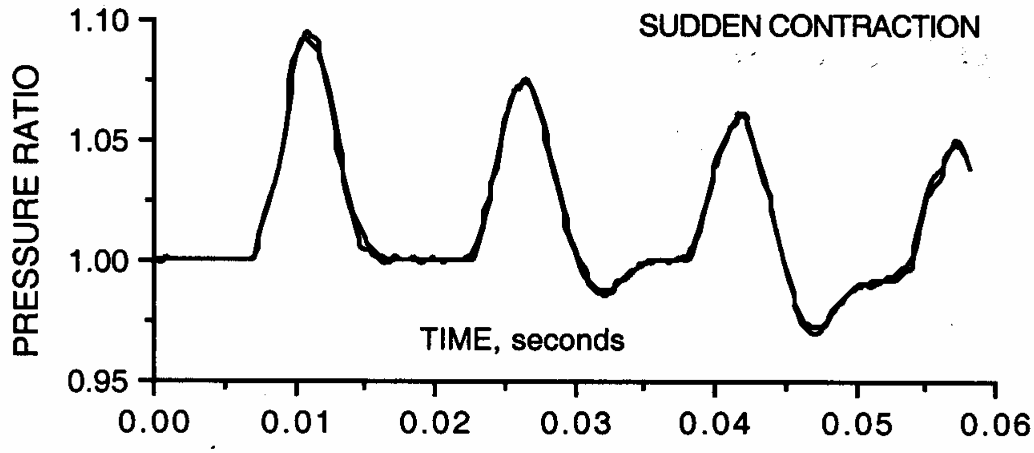


Figure 5.7 Pressure amplitude ratio at station 2 from engine simulation test case 1 in literature [13]

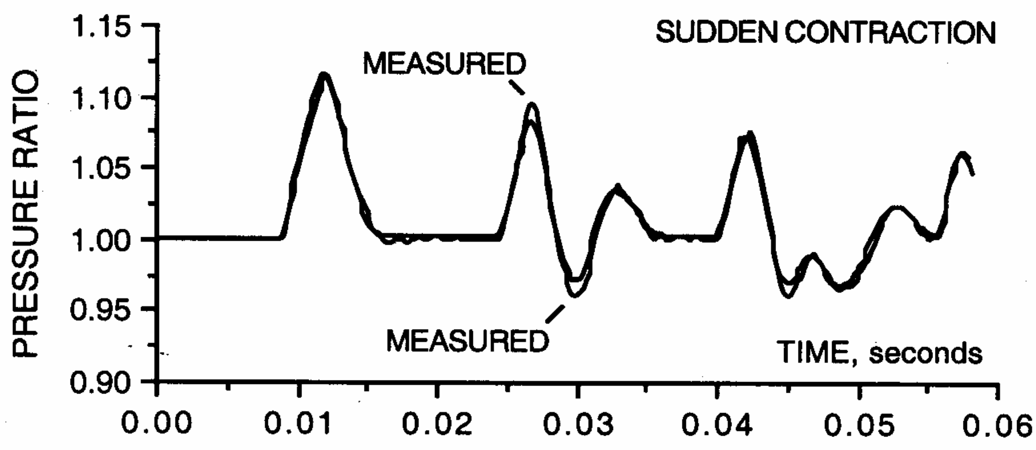


Figure 5.8 Pressure amplitude ratio at station 3 from engine simulation test case 1 in literature [13]

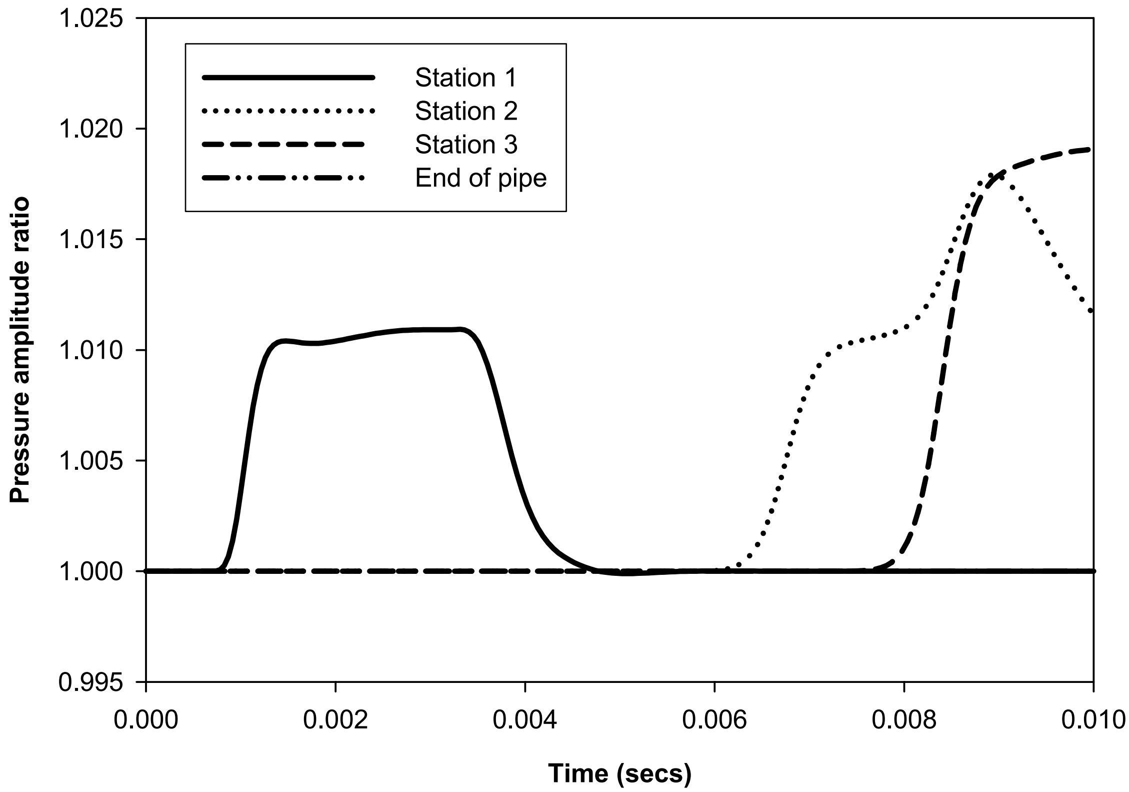


Figure 5.9 Pressure amplitude ratios at transducer locations

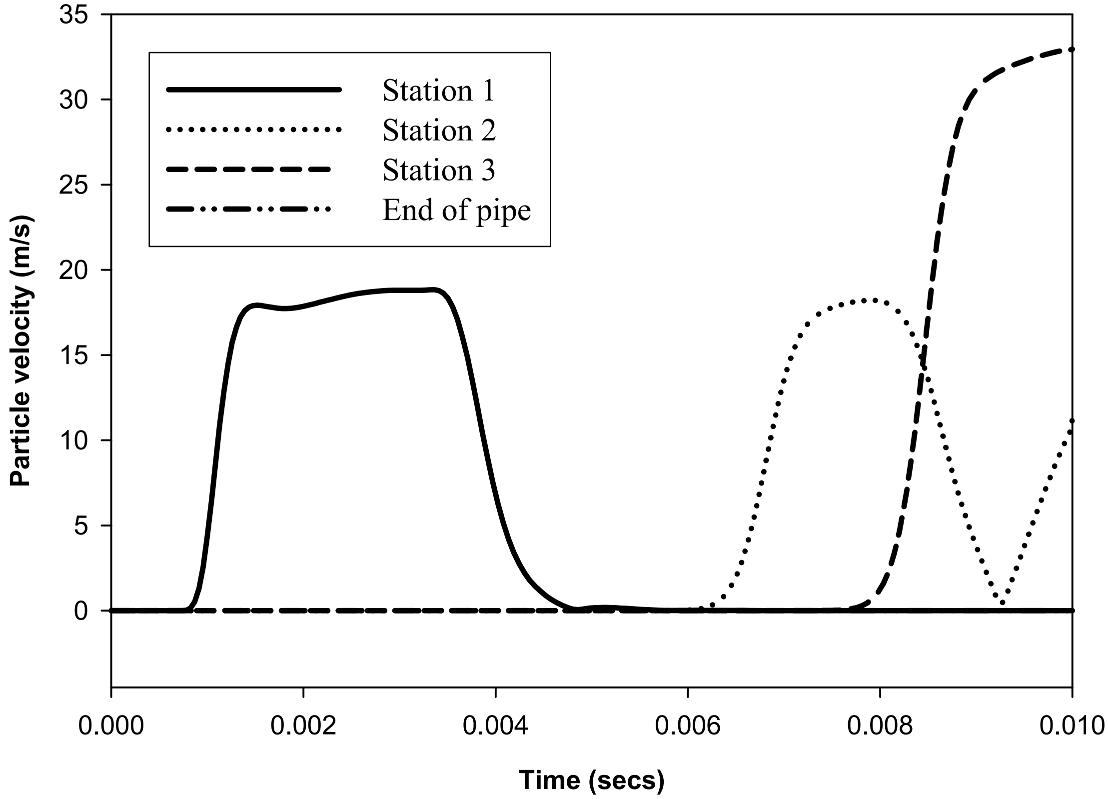


Figure 5.10 Particle velocities at transducer locations

All other parameters such as absolute pressure, temperature, acoustic velocity, density and reference temperature are deduced from the pressure amplitude ratio. Detailed results are plotted for each of the locations in Appendix C.

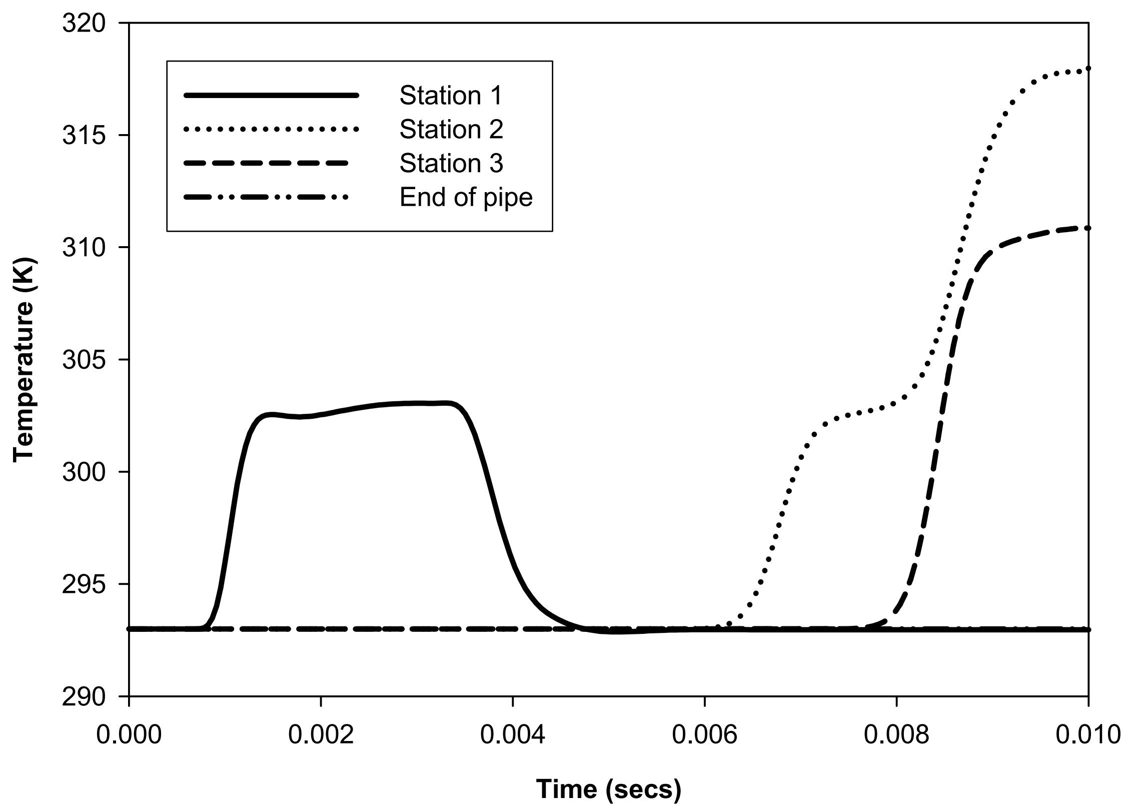


Figure 5.11 Temperature at transducer locations

5.3.2 Test case 2

Another representative configuration modeled which is the test case 2 [13] is shown in figure 5.12. The input parameters in the literature are shown in table 5-6 and for the flow simulation code in table 5-7. Pressure and velocity plots are shown in figures 5.13 and 5.14.

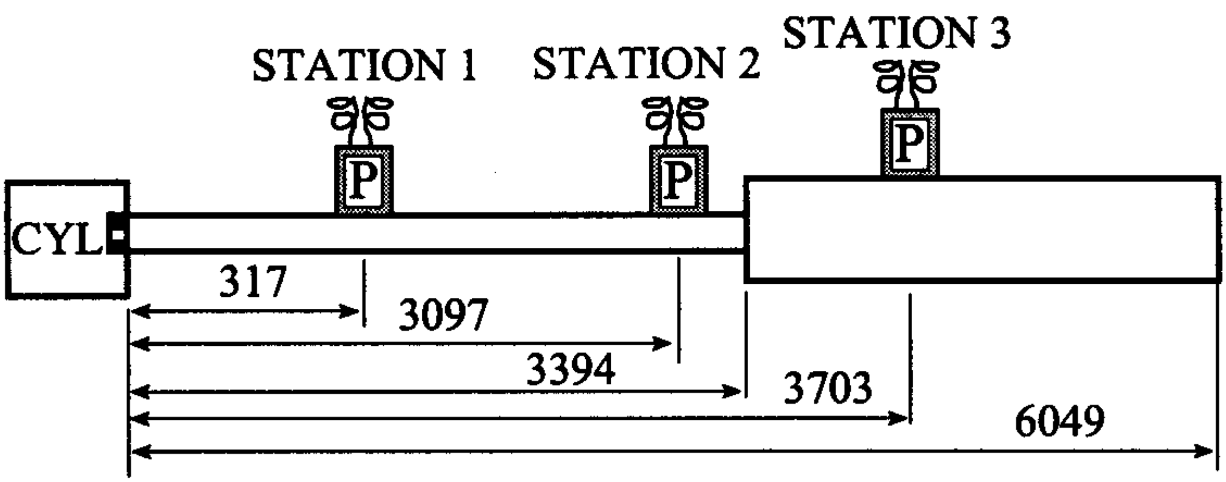


Figure 5.12 configuration for test case 2 [13]

Simulation parameters in literature for test case

PARAMETER	VALUE	UNIT	DESCRIPTION
Pinlet	1.50	bar	pressure in cylinder
medium	air	-------	-------
speed	3000.00	RPM	Engine RPM
Reference temperature	293.00	K	Reference temperature
Port opening	0.008	secs	time during which exhaust port is open during a cycle

Table 5-6. Input parameters for test case 2 in literature [13] [34]

Simulation parameters for flow simulation code

PARAMETER	VALUE	UNIT	DESCRIPTION
Pinlet	150000.00	Pa	pressure at inlet of duct
γ	1.4000	-------	ratio of specific heat capacities
R	287.00	J/kgK	Characterestic gas constant
frequency - f	100.00	Hz	frequency (Hz)
wall temperature - Tw	300.00	K	Temperature of the duct wall
Duty cycle	0.25	-------	fraction of T when Xp is present
Ambient reference temperature	293	K	Initial (undisturbed) temperature
Ambient reference pressure	101325	Pa	Initial (undisturbed) pressure
Exit pressure	101325	Pa	pressure at exit of pipe

Geometric details for flow simulation code

PARAMETER	VALUE	UNIT	DESCRIPTION
L1	3394.00	mm	length of segment 1
D1	25.000	mm	diameter of segment 1
L2	2500.0000	mm	length of segment 2
D2	80.000	mm	diameter of segment 2
L3	155.00	mm	length of segment 3
D3	80.000	mm	diameter of segment 3

Table 5-7. Input parameters for test case 2 for flow simulation code

Discussion of results

Figures 5.13 – 5.15 show the results for this test case from literature [13]. Similar to test case 1, the results are averaged over the entire engine cycle after reaching a periodic steady condition. In our study, the simulation is performed only during the period corresponding to the frequency of the pulse emitted. Hence, the time period is less than for the engine simulation case. The input for our study is a square pulse which is not the case for the engine study.

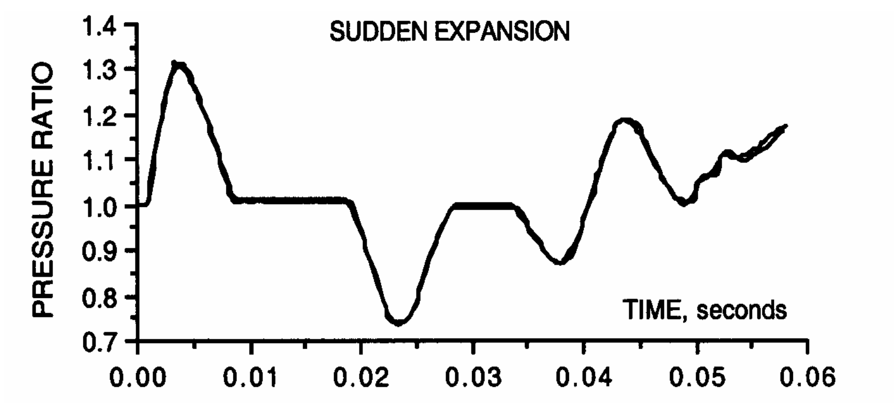


Figure 5.13 Pressure amplitude ratio at station 1 from engine simulation test case 2 in literature [13]

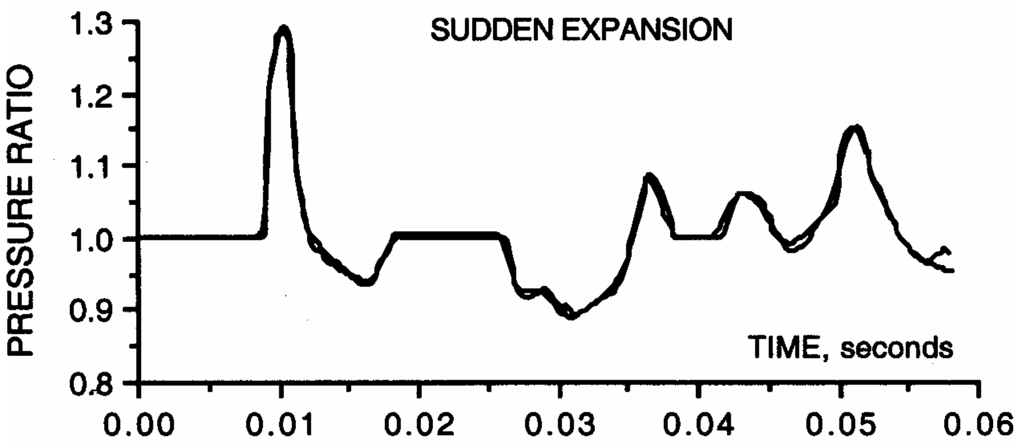


Figure 5.14 Pressure amplitude ratio at station 2 from engine simulation test case 2 in literature [13]

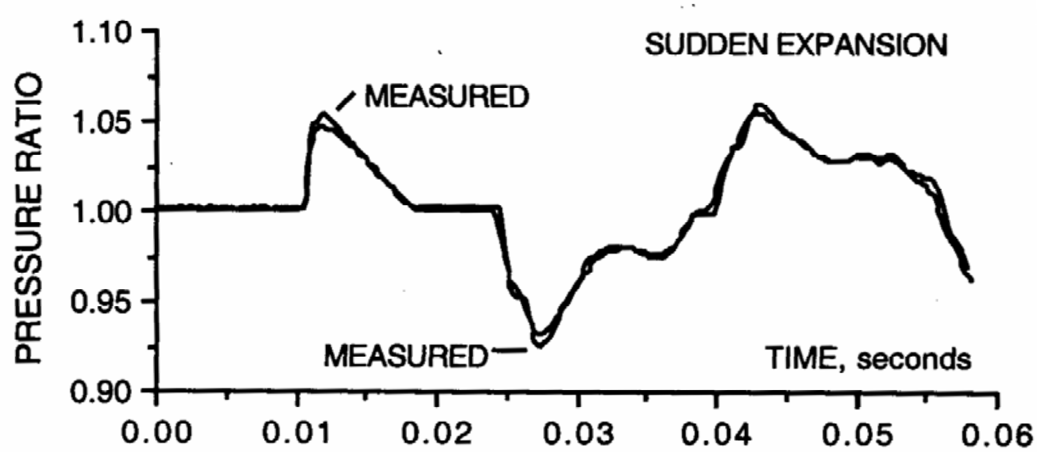


Figure 5.15 Pressure amplitude ratio at station 3 from engine simulation test case 2 in literature [13]

Refer to figure 5.16 for the pressure amplitude ratio plot for the transducer locations. In this test case, the basic action of reflection at a sudden expansion is observed. The pressure pulse passes the station 1 at 0.004 seconds. The spike in pressure at station 2 is due to the reflections from the sudden expansion. Station 3 exhibits a loss in pressure due to the expansion. Again, for the time period of the simulation corresponding to the input frequency, the pressure wave has not reached the end of the pipe and undisturbed conditions exist in the end mesh. Therefore, the pressure amplitude ratio is 1. Figures 5.16 and 5.17 show the pressure and velocity plots at the three stations shown in figure 5.12. The mesh size used is 50 mm. The time step under-relaxation factor is 0.65. A comparison of results with different mesh sizes is discussed in section 5.5. Effect of time step is discussed using the test case 2 in section 5.4.3.

Comparing the two solutions for the time period of 0.01 seconds, it can be seen that the peak pressure is highest at station 2 for both cases. The lowest peak is observed for station 3. The increase in pressure at station 2 is due to the reflection from the sudden expansion and subsequent superposition at station 2.

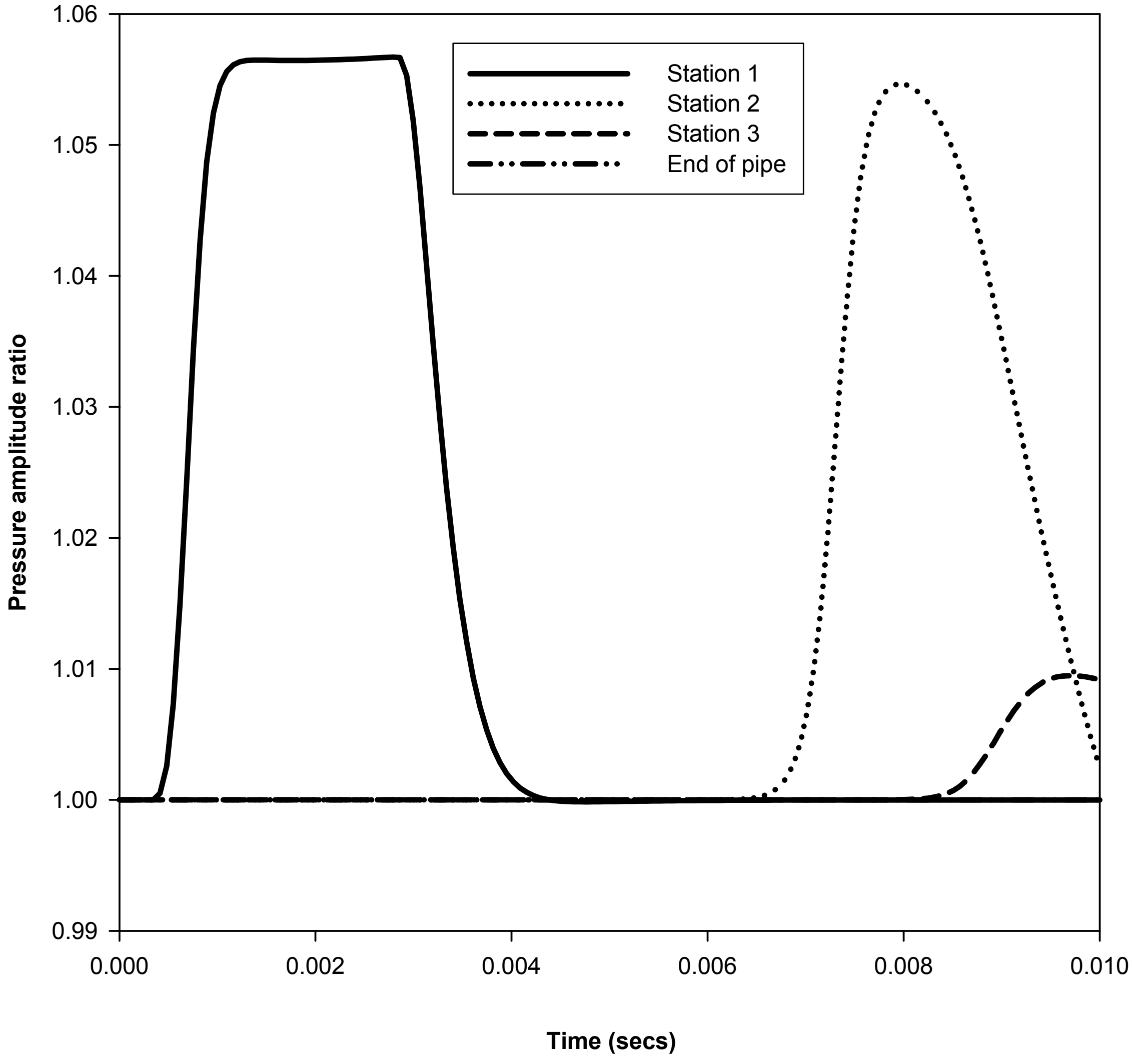


Figure 5.16 Pressure amplitude ratios at transducer locations

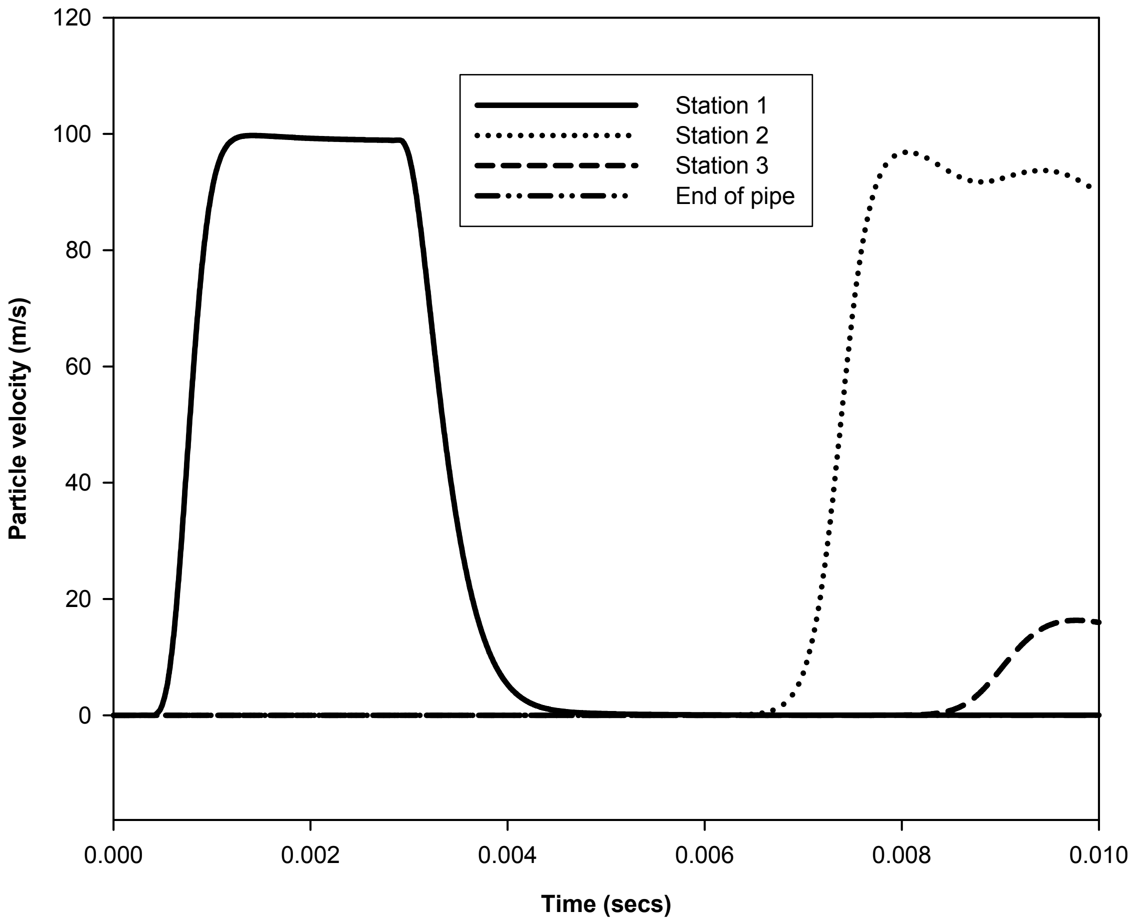


Figure 5.17 Particle velocities at transducer locations

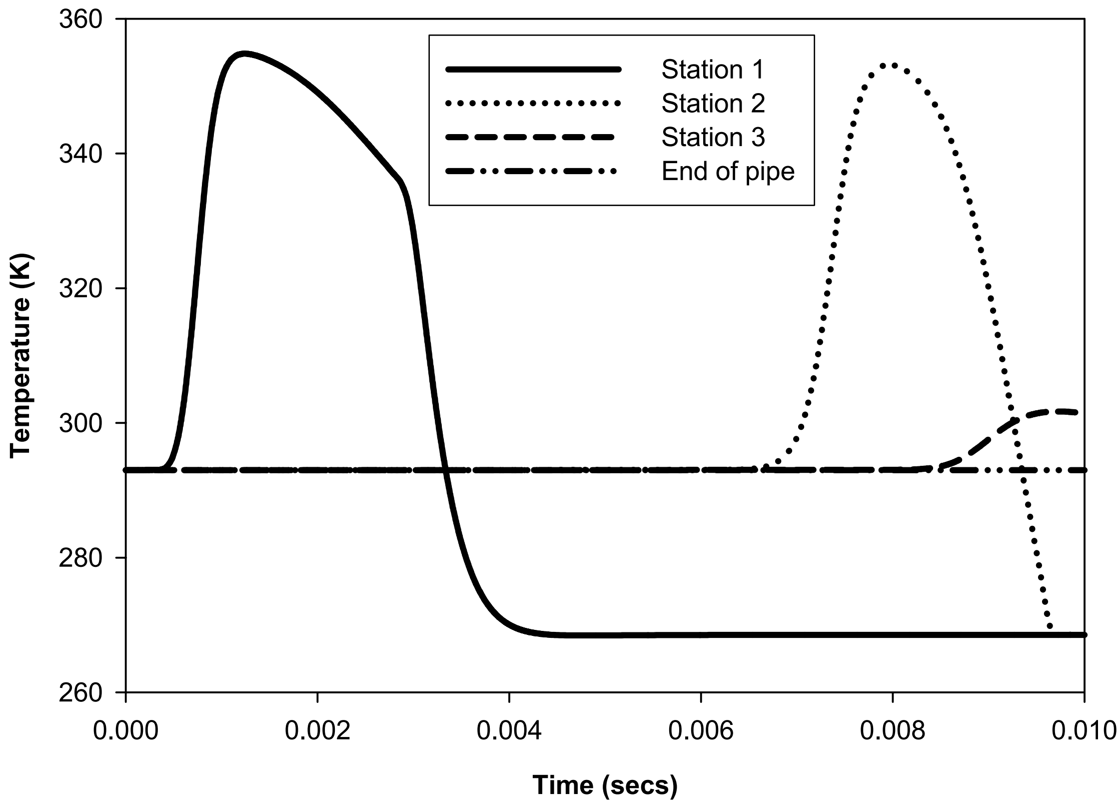


Figure 5.18 Temperature at transducer locations

5.4 Effect of Simulation parameters

The simulation parameters that may affect the stability of the solution for our problem are mesh size, duty cycle and the inlet and exit boundary conditions. Anderson [29] describes a problem for a quasi one-dimensional subsonic-supersonic isentropic nozzle flow. The method of characteristics is used to analyze stability of CFD solutions to such flows. This method necessitates that at a boundary where one characteristic propagates into the domain, one dependent variable must be specified at the boundary and at a boundary where one characteristic propagates out of the domain, one dependent variable must be allowed to float. Further, when a streamline moves into a domain, two values must be specified at the boundary. Thus for a subsonic inflow boundary, two boundary flow variables need to be specified – density and temperature and velocity is allowed to float. In the case where the two characteristics propagate out of the domain and the streamline is moving out, all the variables are allowed to float. Thus for a supersonic outflow boundary, all three variables are allowed to float. Intelligent selection of initial conditions is important because the closer these values are to the final answer, convergence will be faster. A purely subsonic flow is also dealt with by Anderson [29]. A high inlet pressure ratio may cause the solution to become unstable and blow up. This is attributed to the fact that the exit pressure is held constant. Finite compression and reflection waves reflect off this boundary and if these waves are strong enough, they will set up strong oscillations near the downstream boundary leading to a blow up of the calculations. For smaller pressure ratios the unsteady waves created are weaker and hence do not set up an oscillation. Anatomy of a failed solution for these types of problems also can be found in [29]. Such a case will be discussed in sections 5.4.2 and 5.4.3.

To summarize, the simulation parameters that affect the solution are mesh size, input parameters, duty cycle and time step. The effect of mesh size is discussed with a test case in section 5.4.1. Influence of duty cycle is discussed in section 5.4.2 and time step in section 5.4.3. The test case 2 is used to show the effects of all three parameters.

5.4.1 Effect of mesh size

This is the most important parameter affecting the solutions. In performing several numerical experiments with the code, it has been found that for this problem the dependency of solution on mesh size is very high. Unless the correct mesh size is applied or a value in a very close range within a few mm is applied, the solution fails to converge, mostly resulting in a negative reference temperature or the blow up of the solution. Test case 2 can be used to analyze this behavior.

Three mesh sizes were tested – 35mm, 40 mm, 50 mm. Results were in close agreement – maximum variation was 2%. Figures 5.12 – 5.14 show pressure plot comparison for the three cases. The velocity plots are shown in figures 5.15 – 5.17. Other results for this test case can be found in Appendix C.

Smaller mesh sizes of the order of 10mm causes the mass outflow to exceed the mass inflow resulting in negative reference temperatures to be evaluated. This mostly occurs in the mesh just upstream of the expansion. Further, if the difference in individual mesh sizes in the pipe is large, similar results can occur for the smallest mesh. Therefore, the mesher routine is written in such a way as to ensure that the mesh size values are kept as close as possible while retaining the general-purpose nature of the routine.

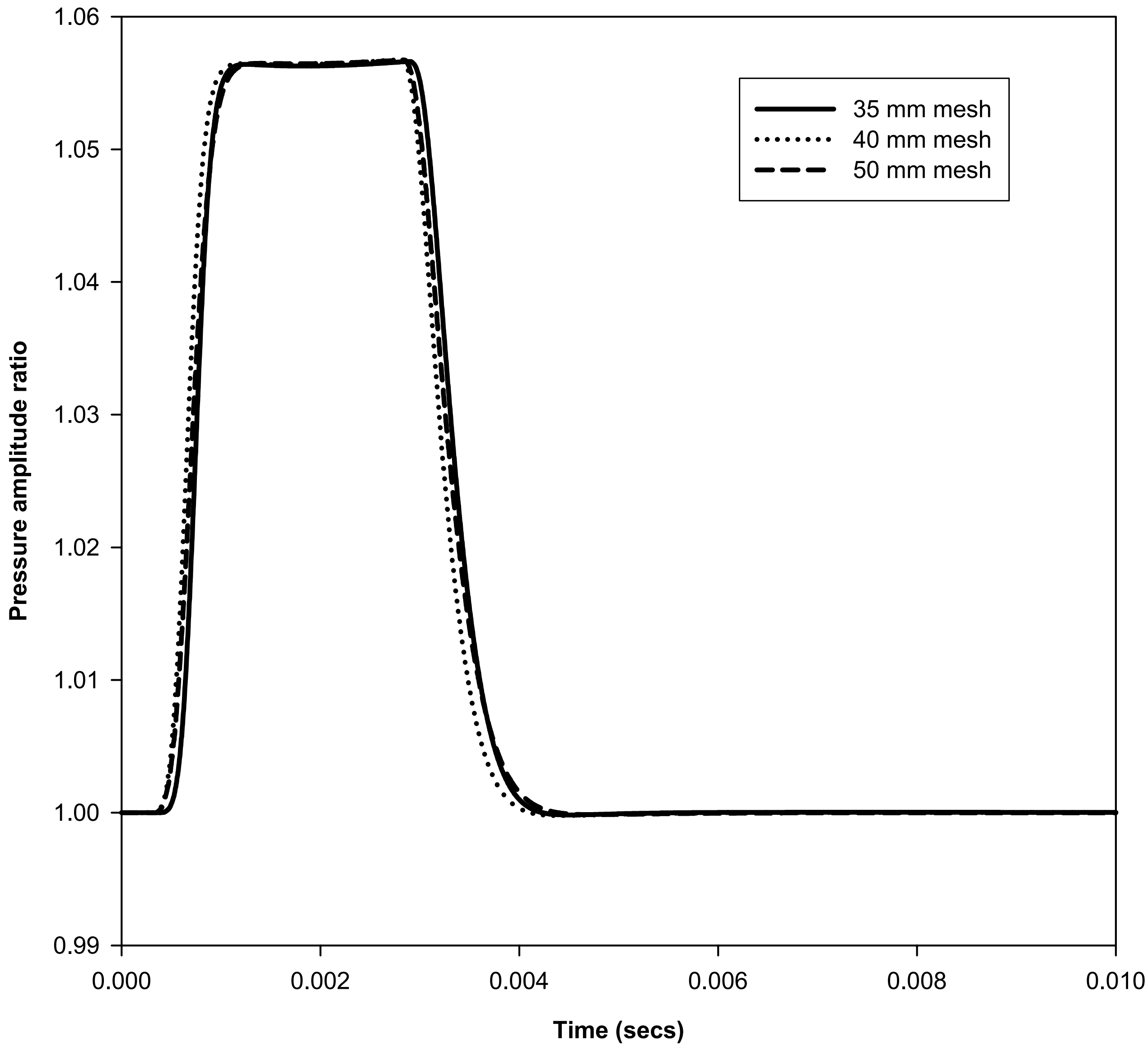


Figure 5.19 Pressure amplitude ratio plots for three mesh sizes, station 1 - comparison

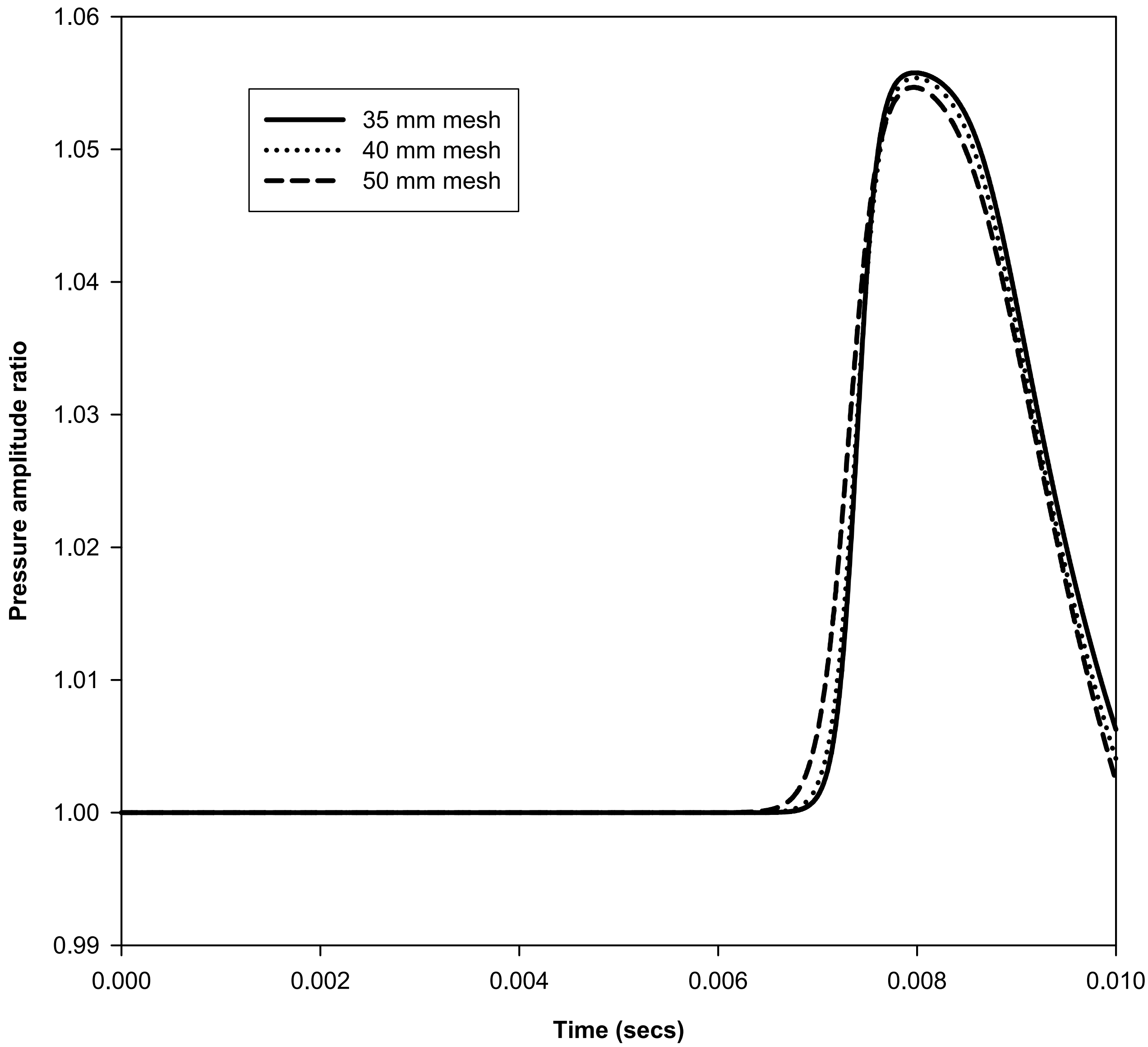


Figure 5.20 Pressure amplitude ratio plots for three mesh sizes, station 2 – comparison

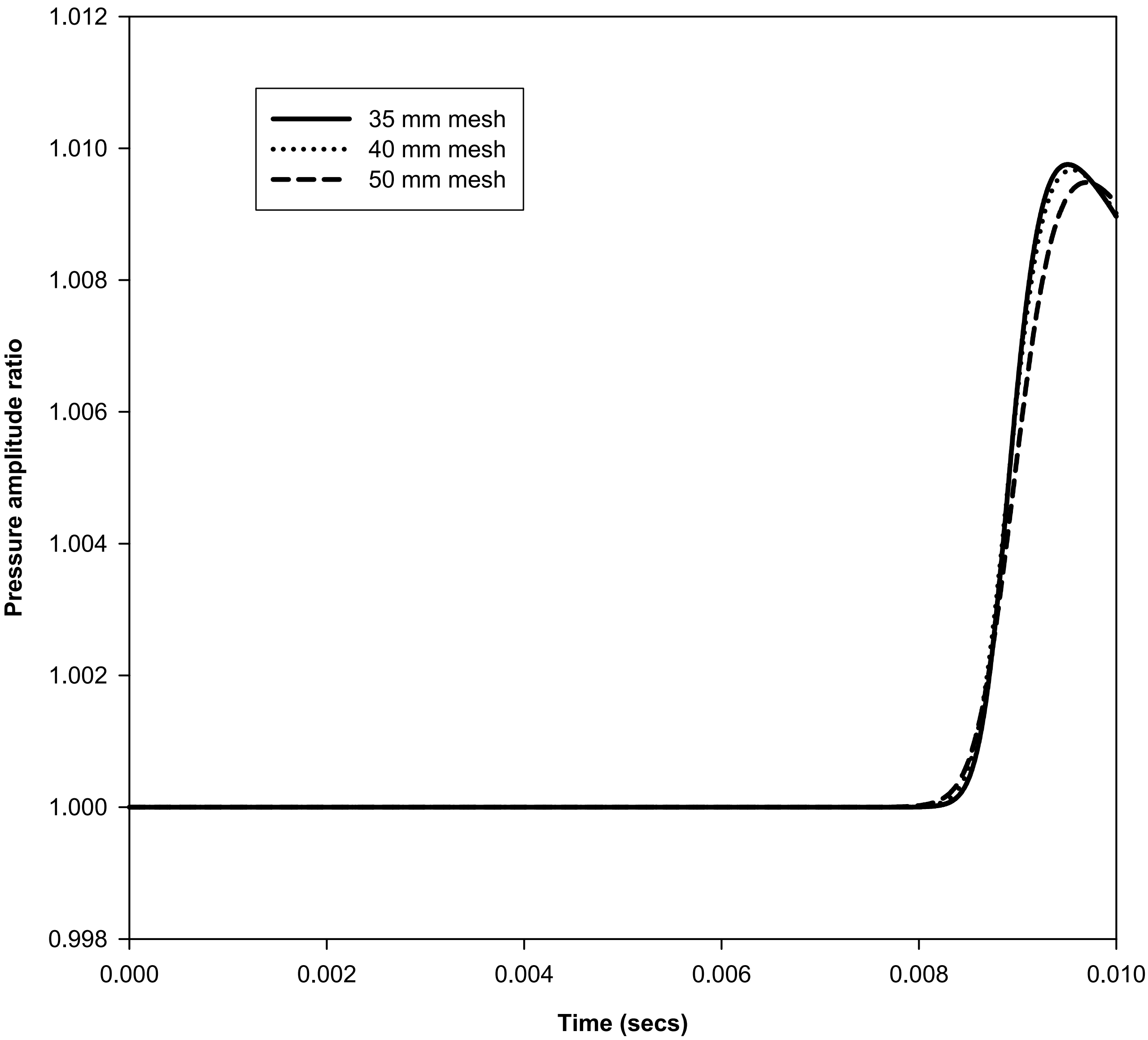


Figure 5.21 Pressure amplitude ratio plots for three mesh sizes, station 3 - comparison

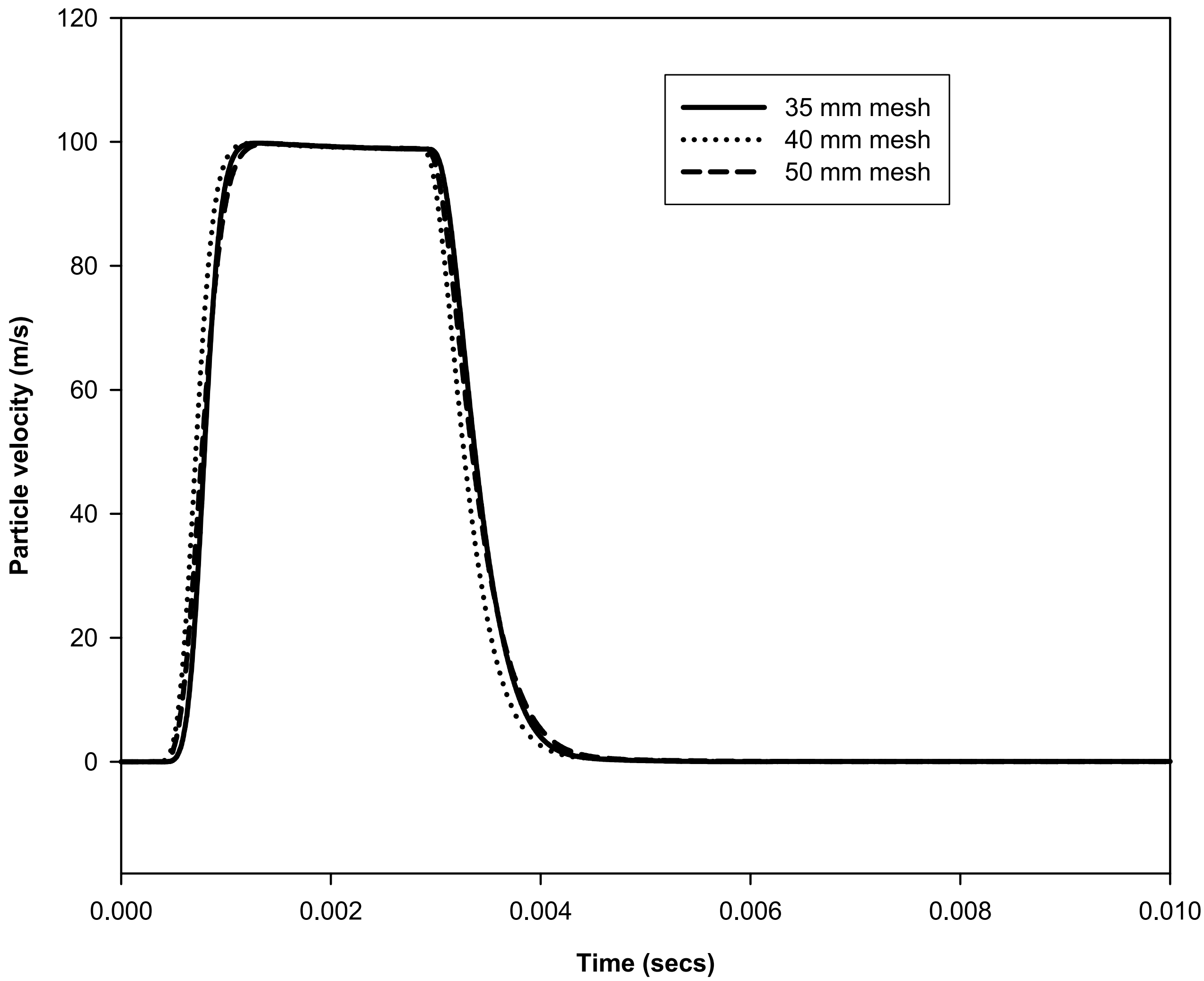


Figure 5.22 Particle velocity plots for three mesh sizes, station 1 – comparison

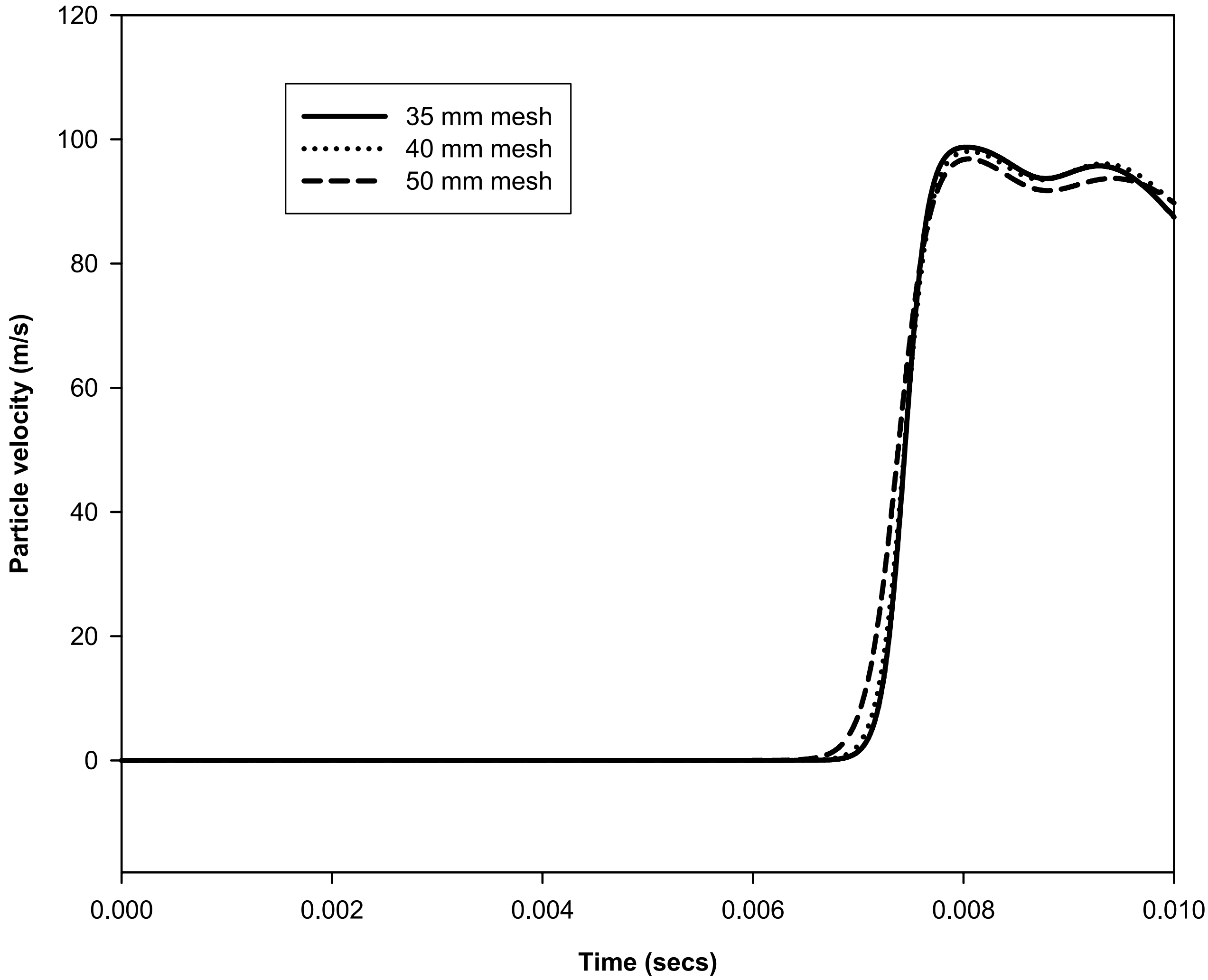


Figure 5.23 Particle velocity plots for three mesh sizes, station 2 – comparison

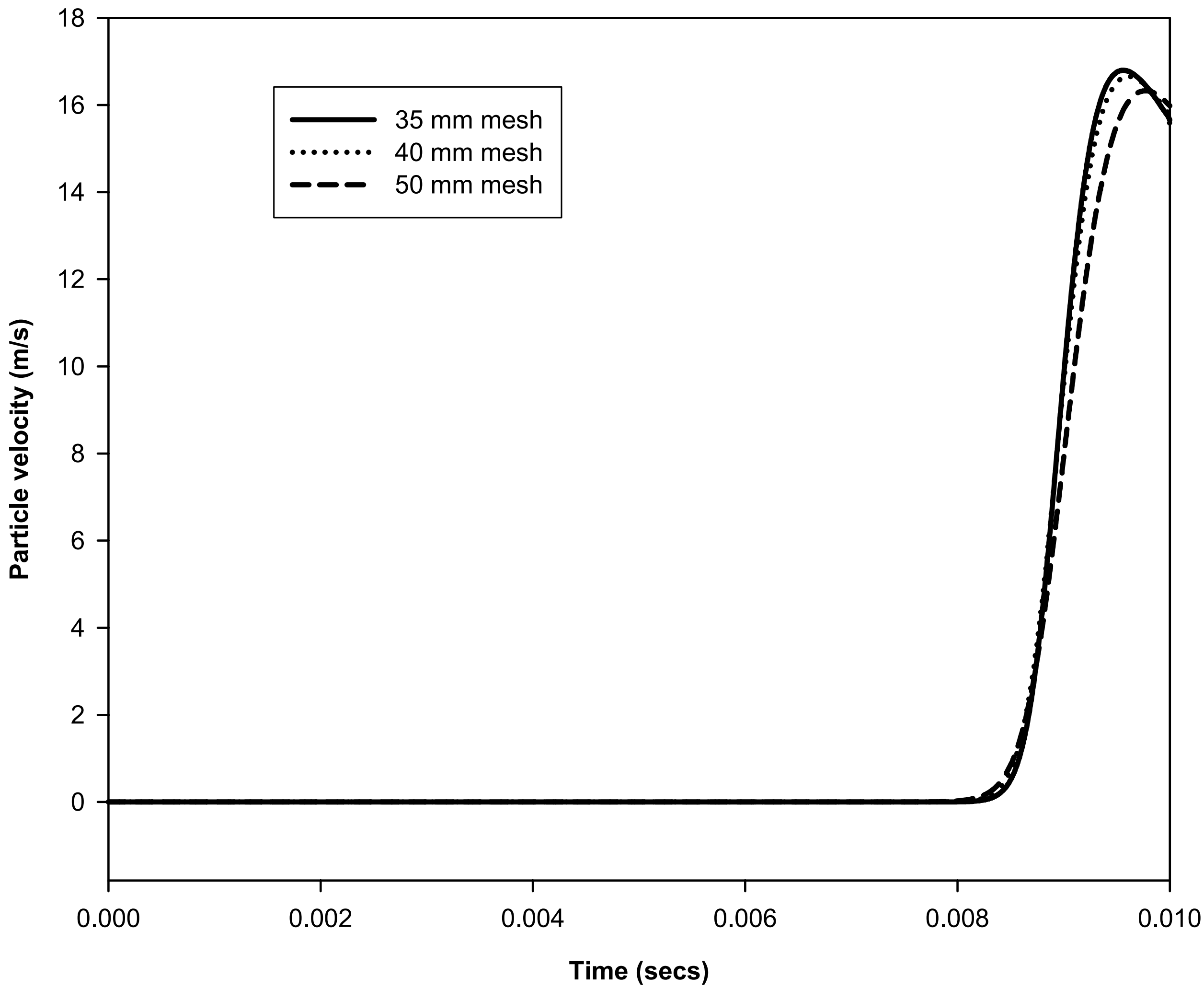


Figure 5.24 Particle velocity plots for three mesh sizes, station 3 – comparison

5.4.2 Effect of duty cycle

The duty cycle is an input parameter. However, at very low duty cycles, it has been seen that the solution may blow up due to strong expansion waves traveling upstream resulting in negative reference temperatures at some arbitrary location in the geometry. This is due to the fact that after a few time steps the pressure amplitude ratio drops to one resulting in the mass outflow from a mesh exceeding the inflow thereby causing negative reference temperatures during the particular time step.

Such a case can be analyzed using test case 2. Here we reduce the duty cycle to 0.1 and increase the inlet pressure to 300,000 Pa. Figure 5.25 shows the reference temperature plot of the failing mesh, in this case it is the mesh just upstream of the sudden expansion. This is true for most cases.

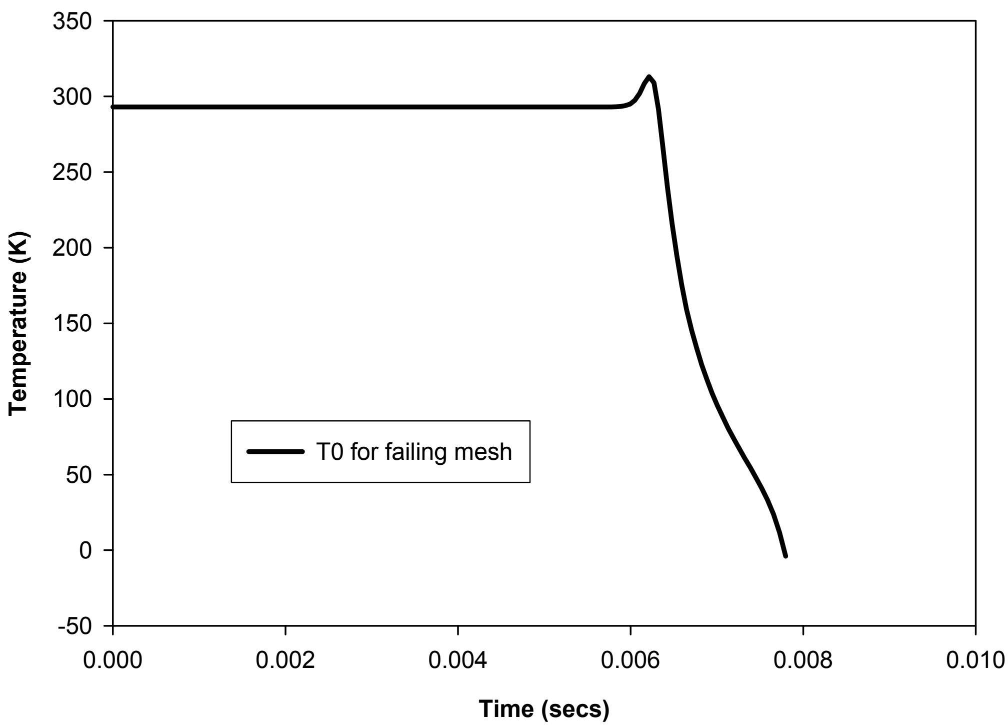


Figure 5.25 Reference temperature plot for failing mesh

The plot of mass flow in and out of the mesh in the course of the simulation is shown in figure 5.26. This plot clearly indicates that the mass flow out of the mesh is more than the mass flow into the mesh. This differential causes a negative reference temperature to be created in the first law evaluation and the code blows up giving an error message. This can be solved in some cases using an under-relaxation in the time step. Such a case showing improvement in solution is discussed in section 5.4.3.

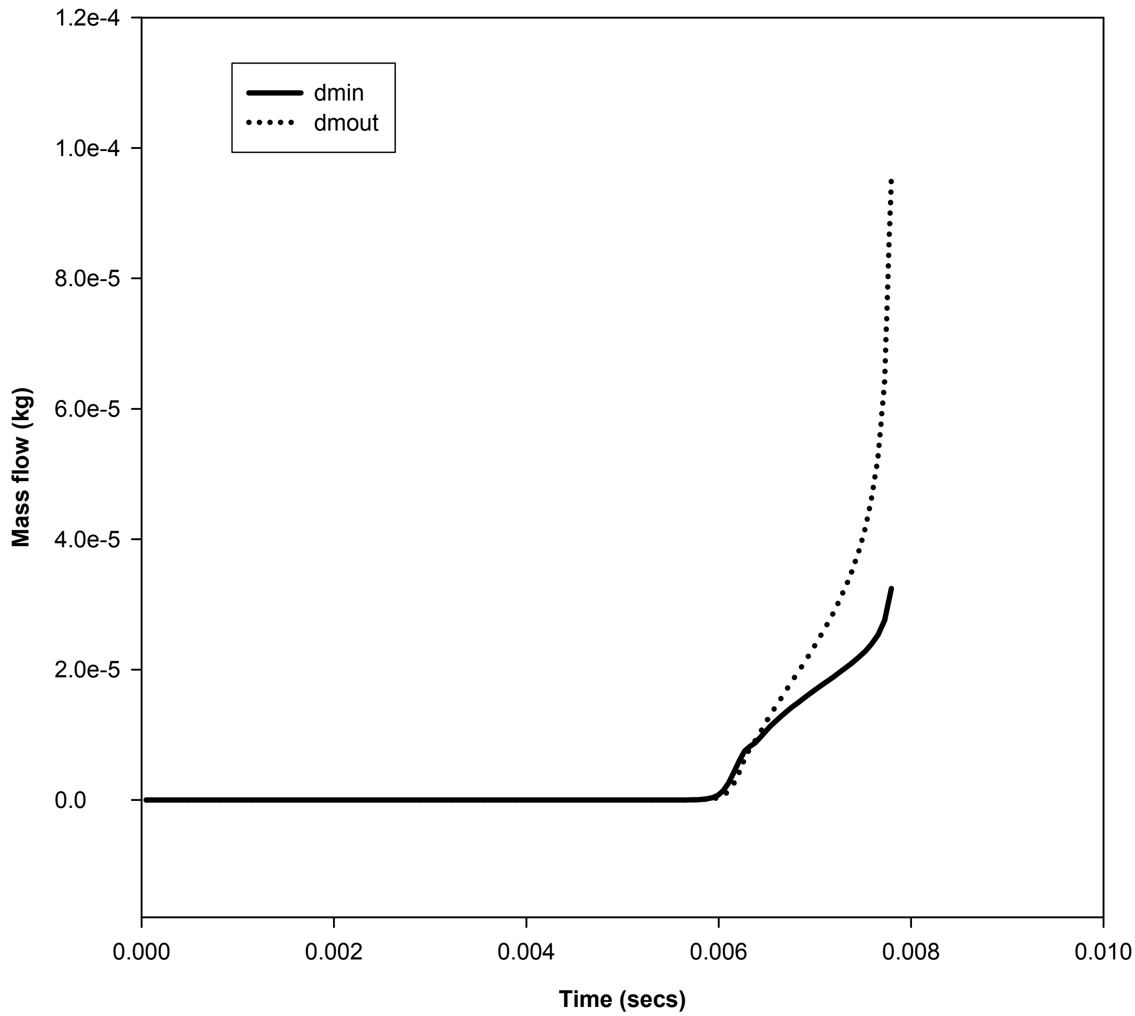


Figure 5.26 Mass inflow and outflow for the failing mesh

5.4.3 Effect of time step and stability

Time step plays a very important role in the accuracy of the solution of hyperbolic equations. A detailed discussion can be found in [29]. Stability of the solution of hyperbolic equations is governed by the Courant, Friedrich and Lewy (CFL) stability criterion [38]. This criterion is mathematically written as

$$C = c\frac{\Delta t}{\Delta x} \qquad (5\text{-}4)$$

where c is the particle velocity and C is the Courant number.

This translates into

$$\Delta t = C\frac{\Delta x}{c} \qquad (5\text{-}5)$$

The Courant Number C is to be kept below 1 but close to 1 for stability and accuracy respectively. A detailed discussion on stability considerations for hyperbolic equations using method of characteristics can be found in [29]. The mesh size is found to be the lone governing factor affecting the stability of the solution. The Courant number needs to be kept below 1 for convergence. However, this is built into the model while evaluating the time step. The 0.99 factor ensures that the Courant number is below 1 and all subsequent iterative procedures are by interpolation. See section 4.1.2.2 for details. The user can vary this factor in the code and check for any stability issues. The author does not feel that this factor is very critical to the outcome of the solution for this problem using this modeling method.

The effect of time step on the solution is analyzed with test case 2. The simulation is performed with under-relaxation in the time step calculation by varying the multiplier (0.99) of the time step in the time step evaluation.

Figure 5.27 shows the pressure amplitude ratio for different under-relaxation factors.

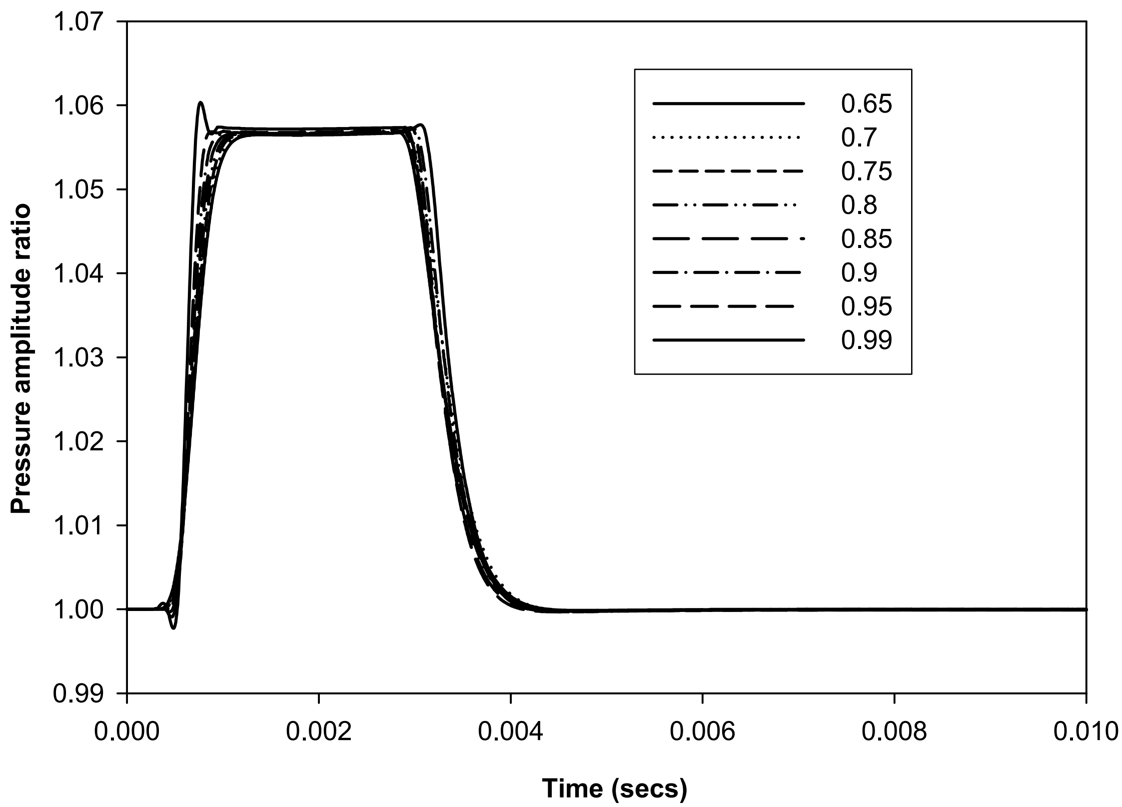


Figure 5.27 Pressure amplitude ratio plots for station 1 with varying under-relaxation factors

The decision on the best solution depends on the reference temperature plots for all three stations. The reference temperature is a good indicator of the stability of the solution. Figure 5.28 shows the velocity plots for the same case. The pressure and velocity plots do not show much variation. However, the temperature plot in figure 5.29 shows a marked divergence in the solution.

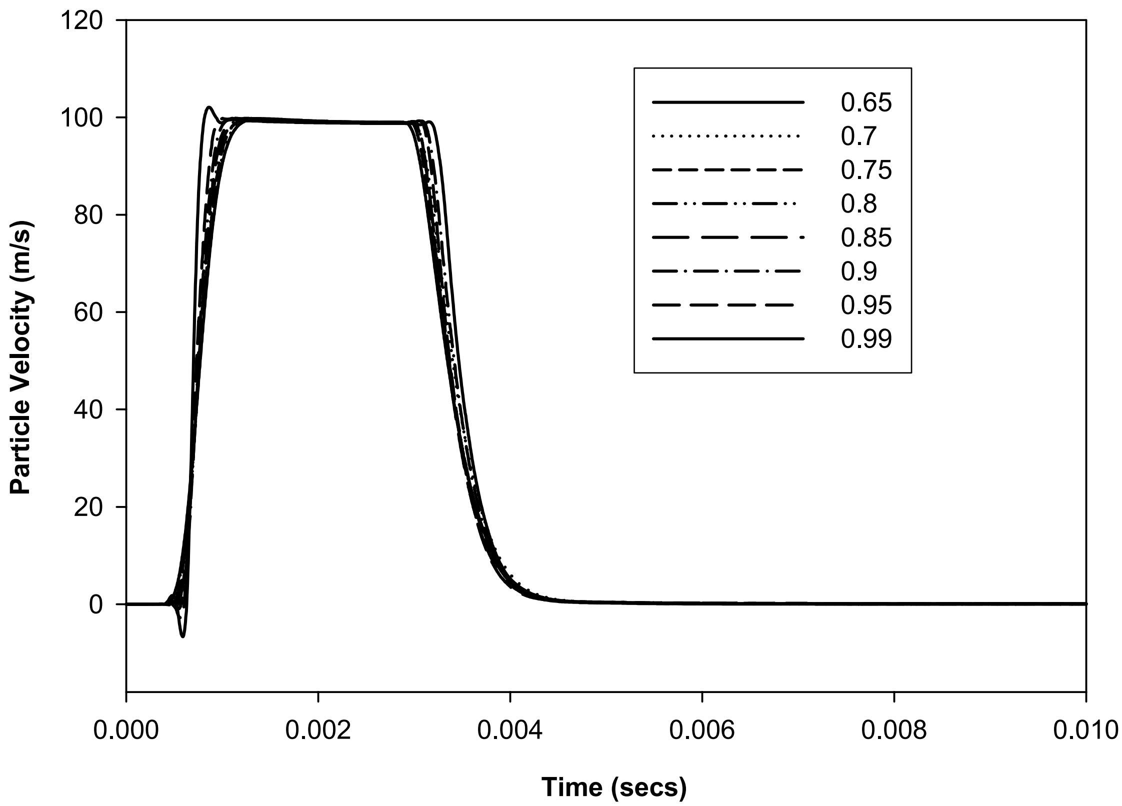


Figure 5.28 Particle velocity plots for station 1 with varying under-relaxation factors

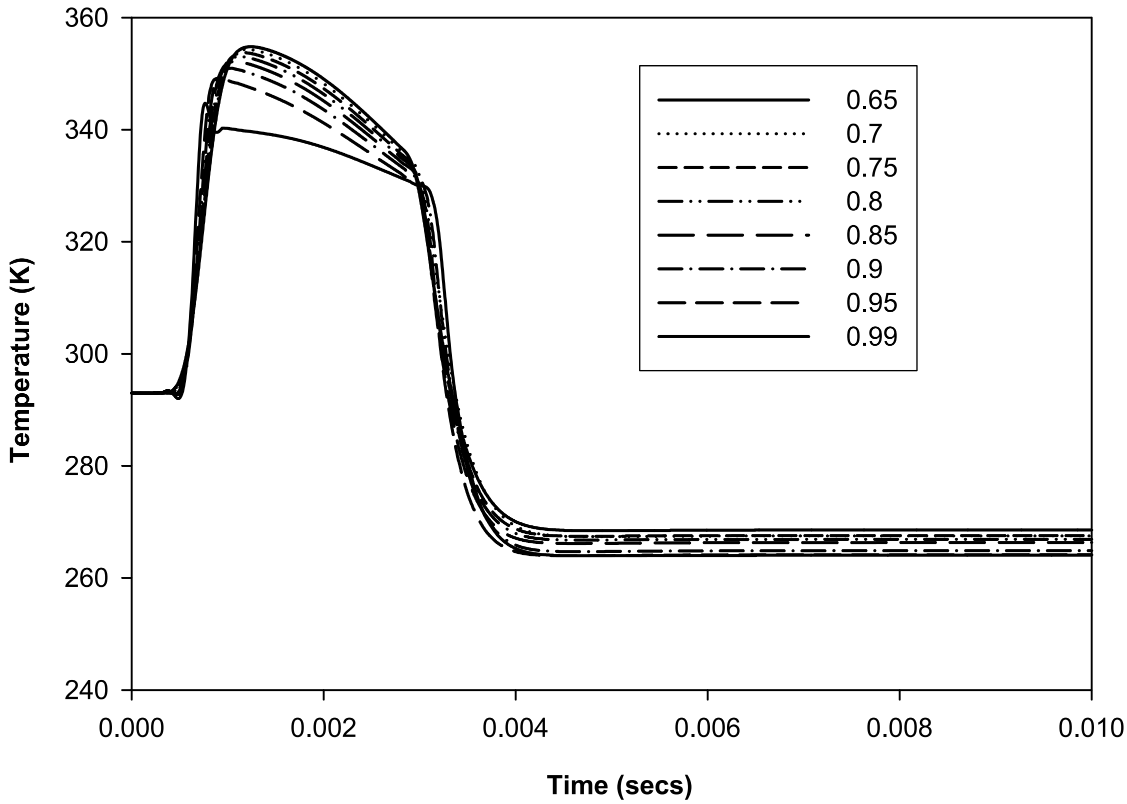


Figure 5.29 Temperature plots for station 1 with varying under-relaxation factors

The next step to find the best solution is to compare the solution for all stations. The best solution selected is the solution that exhibits the smooth solution without any abrupt changes. Figures 5.30 – 5.32 show the results for station 2. Figure 5.22 rules out the solution for the 0.99 factor as this plot shows considerable variation from the other cases. The factor is reduced in steps and as seen from the plot improves the solution. This procedure is repeated until the variation in the variables is acceptable. In this case, this was done until the variation was within 1%. It has been observed that the time step multiplier (0.99 in this case) can cause an error in the interpolation procedure for the mesh. This occurs in the evaluation of the interpolated pressure amplitude ratios at either end of the meshes (X_p and X_q). The selection of an appropriate factor can be done by studying the pressure amplitude and reference temperature plots. The 0.99 case shows undershoot and is asymptotic.

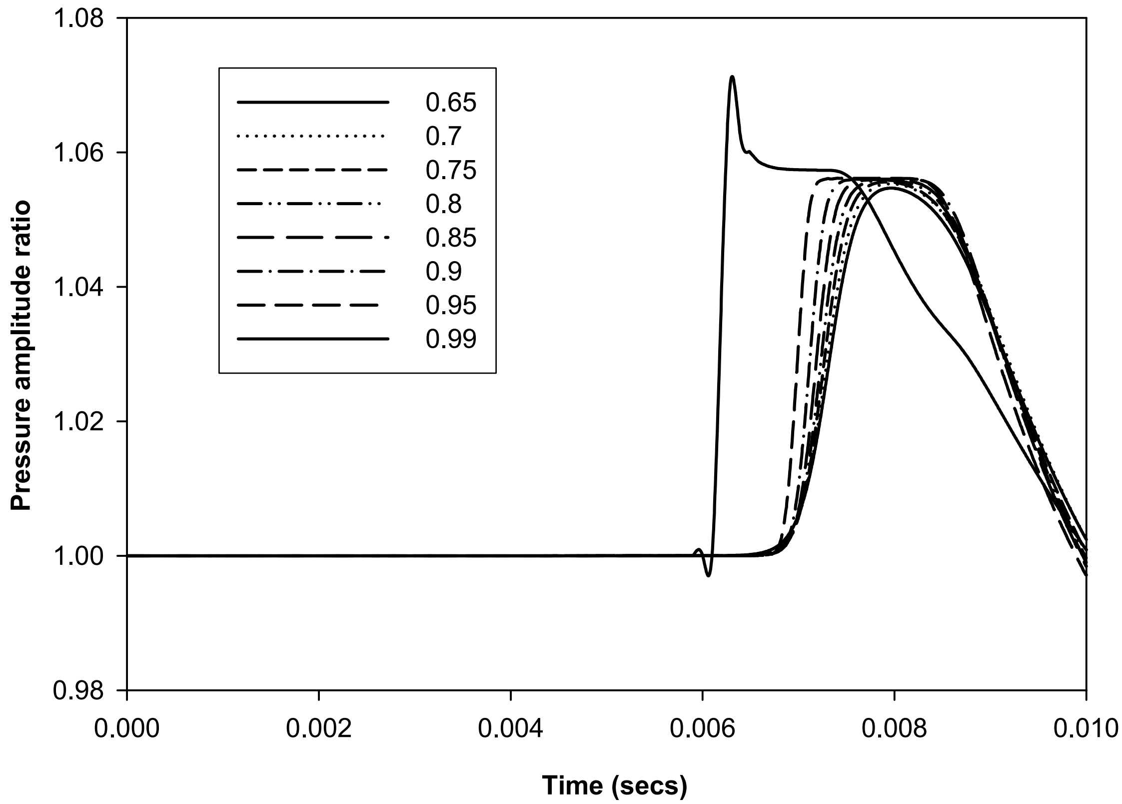


Figure 5.30 Pressure amplitude ratio plots for station 2 with varying under-relaxation factors

Figure 5.32 shows the temperature plot for station 2. The solution for the 0.65 case was selected as the best solution by following the procedure described earlier. The multiplying factor was varied continously until the solution was acceptable. The plots for station 3 are shown in figures 5.33-5.35. Plots for other parameters are shown in Appendix C.

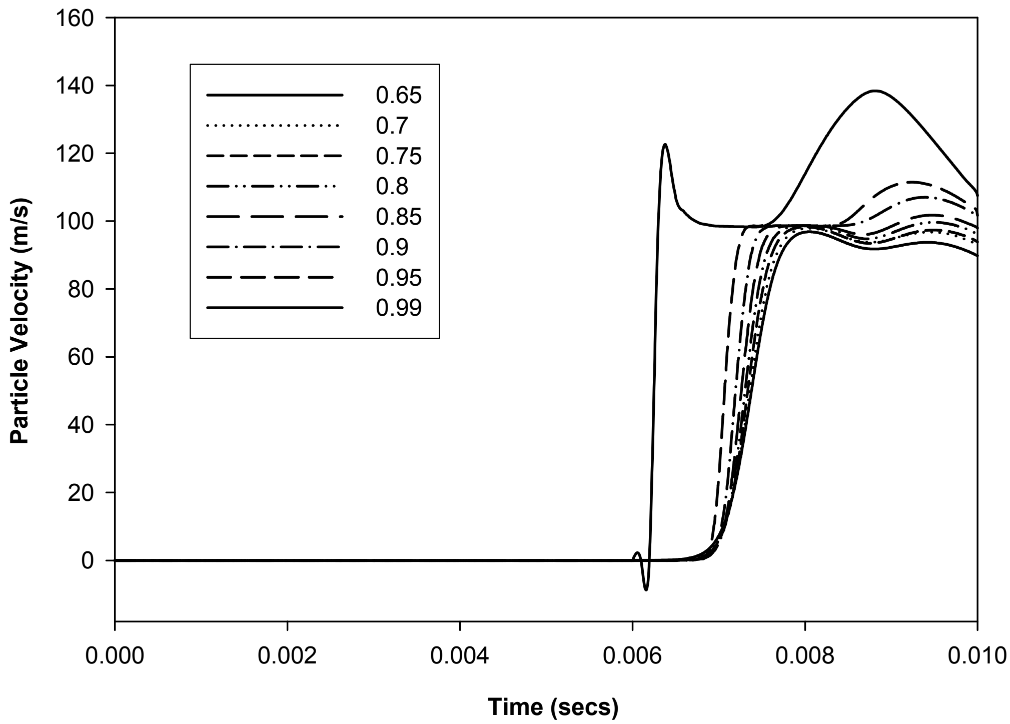


Figure 5.31 Particle velocity plots for station 2 with varying under-relaxation factors

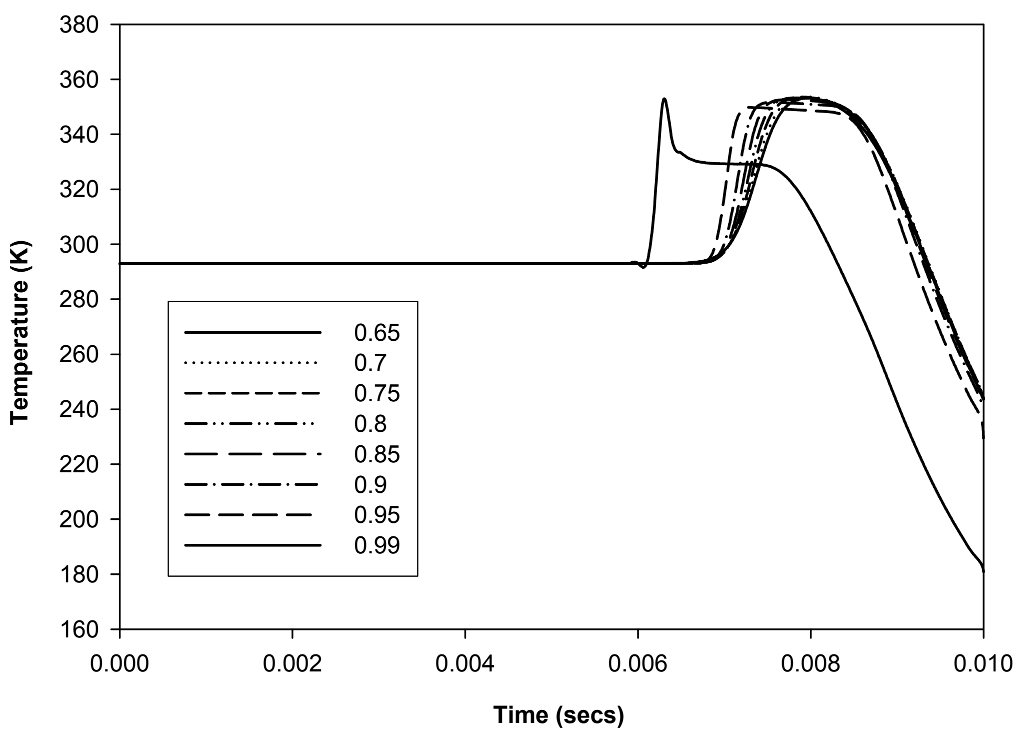


Figure 5.32 Temperature plots for station 2 with varying under-relaxation factors

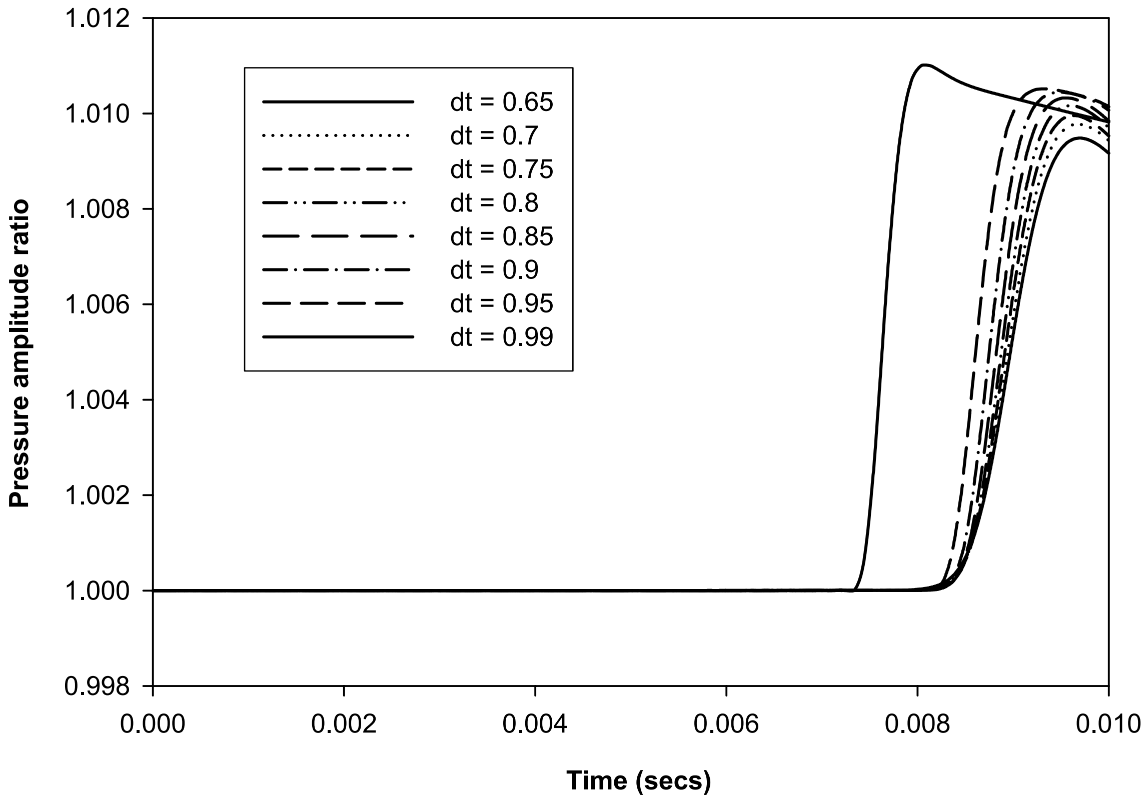


Figure 5.33 Pressure amplitude ratio plots for station 3 with varying under-relaxation factors

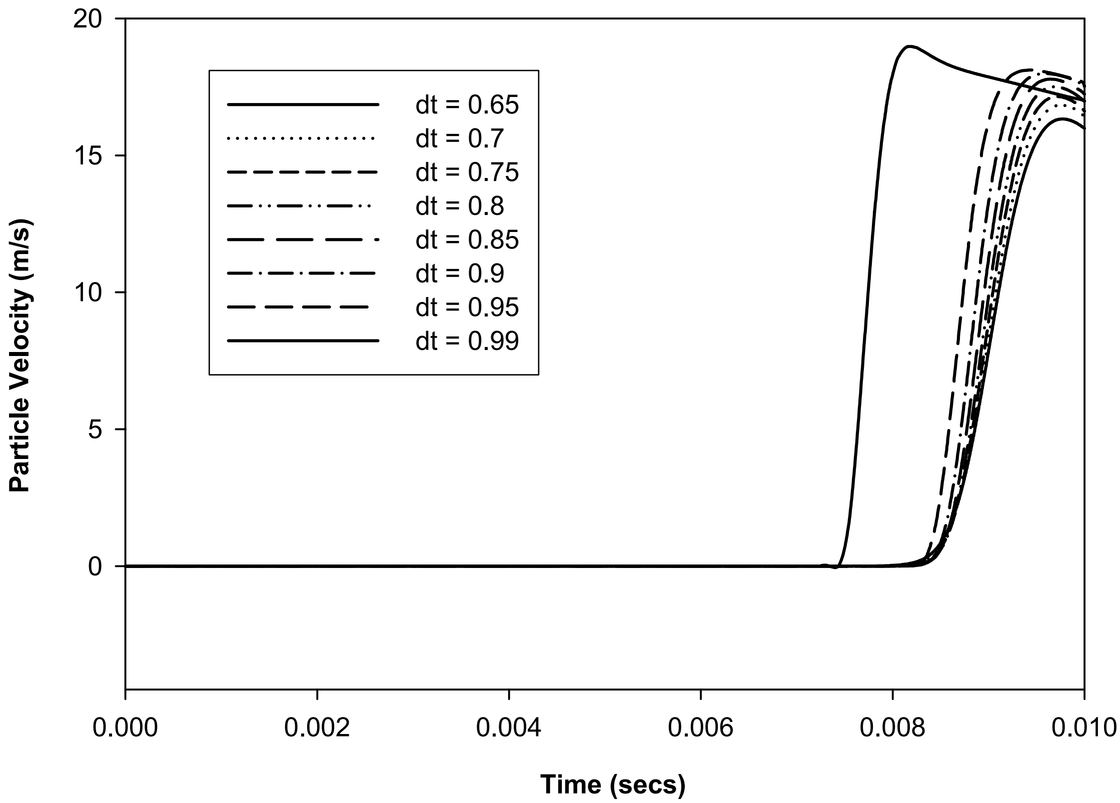


Figure 5.34 Particle velocity plots for station 3 with varying under-relaxation factors

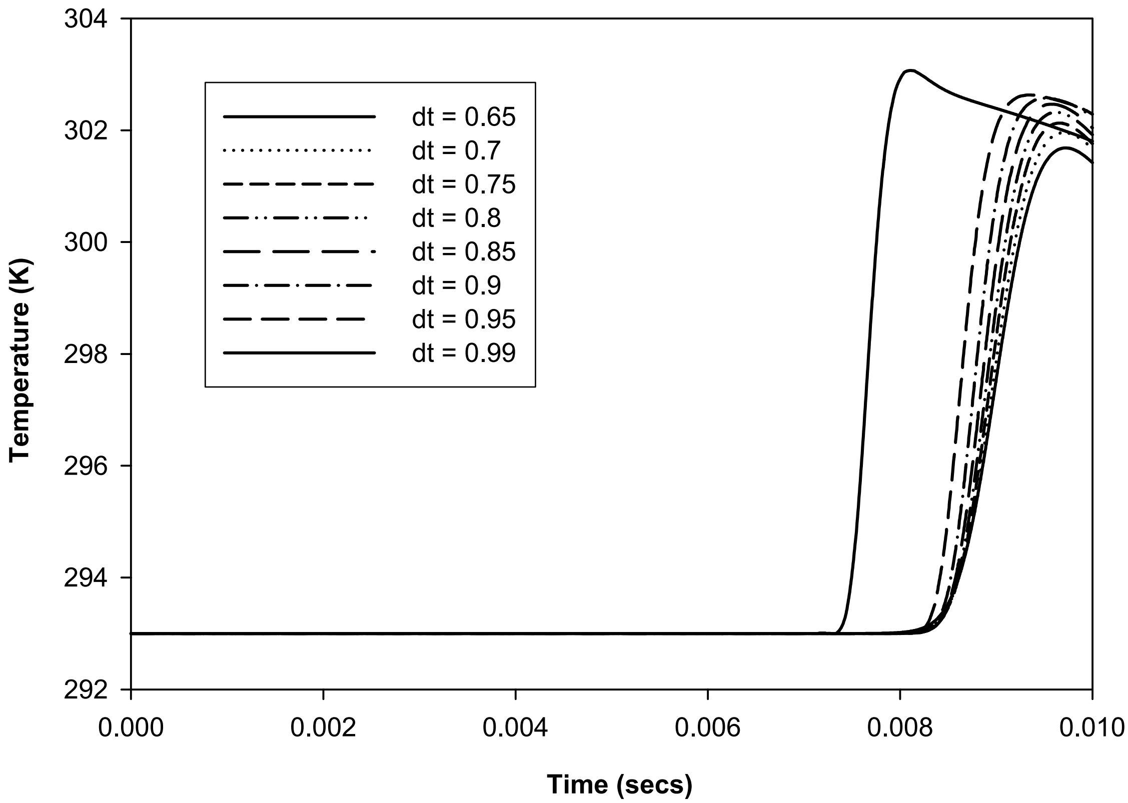


Figure 5.35 Temperature plots for station 3 with varying under-relaxation factors

5.5 Pressure pulse propagation through the pipe

From the results of the simulation, the effect of the transmission tubing on the pressure pulse can be summarized. The square pulse that leaves the actuator at the left end of the tube is distorted due to two reasons. One is the friction pressure loss in the pipe. The other is the reflections that occur inside the tube due to geometric and thermodynamic discontinuities. The one-dimensional model serves as an analysis tool to observe the effects of the tubing on the pressure pulse. The results of the simulation can be used for further aerodynamic analyses.

The pressure pulse propagation through the pipe can be seen in figure 5.25. The square pulse with a duty cycle of 25% is shown with a dashed line in the figure. The pressure amplitude ratio is 1.057 corresponding to an inlet pressure of 150,000 pa. The pulse travels through the pipe and reaches station 1 at 0.000414 seconds. The amplitude ratio of the pulse has reduced to 1.0563 due to friction and reflections. The shape has distorted to the kind of profile shown by the solid lines. As the pulse further undergoes reflections, the distortion increases and a shape similar to a sinusoidal profile is attained as it reaches station 2. This occurs at 0.0065 seconds. The pressure wave crosses station 2 between 0.0065 and 0.01 seconds. The amplitude ratio has decreased further to 1.0547.

This discussion shows that the simulation helps us in visualizing the propagation of the pressure pulse through the pipe. It is possible to quantify the pressure amplitude ratio and the attenuation of the pulse. The distortion of the pulse shape can also be seen from the plots. The square shape is not maintained as the pulse travels through the pipe. This is an important consideration for the design of the control system.

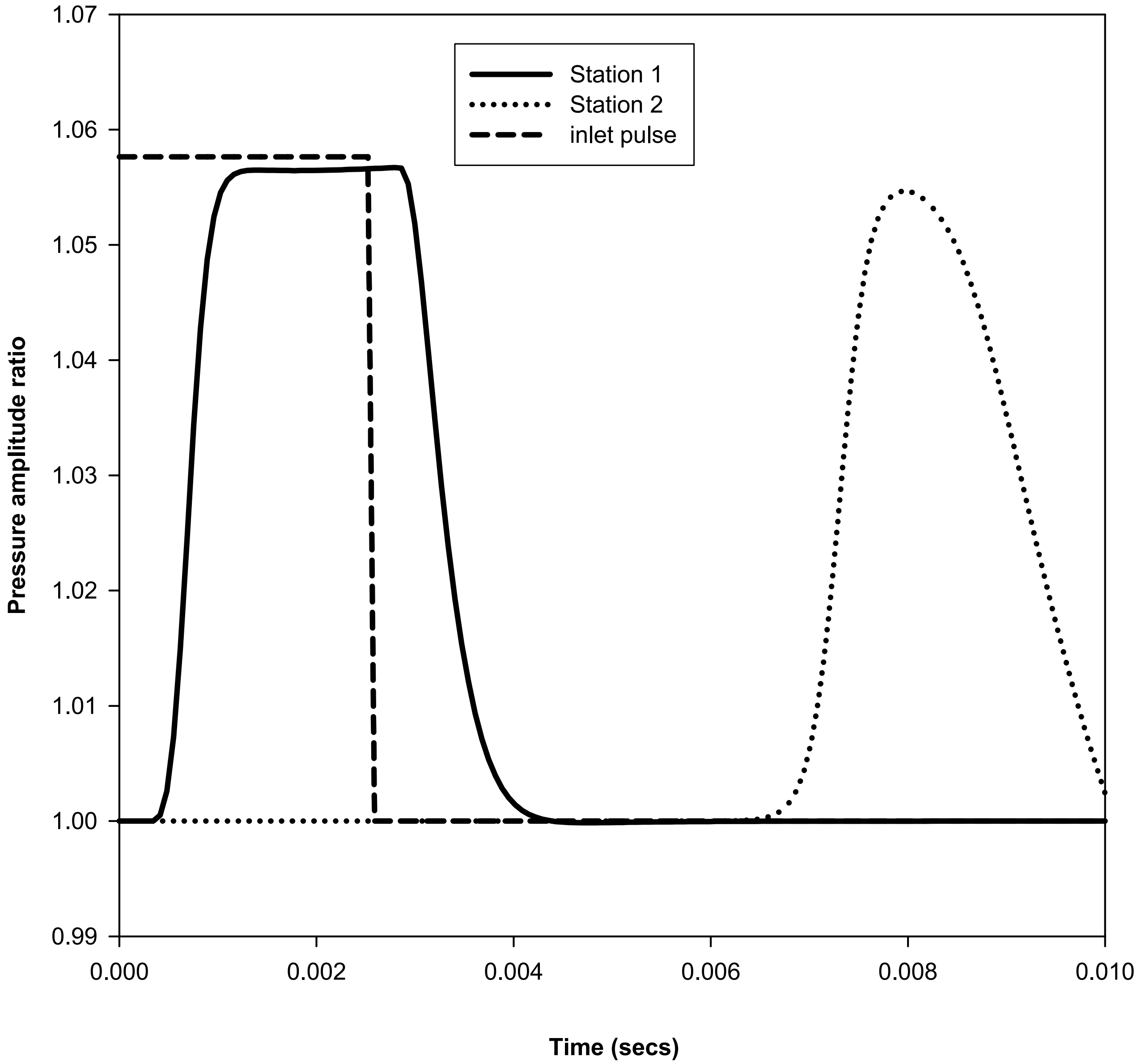


Figure 5.36 Pressure pulse propagation through the pipe

5.6 Comparison to analytical solution

Results from the transmission line analysis of Brown [14] and Karam and Franke [15] could not be directly compared to the flow simulation results because the transmission line analysis assumes a blocked line whereas the application of interest requires an open end. An analytical approach for the pressure pulse propagation through a pipe that is seldom found in the literature has been described by Emmons [39]. The analytical solution assumes an ideal pipe, neglecting friction and heat transfer. Wave reflections are not included in the analysis. However, this analytical solution provides one means of evaluating our numerical solution.

The analytical solution uses a method of analysis that assumes the pressure distribution changes only as a function of the wave velocity. The pressure amplitudes remain invariant as the wave propagates, only the distribution changes. A pure isentropic pulse of finite amplitude has an initial pressure distribution dependent only on distance. For our problem distance traveled by the wave is proportional to time. So the pressure distribution for a whole cycle is assumed as the initial distribution. The wave particle velocities are evaluated for a particular location on the wave from the pressure amplitude ratio. The density is evaluated for the ambient conditions. The formulae for evaluating the variables are described in Chapter III.

The distance traveled by the wave is calculated by $x = (c + a)\,t$. c is the particle velocity and a is the acoustic velocity. This method does not analyze reflections of pressure waves. The distance is calculated for each point on the wave using a suitable time interval. The particle and acoustic velocities are calculated corresponding to the pressure amplitude ratio using the initial pressure distribution at time $t = 0$.

A sinusoidal pulse is selected since a square pulse would not show any variation in the distribution. The mean pressure of the sinusoidal pulse is 150,000 Pa with an amplitude variation of 20%. The frequency of the pulse is 100 Hz. For comparison with the numerical solution, the particle and acoustic velocities obtained from the numerical simulation are used to evaluate the distance traveled by the wave. The numerical scheme employs the algorithm described in chapter IV. Figure 5.37 shows a plot of the analytical solution of the wave pressures as it travels through the pipe. The table of calculations can be found in Appendix C.

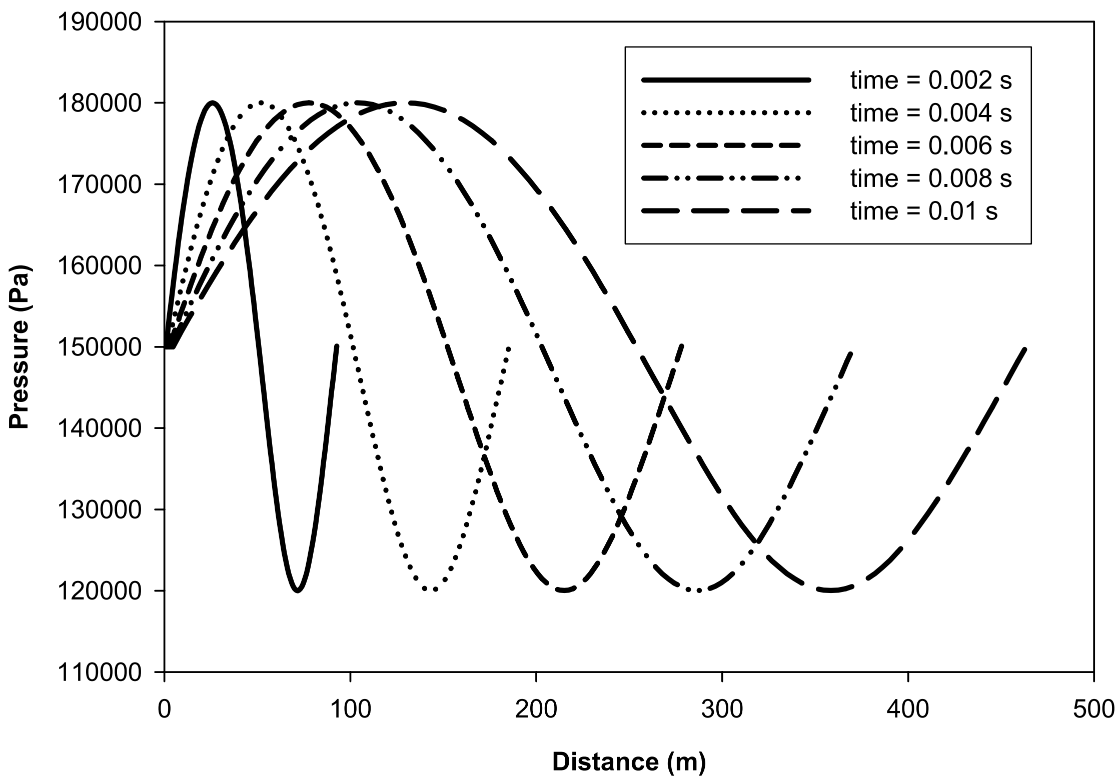


Figure 5.37 Analytical solution of Pressure wave propagation through the pipe

The analytical solution only accounts for the distortion in the pulse profile. This method does not include effects of the variation in reference temperatures and the superposition process. Hence, the pressures and velocities are lower than for the numerical solution. The distortion in pulse profile as it traverses the pipe is the focal point of comparison here. The profile distortion is handled by the analytical solution and more accurately by the numerical solution. The numerical solution involves wave reflections and the first law analysis, which evaluates the variation in reference temperatures. Hence the numerical solution should have more accuracy. The comparison between these solutions further lays stress on the importance of the numerical scheme for the analysis of the pressure pulse propagation. The analytical solution is limited to simple conditions and uses ideal assumptions. The numerical solution can handle boundary conditions that are more realistic and the algorithm can calculate the transient variation in the variables more accurately. Figure 5.38 shows the plot for the numerical case. The table of calculations can be found in Appendix C.

A better comparison can be done by plotting the two solutions on the same plot. Such a plot for two time steps of 0.002 and 0.004 seconds are shown in figures 5.39 and 5.40 respectively. Plots for other time steps are in appendix C.

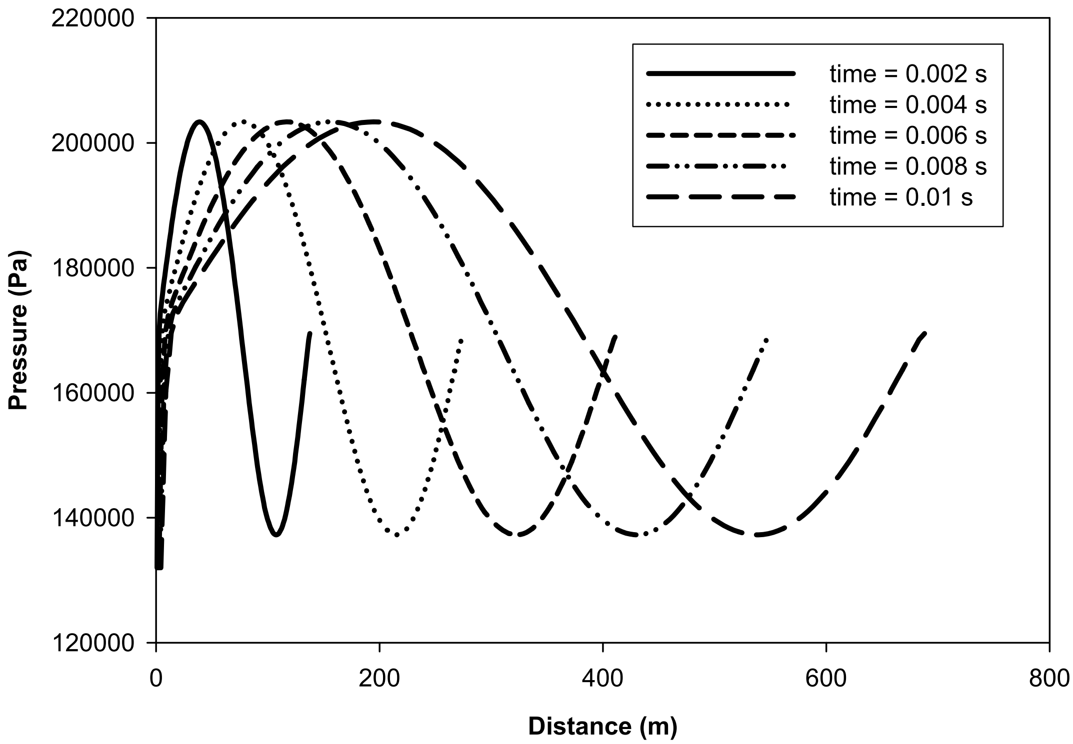


Figure 5.38 Numerical solution of Pressure wave propagation through the pipe

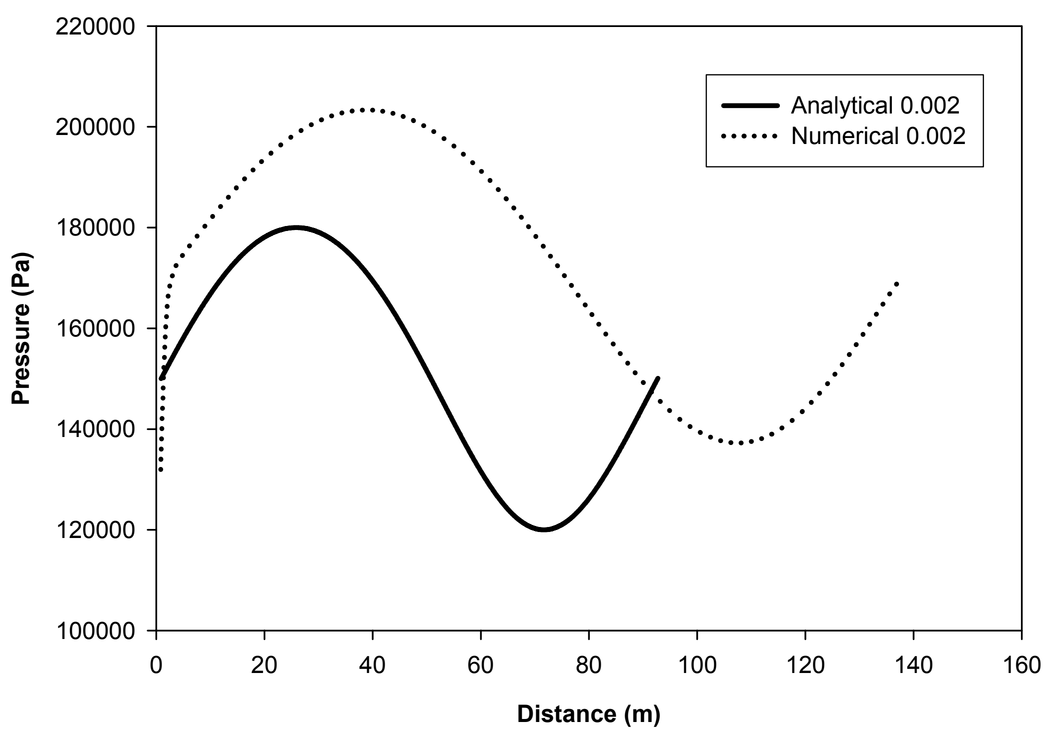


Figure 5.39 Comparison of analytical and numerical solutions for time of 0.002 seconds

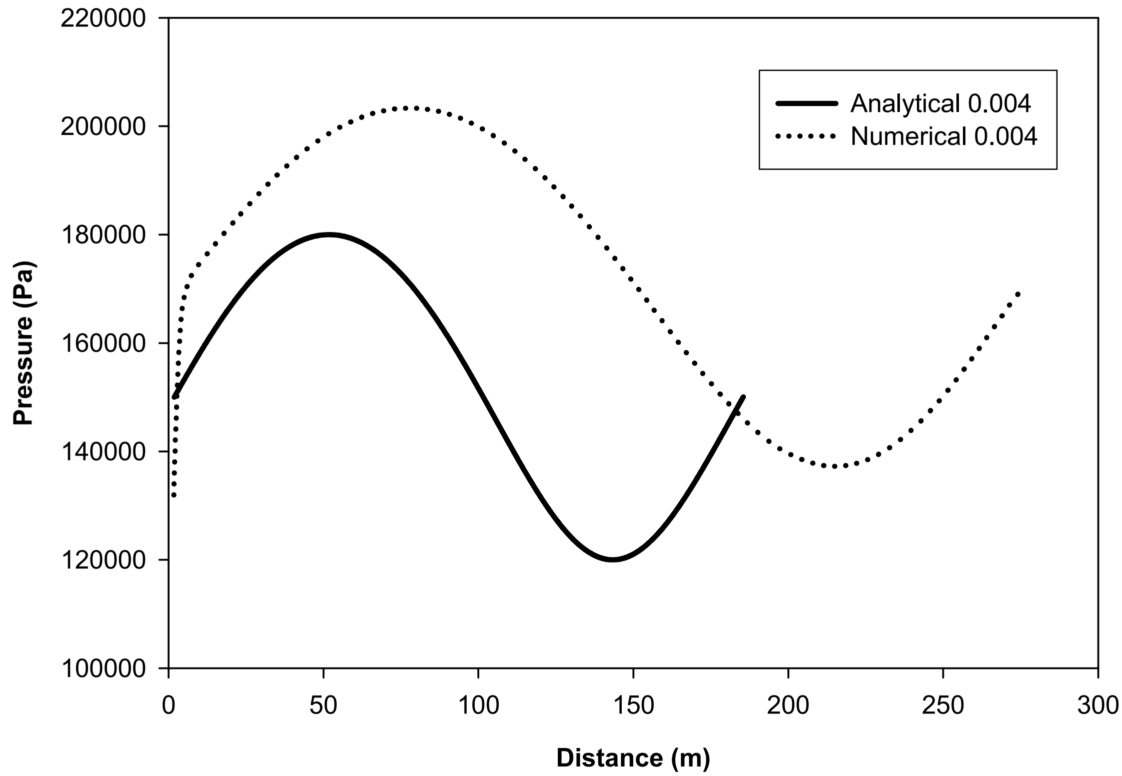


Figure 5.40 Comparison of analytical and numerical solutions for time of 0.004 seconds

A comparison of results between analytical solution and the numerical scheme serves to validate the effectiveness and accuracy of the numerical methodology. The simple wave propagation problem analyzed above shows similar trends for the two solutions. This leads us to the conclusion that the numerical scheme employed may be an effective tool to evaluate the pressure pulse propagation in a duct under realistic conditions involving friction and heat transfer.

5.7 Parameters relevant to the Control system design engineer

To design a control system correctly for the pulsed blowing system, the design engineer needs to know how each parameter can be varied to obtain the required pulse profile at the outlet of the distribution system. The following discussion is aimed at resolving these aspects of the simulation. Tests are performed varying several parameters. The inlet pressure, duty cycle, tube lengths, the expansion and contraction ratios in the geometry are the parameters considered. The model used is the same model used in the software test case discussed in earlier sections.

Pressure amplitude

Increasing the inlet pressure results in higher pressure peaks available at the exit. This is shown in figure 5.41. The inlet pressure is varied from 150000 to 200000 Pa. for the 200000 Pa case, the outlet pressure is considerably higher than the other two cases and higher peaks are available.

Simulation parameters for flow simulation code

PARAMETER	VALUE	UNIT	DESCRIPTION
Pinlet	150000, 175000, 200000	Pa	pressure at inlet of duct
γ	1.4000	-------	ratio of specific heat capacities
R	287.00	J/kgK	Characterestic gas constant
frequency - f	100.00	Hz	frequency (Hz)
wall temperature - Tw	300.00	K	Temperature of the duct wall
Duty cycle	0.1	-------	fraction of T when Xp is present
Ambient reference temperature	293	K	Initial (undisturbed) temperature
Ambient reference pressure	101325	Pa	Initial (undisturbed) pressure
Exit pressure	101325	Pa	pressure at exit of pipe

Geometric details for flow simulation code

PARAMETER	VALUE	UNIT	DESCRIPTION
L1	80.00	mm	length of segment 1
D1	6.858	mm	diameter of segment 1
L2	135.0000	mm	length of segment 2
D2	12.446	mm	diameter of segment 2
L3	80.00	mm	length of segment 3
D3	6.858	mm	diameter of segment 3

Table 5-8. Input parameters for flow simulation code for varied inlet pressure case

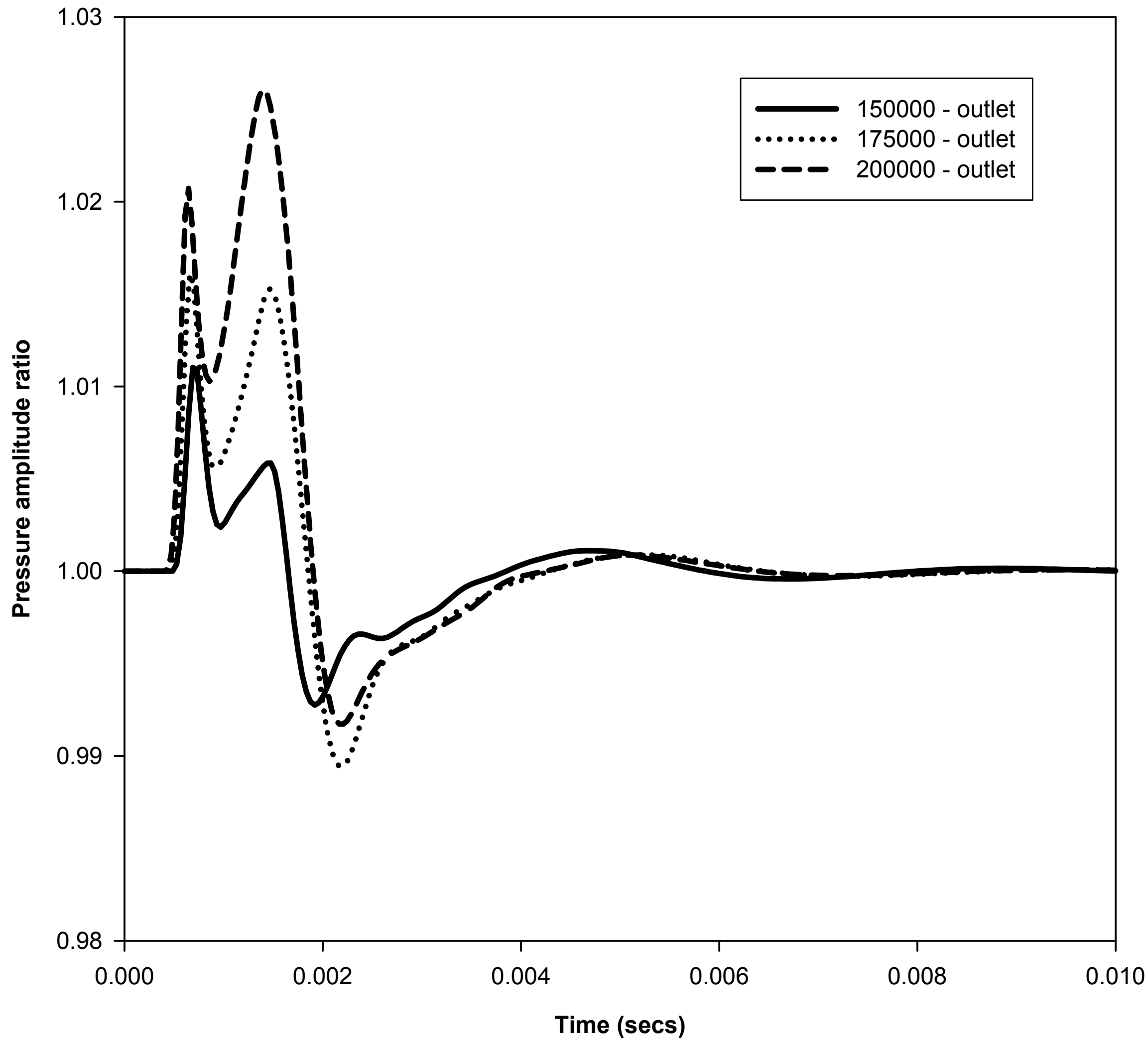


Figure 5.41 – Pressure amplitude plot for various inlet pressures

The plot in figure 5.42 shows the particle velocity variation at the outlet. The higher pressure case results in a more rounded peak and a higher velocity at the outlet. The effect of inlet pressure is to impart a higher particle velocity and results in the velocity peak available for more time.

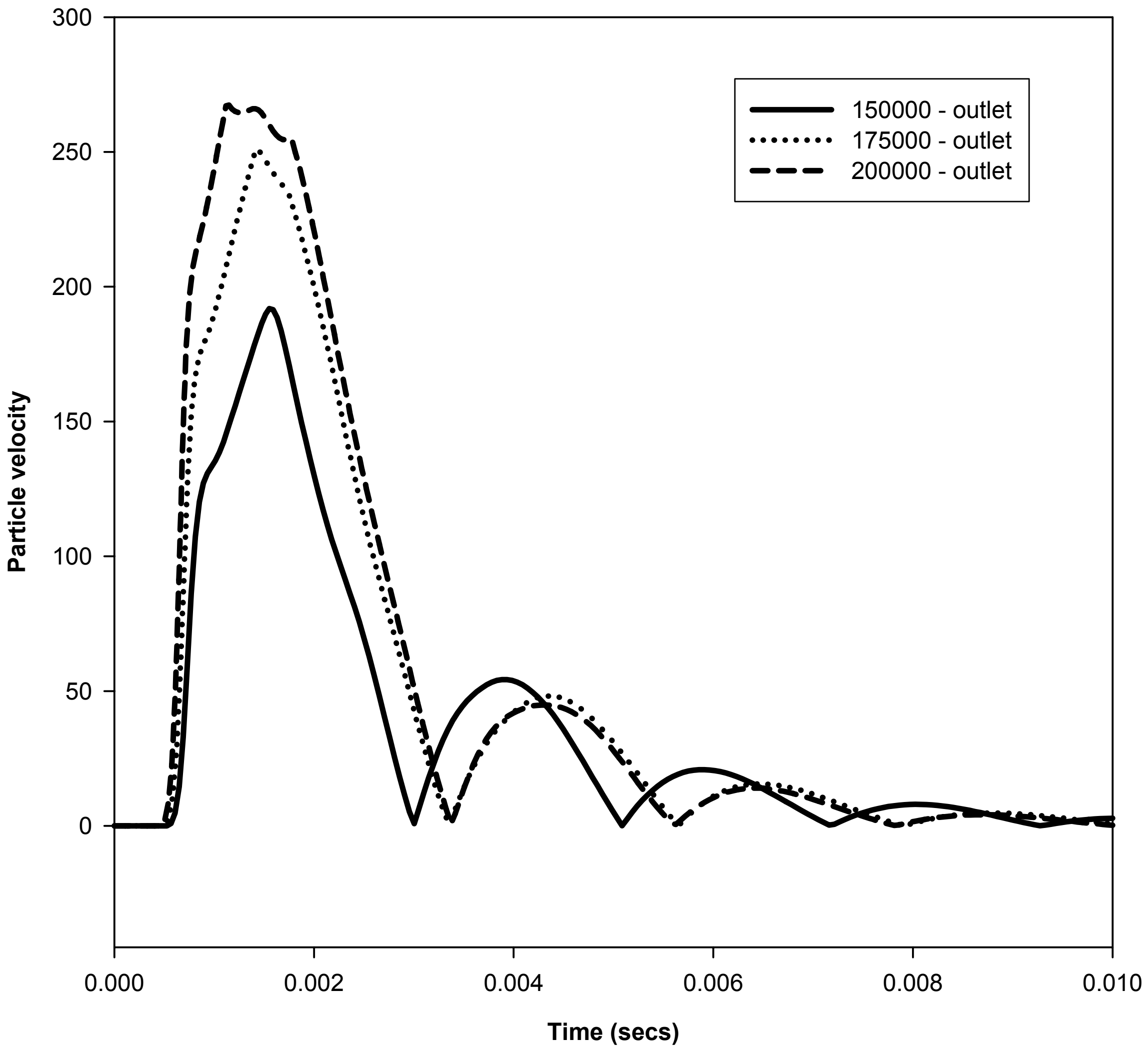


Figure 5.42 – Particle velocity plot for various inlet pressures

Duty cycle

A higher duty cycle at the inlet results in a more sustained pulse available at the exit. Aerodynamic analyses have concluded that a duty cycle in the range of 0.25-0.5 is the most effective. So several duty cycles in the range 0.1 – 0.4 were tested. Figure 5.43 shows that the 0.4 case exhibits a longer retention time for the pulse. Figure 5.44 shows that the peak velocity values are similar. However, the velocity at the outlet for the 0.4 case lasts for more time than all the other cases. A considerably higher and sustained peak pressure is available for a duty cycle of 0.4.

Simulation parameters for flow simulation code

PARAMETER	VALUE	UNIT	DESCRIPTION
Pinlet	150000.00	Pa	pressure at inlet of duct
γ	1.4000	-------	ratio of specific heat capacities
R	287.00	J/kgK	Characterestic gas constant
frequency - f	100.00	Hz	frequency (Hz)
wall temperature - Tw	300.00	K	Temperature of the duct wall
Duty cycle	0.1,0.2,0.3,0.4	-------	fraction of T when Xp is present
Ambient reference temperature	293	K	Initial (undisturbed) temperature
Ambient reference pressure	101325	Pa	Initial (undisturbed) pressure
Exit pressure	101325	Pa	pressure at exit of pipe

Geometric details for flow simulation code

PARAMETER	VALUE	UNIT	DESCRIPTION
L1	80.00	mm	length of segment 1
D1	6.858	mm	diameter of segment 1
L2	135.0000	mm	length of segment 2
D2	12.446	mm	diameter of segment 2
L3	80.00	mm	length of segment 3
D3	6.858	mm	diameter of segment 3

Table 5-9. Input parameters for flow simulation code for varied duty cycle case

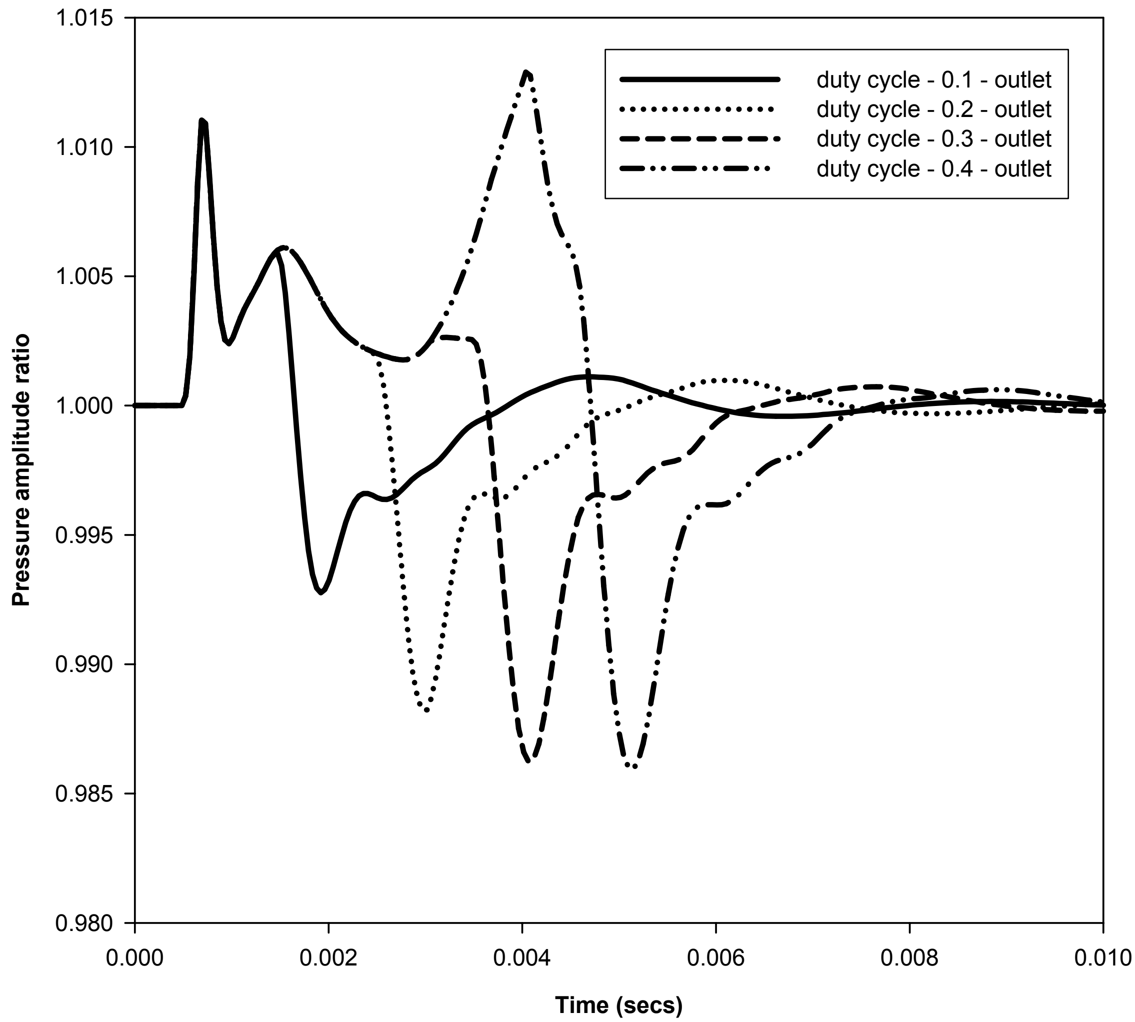


Figure 5.43 – Pressure amplitude plot for various duty cycles

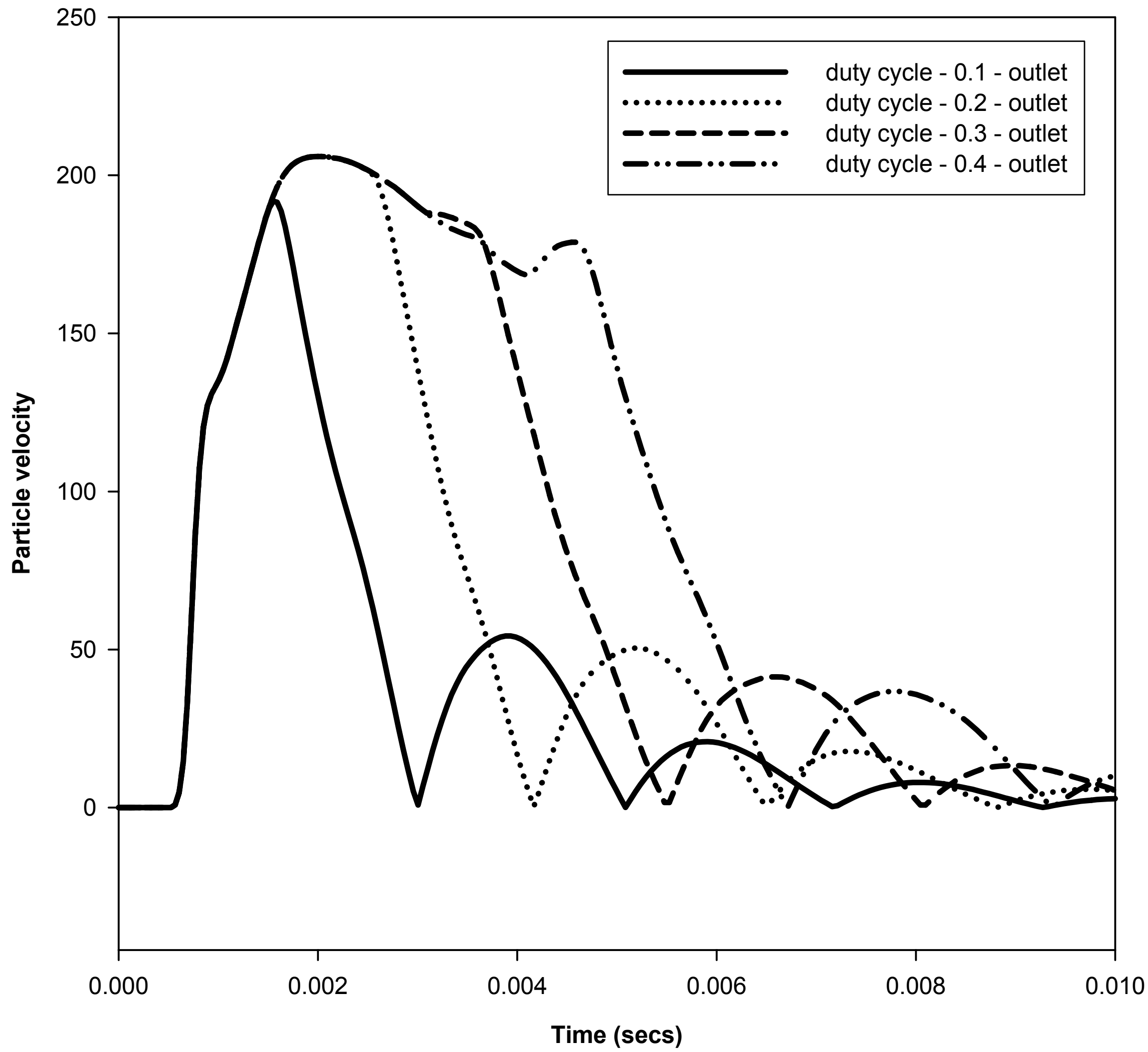


Figure 5.44 – Particle velocity plot for various duty cycles

Tube length

The effect of the tube length was investigated maintaining the other parameters constant. The input parameters used are summarized in the table 5-10.

Simulation parameters for flow simulation code

PARAMETER	VALUE	UNIT	DESCRIPTION
Pinlet	150000.00	Pa	pressure at inlet of duct
γ	1.4000	-------	ratio of specific heat capacities
R	287.00	J/kgK	Characterestic gas constant
frequency - f	100.00	Hz	frequency (Hz)
wall temperature - Tw	300.00	K	Temperature of the duct wall
Duty cycle	0.1	-------	fraction of T when Xp is present
Ambient reference temperature	293	K	Initial (undisturbed) temperature
Ambient reference pressure	101325	Pa	Initial (undisturbed) pressure
Exit pressure	101325	Pa	pressure at exit of pipe

Geometric details for flow simulation code

PARAMETER	VALUE	UNIT	DESCRIPTION
L1	80.00	mm	length of segment 1
D1	6.858	mm	diameter of segment 1
L2	135, 200, 250, 300,2000	mm	length of segment 2
D2	12.446	mm	diameter of segment 2
L3	80.00	mm	length of segment 3
D3	6.858	mm	diameter of segment 3

Table 5-10. Input parameters for flow simulation code for varied tube length case

The pressure amplitude plot in figure 5.45 shows that a longer tube attenuates the pressure pulse more. This is an expected result. The peak pressure is highest for the shortest tube.

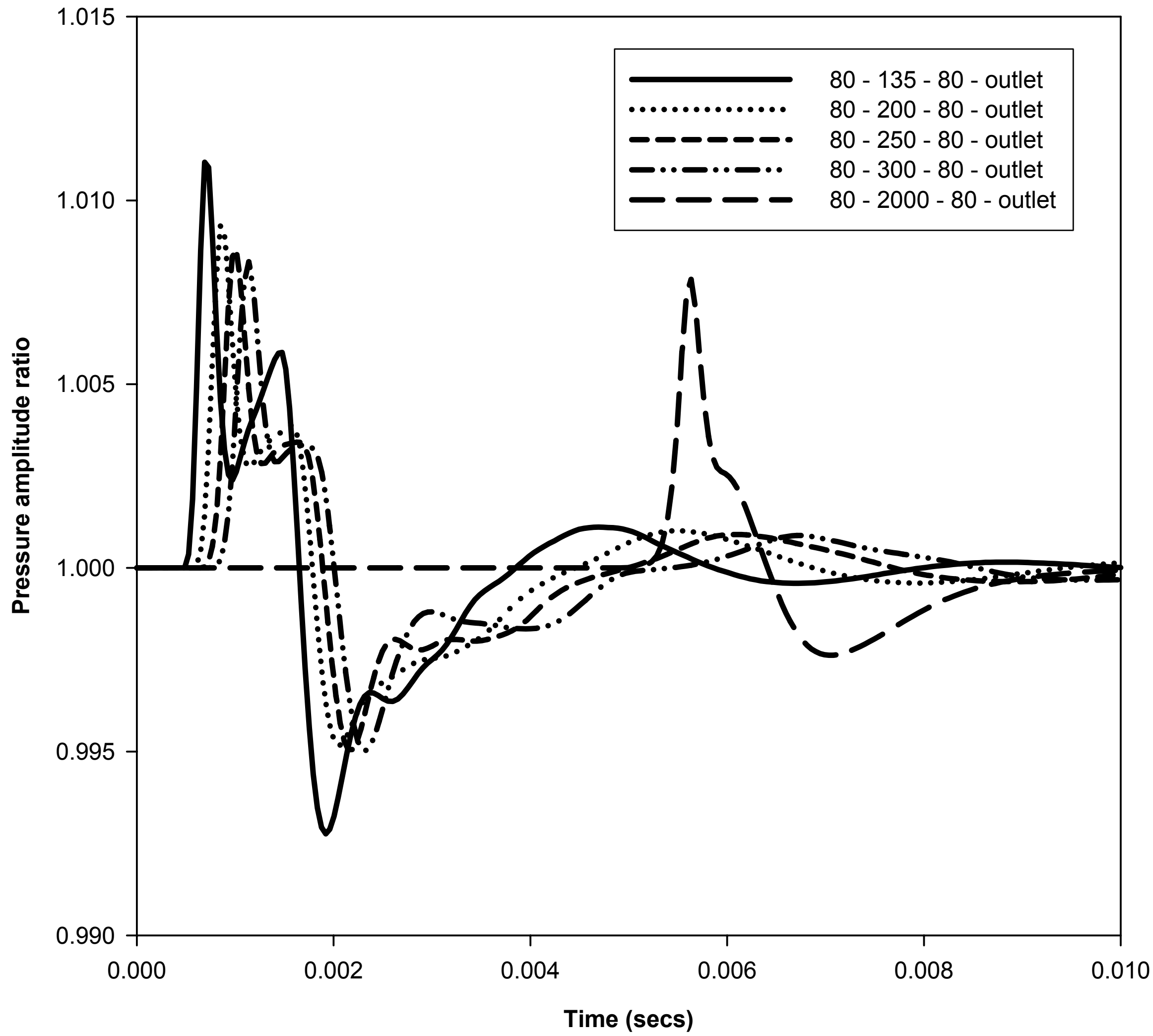


Figure 5.45 – Pressure amplitude plot for various tube lengths

The velocity plot in figure 5.46 exhibits a similar trend. The shorter tube gives a larger particle velocity at the outlet. To minimize the attenuation it is advisable to use a shorter tube for the distribution system.

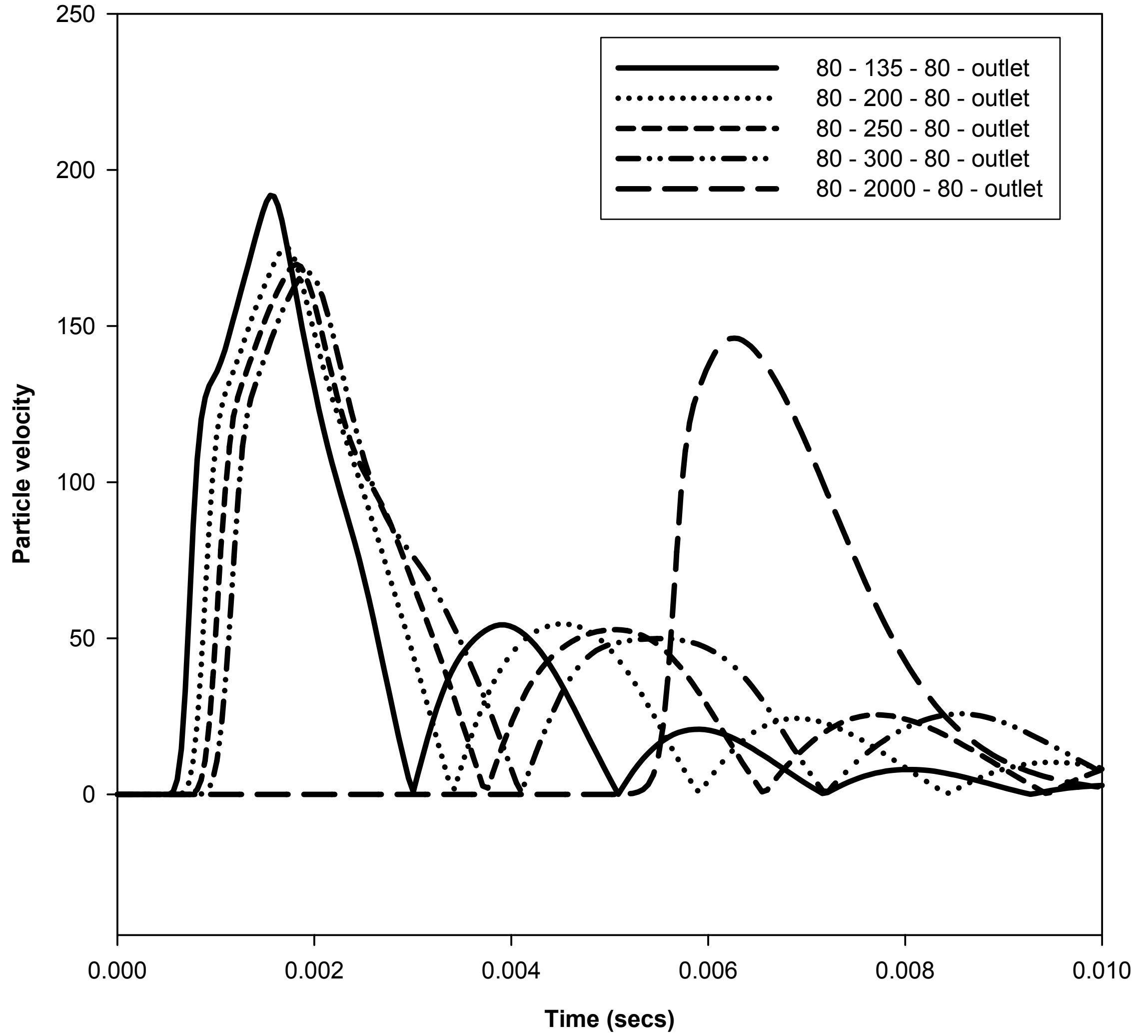


Figure 5.46 – Particle velocity plot for various tube lengths

Effect of area ratio

Several expansion and contraction ratios were tested on the software test case model. The area ratio was increased to 3, 4 and 5. The limit of the area ratio is governed by the Benson's criterion; hence, the maximum value is 6.

Simulation parameters for flow simulation code

PARAMETER	VALUE	UNIT	DESCRIPTION
Pinlet	150000.00	Pa	pressure at inlet of duct
γ	1.4000	-------	ratio of specific heat capacities
R	287.00	J/kgK	Characterestic gas constant
frequency - f	100.00	Hz	frequency (Hz)
wall temperature - Tw	300.00	K	Temperature of the duct wall
Duty cycle	0.1	-------	fraction of T when Xp is present
Ambient reference temperature	293	K	Initial (undisturbed) temperature
Ambient reference pressure	101325	Pa	Initial (undisturbed) pressure
Exit pressure	101325	Pa	pressure at exit of pipe

Geometric details for flow simulation code

PARAMETER	VALUE	UNIT	DESCRIPTION
L1	80.00	mm	length of segment 1
D1	6.858	mm	diameter of segment 1
L2	135.0000	mm	length of segment 2
D2	12.446, 20.574, 27.432, 34.29	mm	diameter of segment 2
L3	80.00	mm	length of segment 3
D3	6.858	mm	diameter of segment 3

Table 5-11. Input parameters for flow simulation code for varied area ratio case

The pressure amplitude plot in figure 5.47 indicates that higher expansion and contraction ratios cause higher attenuation. It is advisable to have a straight pipe with minimal area changes. The velocity plot indicates backflow at several periods in the cycle and higher attenuation with higher area ratio.

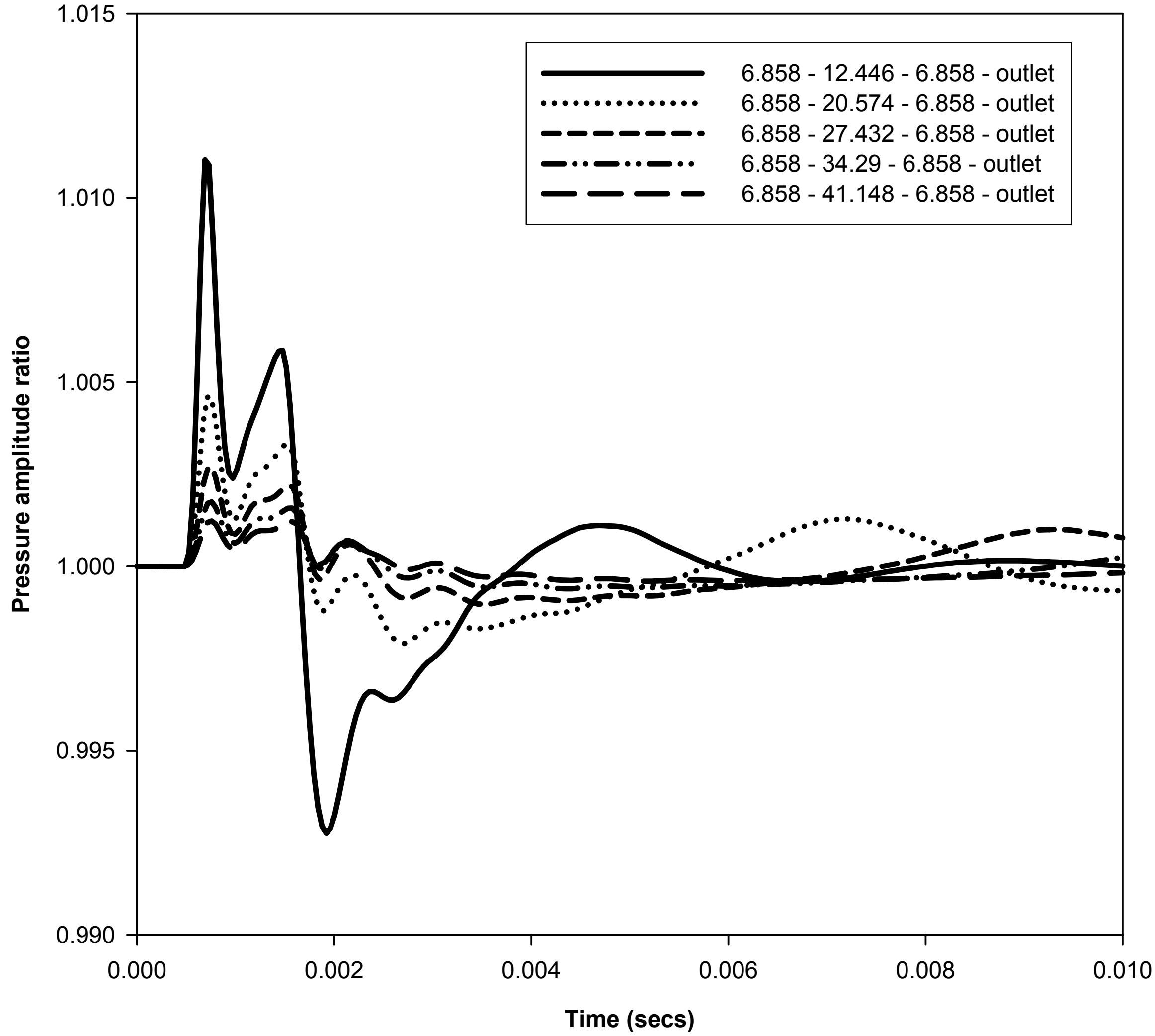


Figure 5.47 – Pressure amplitude plot for varied area ratio case

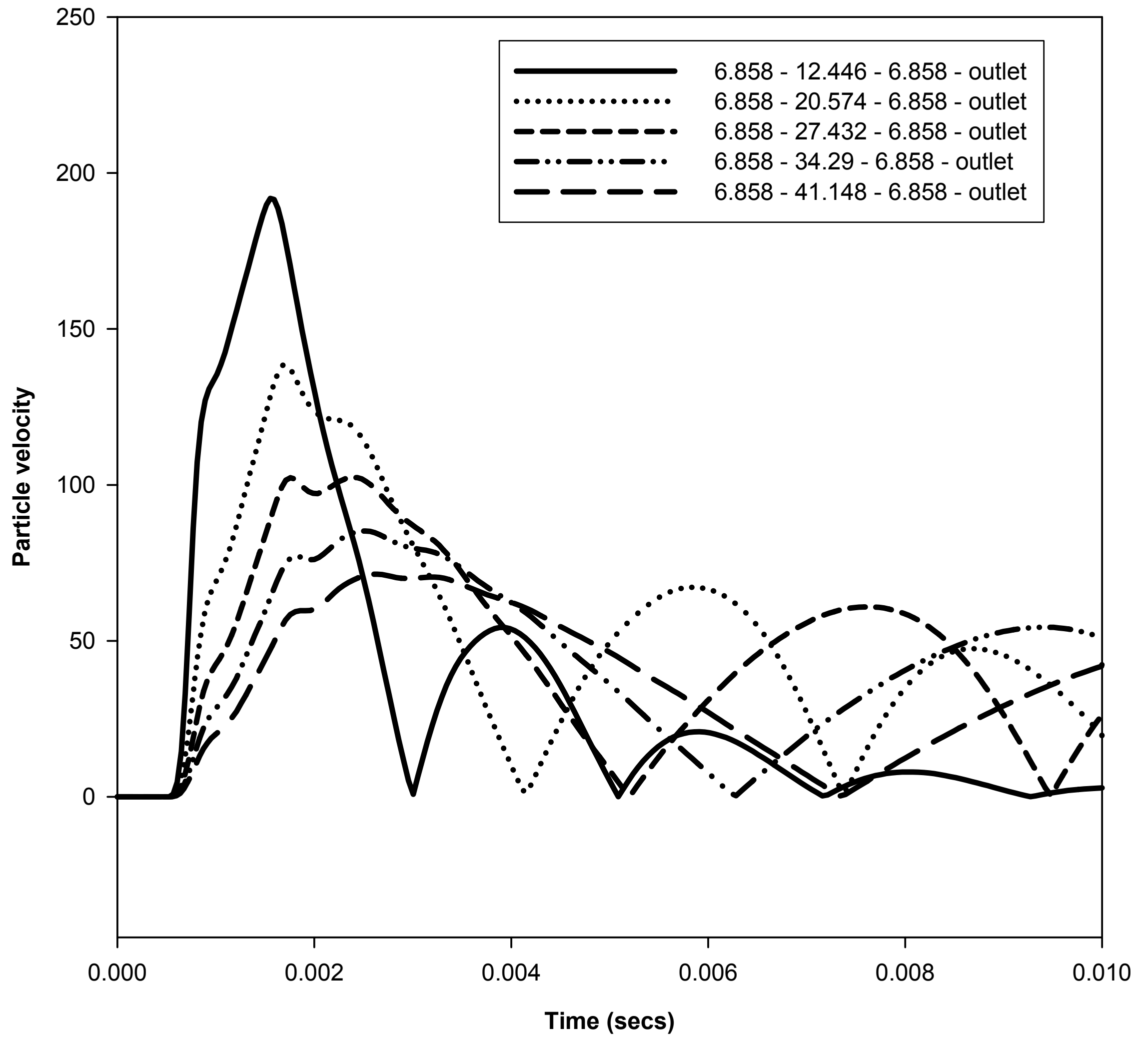


Figure 5.48 – Particle velocity plot for varied area ratio case

CHAPTER VI

CONCLUSION AND RECOMMENDATIONS

In this study, a one-dimensional code is used to analyze unsteady gas flow through ducts. The method is adopted from the analysis techniques used for design and simulation of internal combustion engines. The methodology uses a mesh method of interpolation. The model developed by Dr. G.P. Blair at the Queens University of Belfast is called the GPB model. This model in contrast to earlier methods uses a commercial code - Virtual 4 Stroke® software also developed by Dr. G.P. Blair. This code is used to analyze a pulsed blowing system by simulating an engine giving the necessary pressure pulse at the exhaust port. This analysis is used to study the feasibility of developing a suitable flow simulation code for the specific research problem. Results obtained were encouraging. Therefore, a general-purpose code that can simulate a straight pipe and a straight pipe with one expansion and one contraction was developed. Two test cases described in literature for validation of codes for design of internal combustion engines are used for evaluation of the trends predicted by this code. Effects of parameters like mesh size, time step and duty cycle that affect the solution are discussed with appropriate test cases. The mesh independency issue is addressed by solving the second test case using three different mesh sizes. The attenuation of the pulse amplitude and the distortion of the pulse profile is explained with simulation results.

Summary

The GPB model provides a sufficient explanation of the pulse propagation through the pipe. The dynamic nature of the problem is well captured by this modeling method. The computation time is less compared to other methods and the method appears to predict proper trends. The method is also easier to implement on a digital computer than older methods.

The pulse attenuation and distortion can be visualized in the plots as discussed in section 5.5. This program allows computations to be performed to develop improved exit pulse characteristics. The dynamic nature of unsteady flow is well captured in the model and the variation in properties of the fluid is also reflected in the solution. The simulations show that the pulse shape is not constant as it propagates through the tube. Thus the assumption that the pulse shape is maintained throughout the tube is not true. The exit pulse characteristics are important input for aerodynamic calculations and modeling of the active flow control system. These are difficult to measure experimentally due to the high velocities involved. In this context, this analysis tool is of importance to the designer of the control system in performing design optimization.

More complex algorithms can be developed to model complex geometry. Bends and branches can also be effectively modeled by this method. This is not included in this work and is recommended for future research efforts. The code analyzes the pulse for a time equivalent to one complete cycle of the pulse. This can be modified to cover few more pulse cycles. At the end of the time step the program returns to the location where the pulse generator routine is called. The pulse generator routine needs to be rewritten so that it can evaluate the aggregate time and assign appropriate pressure amplitude ratio values to the left end. The amplitude ratios need to be checked with previous iterations and a suitable convergence criterion needs to be set to end the simulation if the criterion is satisfied. This necessitates that either all the arrays for the calculated variables be changed to two-dimensional ones or all the variable values be stored in temporary variables for comparison to the values in the subsequent time step. If the results need to be reviewed periodically, additional print statements may be needed to print results to new sheets. Further, if the solution diverges, precaution needs to be taken to avoid an infinite loop.

6.1 Conclusions

Investigations in the present study lead to the following major conclusions

(1) A one-dimensional model is suitable for analysis of pulsed blowing systems. The GPB method has advantages of simplicity, ease of modeling and accuracy.

(2) The computational model is already established and experimentally validated for internal combustion engines, hence is acceptable for this application.

(3) The code makes it possible to evaluate the unsteady system parameters where accurate measurement by instruments is difficult.

(4) The code predicts changes in pulse amplitude and shape that agree with trends seen in engine simulation computations.

(5) The output from this simulation can be used for further aerodynamic design calculations.

(6) Like any other computational fluid dynamics code, this code also has inherent limitations. The stability and accuracy of the solution is dependent on input parameters like mesh size, pressure and duty cycle.

6.2 Recommendations

(1) The user may perform numerical experiments with the code to explore further into factors affecting the solution. This will help in refining the code and produce better results. For example, the influence of mesh size on the final solution may be improved by using a different meshing scheme.

(2) The code analyzes the pulse for a time equivalent to one complete cycle of the pulse. This can be modified to cover few more pulse cycles. This needs a more complicated algorithm. The entire set of parameters needs to be stored in temporary variables and used as input to the next time step. The required steps for the modification of the program have already been outlined in the summary section.

(3) Experimental validation for this application is recommended. This could help improve the solution algorithm.

REFERENCES

1. Gad-el-Hak, M., 2001, "Flow control: The Future", *Journal of Aircraft*, Vol. 38, No. 3, pp. 402-415.
2. Gad-el-Hak, M., 2000, *Flow Control – Passive, Active and Reactive Flow Management*, Cambridge University Press, Cambridge, UK.
3. Magill, J.C. and McManus, K.R., 2001, "Exploring the Feasibility of Pulsed Jet Separation Control for Aircraft Configurations", *Journal of Aircraft*, Vol. 38, No.1, pp. 48 -56.
4. Yi Liu, "Numerical Simulations of the Aerodynamic Characteristics of Circulation Control Wing Sections", PhD Thesis, School of Aerospace Engineering, Georgia Institute of technology, April 2003.
5. Kim, B-H., "Modeling Pulsed-Blowing Systems for Active Flow Control", PhD Thesis, Mechanical and Aerospace Engineering, Illinois Institute of Technology, May 2003.
6. Jones, G.S., Viken, S.A., Washburn, A.E., Jenkins, L.N., and Cagle, C.M., 2002, "An Active Flow Circulation Controlled Flap Concept for General Aviation Aircraft Applications," AIAA Paper 2002-3157.
7. Yi Liu, Sankar, L.N., Englar, R.J., Ahuja, K.K., 2001, "Numerical simulations of the steady and unsteady aerodynamic characteristics of a circulation control wing Airfoil," AIAA Paper 2001-0704.
8. Sellers, W.L., III, Jones, G.S., and Moore, M.D., 2002, "Flow Control Research at NASA Langley in Support of High-Lift Augmentation," AIAA Paper 2002-6006.
9. Munjal, M.L., 1987, *Acoustics of Ducts and Mufflers with Application to Exhaust and Ventilation System Design*, John Wiley & Sons, New York.
10. Munjal, M.L., and Doige, A.G., 1991, "On a General Method for Modelling Multi-Source, Multi-Branch, One-Dimensional Acoustical Systems", *Acustica*, Vol. 73, pp. 37-39.
11. Gupta, V.H., Easwaran, V., and Munjal, M.L., 1995, "A Modified Segmentation Approach for Analyzing Plane Wave Propagation in Non-Uniform Ducts with Mean Flow," *Journal of Sound and Vibration*, Vol. 182, No. 5, pp. 697-707.
12. Bulaty, T., and Widenhorn, M., 1993, "Unsteady Flow Calculation of Sophisticated Exhaust Systems Using a Multibranch Junction Model," *Journal of Engineering for Gas Turbines and Power*, Vol. 115, pp. 756-760.

13. Blair, G.P., 1999, *Design and Simulation of Four-Stroke Engines*, Society of Automotive Engineers, Inc., Warrendale, PA.

14. Brown, F.T., 1962, "The Transient Response of Fluid Lines," *J. Basic Engineering, Trans. ASME*, Series D, Vol. 84, pp. 547-553.

15. Karam, J.T., and Franke, M.E., 1967, "The Frequency Response of Pneumatic Lines," *J. Basic Engineering, Trans. ASME*, Series D, Vol. 90, No.2, pp. 371-378.

16. Nichols, N.B., 1962, "The Linear Properties of Pneumatic Transmission Lines," *Trans. of the Instrument Society of America*, Vol. 1, pp. 5-14.

17. Iberall, A.S., 1950, "Attenuation of Oscillatory Pressures in Instrument Lines," *Journal of Research of the National Bureau of Standards*, Vol. 45, pp. 85-108.

18. Tijdeman, H., 1975, "On the Propagation of Sound Waves in Cylindrical Tubes," *Journal of Sound and Vibration*, Vol. 39, pp. 1-33.

19. Hunt, F.V., "Propagation of Sound in Fluids," *American Institute of Physics Handbook*, McGraw-Hill Book Company, Inc., New York, N.Y., 1957.

20. Joslin, R.D., Horta, L.G., Chen, F.J., "Transitioning Active Flow Control to Applications," 30th AIAA Fluid Dynamics Conference, Norfolk, VA, June 28 – July 1,1999, AIAA Paper 99-3575.

21. Perotti, M., 1999, "Speed-up of a DFEM code for unsteady gas dynamics in pipes," *International Journal of Mechanical Sciences*, Vol. 41, pp. 793-813.

22. Gad-el-Hak, M., and Bushnell, D.M., 1991, "Separation Control: Review," *Journal of Fluids Engineering*, Vol. 113, No. 1, pp. 5-30.

23. Choi, H., Moin, P., and Kim, J., 1994, "Active Turbulence Control for Drag Reduction in Wall-Bounded Flows," *Journal of Fluid Mechanics*, Vol. 262, pp. 75-110.

24. Lord, W.K., MacMartin, D.G., and Tillman, T.G., 2000, "Flow Control Opportunities in Gas Turbine Engines," AIAA Paper 2000-2234.

25. Wilson, K.J., Gutmark, E., Schadow, K.C., Smith, R.A., 1995, "Feedback Control of a Dump Combustor with Fuel Modulation," *Journal of Propulsion and Power*, Vol. 11, No. 2, pp. 268-274.

26. Annaswamy, A.M., and Ghoniem, A.F., 2002, "Active Control of Combustion Instability: Theory and Practice," *IEEE Control Systems Magazine*, December 2002, pp. 37-54.

27. Bulaty, T., Niessner, H., 1985, "Calculation of 1-D Unsteady Flows in Pipe systems of I.C.Engines," *Journal of Fluids Engineering*, Vol. 107, pp. 407-412.

28. Benson, R.S., *The Thermodynamics and Gas Dynamics of International Combustion Engines*, (ed. J.H.Horlock, D.E.Winterbone), Vols.1 and 2, Clarendon Press, Oxford, 1982.

29. Anderson, Jr, J., D., *Computational Fluid Dynamics – The Basics with applications*, McGraw-Hill, Inc. 1995.

30. Vandevoorde, M., Vierendeels, J., Sierens, R., Dick, E., Baert, R., 2000, "Comparison of Algorithms for Unsteady Flow Calculations in Inlet and Exhaust Systems of IC Engines," *Journal of Engineering for Gas Turbines and Power*, Vol. 122, pp. 541-548.

31. Gascon, LI., Garcia, J.A., 2003, "About the TVD property for the flux-corrected transport techniques," *Journal of Computational Mechanics*, Vol. 30, pp 281-285.

32. Kirkpatrick, S.J., Blair, G.P., Fleck, R., McMullan, R.K., "Experimental Evaluation of 1-D Computer Codes for the Simulation of Unsteady Gas Flow Through Engines- A First Phase," SAE International Off-Highway and Power plant Congress, Milwaukee, Wisconsin, September 14-16, 1994, SAE Paper No.941685, pp.77-96.

33. Blair, G.P., Kirkpatrick, Fleck, R., "Experimental Validation of a 1-D Modeling Code for a pipe containing Gas of Varying Properties," SAE International Congress, Detroit, Michigan., February 28 - March 3, 1995, SAE Paper No. 950275, p.14.

34. Blair, G.P., Kirkpatrick, S.J., Mackey, D.O., Fleck, R., "Experimental Validation of a 1-D Modeling Code for a Pipe system containing Area Discontinuities," SAE International Congress, Detroit, Michigan., February 28 - March 3, 1995, SAE Paper No. 950276, p.16.

35. Blair, G.P., 1991, "An Alternative Method for the Prediction of Unsteady Gas Flow Through the Internal Combustion Engine" SAE Paper No. 911850.

36. Aksel, M.H., Eralp, O.C., *Gas Dynamics*, Prentice Hall, 1994.

37. Schlichting, H., *Boundary Layer Theory*, 7th edition, McGraw-Hill, New York, 1979.

38. Courant, R., Friedrichs, K., Lewy, H., Translation Report NYO – 7689, Mathematische Annalen, Vol.100, p. 32, 1928.

39. Emmons, H.W., 1958, *Fundamentals of Gas Dynamics*, Princeton University press, Princeton, New Jersey.

APPENDIX A

Derivation of governing equations

Sudden expansion

The governing equations for the sudden expansion case shown in figure A.1 are applied to the superposition stations 1 and 2. The particle flow pattern is shown in figure A.2.

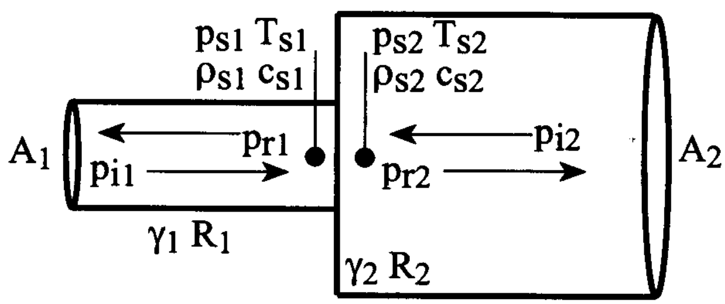


Figure A.1 Sudden expansion in a pipe [13]

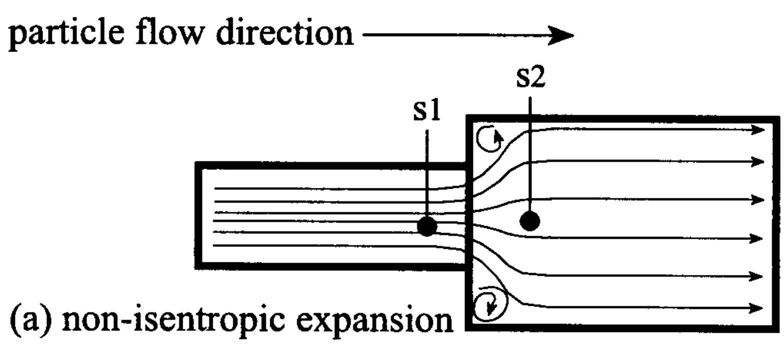


Figure A.2 Particle flow in a sudden expansion in a pipe [13]

Continuity

The continuity equation for a quasi-steady one-dimensional flow can be written as

$$\dot{m}_1 = \dot{m}_2 \tag{A-1}$$

Mass flow rate = density x area x particle velocity = $\rho \times A \times c$ (A-2)

$$\rho = \rho_0 \times X \tag{A-3}$$

$$c = G_5 a_0 \left(X_i - X_r \right) \tag{A-4}$$

With the right direction retained as positive, the continuity equation becomes

$$\rho_{01} X_{s1}^{G_5} A_1 G_5 a_{01} \left(X_{i1} - X_{r1} \right) + \rho_{02} X_{s2}^{G_5} A_2 G_5 a_{02} \left(X_{i2} - X_{r2} \right) = 0 \tag{A-5}$$

$$X_s = X_i + X_r - 1 \tag{A-6}$$

The continuity equation reduces to

$$\rho_{01} \left(X_{i1} + X_{r1} - 1 \right)^{G_5} A_1 G_5 a_{01} \left(X_{i1} - X_{r1} \right) + \rho_{02} \left(X_{i2} + X_{r2} - 1 \right)^{G_5} A_2 G_5 a_{02} \left(X_{i2} - X_{r2} \right) = 0$$

(A-7)

This is the equation solved in section 3.4.5.4 - equation 3-69.

Momentum

The momentum equation for flow from superposition station 1 to superposition 2 can be expressed as

$$A_1 p_{s1} + \left(A_2 - A_1 \right) p_{s1} - A_2 p_{s2} + \left(\dot{m}_{s1} c_{s1} - \dot{m}_{s2} c_{s2} \right) = 0 \tag{A-8}$$

Here the pressure p_{s1} is assumed to act over the annulus area $\left(A_2 - A_1 \right)$. Combining continuity and momentum equations, the momentum equation reduces to

$$A_2\left(p_{s1}-p_{s2}\right)+\dot{m}_{s1}\left(c_{s1}-c_{s2}\right)=0 \tag{A-9}$$

From definitions

$$p_s = p_0 X_s^{G_7} \tag{A-10}$$

$$X_{s1}=\left(X_{i1}+X_{r1}-1\right)^{G_7} \tag{A-11}$$

$$X_{s2}=\left(X_{i2}+X_{r2}-1\right)^{G_7} \tag{A-12}$$

$$m_s = \rho_0 X_s A c_s \tag{A-13}$$

$$c_{s1}=G_5 a_{01}\left(X_{i1}-X_{r1}\right) \tag{A-14}$$

$$c_{s2}=G_5 a_{02}\left(X_{i2}-X_{r2}\right) \tag{A-15}$$

Expanding each term in m_s we get

$$m_{s1}=\rho_{01}\left(X_{i1}+X_{r1}-1\right)^{G_5} A_1 G_5 a_{01}\left(X_{i1}-X_{r1}\right) \tag{A-16}$$

Substituting all these terms in equation (A-9) the expression for the momentum equation becomes

$$\begin{aligned}&p_0 A_2\left[\left(X_{i1}+X_{r1}-1\right)^{G7}-\left(X_{i2}+X_{r2}-1\right)^{G7}\right]+\\&\left[\rho_{01}\left(X_{i1}+X_{r1}-1\right)^{G5} A_1 G_5 a_{01}\left(X_{i1}-X_{r1}\right)\right]\times\left[G_5 a_{01}\left(X_{i1}-X_{r1}\right)+G_5 a_{02}\left(X_{i2}-X_{r2}\right)\right]=0\end{aligned} \tag{A-17}$$

This is the equation solved in section 3.4.5.4 - equation 3-70.

Energy equation (first law of thermodynamics)

The first law of thermodynamics for an open system can be written as

$$\delta Q_{system} + \Delta m_1 \left(h_1 + \frac{c_1^2}{2} \right) = dE_{system} + \Delta m_2 \left(h_2 + \frac{c_2^2}{2} \right) + \delta W_{system} \tag{A-18}$$

Summary of gas flow through a sudden expansion [28]

When gas flows steadily through a sudden enlargement, its static pressure drops slightly and then increases gradually to a maximum value. This pressure difference is called the pressure recovery. The location at which this occurs is called the plane of recovery. For non-steady flow, the plane of recovery moves very close to the geometric area change. This area change zone is of finite length. However, for analysis purposes, the assumption is that the plane of recovery is just downstream of the enlargement and that a quasi-steady model is satisfactory. Therefore, the heat transfer, internal energy and flow work terms disappear from the energy equation. Applying the continuity relation to the energy equation, the equation reduces to

$$h_{s1} + \frac{c_{s1}^2}{2} = h_{s2} + \frac{c_{s2}^2}{2} \tag{A-19}$$

$$h_{s1} - h_{s2} = C_p \left(T_{s1} - T_{s2} \right) = \frac{\gamma R}{\gamma - 1} \left(T_{s1} - T_{s2} \right) = \frac{a_{s1}^2 - a_{s2}^2}{\gamma - 1} \tag{A-20}$$

Substituting this condition in (A-19)

$$\left(c_{s1}^2 + G_5 a_{s1}^2 \right) - \left(c_{s2}^2 + G_5 a_{s2}^2 \right) = 0 \tag{A-21}$$

$$c_{s1} = G_5 a_{01} \left(X_{i1} - X_{r1} \right) \quad \text{and} \quad c_{s2} = G_5 a_{02} \left(X_{i2} - X_{r2} \right) \tag{A-22}$$

$$a_{s1} = a_{01} \left(X_{i1} - X_{r1} \right) \quad \text{and} \quad a_{s2} = a_{02} \left(X_{i2} - X_{r2} \right) \tag{A-23}$$

Therefore, the energy equation (A-21) becomes

$$\left[\left(G_5 a_{01}\left(X_{i1}-X_{r1}\right)\right)^2+G_5 a_{01}^2\left(X_{i1}+X_{r1}-1\right)^2\right]-\left[\left(G_5 a_{02}\left(X_{i2}-X_{r2}\right)\right)^2+G_5 a_{02}^2\left(X_{i2}+X_{r2}-1\right)^2\right]=0$$

This is the equation solved in section 3.4.5.4 - equation 3-71. (A-24)

The unknowns are X_{r1}, X_{r2} and a_{02}.

Sudden contraction

The governing equations for the sudden contraction case shown in figure A.3 are applied to the superposition stations 1 and 2. The particle flow pattern is shown in figure A.4. Unlike a sudden expansion, the contracting flow is assumed to smoothly move from station 1 to 2 without turbulent vortices and particle flow separation. So the flow is assumed isentropic.

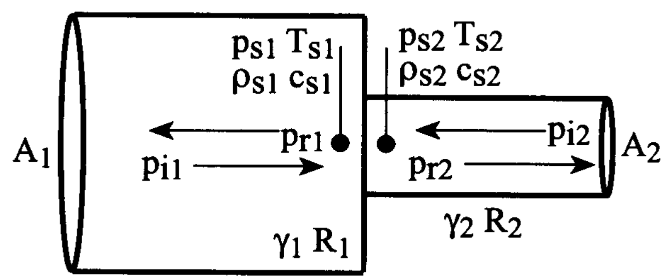


Figure A.3 Sudden contraction in a pipe [13]

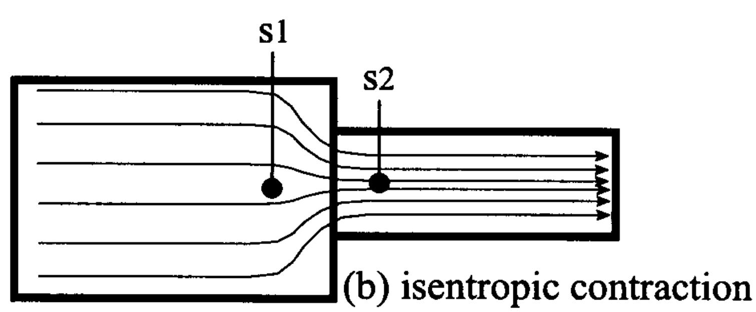


Figure A.4 Particle flow in a sudden contraction in a pipe [13]

With the flow assumed isentropic,

$$T_{01} = T_{02} \tag{A-25}$$

and

$$a_{01} = a_{02} \tag{A-26}$$

Therefore, the number of unknowns reduces to two. The unknowns are X_{r1} and X_{r2}.

<u>Continuity</u>

The continuity equation for a quasi-steady one-dimensional flow can be written as

$$\dot{m}_1 = \dot{m}_2 \tag{A-27}$$

Mass flow rate = density x area x particle velocity = $\rho \times A \times c$ (A-28)

$$\rho = \rho_0 \times X \tag{A-29}$$

$$c = G_5 a_0 \left(X_i - X_r \right) \tag{A-30}$$

With the right direction retained as positive, the continuity equation becomes

$$\rho_{01} X_{s1}^{G_5} A_1 G_5 a_{01} \left(X_{i1} - X_{r1} \right) + \rho_{02} X_{s2}^{G_5} A_2 G_5 a_{02} \left(X_{i2} - X_{r2} \right) = 0 \tag{A-31}$$

$$X_s = X_i + X_r - 1 \tag{A-32}$$

The continuity equation reduces to

$$\rho_{01} \left(X_{i1} + X_{r1} - 1 \right)^{G_5} A_1 G_5 a_{01} \left(X_{i1} - X_{r1} \right) + \rho_{02} \left(X_{i2} + X_{r2} - 1 \right)^{G_5} A_2 G_5 a_{02} \left(X_{i2} - X_{r2} \right) = 0$$

(A-33)

Simplifying applying the condition $a_{01} = a_{02}$

$$\left(X_{i1} + X_{r1} - 1 \right)^{G5} A_1 \left(X_{i1} - X_{r1} \right) + \left(X_{i2} + X_{r2} - 1 \right)^{G5} A_2 \left(X_{i2} - X_{r2} \right) = 0 \tag{A-34}$$

This is the equation solved in section 3.4.5.4 - equation 3-78.

Energy equation (first law of thermodynamics)

The first law of thermodynamics for an open system can be written as

$$\delta Q_{system} + \Delta m_1\left(h_1 + \frac{c_1^2}{2}\right) = dE_{system} + \Delta m_2\left(h_2 + \frac{c_2^2}{2}\right) + \delta W_{system} \tag{A-35}$$

Summary of gas flow through a sudden contraction [28]

Applying a quasi-steady approach as in sudden expansion, the heat transfer, internal energy and flow work terms disappear from the energy equation. Applying the continuity relation to the energy equation, the equation reduces to

$$h_{s1} + \frac{c_{s1}^2}{2} = h_{s2} + \frac{c_{s2}^2}{2} \tag{A-36}$$

$$h_{s1} - h_{s2} = C_p\left(T_{s1} - T_{s2}\right) = \frac{\gamma R}{\gamma - 1}\left(T_{s1} - T_{s2}\right) = \frac{a_{s1}^2 - a_{s2}^2}{\gamma - 1} \tag{A-37}$$

Substituting this condition in (A-36)

$$\left(c_{s1}^2 + G_5 a_{s1}^2\right) - \left(c_{s2}^2 + G_5 a_{s2}^2\right) = 0 \tag{A-38}$$

$$c_{s1} = G_5 a_{01}\left(X_{i1} - X_{r1}\right) \tag{A-39}$$

$$c_{s2} = G_5 a_{02}\left(X_{i2} - X_{r2}\right) \tag{A-40}$$

$$a_{s1} = a_{01}\left(X_{i1} - X_{r1}\right) \tag{A-41}$$

$$a_{s2} = a_{02}\left(X_{i2} - X_{r2}\right) \tag{A-42}$$

Therefore, the energy equation (A-38) becomes

$$\left[\left(G_5 a_{01}\left(X_{i1} - X_{r1}\right)\right)^2 + G_5 a_{01}^2\left(X_{i1} + X_{r1} - 1\right)^2\right] - \left[\left(G_5 a_{02}\left(X_{i2} - X_{r2}\right)\right)^2 + G_5 a_{02}^2\left(X_{i2} + X_{r2} - 1\right)^2\right] = 0 \tag{A-43}$$

Here $a_{01} = a_{02}$ (A-44)

Therefore, the above equation reduces to

$$\left[G_5\left(X_{i1} - X_{r1}\right)^2 + \left(X_{i1} + X_{r1} - 1\right)^2\right] - \left[G_5\left(X_{i2} - X_{r2}\right)^2 + \left(X_{i2} + X_{r2} - 1\right)^2\right] = 0 \quad \text{(A-45)}$$

The unknowns are X_{r1} and X_{r2}.

This is the equation solved in section 3.4.5.4 - equation 3-81.

Interpolation scheme for wave transmission through a mesh [13]

The propagation of pressure waves through a mesh and the notations used in the derivation is shown in figure A.5. The wave transmission theory is briefly explained in section 4.1.2.3. Here the derivation of expressions for the variables X_p and X_q is discussed.

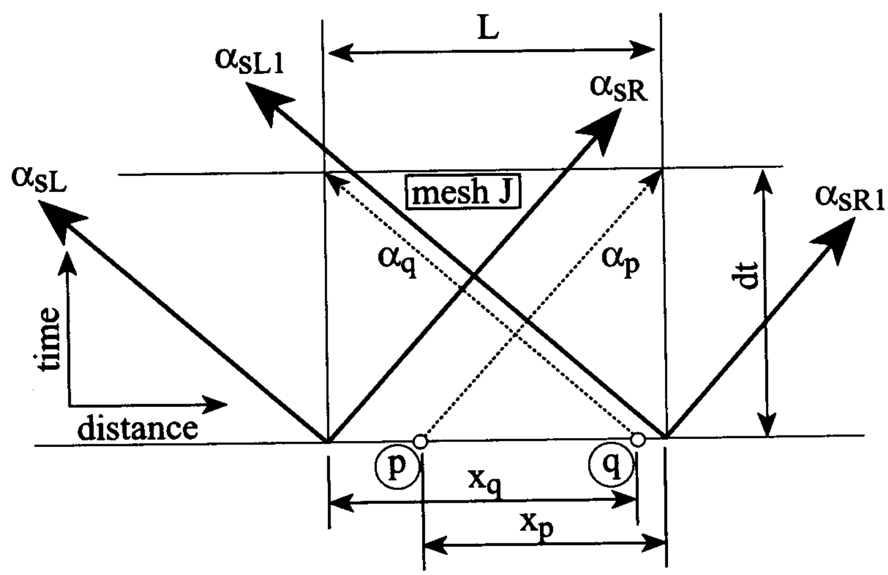


Figure A.5 Propagation of pressure waves in Mesh J [13]

The locations of pressure waves X_p and X_q are 'p' and 'q' respectively. The propagation velocities for these waves are defined by

$$\alpha_p = a_0\left(G_6 X_p - G_4 X_q - 1\right) \tag{A-46}$$

$$\alpha_q = a_0\left(G_6 X_q - G_4 X_p - 1\right) \tag{A-47}$$

The time required for these waves to reach the ends of the mesh is the time step dt.

The distances covered by these waves in this time step are

$$x_p = \alpha_p dt \tag{A-48}$$

$$x_q = \left|\alpha_q\right| dt \tag{A-49}$$

The dimensional values of x_p and x_q relate to the numeric values of X_p and X_q as linear variations of the change of wave pressure between the two ends of the mesh J. Therefore, the relations of X_p and X_q can be derived from

$$X_p = X_R + \left(X_{R1} - X_R\right)\frac{L - x_p}{L} \tag{A-50}$$

$$X_q = X_{L1} + \left(X_L - X_{L1}\right)\frac{L - x_q}{L} \tag{A-51}$$

Eliminating x_p and x_q from the above relations

$$\frac{X_{R1} - X_p}{X_{R1} - X_R} = \frac{a_0 dt}{L}\left(G_6 X_p - G_4 X_q - 1\right) \tag{A-52}$$

$$\frac{X_L - X_q}{X_L - X_{L1}} = \frac{a_0 dt}{L}\left(G_6 X_q - G_4 X_p - 1\right) \tag{A-53}$$

Defining the following variables makes the expressions more convenient

$$A = E\left(X_{R1} - X_R\right) \tag{A-54}$$

$$B = E\left(X_L - X_{L1}\right) \tag{A-55}$$

$$C = \frac{X_{R1}}{A} \tag{A-56}$$

$$D = \frac{X_L}{B} \tag{A-57}$$

$$E = \frac{a_0\, dt}{L} \tag{A-58}$$

Using these variables, the equations (A-52) and (A-53) become

$$X_p \left(G_6 + \frac{1}{A} \right) - G_4 X_q - C - 1 = 0 \tag{A-59}$$

$$X_q \left(G_6 + \frac{1}{B} \right) - G_4 X_p - D - 1 = 0 \tag{A-60}$$

Defining two more variables

$$F_R = \frac{G_6 + \dfrac{1}{A}}{G_4} \tag{A-61}$$

$$F_L = \frac{G_6 + \dfrac{1}{B}}{G_4} \tag{A-62}$$

The final expressions employing the above condensed variables are

$$X_p = \frac{1 + D + F_L + F_L C}{G_4 \left(F_R F_L - 1 \right)} \tag{A-63}$$

$$X_q = \frac{1 + C + F_R + F_R D}{G_4 \left(F_R F_L - 1 \right)} \tag{A-64}$$

Rankine-Hugoniot relations across a shock [13]

In the computational model, whenever a supersonic velocity is reached at any location in a duct, a normal shock is applied to reduce this velocity to a subsonic value. This method is called a 'shock fitting' scheme in computational fluid dynamics. In this scheme, at any instant the particle velocity is calculated, a supersonic check routine is called to check the Mach number. If the value of Mach number exceeds 1, the Rankine-Hugoniot relations are used to calculate the modified pressure amplitudes and reduce the particle velocity to a subsonic value. The reason behind this scheme is that in unsteady flow the particle velocity cannot exceed the local velocity of the forcing pulse. For the pulse, the limiting velocity is the local acoustic velocity. The Rankine-Hugoniot relations are applied to this shock to obtain the modified pressure amplitude ratios.

Consider two oppositely moving waves with amplitudes X_1 and X_2 in a superposition condition. The following relations apply.

$$X_s = X_1 + X_2 - 1 \tag{A-65}$$

$$a_s = a_0 X_s \tag{A-66}$$

$$c_s = G_5 a_0 \left(X_1 - X_2 \right) \tag{A-67}$$

Mach number $$M_s = \frac{c_s}{a_s} = \left| \frac{G_5 a_0 \left(X_1 - X_2 \right)}{a_0 X_s} \right| \tag{A-68}$$

If the Mach number exceeds unity the Rankine-Hugoniot relations are applied to the waves X_1 and X_2. These relations describe the internal reflections across a shock and give the expressions for the new pressure amplitude ratios X_{1new} and X_{2new}.

The expressions for X_{1new} and X_{2new} are

$$X_{1new} = \frac{1+\Gamma_4+\Gamma_3\Gamma_4}{2} \tag{A-69}$$

and

$$X_{2new} = \frac{1+\Gamma_4-\Gamma_3\Gamma_4}{2} \tag{A-70}$$

where

$$\Gamma_1 = \frac{M_s^2+\frac{2}{\gamma-1}}{\frac{2\gamma}{\gamma-1}M_s^2-1} \tag{A-71}$$

$$\Gamma_2 = \frac{2\gamma}{\gamma+1}M_s^2-\frac{\gamma-1}{\gamma+1} \tag{A-72}$$

$$\Gamma_3 = \frac{\gamma-1}{2}\sqrt{\Gamma_1} \tag{A-73}$$

$$\Gamma_4 = X_s\Gamma_2^{\frac{\gamma-1}{2\gamma}} \tag{A-74}$$

The new pressures are given by

$$p_{1new} = p_0 X_{1new}^{G_7} \tag{A-75}$$

$$p_{2new} = p_0 X_{2new}^{G_7} \tag{A-76}$$

APPENDIX B

PROGRAM LISTING

' NASA EPSCOR RESEARCH PROJECT - SUMMER 2003

'

' Mechanical and Aerospace Engineering, Oklahoma State University

' Principal investigator - Dr. Frank.W.Chambers

' Thesis : One-dimensional analysis techniques for pulsed jet flow distribution systems

' Simulation of transient flow

' Reference source : Gordon.P.Blair, Design and Simulation of Four Stroke Engines - Chapter 2,3, pp 153 - 213

' Developer : Krishnan, Kalyanasundaram - Master of Science student, MAE

' Dates : 05/15/2003-12/15/2004

'

```
Option Explicit    'prompts to declare all variables
Option Base 1      'prompts all arrays to start from value 1

'Declaration of variables

Public pinlet As Double
Public T0(300) As Double
Public dia(300) As Double
Public pipedia As Double
Public gamma As Double
Public gc As Double
Public F As Double
Public Tw As Double
Public p0 As Double
Public G3 As Double
Public G4 As Double
Public G5 As Double
```

```
Public G6 As Double
Public G7 As Double
Public G17 As Double
Public G35 As Double
Public G67 As Double
Public Pi As Double
Public Pratio As Double
Public Xinlet As Double
Public Rho0(300) As Double
Public Area(300) As Double
Public a0(300) As Double
Public j As Integer
Public L(300) As Double
Public XR(300) As Double
Public XL(300) As Double
Public XR1(300) As Double
Public XL1(300) As Double
Public Xp(300) As Double
Public Xq(300) As Double
Public totaltime As Double
Public aggtime As Double
Public dt As Double
Public Rhozero As Double
Public azero As Double
Public n As Integer
Public alphasR(300) As Double
Public alphasL(300) As Double
Public alphasR1(300) As Double
Public alphasL1(300) As Double
Public dtL(300) As Double
Public dtR(300) As Double
Public dtL1(300) As Double
Public dtR1(300) As Double
Public dttemp(300) As Double
Public dtmin As Double
Public A(300) As Double
Public B(300) As Double
Public C(300) As Double
Public D(300) As Double
Public E(300) As Double
Public FR(300) As Double
Public FL(300) As Double
Public Tzero As Double
Public Xs(300) As Double
Public ps(300) As Double
Public Rhos(300) As Double
```

```
Public Ts(300) As Double
Public cs(300) As Double
Public Ck(300) As Double
Public mhu(300) As Double
Public Re(300) As Double
Public Cf(300) As Double
Public Ch(300) As Double
Public Areasurf(300) As Double
Public dQf(300) As Double
Public dQh(300) As Double
Public Cp As Double
Public Cv As Double
Public X(300) As Double
Public p(300) As Double
Public Rho(300) As Double
Public Temp(300) As Double
Public ma(300)
Public mb(300) As Double
Public Xin(300) As Double
Public cin(300) As Double
Public cout(300) As Double
Public ca(300) As Double
Public cb(300) As Double
Public Rhoin(300) As Double
Public dhin(300) As Double
Public dmassin(300) As Double
Public dmassout(300) As Double
Public denthalpyin(300) As Double
Public denthalpyout(300) As Double
Public Xin1(300) As Double
Public cin1(300) As Double
Public Rhoin1(300) As Double
Public dhin1(300) As Double
Public dmassin1(300) As Double
Public denthalpyin1(300) As Double
Public Xin2(300) As Double
Public cin2(300) As Double
Public Rhoin2(300) As Double
Public dhin2(300) As Double
Public dmassin2(300) As Double
Public denthalpyin2(300) As Double
Public Xout3(300) As Double
Public cout3(300) As Double
Public Rhoout3(300) As Double
Public dhout3(300) As Double
Public dmassout3(300) As Double
```

```
Public denthalpyout3(300) As Double
Public Xout4(300) As Double
Public cout4(300) As Double
Public Rhoout4(300) As Double
Public dhout4(300) As Double
Public dmassout4(300) As Double
Public denthalpyout4(300) As Double
Public dU(300) As Double
Public Ta(300) As Double
Public Tb(300) As Double
Public XR1new(300) As Double
Public XLnew(300) As Double
Public i As Integer
Public x1(300) As Double
Public x2(300) As Double
Public X2d(300) As Double
Public X1d(300) As Double
Public Xfinal(300) As Double
Public Pfinal(300) As Double
Public Rhofinal(300) As Double
Public Tfinal(300) As Double
Public afinal(300) As Double
Public cfinal(300) As Double
Public counter As Integer
Public Tinlet As Double
Public ainitial(300) As Double
Public XRmod As Double
Public XLmod As Double
Public casenumber As Integer
Public n1initial As Double
Public n1x As Double
Public Ln1 As Double
Public n1 As Integer
Public n2initial As Double
Public n2x As Double
Public Ln2 As Double
Public n2 As Integer
Public n3initial As Double
Public n3x As Double
Public Ln3 As Double
Public n3 As Integer
Public L1 As Double
Public L2 As Double
Public L3 As Double
Public d1 As Double
Public d2 As Double
```

```
Public d3 As Double
Public Tp As Double
Public k As Integer
Public Xi1 As Double
Public Xi2 As Double
Public a01 As Double
Public T02 As Double
Public A1 As Double
Public A2 As Double
Public Rho01 As Double
Public Rho02 As Double
Public Xref1 As Double
Public Xref2 As Double
Public a02 As Double
```

```
'******************************************************************
'SUBROUTINE- INITIALIZATION OF VARIABLES AND READING INPUT
'This subroutine does all the input reading and initialization
'******************************************************************

Sub input_read()

 'READS ALL INPUT FROM THE INPUT SHEET

   pinlet = Worksheets("input").Cells(8, 3) 'Pa
   gamma = Worksheets("input").Cells(9, 3)
   gc = Worksheets("input").Cells(10, 3)
   F = Worksheets("input").Cells(11, 3)
   Tw = Worksheets("input").Cells(12, 3)
   Tp = Worksheets("input").Cells(14, 3)


   L1 = Worksheets("input").Cells(21, 3) 'mm
   L2 = Worksheets("input").Cells(23, 3) 'mm
   L3 = Worksheets("input").Cells(25, 3) 'mm

   d1 = (Worksheets("input").Cells(22, 3)) / 1000  'M
   d2 = (Worksheets("input").Cells(24, 3)) / 1000  'M
   d3 = (Worksheets("input").Cells(26, 3)) / 1000  'M



   'ASSIGNING CONSTANTS TO BE USED

   p0 = 101325         'reference pressure, Pa
   Tzero = 293         'reference temperature, K
   Pi = 22 / 7
   Cp = 1005           'Specific heat at constant pressure, J/kg K
   Cv = Cp / gamma     'Specific heat at constant volume, J/kg K


   'Calculates all derived functions of gamma

    G3 = (4 - 2 * gamma) / (gamma - 1)
    G4 = (3 - gamma) / (gamma - 1)
    G5 = 2 / (gamma - 1)
    G6 = (gamma + 1) / (gamma - 1)
    G7 = (2 * gamma) / (gamma - 1)
    G17 = (gamma - 1) / (2 * gamma)
    G35 = gamma / (gamma - 1)
```

```
G67 = (gamma + 1) / (2 * gamma)



'Calculates all derived variables

 Pratio = pinlet / p0
 Xinlet = Pratio ^ G17
 Tinlet = Tzero * Xinlet ^ 2
 Rhozero = p0 / (gc * Tzero)
 azero = Sqr(gamma * gc * Tzero)
```

'***

'MESHING AND INITIALISATION OF VARIABLES

'***

```
 'Meshing - sets up the mesh, length, diameter and area of each mesh by calling Mesher
  subroutine

 Call mesher


 'Initialises all the pressure ratios, acoustic velocities, densities for all the meshes

For j = 1 To n

 'For the first mesh the pressure ratio of wave towards right at left end of mesh
 'is initialized as Xinlet and for all other meshes initialized as 1

   If j = 1 Then
      XR(j) = Xinlet
   Else
      XR(j) = 1
   End If

 'For all meshes the pressure ratio of wave towards left at left end of mesh
 'and pressure ratio of wave towards left and right at right end of mesh
 'is initialized as 1

   XL(j) = 1
   XR1(j) = 1
   XL1(j) = 1

  'The acoustic velocity and density are initialized here
```

```
      a0(j) = azero
      Rho0(j) = Rhozero
      T0(j) = Tzero


   Next j

'*******************************************************************
      'PRINT ALL VARIABLES FROM READ INPUT - INITIAL VALUES
'*******************************************************************

      Worksheets("output_variables").Cells(5, 3) = XR(1)
      Worksheets("output_variables").Cells(6, 3) = XR(n1)
      Worksheets("output_variables").Cells(7, 3) = XR(n1 + 1)
      Worksheets("output_variables").Cells(8, 3) = XR(n1 + n2)
      Worksheets("output_variables").Cells(9, 3) = XR(n1 + n2 + 1)
      Worksheets("output_variables").Cells(10, 3) = XR(n)

      Worksheets("output_variables").Cells(12, 3) = XR1(1)
      Worksheets("output_variables").Cells(13, 3) = XR1(n1)
      Worksheets("output_variables").Cells(14, 3) = XR1(n1 + 1)
      Worksheets("output_variables").Cells(15, 3) = XR1(n1 + n2)
      Worksheets("output_variables").Cells(16, 3) = XR1(n1 + n2 + 1)
      Worksheets("output_variables").Cells(17, 3) = XR1(n)

      Worksheets("output_variables").Cells(19, 3) = XL(1)
      Worksheets("output_variables").Cells(20, 3) = XL(n1)
      Worksheets("output_variables").Cells(21, 3) = XL(n1 + 1)
      Worksheets("output_variables").Cells(22, 3) = XL(n1 + n2)
      Worksheets("output_variables").Cells(23, 3) = XL(n1 + n2 + 1)
      Worksheets("output_variables").Cells(24, 3) = XL(n)

      Worksheets("output_variables").Cells(26, 3) = XL1(1)
      Worksheets("output_variables").Cells(27, 3) = XL1(n1)
      Worksheets("output_variables").Cells(28, 3) = XL1(n1 + 1)
      Worksheets("output_variables").Cells(29, 3) = XL1(n1 + n2)
      Worksheets("output_variables").Cells(30, 3) = XL1(n1 + n2 + 1)
      Worksheets("output_variables").Cells(31, 3) = XL1(n)

      Worksheets("output_variables").Cells(33, 3) = Rho0(1)
      Worksheets("output_variables").Cells(34, 3) = Rho0(n1)
      Worksheets("output_variables").Cells(35, 3) = Rho0(n1 + 1)
      Worksheets("output_variables").Cells(36, 3) = Rho0(n1 + n2)
```

```
Worksheets("output_variables").Cells(37, 3) = Rho0(n1 + n2 + 1)
Worksheets("output_variables").Cells(38, 3) = Rho0(n)

Worksheets("output_variables").Cells(40, 3) = a0(1)
Worksheets("output_variables").Cells(41, 3) = a0(n1)
Worksheets("output_variables").Cells(42, 3) = a0(n1 + 1)
Worksheets("output_variables").Cells(43, 3) = a0(n1 + n2)
Worksheets("output_variables").Cells(44, 3) = a0(n1 + n2 + 1)
Worksheets("output_variables").Cells(45, 3) = a0(n)

Worksheets("output_variables").Cells(47, 3) = T0(1)
Worksheets("output_variables").Cells(48, 3) = T0(n1)
Worksheets("output_variables").Cells(49, 3) = T0(n1 + 1)
Worksheets("output_variables").Cells(50, 3) = T0(n1 + n2)
Worksheets("output_variables").Cells(51, 3) = T0(n1 + n2 + 1)
Worksheets("output_variables").Cells(52, 3) = T0(n)



For j = 1 To n

    Worksheets("output_constants").Cells(j + 4, 5) = "L"
    Worksheets("output_constants").Cells(j + 4, 6) = j
    Worksheets("output_constants").Cells(j + 4, 7) = L(j)

    Worksheets("output_constants").Cells(j + 4, 8) = "diameter"
    Worksheets("output_constants").Cells(j + 4, 9) = j
    Worksheets("output_constants").Cells(j + 4, 10) = dia(j)

    Worksheets("output_constants").Cells(j + 4, 11) = "C/S Area"
    Worksheets("output_constants").Cells(j + 4, 12) = j
    Worksheets("output_constants").Cells(j + 4, 13) = Area(j)

Next j
```

```
'*****************************************************************************
        'PRINT ALL CONSTANTS FROM READ INPUT

'*****************************************************************************

        Worksheets("output_constants").Cells(5, 2) = pinlet
        Worksheets("output_constants").Cells(6, 2) = pipedia
        Worksheets("output_constants").Cells(7, 2) = gamma
        Worksheets("output_constants").Cells(8, 2) = gc
        Worksheets("output_constants").Cells(9, 2) = F
        Worksheets("output_constants").Cells(10, 2) = Tw
        Worksheets("output_constants").Cells(11, 2) = p0
        Worksheets("output_constants").Cells(12, 2) = Tzero
        Worksheets("output_constants").Cells(13, 2) = Pi
        Worksheets("output_constants").Cells(14, 2) = Cp
        Worksheets("output_constants").Cells(15, 2) = Cv
        Worksheets("output_constants").Cells(16, 2) = Pratio
        Worksheets("output_constants").Cells(17, 2) = Xinlet
        Worksheets("output_constants").Cells(18, 2) = Rhozero
        Worksheets("output_constants").Cells(19, 2) = azero

        Worksheets("output_constants").Cells(24, 2) = G3
        Worksheets("output_constants").Cells(25, 2) = G4
        Worksheets("output_constants").Cells(26, 2) = G5
        Worksheets("output_constants").Cells(27, 2) = G6
        Worksheets("output_constants").Cells(28, 2) = G7
        Worksheets("output_constants").Cells(29, 2) = G17
        Worksheets("output_constants").Cells(30, 2) = G35
        Worksheets("output_constants").Cells(31, 2) = G67
```

```
'**********************************************************************
'ALL INITIAL PRINTING OF MESH NUMBERS, TIME IS DONE HERE
'**********************************************************************
'At the start the representative mesh numbers are printed to several sheets

'**********************************************************************
'SIMULATION LOOP VARIABLES SHEET
'**********************************************************************

k = 1

Do While k <= 62

    Worksheets("simulation_loop_variables").Cells(k + 11, 2) = "1"
    Worksheets("simulation_loop_variables").Cells(k + 12, 2) = n1
    Worksheets("simulation_loop_variables").Cells(k + 13, 2) = n1 + 1
    Worksheets("simulation_loop_variables").Cells(k + 14, 2) = n1 + n2
    Worksheets("simulation_loop_variables").Cells(k + 15, 2) = n1 + n2 + 1
    Worksheets("simulation_loop_variables").Cells(k + 16, 2) = n

    k = k + 7

Loop

k = 67

Do While k <= 326

    Worksheets("simulation_loop_variables").Cells(k + 11, 2) = "1"
    Worksheets("simulation_loop_variables").Cells(k + 12, 2) = n1
    Worksheets("simulation_loop_variables").Cells(k + 13, 2) = n1 + 1
    Worksheets("simulation_loop_variables").Cells(k + 14, 2) = n1 + n2
    Worksheets("simulation_loop_variables").Cells(k + 15, 2) = n1 + n2 + 1
    Worksheets("simulation_loop_variables").Cells(k + 16, 2) = n

    k = k + 7

Loop
```

```
'*********************************************************************
                    'OUTPUT VARIABLES SHEET
'*********************************************************************


    k = 1

    Do While k <= 85

        Worksheets("output_variables").Cells(k + 4, 2) = "1"
        Worksheets("output_variables").Cells(k + 5, 2) = n1
        Worksheets("output_variables").Cells(k + 6, 2) = n1 + 1
        Worksheets("output_variables").Cells(k + 7, 2) = n1 + n2
        Worksheets("output_variables").Cells(k + 8, 2) = n1 + n2 + 1
        Worksheets("output_variables").Cells(k + 9, 2) = n

        k = k + 7

    Loop


'*********************************************************************
                'X,C,T,Rho,P,a,T0_Check SHEETS
'*********************************************************************


   For i = 1 To n

      Worksheets("X").Cells(8, i + 3) = i
      Worksheets("C").Cells(8, i + 3) = i
      Worksheets("T").Cells(8, i + 3) = i
      Worksheets("Rho").Cells(8, i + 3) = i
      Worksheets("P").Cells(8, i + 3) = i
      Worksheets("a").Cells(8, i + 3) = i
      Worksheets("T0_Check").Cells(8, i + 3) = i

   Next i

End Sub
```

```
'*********************************************************************
                    'SIMULATION SUBROUTINE
              'ENTER TIMELOOP TO START SIMULATION
'*********************************************************************

Sub simulate()

Call input_read

Dim Msg, Style, Title, Response

If d2 < d1 Then

  Msg = "                              d2 must be >= d1                    "

      Style = vbOKOnly
      Title = "Error"

      Response = MsgBox(Msg, Style, Title)

      GoTo endofprogram

End If


If d3 > d2 Then

  Msg = "                              d3 must be <= d2                    "

      Style = vbOKOnly
      Title = "Error"

      Response = MsgBox(Msg, Style, Title)

      GoTo endofprogram

End If
```

```
'This section evaluates the total time cycle for which the simulation has to be
 performed and also the aggregate time completed after each time step

    totaltime = 1 / F  'seconds
    aggtime = 0  'seconds


  'Prints all initial values of aggtime and total time to several output sheets

    Worksheets("simulation_loop_variables").Cells(5, 3) = totaltime
    Worksheets("X").Cells(4, 3) = totaltime
    Worksheets("C").Cells(4, 3) = totaltime
    Worksheets("T").Cells(4, 3) = totaltime
    Worksheets("Rho").Cells(4, 3) = totaltime
    Worksheets("P").Cells(4, 3) = totaltime
    Worksheets("a").Cells(4, 3) = totaltime
    Worksheets("T0_Check").Cells(4, 3) = totaltime


    Worksheets("output_variables").Cells(3, 3) = aggtime
    Worksheets("X").Cells(10, 2) = aggtime
    Worksheets("C").Cells(10, 2) = aggtime
    Worksheets("T").Cells(10, 2) = aggtime
    Worksheets("Rho").Cells(10, 2) = aggtime
    Worksheets("P").Cells(10, 2) = aggtime
    Worksheets("a").Cells(10, 2) = aggtime
    Worksheets("T0_Check").Cells(10, 2) = aggtime
```

```
'*********************************************************************
                'THE MAIN SIMULATION DO LOOP STARTS HERE
'*********************************************************************

  counter = 0    'This is the counter for the time steps completed

  'Do loop until total time is reached

  Do Until aggtime >= totaltime


 'Here the pulse generator routine assigns XR for mesh 1 checking aggregate time v/s
  time of pulse

  Call pulse_generator


  'find the superposition velocities at either end of meshes for all meshes


'*********************************************************************
                'THE TIME STEP CALCULATION STARTS HERE - dt
                'PAGES 259-260 OF BOOK
'*********************************************************************


     For j = 1 To n

          'find the wave superposition velocities at either ends of the mesh
          '(text p.178-179, eqn 2.2.9, 2.2.10)

          alphasR(j) = a0(j) * (G6 * XR(j) - G4 * XL(j) - 1)
          alphasL(j) = -a0(j) * (G6 * XL(j) - G4 * XR(j) - 1)
          alphasR1(j) = a0(j) * (G6 * XR1(j) - G4 * XL1(j) - 1)
          alphasL1(j) = -a0(j) * (G6 * XL1(j) - G4 * XR1(j) - 1)


          'find the times for travel in the meshes (text p.260, eqn 2.20.7)

          dtL(j) = L(j) / alphasL(j)
          dtR(j) = L(j) / alphasR(j)
          dtL1(j) = L(j) / alphasL1(j)
          dtR1(j) = L(j) / alphasR1(j)

          'dttemp(j) is the min dt value among the four above dt's for each mesh j
```

```
        dttemp(j) = Application.WorksheetFunction.Min(Abs(dtL(j)), Abs(dtR(j)),
        Abs(dtL1(j)), Abs(dtR1(j)))


 Next j

'CALCULATE TIMESTEP dt

'find smallest dt among the dt's for all meshes

    dtmin = dttemp(1)

For j = 2 To n

'This loop compares the times 'dt' for all the meshes from 1 to n and outputs least dt


        If dtmin < dttemp(j) Then
            dtmin = dtmin
        Else
            dtmin = dttemp(j)
        End If

Next j

'calculating the actual dt acc eqn 2.20.7 p260

    dt = 0.99 * dtmin


'Here the time step is calculated in each loop and then added to aggregate time
'If the difference between total time and aggregate time is less than 'dt' then
'this difference value is assigned to 'dt' so as to complete the simulation in
'the total time

 If (totaltime - aggtime) >= dt Then

    dt = dt
 Else

    dt = (totaltime - aggtime)

 End If

    aggtime = aggtime + dt
```

'**

'PRINT ALL VARIABLES FROM TIME CALCULATION

'**

```
'Here the initial counter value is printed to X,C,P,T,Rho,T0_Check sheets

    If counter = 0 Then

        Worksheets("X").Cells(10, 1) = counter
        Worksheets("C").Cells(10, 1) = counter
        Worksheets("T").Cells(10, 1) = counter
        Worksheets("Rho").Cells(10, 1) = counter
        Worksheets("P").Cells(10, 1) = counter
        Worksheets("a").Cells(10, 1) = counter
        Worksheets("T0_Check").Cells(10, 1) = counter

    End If

'Here the counter, dtmin, dt and aggtime values are printed to the simulation loop
 variables sheet


 If counter < 253 Then 'This is to avoid the program crashing after it has reached the
   'limit of columns of excel sheet for printing


    Worksheets("simulation_loop_variables").Cells(6, counter + 3) = counter + 1
    Worksheets("simulation_loop_variables").Cells(7, counter + 3) = dtmin
    Worksheets("simulation_loop_variables").Cells(8, counter + 3) = dt
    Worksheets("simulation_loop_variables").Cells(10, counter + 3) = aggtime


    'Here the variables used in 'dt' calculation are printed

    Worksheets("simulation_loop_variables").Cells(12, counter + 3) = alphasR(1)
    Worksheets("simulation_loop_variables").Cells(13, counter + 3) = alphasR(n1)
    Worksheets("simulation_loop_variables").Cells(14, counter + 3) = alphasR(n1 + 1)
    Worksheets("simulation_loop_variables").Cells(15, counter + 3) = alphasR(n1 + n2)
    Worksheets("simulation_loop_variables").Cells(16, counter + 3) = alphasR(n1 + n2 + 1)
    Worksheets("simulation_loop_variables").Cells(17, counter + 3) = alphasR(n)
    Worksheets("simulation_loop_variables").Cells(19, counter + 3) = alphasL(1)
    Worksheets("simulation_loop_variables").Cells(20, counter + 3) = alphasL(n1)
    Worksheets("simulation_loop_variables").Cells(21, counter + 3) = alphasL(n1 + 1)
    Worksheets("simulation_loop_variables").Cells(22, counter + 3) = alphasL(n1 + n2)
```

```
Worksheets("simulation_loop_variables").Cells(23, counter + 3) = alphasL(n1 + n2 + 1)
Worksheets("simulation_loop_variables").Cells(24, counter + 3) = alphasL(n)
Worksheets("simulation_loop_variables").Cells(26, counter + 3) = alphasR1(1)
Worksheets("simulation_loop_variables").Cells(27, counter + 3) = alphasR1(n1)
Worksheets("simulation_loop_variables").Cells(28, counter + 3) = alphasR1(n1 + 1)
Worksheets("simulation_loop_variables").Cells(29, counter + 3) = alphasR1(n1 + n2)
Worksheets("simulation_loop_variables").Cells(30, counter + 3) = alphasR1(n1 + n2 + 1)
Worksheets("simulation_loop_variables").Cells(31, counter + 3) = alphasR1(n)
Worksheets("simulation_loop_variables").Cells(33, counter + 3) = alphasL1(1)
Worksheets("simulation_loop_variables").Cells(34, counter + 3) = alphasL1(n1)
Worksheets("simulation_loop_variables").Cells(35, counter + 3) = alphasL1(n1 + 1)
Worksheets("simulation_loop_variables").Cells(36, counter + 3) = alphasL1(n1 + n2)
Worksheets("simulation_loop_variables").Cells(37, counter + 3) = alphasL1(n1 + n2 + 1)
Worksheets("simulation_loop_variables").Cells(38, counter + 3) = alphasL1(n)

Worksheets("simulation_loop_variables").Cells(40, counter + 3) = dtL(1)
Worksheets("simulation_loop_variables").Cells(41, counter + 3) = dtL(n1)
Worksheets("simulation_loop_variables").Cells(42, counter + 3) = dtL(n1 + 1)
Worksheets("simulation_loop_variables").Cells(43, counter + 3) = dtL(n1 + n2)
Worksheets("simulation_loop_variables").Cells(44, counter + 3) = dtL(n1 + n2 + 1)
Worksheets("simulation_loop_variables").Cells(45, counter + 3) = dtL(n)

Worksheets("simulation_loop_variables").Cells(47, counter + 3) = dtR(1)
Worksheets("simulation_loop_variables").Cells(48, counter + 3) = dtR(n1)
Worksheets("simulation_loop_variables").Cells(49, counter + 3) = dtR(n1 + 1)
Worksheets("simulation_loop_variables").Cells(50, counter + 3) = dtR(n1 + n2)
Worksheets("simulation_loop_variables").Cells(51, counter + 3) = dtR(n1 + n2 + 1)
Worksheets("simulation_loop_variables").Cells(52, counter + 3) = dtR(n)

Worksheets("simulation_loop_variables").Cells(54, counter + 3) = dtL1(1)
Worksheets("simulation_loop_variables").Cells(55, counter + 3) = dtL1(n1)
Worksheets("simulation_loop_variables").Cells(56, counter + 3) = dtL1(n1 + 1)
Worksheets("simulation_loop_variables").Cells(57, counter + 3) = dtL1(n1 + n2)
Worksheets("simulation_loop_variables").Cells(58, counter + 3) = dtL1(n1 + n2 + 1)
Worksheets("simulation_loop_variables").Cells(59, counter + 3) = dtL1(n)

Worksheets("simulation_loop_variables").Cells(61, counter + 3) = dtR1(1)
Worksheets("simulation_loop_variables").Cells(62, counter + 3) = dtR1(n1)
Worksheets("simulation_loop_variables").Cells(63, counter + 3) = dtR1(n1 + 1)
Worksheets("simulation_loop_variables").Cells(64, counter + 3) = dtR1(n1 + n2)
Worksheets("simulation_loop_variables").Cells(65, counter + 3) = dtR1(n1 + n2 + 1)
Worksheets("simulation_loop_variables").Cells(66, counter + 3) = dtR1(n)
```

```
    Worksheets("simulation_loop_variables").Cells(68, counter + 3) = dttemp(1)
    Worksheets("simulation_loop_variables").Cells(69, counter + 3) = dttemp(n1)
    Worksheets("simulation_loop_variables").Cells(70, counter + 3) = dttemp(n1 + 1)
    Worksheets("simulation_loop_variables").Cells(71, counter + 3) = dttemp(n1 + n2)
    Worksheets("simulation_loop_variables").Cells(72, counter + 3) = dttemp(n1 + n2 + 1)
    Worksheets("simulation_loop_variables").Cells(73, counter + 3) = dttemp(n)


End If
```

```
'**********************************************************************
'THE FIRST LAW THEORY STARTS HERE - TO CALCULATE NEW T0, a0
'PAGES 271-279 OF BOOK
'**********************************************************************

'CALCULATE VARIABLES FOR THE MESH SPACE - ASSUMED TO BE THE
'AVERAGEOF THE SUPERPOSITION VALUES - PAGE 258


For j = 1 To n

    X(j) = ((XR(j) + XL(j) - 1) + (XR1(j) + XL1(j) - 1)) / 2    'pressure amplitude ratio
    p(j) = p0 * X(j) ^ G7                                        'average pressure
    Rho(j) = Rho0(j) * X(j) ^ G5                                 'Density
    Temp(j) = T0(j) * X(j) ^ 2                                   'Temperature
    Tb(j) = Temp(j)
    ainitial(j) = a0(j) * X(j)
    mb(j) = Rho(j) * Area(j) * L(j)                              'Mass in the mesh
    cb(j) = G5 * a0(j) * (X(j) - 1)

Next j



'PRINTING ALL INITIAL REPRESENTATIVE VALUES

If counter = 0 Then

    Worksheets("output_variables").Cells(54, 3) = X(1)
    Worksheets("output_variables").Cells(55, 3) = X(n1)
    Worksheets("output_variables").Cells(56, 3) = X(n1 + 1)
    Worksheets("output_variables").Cells(57, 3) = X(n1 + n2)
    Worksheets("output_variables").Cells(58, 3) = X(n1 + n2 + 1)
    Worksheets("output_variables").Cells(59, 3) = X(n)

    Worksheets("output_variables").Cells(61, 3) = p(1)
    Worksheets("output_variables").Cells(62, 3) = p(n1)
    Worksheets("output_variables").Cells(63, 3) = p(n1 + 1)
    Worksheets("output_variables").Cells(64, 3) = p(n1 + n2)
    Worksheets("output_variables").Cells(65, 3) = p(n1 + n2 + 1)
    Worksheets("output_variables").Cells(66, 3) = p(n)
```

```
    Worksheets("output_variables").Cells(68, 3) = Rho(1)
    Worksheets("output_variables").Cells(69, 3) = Rho(n1)
    Worksheets("output_variables").Cells(70, 3) = Rho(n1 + 1)
    Worksheets("output_variables").Cells(71, 3) = Rho(n1 + n2)
    Worksheets("output_variables").Cells(72, 3) = Rho(n1 + n2 + 1)
    Worksheets("output_variables").Cells(73, 3) = Rho(n)

    Worksheets("output_variables").Cells(75, 3) = Temp(1)
    Worksheets("output_variables").Cells(76, 3) = Temp(n1)
    Worksheets("output_variables").Cells(77, 3) = Temp(n1 + 1)
    Worksheets("output_variables").Cells(78, 3) = Temp(n1 + n2)
    Worksheets("output_variables").Cells(79, 3) = Temp(n1 + n2 + 1)
    Worksheets("output_variables").Cells(80, 3) = Temp(n)

    Worksheets("output_variables").Cells(82, 3) = ainitial(1)
    Worksheets("output_variables").Cells(83, 3) = ainitial(n1)
    Worksheets("output_variables").Cells(84, 3) = ainitial(n1 + 1)
    Worksheets("output_variables").Cells(85, 3) = ainitial(n1 + n2)
    Worksheets("output_variables").Cells(86, 3) = ainitial(n1 + n2 + 1)
    Worksheets("output_variables").Cells(87, 3) = ainitial(n)

    Worksheets("output_variables").Cells(89, 3) = cb(1)
    Worksheets("output_variables").Cells(90, 3) = cb(n1)
    Worksheets("output_variables").Cells(91, 3) = cb(n1 + 1)
    Worksheets("output_variables").Cells(92, 3) = cb(n1 + n2)
    Worksheets("output_variables").Cells(93, 3) = cb(n1 + n2 + 1)
    Worksheets("output_variables").Cells(94, 3) = cb(n)


    For j = 1 To n

      Worksheets("X").Cells(10, j + 3) = X(j)
      Worksheets("C").Cells(10, j + 3) = cb(j)
      Worksheets("T").Cells(10, j + 3) = Temp(j)
      Worksheets("Rho").Cells(10, j + 3) = Rho(j)
      Worksheets("P").Cells(10, j + 3) = p(j)
      Worksheets("a").Cells(10, j + 3) = ainitial(j)
      Worksheets("T0_Check").Cells(10, j + 3) = T0(j)

    Next j



End If
```

'***

'HERE WE SELECT A COMBINATION OF CASES FROM THE FOUR CASES PP 274-275 'BASED ON THE VALUE OFDIFFERENCE IN PRESSURE AMPLITUDE RATIOS OF 'RIGHT AND LEFT WAVES AT BOTH ENDS OF THE MESH

'BOOK PP 274-275

'***

```
    'Calls a subroutine "case_selector_1_2" to output case number. If case number is 1, it
    'corresponds to case I. if 2, it corresponds to case II

     For j = 1 To n

       Call case_selector_1_2(XR(j), XL(j), casenumber)


       If casenumber = 1 Then
          GoTo 1
       ElseIf casenumber = 2 Then
          GoTo 2
       End If


1:    'CASE I - OUTFLOW FROM LEFT END - "IN" SIDE OF ALL MESHES


'SUPERSONIC CHECK DONE HERE, IF SUPERSONIC THE X VALUES NEED TO
'BE RECALCULATED

       cin1(j) = G5 * a0(j) * (XR(j) - XL(j))          'Particle velocity

       Call supersonic_check(XR(j), XL(j), XRmod, XLmod)

       XR(j) = XRmod
       XL(j) = XLmod


    'Checks back again for the appropriate case

       Call case_selector_1_2(XR(j), XL(j), casenumber)
```

```
    If casenumber = 1 Then

        cin1(j) = G5 * a0(j) * (XR(j) - XL(j))            'Particle velocity

    Else

    GoTo 2

    End If

'CALCULATE ALL VARIABLES

    Xin1(j) = XR(j) + XL(j) - 1                           'Pressure
    Rhoin1(j) = Rho0(j) * Xin1(j) ^ G5                    'Density
    dhin1(j) = Cp * Temp(j) + ((cin1(j) ^ 2) / 2)         'Specific enthalpy
    dmassin1(j) = Rhoin1(j) * Area(j) * cin1(j) * dt      'Mass flow increment
    denthalpyin1(j) = dhin1(j) * dmassin1(j)              'Enthalpy increment


    dmassin(j) = dmassin1(j)
    denthalpyin(j) = denthalpyin1(j)
    cin(j) = cin1(j)

  If casenumber = 1 Then

    GoTo case_selection_3_4

  End If
```

```
2:  'CASE II - INFLOW FROM LEFT END - "IN" SIDE OF ALL MESHES


    If j = 1 Then

    'SUPERSONIC CHECK DONE HERE, IF SUPERSONIC THE X VALUES NEED
    'TO BE RECALCULATED

        cin2(j) = G5 * azero * (XR(j) - XL(j))          'Particle velocity

        Call supersonic_check(XR(j), XL(j), XRmod, XLmod)

        XR(j) = XRmod
        XL(j) = XLmod


    'Checks back again for the appropriate case

        Call case_selector_1_2(XR(j), XL(j), casenumber)

        If casenumber = 2 Then

          cin2(j) = G5 * azero * (XR(j) - XL(j))          'Particle velocity

        Else

        GoTo 1

        End If

    'CALCULATE ALL VARIABLES

        Xin2(j) = XR(j) + XL(j) - 1                          'Pressure
        Rhoin2(j) = Rhozero * Xin2(j) ^ G5                   'Density
       dhin2(j) = Cp * Tinlet + ((cin2(j) ^ 2) / 2)          'Specific enthalpy
        dmassin2(j) = Rhoin2(j) * Area(j) * cin2(j) * dt     'Mass flow increment
        denthalpyin2(j) = dhin2(j) * dmassin2(j)             'Enthalpy increment

      Else
```

```
'SUPERSONIC CHECK DONE HERE, IF SUPERSONIC THE X VALUES NEED
' TO BE RECALCULATED


        cin2(j) = G5 * a0(j - 1) * (XR(j) - XL(j))         'Particle velocity

        Call supersonic_check(XR(j), XL(j), XRmod, XLmod)

        XR(j) = XRmod
        XL(j) = XLmod


'Checks back again for the appropriate case

        Call case_selector_1_2(XR(j), XL(j), casenumber)

        If casenumber = 2 Then

        cin2(j) = G5 * a0(j - 1) * (XR(j) - XL(j))         'Particle velocity

        Else

        GoTo 1

        End If


'CALCULATE ALL VARIABLES

        Xin2(j) = XR(j) + XL(j) - 1                         'Pressure
        Rhoin2(j) = Rho0(j - 1) * Xin2(j) ^ G5             'Density
        dhin2(j) = Cp * Temp(j - 1) + ((cin2(j) ^ 2) / 2)   'Specific enthalpy
        dmassin2(j) = Rhoin2(j) * Area(j) * cin2(j) * dt    'Mass flow increment
        denthalpyin2(j) = dhin2(j) * dmassin2(j)           'Enthalpy increment


    End If


    dmassin(j) = dmassin2(j)
    denthalpyin(j) = denthalpyin2(j)
    cin(j) = cin2(j)
```

```
case_selection_3_4:
'*********************************************************************
'HERE WE USE A SELECT CASE OPTION FOR THE FOUR CASES AND
'SELECT THE CORRECT COMBINATION OF CASES BASED ON THE
'VALUE OF DIFFERENCE IN PRESSURE AMPLITUDE RATIOS OF RIGHT
'AND LEFT WAVES AT BOTH ENDS OF THE MESH
'*********************************************************************

    'Calls a subroutine "case_selector_3_4" to output case number.If case number is 3, it
     corresponds to case III. if 4, it corresponds to case IV

        Call case_selector_3_4(XR1(j), XL1(j), casenumber)

        If casenumber = 3 Then
           GoTo 3
        ElseIf casenumber = 4 Then
           GoTo 4
        End If

3:     'CASE III - INFLOW FROM RIGHT END - "OUT" SIDE OF ALL MESHES

        If j = n Then

    'SUPERSONIC CHECK DONE HERE, IF SUPERSONIC THE X VALUES NEED
     TO BE RECALCULATED

           cout3(j) = G5 * azero * (XR1(j) - XL1(j))           'Particle velocity

           Call supersonic_check(XR1(j), XL1(j), XRmod, XLmod)

           XR1(j) = XRmod
           XL1(j) = XLmod

    'Checks back again for the appropriate case

           Call case_selector_3_4(XR1(j), XL1(j), casenumber)

           If casenumber = 3 Then

              cout3(j) = G5 * azero * (XR1(j) - XL1(j))           'Particle velocity

           Else
           GoTo 4

           End If
```

```
'CALCULATE ALL VARIABLES

    Xout3(j) = XR1(j) + XL1(j) - 1                      'Pressure
    Rhoout3(j) = Rhozero * Xout3(j) ^ G5                'Density
    dhout3(j) = Cp * Tzero + ((cout3(j) ^ 2) / 2)       'Specific enthalpy
    dmassout3(j) = Rhoout3(j) * Area(j) * cout3(j) * dt 'Mass flow increment
    denthalpyout3(j) = dhout3(j) * dmassout3(j)         'Enthalpy increment

  Else


'SUPERSONIC CHECK DONE HERE, IF SUPERSONIC THE X VALUES NEED
 TO BE RECALCULATED

    cout3(j) = G5 * a0(j + 1) * (XR1(j) - XL1(j))       'Particle velocity

    Call supersonic_check(XR1(j), XL1(j), XRmod, XLmod)

    XR1(j) = XRmod
    XL1(j) = XLmod

'Checks back again for the appropriate case

    Call case_selector_3_4(XR1(j), XL1(j), casenumber)

    If casenumber = 3 Then

    cout3(j) = G5 * a0(j + 1) * (XR1(j) - XL1(j))       'Particle velocity

    Else

    GoTo 4

    End If

'CALCULATE ALL VARIABLES


    Xout3(j) = XR1(j) + XL1(j) - 1                      'Pressure
    Rhoout3(j) = Rho0(j + 1) * Xout3(j) ^ G5            'Density
    dhout3(j) = Cp * Temp(j + 1) + ((cout3(j) ^ 2) / 2) 'Specific enthalpy
    dmassout3(j) = Rhoout3(j) * Area(j) * cout3(j) * dt 'Mass flow increment
    denthalpyout3(j) = dhout3(j) * dmassout3(j)         'Enthalpy increment

  End If
```

```
dmassout(j) = dmassout3(j)
denthalpyout(j) = denthalpyout3(j)
cout(j) = cout3(j)

If casenumber = 3 Then
   GoTo firstlaw
End If


4:    'CASE IV - OUTFLOW FROM RIGHT END - "OUT" SIDE OF ALL MESHES


    'SUPERSONIC CHECK DONE HERE, IF SUPERSONIC THE X VALUES NEED
     TO BE RECALCULATED

cout4(j) = G5 * a0(j) * (XR1(j) - XL1(j))              'Particle velocity

Call supersonic_check(XR1(j), XL1(j), XRmod, XLmod)

XR1(j) = XRmod
XL1(j) = XLmod


Call case_selector_3_4(XR1(j), XL1(j), casenumber)

If casenumber = 4 Then

cout4(j) = G5 * a0(j) * (XR1(j) - XL1(j))              'Particle velocity

Else

GoTo 3

End If


Xout4(j) = XR1(j) + XL1(j) - 1                         'Pressure
Rhoout4(j) = Rho0(j) * Xout4(j) ^ G5                   'Density
dhout4(j) = Cp * Temp(j) + ((cout4(j) ^ 2) / 2)        'Specific enthalpy
dmassout4(j) = Rhoout4(j) * Area(j) * cout4(j) * dt    'Mass flow increment
denthalpyout4(j) = dhout4(j) * dmassout4(j)            'Enthalpy increment


dmassout(j) = dmassout4(j)
denthalpyout(j) = denthalpyout4(j)
cout(j) = cout4(j)
```

firstlaw:

```
'*******************************************************************
  '   APPLICATION OF FIRST LAW OF THERMODYNAMICS FOR EACH
  '   MESH TO UPDATE REFERENCE TEMPERATURES
'*******************************************************************

    'FIRST LAW OF THERMODYNAMICS APPLICATION page 276-277

    'New system mass and velocity in Mesh J derived from the continuity equation


       ma(j) = mb(j) + dmassin(j) - dmassout(j)
       ca(j) = Sqr((cin(j) ^ 2 / 2) + (cout(j) ^ 2 / 2))


    Next j

'*******************************************************************
 '   PRINT ALL VARIABLES FROM FIRST LAW OF THERMODYNAMICS
 '   CALCULATION
'*******************************************************************


    'If counter < 253 Then 'This is to avoid the program crashing after it has reached the
    'Limit of columns of excel sheet for printing

       For j = 1 To n


          Worksheets("simulation_loop_variables").Cells(78, counter + 3) = X(1)
          Worksheets("simulation_loop_variables").Cells(79, counter + 3) = X(n1)
          Worksheets("simulation_loop_variables").Cells(80, counter + 3) = X(n1 + 1)
          Worksheets("simulation_loop_variables").Cells(81, counter + 3) = X(n1 + n2)
          Worksheets("simulation_loop_variables").Cells(82, counter + 3) = X(n1 + n2 + 1)
          Worksheets("simulation_loop_variables").Cells(83, counter + 3) = X(n)


          Worksheets("simulation_loop_variables").Cells(85, counter + 3) = p(1)
          Worksheets("simulation_loop_variables").Cells(86, counter + 3) = p(n1)
          Worksheets("simulation_loop_variables").Cells(87, counter + 3) = p(n1 + 1)
          Worksheets("simulation_loop_variables").Cells(88, counter + 3) = p(n1 + n2)
```

```
Worksheets("simulation_loop_variables").Cells(89, counter + 3) = p(n1 + n2 + 1)
Worksheets("simulation_loop_variables").Cells(90, counter + 3) = p(n)

Worksheets("simulation_loop_variables").Cells(92, counter + 3) = Rho(1)
Worksheets("simulation_loop_variables").Cells(93, counter + 3) = Rho(n1)
Worksheets("simulation_loop_variables").Cells(94, counter + 3) = Rho(n1 + 1)
Worksheets("simulation_loop_variables").Cells(95, counter + 3) = Rho(n1 + n2)
Worksheets("simulation_loop_variables").Cells(96, counter + 3) = Rho(n1 + n2 + 1)
Worksheets("simulation_loop_variables").Cells(97, counter + 3) = Rho(n)


Worksheets("simulation_loop_variables").Cells(99, counter + 3) = Temp(1)
Worksheets("simulation_loop_variables").Cells(100, counter + 3) = Temp(n1)
Worksheets("simulation_loop_variables").Cells(101, counter + 3) = Temp(n1 + 1)
Worksheets("simulation_loop_variables").Cells(102, counter + 3) = Temp(n1 + n2)
Worksheets("simulation_loop_variables").Cells(103, counter + 3) = Temp(n1 + n2 + 1)
Worksheets("simulation_loop_variables").Cells(104, counter + 3) = Temp(n)


Worksheets("simulation_loop_variables").Cells(106, counter + 3) = mb(1)
Worksheets("simulation_loop_variables").Cells(107, counter + 3) = mb(n1)
Worksheets("simulation_loop_variables").Cells(108, counter + 3) = mb(n1 + 1)
Worksheets("simulation_loop_variables").Cells(109, counter + 3) = mb(n1 + n2)
Worksheets("simulation_loop_variables").Cells(110, counter + 3) = mb(n1 + n2 + 1)
Worksheets("simulation_loop_variables").Cells(111, counter + 3) = mb(n)


Worksheets("simulation_loop_variables").Cells(113, counter + 3) = cb(1)
Worksheets("simulation_loop_variables").Cells(114, counter + 3) = cb(n1)
Worksheets("simulation_loop_variables").Cells(115, counter + 3) = cb(n1 + 1)
Worksheets("simulation_loop_variables").Cells(116, counter + 3) = cb(n1 + n2)
Worksheets("simulation_loop_variables").Cells(117, counter + 3) = cb(n1 + n2 + 1)
Worksheets("simulation_loop_variables").Cells(120, counter + 3) = cb(n)


Worksheets("simulation_loop_variables").Cells(120, counter + 3) = Xin1(1)
Worksheets("simulation_loop_variables").Cells(121, counter + 3) = Xin1(n1)
Worksheets("simulation_loop_variables").Cells(122, counter + 3) = Xin1(n1 + 1)
Worksheets("simulation_loop_variables").Cells(123, counter + 3) = Xin1(n1 + n2)
Worksheets("simulation_loop_variables").Cells(124, counter + 3) = Xin1(n1 + n2 + 1)
Worksheets("simulation_loop_variables").Cells(125, counter + 3) = Xin1(n)


Worksheets("simulation_loop_variables").Cells(127, counter + 3) = cin1(1)
Worksheets("simulation_loop_variables").Cells(128, counter + 3) = cin1(n1)
Worksheets("simulation_loop_variables").Cells(129, counter + 3) = cin1(n1 + 1)
```

```
Worksheets("simulation_loop_variables").Cells(130, counter + 3) = cin1(n1 + n2)
Worksheets("simulation_loop_variables").Cells(131, counter + 3) = cin1(n1 + n2 + 1)
Worksheets("simulation_loop_variables").Cells(132, counter + 3) = cin1(n)


Worksheets("simulation_loop_variables").Cells(134, counter + 3) = Rhoin1(1)
Worksheets("simulation_loop_variables").Cells(135, counter + 3) = Rhoin1(n1)
Worksheets("simulation_loop_variables").Cells(136, counter + 3) = Rhoin1(n1 + 1)
Worksheets("simulation_loop_variables").Cells(137, counter + 3) = Rhoin1(n1 + n2)
Worksheets("simulation_loop_variables").Cells(138, counter + 3) = Rhoin1(n1 + n2 + 1)
Worksheets("simulation_loop_variables").Cells(139, counter + 3) = Rhoin1(n)


Worksheets("simulation_loop_variables").Cells(141, counter + 3) = dhin1(1)
Worksheets("simulation_loop_variables").Cells(142, counter + 3) = dhin1(n1)
Worksheets("simulation_loop_variables").Cells(143, counter + 3) = dhin1(n1 + 1)
Worksheets("simulation_loop_variables").Cells(144, counter + 3) = dhin1(n1 + n2)
Worksheets("simulation_loop_variables").Cells(145, counter + 3) = dhin1(n1 + n2 + 1)
Worksheets("simulation_loop_variables").Cells(146, counter + 3) = dhin1(n)


Worksheets("simulation_loop_variables").Cells(148, counter + 3) = dmassin1(1)
Worksheets("simulation_loop_variables").Cells(149, counter + 3) = dmassin1(n1)
Worksheets("simulation_loop_variables").Cells(150, counter + 3) = dmassin1(n1 + 1)
Worksheets("simulation_loop_variables").Cells(151, counter + 3) = dmassin1(n1 + n2)
Worksheets("simulation_loop_variables").Cells(152, counter + 3) = dmassin1(n1 + n2 + 1)
Worksheets("simulation_loop_variables").Cells(153, counter + 3) = dmassin1(n)


Worksheets("simulation_loop_variables").Cells(155, counter + 3) = denthalpyin1(1)
Worksheets("simulation_loop_variables").Cells(156, counter + 3) = denthalpyin1(n1)
Worksheets("simulation_loop_variables").Cells(157, counter + 3) = denthalpyin1(n1 + 1)
Worksheets("simulation_loop_variables").Cells(158, counter + 3) = denthalpyin1(n1 + n2)
Worksheets("simulation_loop_variables").Cells(159, counter + 3) = denthalpyin1(n1 + n2+1)
Worksheets("simulation_loop_variables").Cells(160, counter + 3) = denthalpyin1(n)


Worksheets("simulation_loop_variables").Cells(162, counter + 3) = Xin2(1)
Worksheets("simulation_loop_variables").Cells(163, counter + 3) = Xin2(n1)
Worksheets("simulation_loop_variables").Cells(164, counter + 3) = Xin2(n1 + 1)
Worksheets("simulation_loop_variables").Cells(165, counter + 3) = Xin2(n1 + n2)
Worksheets("simulation_loop_variables").Cells(166, counter + 3) = Xin2(n1 + n2 + 1)
Worksheets("simulation_loop_variables").Cells(167, counter + 3) = Xin2(n)


Worksheets("simulation_loop_variables").Cells(169, counter + 3) = cin2(1)
```

```
Worksheets("simulation_loop_variables").Cells(170, counter + 3) = cin2(n1)
Worksheets("simulation_loop_variables").Cells(171, counter + 3) = cin2(n1 + 1)
Worksheets("simulation_loop_variables").Cells(172, counter + 3) = cin2(n1 + n2)
Worksheets("simulation_loop_variables").Cells(173, counter + 3) = cin2(n1 + n2 + 1)
Worksheets("simulation_loop_variables").Cells(174, counter + 3) = cin2(n)


Worksheets("simulation_loop_variables").Cells(176, counter + 3) = Rhoin2(1)
Worksheets("simulation_loop_variables").Cells(177, counter + 3) = Rhoin2(n1)
Worksheets("simulation_loop_variables").Cells(178, counter + 3) = Rhoin2(n1 + 1)
Worksheets("simulation_loop_variables").Cells(179, counter + 3) = Rhoin2(n1 + n2)
Worksheets("simulation_loop_variables").Cells(200, counter + 3) = Rhoin2(n1 + n2 + 1)
Worksheets("simulation_loop_variables").Cells(201, counter + 3) = Rhoin2(n)


Worksheets("simulation_loop_variables").Cells(203, counter + 3) = dhin2(1)
Worksheets("simulation_loop_variables").Cells(204, counter + 3) = dhin2(n1)
Worksheets("simulation_loop_variables").Cells(205, counter + 3) = dhin2(n1 + 1)
Worksheets("simulation_loop_variables").Cells(206, counter + 3) = dhin2(n1 + n2)
Worksheets("simulation_loop_variables").Cells(207, counter + 3) = dhin2(n1 + n2 + 1)
Worksheets("simulation_loop_variables").Cells(208, counter + 3) = dhin2(n)


Worksheets("simulation_loop_variables").Cells(190, counter + 3) = dmassin2(1)
Worksheets("simulation_loop_variables").Cells(191, counter + 3) = dmassin2(n1)
Worksheets("simulation_loop_variables").Cells(192, counter + 3) = dmassin2(n1 + 1)
Worksheets("simulation_loop_variables").Cells(193, counter + 3) = dmassin2(n1 + n2)
Worksheets("simulation_loop_variables").Cells(194, counter + 3) = dmassin2(n1 + n2 + 1)
Worksheets("simulation_loop_variables").Cells(195, counter + 3) = dmassin2(n)

Worksheets("simulation_loop_variables").Cells(197, counter + 3) = denthalpyin2(1)
Worksheets("simulation_loop_variables").Cells(198, counter + 3) = denthalpyin2(n1)
Worksheets("simulation_loop_variables").Cells(199, counter + 3) = denthalpyin2(n1 + 1)
Worksheets("simulation_loop_variables").Cells(200, counter + 3) = denthalpyin2(n1 + n2)
Worksheets("simulation_loop_variables").Cells(201, counter + 3) = denthalpyin2(n1 + n2 + 1)
Worksheets("simulation_loop_variables").Cells(202, counter + 3) = denthalpyin2(n)


Worksheets("simulation_loop_variables").Cells(204, counter + 3) = Xout3(1)
Worksheets("simulation_loop_variables").Cells(205, counter + 3) = Xout3(n1)
Worksheets("simulation_loop_variables").Cells(206, counter + 3) = Xout3(n1 + 1)
Worksheets("simulation_loop_variables").Cells(207, counter + 3) = Xout3(n1 + n2)
Worksheets("simulation_loop_variables").Cells(208, counter + 3) = Xout3(n1 + n2 + 1)
Worksheets("simulation_loop_variables").Cells(209, counter + 3) = Xout3(n)
```

```
Worksheets("simulation_loop_variables").Cells(211, counter + 3) = cout3(1)
Worksheets("simulation_loop_variables").Cells(212, counter + 3) = cout3(n1)
Worksheets("simulation_loop_variables").Cells(213, counter + 3) = cout3(n1 + 1)
Worksheets("simulation_loop_variables").Cells(214, counter + 3) = cout3(n1 + n2)
Worksheets("simulation_loop_variables").Cells(215, counter + 3) = cout3(n1 + n2 + 1)
Worksheets("simulation_loop_variables").Cells(216, counter + 3) = cout3(n)


Worksheets("simulation_loop_variables").Cells(220, counter + 3) = Rhoout3(1)
Worksheets("simulation_loop_variables").Cells(219, counter + 3) = Rhoout3(n1)
Worksheets("simulation_loop_variables").Cells(220, counter + 3) = Rhoout3(n1 + 1)
Worksheets("simulation_loop_variables").Cells(221, counter + 3) = Rhoout3(n1 + n2)
Worksheets("simulation_loop_variables").Cells(222, counter + 3) = Rhoout3(n1 + n2 + 1)
Worksheets("simulation_loop_variables").Cells(223, counter + 3) = Rhoout3(n)


Worksheets("simulation_loop_variables").Cells(225, counter + 3) = dhout3(1)
Worksheets("simulation_loop_variables").Cells(226, counter + 3) = dhout3(n1)
Worksheets("simulation_loop_variables").Cells(227, counter + 3) = dhout3(n1 + 1)
Worksheets("simulation_loop_variables").Cells(228, counter + 3) = dhout3(n1 + n2)
Worksheets("simulation_loop_variables").Cells(229, counter + 3) = dhout3(n1 + n2 + 1)
Worksheets("simulation_loop_variables").Cells(230, counter + 3) = dhout3(n)

Worksheets("simulation_loop_variables").Cells(232, counter + 3) = dmassout3(1)
Worksheets("simulation_loop_variables").Cells(233, counter + 3) = dmassout3(n1)
Worksheets("simulation_loop_variables").Cells(234, counter + 3) = dmassout3(n1 + 1)
Worksheets("simulation_loop_variables").Cells(235, counter + 3) = dmassout3(n1 + n2)
Worksheets("simulation_loop_variables").Cells(236, counter + 3) = dmassout3(n1 + n2 + 1)
Worksheets("simulation_loop_variables").Cells(237, counter + 3) = dmassout3(n)


Worksheets("simulation_loop_variables").Cells(239, counter + 3) = denthalpyout3(1)
Worksheets("simulation_loop_variables").Cells(240, counter + 3) =denthalpyout3(n1)
Worksheets("simulation_loop_variables").Cells(241, counter + 3) = denthalpyout3(n1 + 1)
Worksheets("simulation_loop_variables").Cells(242, counter + 3) = denthalpyout3(n1 + n2)
Worksheets("simulation_loop_variables").Cells(243, counter + 3) = denthalpyout3(n1 + n2 + 1)
Worksheets("simulation_loop_variables").Cells(244, counter + 3) = denthalpyout3(n)


Worksheets("simulation_loop_variables").Cells(246, counter + 3) = Xout4(j)
Worksheets("simulation_loop_variables").Cells(247, counter + 3) = Xout4(n1)
Worksheets("simulation_loop_variables").Cells(248, counter + 3) = Xout4(n1 + 1)
Worksheets("simulation_loop_variables").Cells(249, counter + 3) = Xout4(n1 + n2)
Worksheets("simulation_loop_variables").Cells(250, counter + 3) = Xout4(n1 + n2 + 1)
Worksheets("simulation_loop_variables").Cells(251, counter + 3) = Xout4(n)
```

```
Worksheets("simulation_loop_variables").Cells(253, counter + 3) = cout4(1)
Worksheets("simulation_loop_variables").Cells(254, counter + 3) = cout4(n1)
Worksheets("simulation_loop_variables").Cells(255, counter + 3) = cout4(n1 + 1)
Worksheets("simulation_loop_variables").Cells(256, counter + 3) = cout4(n1 + n2)
Worksheets("simulation_loop_variables").Cells(257, counter + 3) = cout4(n1 + n2 + 1)
Worksheets("simulation_loop_variables").Cells(258, counter + 3) = cout4(n)


Worksheets("simulation_loop_variables").Cells(260, counter + 3) = Rhoout4(1)
Worksheets("simulation_loop_variables").Cells(261, counter + 3) = Rhoout4(n1)
Worksheets("simulation_loop_variables").Cells(262, counter + 3) = Rhoout4(n1 + 1)
Worksheets("simulation_loop_variables").Cells(263, counter + 3) = Rhoout4(n1 + n2)
Worksheets("simulation_loop_variables").Cells(264, counter + 3) = Rhoout4(n1 + n2 + 1)
Worksheets("simulation_loop_variables").Cells(265, counter + 3) = Rhoout4(n)


Worksheets("simulation_loop_variables").Cells(267, counter + 3) = dhout4(1)
Worksheets("simulation_loop_variables").Cells(268, counter + 3) = dhout4(n1)
Worksheets("simulation_loop_variables").Cells(269, counter + 3) = dhout4(n1 + 1)
Worksheets("simulation_loop_variables").Cells(270, counter + 3) = dhout4(n1 + n2)
Worksheets("simulation_loop_variables").Cells(271, counter + 3) = dhout4(n1 + n2 + 1)
Worksheets("simulation_loop_variables").Cells(272, counter + 3) = dhout4(n)


Worksheets("simulation_loop_variables").Cells(274, counter + 3) = dmassout4(1)
Worksheets("simulation_loop_variables").Cells(275, counter + 3) = dmassout4(n1)
Worksheets("simulation_loop_variables").Cells(276, counter + 3) = dmassout4(n1 + 1)
Worksheets("simulation_loop_variables").Cells(277, counter + 3) = dmassout4(n1 + n2)
Worksheets("simulation_loop_variables").Cells(278, counter + 3) = dmassout4(n1 + n2 + 1)
Worksheets("simulation_loop_variables").Cells(279, counter + 3) = dmassout4(n)


Worksheets("simulation_loop_variables").Cells(281, counter + 3) = denthalpyout4(1)
Worksheets("simulation_loop_variables").Cells(282, counter + 3) = denthalpyout4(n1)
Worksheets("simulation_loop_variables").Cells(283, counter + 3) = denthalpyout4(n1 + 1)
Worksheets("simulation_loop_variables").Cells(284, counter + 3) = denthalpyout4(n1 + n2)
Worksheets("simulation_loop_variables").Cells(285, counter + 3) = denthalpyout4(n1 + n2 + 1)
Worksheets("simulation_loop_variables").Cells(286, counter + 3) = denthalpyout4(n)


Worksheets("simulation_loop_variables").Cells(288, counter + 3) = dmassin(1)
Worksheets("simulation_loop_variables").Cells(289, counter + 3) = dmassin(n1)
Worksheets("simulation_loop_variables").Cells(290, counter + 3) = dmassin(n1 + 1)
Worksheets("simulation_loop_variables").Cells(291, counter + 3) = dmassin(n1 + n2)
Worksheets("simulation_loop_variables").Cells(292, counter + 3) = dmassin(n1 + n2 + 1)
```

```
Worksheets("simulation_loop_variables").Cells(293, counter + 3) = dmassin(n)


Worksheets("simulation_loop_variables").Cells(295, counter + 3) = dmassout(1)
Worksheets("simulation_loop_variables").Cells(296, counter + 3) = dmassout(n1)
Worksheets("simulation_loop_variables").Cells(297, counter + 3) = dmassout(n1 + 1)
Worksheets("simulation_loop_variables").Cells(298, counter + 3) = dmassout(n1 + n2)
Worksheets("simulation_loop_variables").Cells(299, counter + 3) = dmassout(n1 + n2 + 1)
Worksheets("simulation_loop_variables").Cells(300, counter + 3) = dmassout(n)


Worksheets("simulation_loop_variables").Cells(302, counter + 3) = denthalpyin(1)
Worksheets("simulation_loop_variables").Cells(303, counter + 3) = denthalpyin(n1)
Worksheets("simulation_loop_variables").Cells(304, counter + 3) = denthalpyin(n1 + 1)
Worksheets("simulation_loop_variables").Cells(305, counter + 3) = denthalpyin(n1 + n2)
Worksheets("simulation_loop_variables").Cells(306, counter + 3) = denthalpyin(n1 + n2 + 1)
Worksheets("simulation_loop_variables").Cells(307, counter + 3) = denthalpyin(n)


Worksheets("simulation_loop_variables").Cells(309, counter + 3) = denthalpyout(1)
Worksheets("simulation_loop_variables").Cells(310, counter + 3) = denthalpyout(n1)
Worksheets("simulation_loop_variables").Cells(311, counter + 3) = denthalpyout(n1 + 1)
Worksheets("simulation_loop_variables").Cells(312, counter + 3) = denthalpyout(n1 + n2)
Worksheets("simulation_loop_variables").Cells(313, counter + 3) = denthalpyout(n1 + n2 + 1)
Worksheets("simulation_loop_variables").Cells(314, counter + 3) = denthalpyout(n)
```

```
Worksheets("simulation_loop_variables").Cells(316, counter + 3) = cin(j)
Worksheets("simulation_loop_variables").Cells(317, counter + 3) = cin(n1)
Worksheets("simulation_loop_variables").Cells(320, counter + 3) = cin(n1 + 1)
Worksheets("simulation_loop_variables").Cells(319, counter + 3) = cin(n1 + n2)
Worksheets("simulation_loop_variables").Cells(320, counter + 3) = cin(n1 + n2 + 1)
Worksheets("simulation_loop_variables").Cells(321, counter + 3) = cin(n)


Worksheets("simulation_loop_variables").Cells(323, counter + 3) = cout(1)
Worksheets("simulation_loop_variables").Cells(324, counter + 3) = cout(n1)
Worksheets("simulation_loop_variables").Cells(325, counter + 3) = cout(n1 + 1)
Worksheets("simulation_loop_variables").Cells(326, counter + 3) = cout(n1 + n2)
Worksheets("simulation_loop_variables").Cells(327, counter + 3) = cout(n1 + n2 + 1)
Worksheets("simulation_loop_variables").Cells(328, counter + 3) = cout(n)


Worksheets("simulation_loop_variables").Cells(330, counter + 3) = ma(1)
Worksheets("simulation_loop_variables").Cells(331, counter + 3) = ma(n1)
Worksheets("simulation_loop_variables").Cells(332, counter + 3) = ma(n1 + 1)
Worksheets("simulation_loop_variables").Cells(333, counter + 3) = ma(n1 + n2)
Worksheets("simulation_loop_variables").Cells(334, counter + 3) = ma(n1 + n2 + 1)
Worksheets("simulation_loop_variables").Cells(335, counter + 3) = ma(n)


Worksheets("simulation_loop_variables").Cells(337, counter + 3) = ca(1)
Worksheets("simulation_loop_variables").Cells(338, counter + 3) = ca(n1)
Worksheets("simulation_loop_variables").Cells(339, counter + 3) = ca(n1 + 1)
Worksheets("simulation_loop_variables").Cells(340, counter + 3) = ca(n1 + n2)
Worksheets("simulation_loop_variables").Cells(341, counter + 3) = ca(n1 + n2 + 1)
Worksheets("simulation_loop_variables").Cells(342, counter + 3) = ca(n)

 Next j

 End If
```

```
'******************************************************************
'THE Xp, Xq CALCULATIONS START HERE
'EVALUATION OF dQf AND dQh IS ALSO DONE HERE
'PAGES 260-265 OF BOOK
'******************************************************************


    For j = 1 To n

      If XR(j) = XR1(j) And XL(j) = XL1(j) Then

        Xq(j) = XL1(j)
        Xp(j) = XR1(j)

        ElseIf XR(j) = XR1(j) Then

          E(j) = a0(j) * dt / L(j)

          B(j) = E(j) * (XL(j) - XL1(j))
          D(j) = XL(j) / B(j)

          Xp(j) = XR1(j)

          Xq(j) = (1 + D(j) + G4 * Xp(j)) / (G6 + (1 / B(j)))

        ElseIf XL(j) = XL1(j) Then

          E(j) = a0(j) * dt / L(j)

          A(j) = E(j) * (XR1(j) - XR(j))

          C(j) = XR1(j) / A(j)


          Xq(j) = XL1(j)

          Xp(j) = (1 + C(j) + G4 * Xq(j)) / (G6 + (1 / A(j)))

        Else

          E(j) = a0(j) * dt / L(j)

          A(j) = E(j) * (XR1(j) - XR(j))
          B(j) = E(j) * (XL(j) - XL1(j))
```

```
        C(j) = XR1(j) / A(j)
        D(j) = XL(j) / B(j)

        FR(j) = (G6 + (1 / A(j))) / G4
        FL(j) = (G6 + (1 / B(j))) / G4


        Xp(j) = (1 + D(j) + FL(j) + FL(j) * C(j)) / (G4 * (FR(j) * FL(j) - 1))

        Xq(j) = (1 + C(j) + FR(j) + FR(j) * D(j)) / (G4 * (FR(j) * FL(j) - 1))

    End If

        XR1new(j) = Xp(j)
        XLnew(j) = Xq(j)

'SUPERPOSITION VARIABLE CALCULATIONS

'SUPERSONIC CHECK DONE HERE, IF SUPERSONIC THE X VALUES NEED
'TO BE RECALCULATED

      cs(j) = G5 * a0(j) * (Xp(j) - Xq(j))          'Superposition velocity

      Call supersonic_check(Xp(j), Xq(j), XRmod, XLmod)

      Xp(j) = XRmod
      Xq(j) = XLmod

      cs(j) = G5 * a0(j) * (Xp(j) - Xq(j))          'Superposition velocity


   'Calculation of Updated Representative parameters in the mesh is calculated here
   'The pressure amplitude ratio is assumed to be the average of the superposition
   'pressures at either end of the mesh


        XR1new(j) = Xp(j)
        XLnew(j) = Xq(j)

    'pressure amplitude ratio

        X(j) = ((XR(j) + XLnew(j) - 1) + (XR1new(j) + XL1(j) - 1)) / 2
        p(j) = p0 * X(j) ^ G7                         'average pressure
        Rho(j) = Rho0(j) * X(j) ^ G5                  'Density
        Temp(j) = T0(j) * X(j) ^ 2                    'Temperature
```

```
'Calculation of Superposition variables for each mesh is done here

        Xs(j) = Xp(j) + Xq(j) - 1
        ps(j) = p0 * Xs(j) ^ G7
        Rhos(j) = Rho0(j) * Xs(j) ^ G5
        Ts(j) = T0(j) * Xs(j) ^ 2

'Calculating the Thermal conductivity coefficient,viscosity coefficient
'friction factor and Convection heat transfer coefficient from
'Superposition Temperature values


'Thermal conductivity

Ck(j) = 6.1944 * 0.001 + 7.3814 * 0.00001 * Ts(j) - 1.2491 * 0.00000001 * Ts(j) ^ 2
                                                                            'W/mK

'Coefficient of Viscosity

mhu(j) = 0.000007457 + 0.000000041547 * Ts(j) - 7.4793 * (10 ^ -12) * Ts(j) ^ 2
                                                                            'kg/ms

'Reynolds number

Re(j) = Rhos(j) * dia(j) * cs(j) / (mhu(j))


'Checking for turbulent or laminar flow and calculating appropriate friction factor

If Re(j) >= 4000 Then

    Cf(j) = 0.0791 / (Re(j) ^ 0.25)

Else

    Cf(j) = 0.01

End If

Ch(j) = (Ck(j) * Cf(j) * Re(j)) / (2 * dia(j))
```

```
'Here the Heat transfer due to friction and Convective heat transfer from the wall to
'the gas is calculated


        Areasurf(j) = Pi * dia(j) * L(j) 'Surface area of wall

 'Heat transfer due to friction

        dQf(j) = Abs(Cf(j) * Areasurf(j) * Rhos(j) * (cs(j) ^ 3) * dt / 2)

'Convective Heat transfer from wall

        dQh(j) = Ch(j) * Areasurf(j) * (Tw - Ts(j)) * dt

    Next j

'*********************************************************************
'THE FIRST LAW CALCULATIONS START HERE
'CALCULATING NEW REFERENCE CONDITIONS FOR EACH MESH SPACE
'PAGES 276-279 OF BOOK
'*********************************************************************

    For j = 1 To n

        dU(j) = dQf(j) + dQh(j) + denthalpyin(j) - denthalpyout(j)

    Ta(j) = (mb(j) * Tb(j) + ((dU(j) - 0.5 * (ma(j) * ca(j) ^ 2 - mb(j) * cb(j) ^ 2)) / Cv)) / ma(j)

        T0(j) = Ta(j) / X(j) ^ 2        'New reference temperature


        Worksheets("T0_Check").Cells(counter + 11, j + 3) = T0(j)

        If T0(j) < 0 Then

            Msg = "                      T0 value negative                      "
            Style = vbOKOnly
            Title = "Error"

            Response = MsgBox(Msg, Style, Title)

            GoTo endofprogram

        End If
```

```
    'Here new reference acoustic velocity and density are calculated

     a0(j) = Sqr(gamma * gc * T0(j))  'New reference acoustic velocity

     Rho0(j) = p0 / (gc * T0(j))      'New reference density


  Next j

'*********************************************************************
' THE Xp, Xq CALCULATIONS ARE REDONE HERE TO UPDATE THE
' PRESSURE AMPLITUDE VALUES WITH NEW
' ACOUSTIC VELOCITY FOR EACH MESH AS IN PAGES 260-265 OF BOOK

 ' XR1new = XR1 (+/-) friction (+/-) heat transfer effects
 ' XLnew = XL (+/-) friction (+/-) heat transfer effects

'*********************************************************************


  For j = 1 To n

     If XR(j) = XR1(j) And XL(j) = XL1(j) Then

        Xq(j) = XL1(j)
        Xp(j) = XR1(j)

        ElseIf XR(j) = XR1(j) Then

           E(j) = a0(j) * dt / L(j)

           B(j) = E(j) * (XL(j) - XL1(j))
           D(j) = XL(j) / B(j)

           Xp(j) = XR1(j)

           Xq(j) = (1 + D(j) + G4 * Xp(j)) / (G6 + (1 / B(j)))

        ElseIf XL(j) = XL1(j) Then

           E(j) = a0(j) * dt / L(j)

           A(j) = E(j) * (XR1(j) - XR(j))

           C(j) = XR1(j) / A(j)
```

```
        Xq(j) = XL1(j)
        Xp(j) = (1 + C(j) + G4 * Xq(j)) / (G6 + (1 / A(j)))

      Else

        E(j) = a0(j) * dt / L(j)

        A(j) = E(j) * (XR1(j) - XR(j))
        B(j) = E(j) * (XL(j) - XL1(j))
        C(j) = XR1(j) / A(j)
        D(j) = XL(j) / B(j)

        FR(j) = (G6 + (1 / A(j))) / G4
        FL(j) = (G6 + (1 / B(j))) / G4


        Xp(j) = (1 + D(j) + FL(j) + FL(j) * C(j)) / (G4 * (FR(j) * FL(j) - 1))
        Xq(j) = (1 + C(j) + FR(j) + FR(j) * D(j)) / (G4 * (FR(j) * FL(j) - 1))

    End If

        XR1(j) = Xp(j)
        XL(j) = Xq(j)

  Next j
```

```
'*********************************************************************
' THE WAVE REFLECTION AT INTERSECTIONS /EXPANSIONS /
' CONTRACTIONS/ OPEN END CALCULATIONS START HERE AS
' IN PAGES 194-195,213-217 OF BOOK
'*********************************************************************

    'INTERSECTIONS BETWEEN MESHES (PAGES 194-195)

    For i = 1 To n - 1

'*********************************************************************
 'HERE THE WAVE EXPANSION AND CONTRACTION ROUTINES ARE
 ' INCORPORATED
 ' If d2>d1 then call expansion,d3<d2 call contraction
 ' IF i = n1 then call expansion, i = n1 + n2 call contraction
 ' OUTPUT WILL BE REFLECTED PRESSURE WAVE VALUES X1d AND X2d
'*********************************************************************

     If i = n1 Then

        If d2 > d1 Then

          Call sudden_expansion

          XL1(i) = Xref1

          XR(i + 1) = Xref2

          T0(i + 1) = (a02 ^ 2) / (gamma * gc)

          'Here new reference acoustic velocity and density are recalculated

          a0(i + 1) = Sqr(gamma * gc * T0(i + 1)) 'New reference acoustic velocity

          Rho0(i + 1) = p0 / (gc * T0(i + 1))  'New reference density

        Else
```

```
   'Assigning values to temporary variables

     x1(i) = XR1(i)
     x2(i) = XL(i + 1)

     'Reflection calculations

     X2d(i) = (2 * x2(i) - x1(i) * (1 - ((a0(i) * G5) / (a0(i + 1) * G5)))) / (1 +
                                            ((a0(i) * G5) / (a0(i + 1) * G5)))

     X1d(i) = x1(i) + X2d(i) - x2(i)

     'Reassigning the reflected values to the appropriate variables

     XR(i + 1) = X1d(i)
     XL1(i) = X2d(i)


   End If

 Else

'********************************************************************

 If i = (n1 + n2) Then

   If d3 < d2 Then

     Call sudden_contraction

     XL1(i) = Xref1

     XR(i + 1) = Xref2


   Else

   'Assigning values to temporary variables

     x1(i) = XR1(i)
     x2(i) = XL(i + 1)

     'Reflection calculations

     X2d(i) = (2 * x2(i) - x1(i) * (1 - ((a0(i) * G5) / (a0(i + 1) * G5)))) / (1 +
                                            ((a0(i) * G5) / (a0(i + 1) * G5)))
```

```
            X1d(i) = x1(i) + X2d(i) - x2(i)

            'Reassigning the reflected values to the appropriate variables

            XR(i + 1) = X1d(i)
            XL1(i) = X2d(i)


        End If



    Else

            'Assigning values to temporary variables

            x1(i) = XR1(i)
            x2(i) = XL(i + 1)

            'Reflection calculations

            X2d(i) = (2 * x2(i) - x1(i) * (1 - ((a0(i) * G5) / (a0(i + 1) * G5)))) / (1 +
                                                    ((a0(i) * G5) / (a0(i + 1) * G5)))

            X1d(i) = x1(i) + X2d(i) - x2(i)

            'Reassigning the reflected values to the appropriate variables

            XR(i + 1) = X1d(i)
            XL1(i) = X2d(i)

      End If

    End If


Next i

  'OPEN END OF LAST MESH (PAGE 200)

        XL1(n) = 2 - XR1(n)
```

```
'*********************************************************************
' TIMESTEP CALCULATIONS END HERE
'*********************************************************************

' CALCULATING REPRESENTATIVE VARIABLES FOR EACH MESH
' (FINAL OUTPUT)

    For j = 1 To n

        'pressure amplitude ratio

        Xfinal(j) = ((XR(j) + XL(j) - 1) + (XR1(j) + XL1(j) - 1)) / 2

        Pfinal(j) = p0 * Xfinal(j) ^ G7                   'average pressure
        Rhofinal(j) = Rho0(j) * Xfinal(j) ^ G5            'Density
        Tfinal(j) = T0(j) * Xfinal(j) ^ 2                 'Temperature
        afinal(j) = a0(j) * Xfinal(j)                     'Acoustic velocity
        cfinal(j) = ca(j)                                 'Particle velocity


        'cfinal(j) = G5 * a0(j) * (Xfinal(j) - 1)         'Velocity

    Next j
```

```
'*********************************************************************
                ' PRINTING RESULTS TO OUTPUT SHEET
'*********************************************************************

    For j = 1 To n

        Worksheets("output").Cells(j + 6, 1) = j
        Worksheets("output").Cells(j + 6, 2) = Xfinal(j)
        Worksheets("output").Cells(j + 6, 3) = Pfinal(j)
        Worksheets("output").Cells(j + 6, 4) = Rhofinal(j)
        Worksheets("output").Cells(j + 6, 5) = Tfinal(j)
        Worksheets("output").Cells(j + 6, 6) = afinal(j)
        Worksheets("output").Cells(j + 6, 7) = cfinal(j)


    Next j



'*********************************************************************
' PRINTING OUTPUT VARIABLES FOR SELECT MESH LOCATIONS -1, n1,
n1+1, ' n1+n2,n1+n2+1,n
'*********************************************************************

        If counter < 253 Then

         'This is to avoid the program crashing after it has reached the limit
         'of columns of excel sheet for printing

            Worksheets("output_variables").Cells(5, counter + 4) = XR(1)
            Worksheets("output_variables").Cells(6, counter + 4) = XR(n1)
            Worksheets("output_variables").Cells(7, counter + 4) = XR(n1 + 1)
            Worksheets("output_variables").Cells(8, counter + 4) = XR(n1 + n2)
            Worksheets("output_variables").Cells(9, counter + 4) = XR(n1 + n2 + 1)
            Worksheets("output_variables").Cells(10, counter + 4) = XR(n)



            Worksheets("output_variables").Cells(12, counter + 4) = XR1(1)
            Worksheets("output_variables").Cells(13, counter + 4) = XR1(n1)
            Worksheets("output_variables").Cells(14, counter + 4) = XR1(n1 + 1)
```

```
Worksheets("output_variables").Cells(15, counter + 4) = XR1(n1 + n2)
Worksheets("output_variables").Cells(16, counter + 4) = XR1(n1 + n2 + 1)
Worksheets("output_variables").Cells(17, counter + 4) = XR1(n)


Worksheets("output_variables").Cells(19, counter + 4) = XL(1)
Worksheets("output_variables").Cells(20, counter + 4) = XL(n1)
Worksheets("output_variables").Cells(21, counter + 4) = XL(n1 + 1)
Worksheets("output_variables").Cells(22, counter + 4) = XL(n1 + n2)
Worksheets("output_variables").Cells(23, counter + 4) = XL(n1 + n2 + 1)
Worksheets("output_variables").Cells(24, counter + 4) = XL(n)

Worksheets("output_variables").Cells(26, counter + 4) = XL1(1)
Worksheets("output_variables").Cells(27, counter + 4) = XL1(n1)
Worksheets("output_variables").Cells(28, counter + 4) = XL1(n1 + 1)
Worksheets("output_variables").Cells(29, counter + 4) = XL1(n1 + n2)
Worksheets("output_variables").Cells(30, counter + 4) = XL1(n1 + n2 + 1)
Worksheets("output_variables").Cells(31, counter + 4) = XL1(n)

Worksheets("output_variables").Cells(33, counter + 4) = Rho0(1)
Worksheets("output_variables").Cells(34, counter + 4) = Rho0(n1)
Worksheets("output_variables").Cells(35, counter + 4) = Rho0(n1 + 1)
Worksheets("output_variables").Cells(36, counter + 4) = Rho0(n1 + n2)
Worksheets("output_variables").Cells(37, counter + 4) = Rho0(n1 + n2 + 1)
Worksheets("output_variables").Cells(38, counter + 4) = Rho0(n)


Worksheets("output_variables").Cells(40, counter + 4) = a0(1)
Worksheets("output_variables").Cells(41, counter + 4) = a0(n1)
Worksheets("output_variables").Cells(42, counter + 4) = a0(n1 + 1)
Worksheets("output_variables").Cells(43, counter + 4) = a0(n1 + n2)
Worksheets("output_variables").Cells(44, counter + 4) = a0(n1 + n2 + 1)
Worksheets("output_variables").Cells(45, counter + 4) = a0(n)

Worksheets("output_variables").Cells(47, counter + 4) = T0(1)
Worksheets("output_variables").Cells(48, counter + 4) = T0(n1)
Worksheets("output_variables").Cells(49, counter + 4) = T0(n1 + 1)
Worksheets("output_variables").Cells(50, counter + 4) = T0(n1 + n2)
Worksheets("output_variables").Cells(51, counter + 4) = T0(n1 + n2 + 1)
Worksheets("output_variables").Cells(52, counter + 4) = T0(n)


Worksheets("output_variables").Cells(54, counter + 4) = Xfinal(1)
Worksheets("output_variables").Cells(55, counter + 4) = Xfinal(n1)
Worksheets("output_variables").Cells(56, counter + 4) = Xfinal(n1 + 1)
Worksheets("output_variables").Cells(57, counter + 4) = Xfinal(n1 + n2)
```

```
    Worksheets("output_variables").Cells(58, counter + 4) = Xfinal(n1 + n2 + 1)
    Worksheets("output_variables").Cells(59, counter + 4) = Xfinal(n)


    Worksheets("output_variables").Cells(61, counter + 4) = Pfinal(1)
    Worksheets("output_variables").Cells(62, counter + 4) = Pfinal(n1)
    Worksheets("output_variables").Cells(63, counter + 4) = Pfinal(n1 + 1)
    Worksheets("output_variables").Cells(64, counter + 4) = Pfinal(n1 + n2)
    Worksheets("output_variables").Cells(65, counter + 4) = Pfinal(n1 + n2 + 1)
    Worksheets("output_variables").Cells(66, counter + 4) = Pfinal(n)


    Worksheets("output_variables").Cells(68, counter + 4) = Rhofinal(1)
    Worksheets("output_variables").Cells(69, counter + 4) = Rhofinal(n1)
    Worksheets("output_variables").Cells(70, counter + 4) = Rhofinal(n1 + 1)
    Worksheets("output_variables").Cells(71, counter + 4) = Rhofinal(n1 + n2)
    Worksheets("output_variables").Cells(72, counter + 4) = Rhofinal(n1 + n2 + 1)
    Worksheets("output_variables").Cells(73, counter + 4) = Rhofinal(n)


    Worksheets("output_variables").Cells(75, counter + 4) = Tfinal(1)
    Worksheets("output_variables").Cells(76, counter + 4) = Tfinal(n1)
    Worksheets("output_variables").Cells(77, counter + 4) = Tfinal(n1 + 1)
    Worksheets("output_variables").Cells(78, counter + 4) = Tfinal(n1 + n2)
    Worksheets("output_variables").Cells(79, counter + 4) = Tfinal(n1 + n2 + 1)
    Worksheets("output_variables").Cells(80, counter + 4) = Tfinal(n)


    Worksheets("output_variables").Cells(82, counter + 4) = afinal(1)
    Worksheets("output_variables").Cells(83, counter + 4) = afinal(n1)
    Worksheets("output_variables").Cells(84, counter + 4) = afinal(n1 + 1)
    Worksheets("output_variables").Cells(85, counter + 4) = afinal(n1 + n2)
    Worksheets("output_variables").Cells(86, counter + 4) = afinal(n1 + n2 + 1)
    Worksheets("output_variables").Cells(87, counter + 4) = afinal(n)


    Worksheets("output_variables").Cells(89, counter + 4) = cfinal(1)
    Worksheets("output_variables").Cells(90, counter + 4) = cfinal(n1)
    Worksheets("output_variables").Cells(91, counter + 4) = cfinal(n1 + 1)
    Worksheets("output_variables").Cells(92, counter + 4) = cfinal(n1 + n2)
    Worksheets("output_variables").Cells(93, counter + 4) = cfinal(n1 + n2 + 1)
    Worksheets("output_variables").Cells(94, counter + 4) = cfinal(n)

End If
```

```
For j = 1 To n

        Worksheets("X").Cells(counter + 11, j + 3) = Xfinal(j)
        Worksheets("C").Cells(counter + 11, j + 3) = cfinal(j)
        Worksheets("T").Cells(counter + 11, j + 3) = Tfinal(j)
        Worksheets("Rho").Cells(counter + 11, j + 3) = Rhofinal(j)
        Worksheets("P").Cells(counter + 11, j + 3) = Pfinal(j)
        Worksheets("a").Cells(counter + 11, j + 3) = afinal(j)

      Next j

    counter = counter + 1

      If counter < 253 Then

        Worksheets("output_variables").Cells(3, counter + 3) = aggtime

      End If

      Worksheets("X").Cells(counter + 10, 2) = aggtime
      Worksheets("C").Cells(counter + 10, 2) = aggtime
      Worksheets("T").Cells(counter + 10, 2) = aggtime
      Worksheets("Rho").Cells(counter + 10, 2) = aggtime
      Worksheets("P").Cells(counter + 10, 2) = aggtime
      Worksheets("a").Cells(counter + 10, 2) = aggtime
      Worksheets("T0_Check").Cells(counter + 10, 2) = aggtime

      Worksheets("X").Cells(counter + 10, 1) = counter
      Worksheets("C").Cells(counter + 10, 1) = counter
      Worksheets("T").Cells(counter + 10, 1) = counter
      Worksheets("Rho").Cells(counter + 10, 1) = counter
      Worksheets("P").Cells(counter + 10, 1) = counter
      Worksheets("a").Cells(counter + 10, 1) = counter
      Worksheets("T0_Check").Cells(counter + 10, 1) = counter

  Loop

        Msg = "                    Solution is done!                    "
        Style = vbOKOnly
        Title = "Simulator"
        Response = MsgBox(Msg, Style, Title)

endofprogram:

  End Sub
```

'***
'MESHER SUBROUTINE

' This routine sets up the mesh. The inputs required are lengths and diameters of
' individual segments of the geometry.

' The output is the number of meshes for each segment and total number of meshes and
' length and diameter of each mesh
'***

'Here the meshes are as closest as possible in length while maintaining a general purpose routine

```
Sub mesher ()

    If L1 > 25 Then
```

'Here the number of meshes is calculated by dividing segment lengths by 25.Then the integer part is taken and subtracted from the actual float value. This is multiplied by 25mm to get the size of the last mesh. if the difference is zero then the no of meshes is equal to the integer part else it is one more than the integer part.

```
          'eg : for 250mm length n1initial = 10
              'n1x = 0, n1 = 10

          '   : for 255mm length n1initial = 10.2
              'n1x = 0.2, n1 = 26


          n1initial = L1 / 25
          n1x = n1initial - Int(n1initial)


          If n1x = 0 Then

             n1 = Int(n1initial)

          Else

             n1 = Int(n1initial) + 1

          End If
```

```
For i = 1 To n1

        L(i) = (L1 / n1) * 0.001

      Next i

    Else

      n1 = 1
      L(n1) = L1 * 0.001

    End If


    For i = 1 To n1

      dia(i) = d1
      Area(i) = Pi * dia(i) ^ 2 / 4

    Next i

'CALCULATES THE MESH LENGTH AND NUMBER OF MESHES FOR SEGMENT 2
' AND APPENDS THE MESH LENGTH, DIAMETER AND AREA TO GLOBAL MESH
'  LENGTH ARRAY L(i) AND dia(i)


       If L2 > 25 Then

'Here the number of meshes is calculated by dividing segment lengths by 25.Then the
integer part is taken and subtracted from the actual float value. This is multiplied by
25mm to get the size of the last mesh. if the difference is zero then the no of meshes is
equal to the integer part else it is one more than the integer part.

      'eg : for 250mm length n1initial = 10
         'n1x = 0, n1 = 10

      '   : for 255mm length n1initial = 10.2
         'n1x = 0.2, n1 = 26

      n2initial = L2 / 25
      n2x = n2initial - Int(n2initial)


      If n2x = 0 Then

        n2 = Int(n2initial)
```

```
        Else

            n2 = Int(n2initial) + 1

        End If

        For i = (n1 + 1) To (n1 + n2)

            L(i) = (L2 / n2) * 0.001

        Next i

    Else

        n2 = 1
        L(n1 + n2) = L2 * 0.001

    End If

    For i = (n1 + 1) To (n1 + n2)

        dia(i) = d2
        Area(i) = Pi * dia(i) ^ 2 / 4

    Next i

' CALCULATES THE MESH LENGTH AND NUMBER OF MESHES FOR SEGMENT 3
' AND APPENDS THE MESH LENGTH, DIAMETER AND AREA TO GLOBAL MESH
' LENGTH ARRAY L(i) AND dia(i)

    If L3 > 25 Then

'Here the number of meshes is calculated by dividing segment lengths by 25.Then the
integer part is taken and subtracted from the actual float value. This is multiplied by
25mm to get the size of the last mesh. if the difference is zero then the no of meshes is
equal to the integer part else it is one more than the integer part.

        'eg : for 250mm length n1initial = 10
            'n1x = 0, n1 = 10

        '   : for 255mm length n1initial = 10.2
            'n1x = 0.2, n1 = 26
```

```
n3initial = L3 / 25
        n3x = n3initial - Int(n3initial)


        If n3x = 0 Then

           n3 = Int(n3initial)

        Else

           n3 = Int(n3initial) + 1

        End If

        For i = (n1 + n2 + 1) To (n1 + n2 + n3)

           L(i) = (L3 / n3) * 0.001

        Next i

     Else

        n3 = 1
        L(n1 + n2 + n3) = L3 * 0.001

     End If

     For i = (n1 + n2 + 1) To (n1 + n2 + n3)

        dia(i) = d3
        Area(i) = Pi * dia(i) ^ 2 / 4

     Next i

  'CALCULATES THE TOTAL NUMBER OF MESHES

     n = n1 + n2 + n3


End Sub
```

```
'*********************************************************************
                    'PULSE GENERATOR SUBROUTINE
'This routine assigns value to XR(1) by comparing values of aggregate time and time
of pulse (Tp).
'If aggregate time is greater than Tp, then XR(1)= 1. Else XR(1) = Xinlet
'*********************************************************************


Sub pulse_generator()

    If aggtime > Tp Then

        XR(1) = 1

    Else

        XR(1) = Xinlet

    End If

End Sub
```

'***

'SUPERSONIC CHECK SUBROUTINE

'This routine calculates the particle Mach number and checks for supersonic condition.

'If found to be supersonic, modifies the opposite moving pressure wave amplitudes for a shock condition and outputs the new amplitudes

'***

```
Sub supersonic_check(Xa As Double, Xb As Double, Xanew As Double, Xbnew As
                                                                  Double)

    'Rankine - Hugoniot equations

    'Xa and Xb are existing Pressure amplitude ratios
    'Xanew and Xbnew are modified Pressure amplitude ratios

    Dim Xsup As Double
    Dim Msup As Double
    Dim tau1 As Double
    Dim tau2 As Double
    Dim tau3 As Double
    Dim tau4 As Double


    Xsup = (Xa + Xb - 1)
    Msup = Abs(G5 * (Xa - Xb) / Xsup)

    If Msup > 1 Then

      tau1 = ((Msup ^ 2) + (2 / (gamma - 1))) /
                                    (((2 * gamma * Msup ^ 2) / (gamma – 1)) - 1)

      tau2 = ((2 * gamma * Msup ^ 2) / (gamma + 1)) - ((gamma - 1) / (gamma + 1))

      tau3 = (gamma - 1) * (Sqr(tau1)) / 2

      tau4 = Xsup * (tau2 ^ ((gamma - 1) / (2 * gamma)))


      Xanew = (1 + tau4 + tau3 * tau4) / 2

      Xbnew = (1 + tau4 - tau3 * tau4) / 2
```

```
        'new mach number

        Xsup = (Xanew + Xbnew - 1)
        Msup = Abs(G5 * (Xanew - Xbnew) / Xsup)

      Else

      Xanew = Xa
      Xbnew = Xb

      End If

   End Sub
```

'***

'CASE SELECTOR SUBROUTINE

' This routine decides which case is applicable from among the two values possible

' for the first law application at the "in" side of all meshes.

'***

```
Sub case_selector_1_2(Xc As Double, Xd As Double, caseno As Integer)

   If (Abs(Xc) - Abs(Xd)) >= 0 Then

      caseno = 2

   Else

      caseno = 1

   End If

End Sub
```

```
'*********************************************************************
                    'CASE SELECTOR SUBROUTINE
'This routine decides which case is applicable from among the two values possible
' for the first law application at the "out" side of all meshes.
'*********************************************************************
Sub case_selector_3_4(Xe As Double, Xf As Double, caseno As Integer)

   If (Abs(Xe) - Abs(Xf)) >= 0 Then

      caseno = 4

   Else

      caseno = 3

   End If

End Sub
```

```
'*********************************************************************
'CLEAR SUBROUTINE

'This routine clears all the output cells at user discretion.

'*********************************************************************

Sub clearcells()

    Worksheets("output").Range("A7:G100").ClearContents
    Worksheets("output_variables").Range("B3: IV94").ClearContents
    Worksheets("simulation_loop_variables").Range("B5: IV342").ClearContents
    Worksheets("output_constants").Range("B5: B19").ClearContents
    Worksheets("output_constants").Range("B24: B31").ClearContents
    Worksheets("output_constants").Range("E5: M100").ClearContents
    Worksheets("X").Range("A10: IV1200").ClearContents
    Worksheets("X").Range("D8: IV8").ClearContents
    Worksheets("X").Cells(4, 3).ClearContents
    Worksheets("C").Range("A10: IV1200").ClearContents
    Worksheets("C").Range("D8: IV8").ClearContents
    Worksheets("C").Cells(4, 3).ClearContents
    Worksheets("T").Range("A10: IV1200").ClearContents
    Worksheets("T").Range("D8: IV8").ClearContents
    Worksheets("T").Cells(4, 3).ClearContents
    Worksheets("Rho").Range("A10: IV1200").ClearContents
    Worksheets("Rho").Range("D8: IV8").ClearContents
    Worksheets("Rho").Cells(4, 3).ClearContents
    Worksheets("P").Range("A10: IV1200").ClearContents
    Worksheets("P").Range("D8: IV8").ClearContents
    Worksheets("P").Cells(4, 3).ClearContents
    Worksheets("a").Range("A10: IV1200").ClearContents
    Worksheets("a").Range("D8: IV8").ClearContents
    Worksheets("a").Cells(4, 3).ClearContents
    Worksheets("T0_Check").Range("A10: IV1200").ClearContents
    Worksheets("T0_Check").Range("D8: IV8").ClearContents
    Worksheets("T0_Check").Cells(4, 3).ClearContents


End Sub
```

```
'*********************************************************************
'            NEWTON-RAPHSON WITH GAUSS-ELIMINATION
'            EXPANSION IN PIPE AREA - EQNS 2.10.7, 2.10.8, 2.10.9 PP 213-214
'            PROGRAM FOR SOLUTION OF SYSTEM OF NONLINEAR
'            EQUATIONS USING NEWTON-RAPHSON ITERATIVE METHOD
'*********************************************************************

'Ref Source :   Gordon.P.Blair,Design and simulation of four stroke engines-ch.2

' A subroutine for sudden expansion in pipe area is developed. This routine solves the
' equations denoted by the equation numbers above. A newton - raphson methodology is
' used. initial guesses are obtained by benson's method as in pp207 - 208 of ch2.

'The equations are

'CONTINUITY

'Rho01(Xi1+Xr1-1)^G5 A1 G5 a01(Xi1-Xr1) + Rho02(Xi2+Xr2-1)^G5 A2 G5 a02(Xi2-
Xr2) = 0

'FIRST LAW OF THERMODYNAMICS

'[(G5 a01(Xi1-Xr1))^2 + G5 a01^2 (Xi1+Xr1-1)^2 ]- [(G5 a02(Xi2-Xr2))^2 + G5 a02^2
(Xi2+Xr2-1)^2 ] = 0

'MOMENTUM

'p0 A2 [(Xi1+Xr1-1)^G7 - (Xi2+Xr2-1)^G7 ] + [Rho01(Xi1+Xr1-1)^G5 A1 G5 a01(Xi1-
Xr1) ]
'                              x[G5 a01(Xi1-Xr1)+ G5 a02 (Xi2-Xr2)] = 0


'The unknowns are Xr1, Xr2, a02.

'Initial guesses by Bensons solution eq 2.9.7 and 2.9.8 is as follows

'Xr1 = [(1-Ar)Xi1 + 2Xi2 Ar ]/[1+Ar]

'Xr2 = [2Xi1-Xi2(1-Ar)]/[1+Ar]

'a02 = sqrt (gamma * R * T02)
```

'Sonic velocity case

'If the Mach number at station1 (Ms1) i.e the smaller pipe exceeds unity it is brought back to unity
'and this directly gives Xr1 and then this value is fed into the next iteration

```
Sub sudden_expansion()


Xi1 = XR1(n1)
Xi2 = XL(n1 + 1)
a01 = a0(n1)
T02 = T0(n1 + 1)

A1 = Area(n1)
A2 = Area(n1 + 1)
Rho01 = Rho0(n1)
Rho02 = Rho0(n1 + 1)

a02 = Sqr(gamma * gc * T02)


    'The input variables are Xi1,Xi2,a01,T02,A1,A2,gc,Rho01, Rho02.
    'The output variables are Xref1,Xref2,a02.



    'Declaration of variables

    Dim Ar As Double, Ms1 As Double

    Dim n As Integer, m As Integer, i As Integer, j As Integer
    Dim maxi As Integer, p As Integer, k As Integer
    Dim AM(3, 4) As Double, XM(3) As Double, HM(3) As Double, DX(3) As Double
    Dim xmtemp(3) As Double

    'Number of equations

    n = 3

    'Initialising the variables

'Calculation of  initial guesses using benson's approximations
```

```
    Ar = A2 / A1

    'XM(1) = Xr1
    'XM(2) = Xr2
    'XM(3) = a02

'***********************************************************************
'Initial guesses by Bensons solution eq 2.9.7 and 2.9.8 is as follows
'***********************************************************************

    XM(1) = ((1 - Ar) * Xi1 + 2 * Xi2 * Ar) / (1 + Ar) 'Xr1
    XM(2) = (2 * Xi1 - Xi2 * (1 - Ar)) / (1 + Ar) 'Xr2
    XM(3) = Sqr(gamma * gc * T02)   'a02

'Number of iterations
    maxi = 1000
    m = n + 1
    i = 0

'***********************************************************************
' HERE THE MACH NUMBER CHECK AT STATION 1 NEEDS TO BE
' PERFORMED AND XM(1) NEEDS TO BE MODIFIED IF REQD
'***********************************************************************

'Mach number check at station 1 - eqn 2.10.10


    Ms1 = G5 * (Xi1 - XM(1)) / (Xi1 + XM(1) - 1)

    If Ms1 >= 1 Then

      XM(1) = (1 + G4 * Xi1) / G6 'Xr1

    End If
```

```
For i = 1 To maxi

'    function evaluation

        Call FUN_EXPN(XM, HM)


'    compute partial derivatives
        Call PAR_EXPN(XM, n, AM)

'    form the matrix by augmenting function values
     For k = 1 To n
     AM(k, m) = -HM(k)
     Next

'    solve the jacobian matrix
        Call GAUSS_EXPN(AM, n, DX)

'    apply the correction to XM values
      For k = 1 To n
        XM(k) = XM(k) + (DX(k))
      Next

      i = i + 1

      'Mach number check at station 1 - eqn 2.10.10


       Ms1 = G5 * (Xi1 - XM(1)) / (Xi1 + XM(1) - 1)

       If Ms1 >= 1 Then

          XM(1) = (1 + G4 * Xi1) / G6

       End If

      For k = 1 To n
          If Abs(XM(k) - xmtemp(k)) <= 0.01 Then Exit For
          xmtemp(k) = XM(k)
      Next

     ' if xmtemp
Next i

'    error upon substituting the solution in the equations
```

```
'     Call FUN_EXPN(XM, HM)

      Xref1 = XM(1) 'Xr1
      Xref2 = XM(2) 'Xr2

      a02 = XM(3)   'a02


End Sub

'*********************************************************************
'                              GAUSS ELIMINATION
'*********************************************************************

Sub GAUSS_EXPN(AM, n, YM)

     Dim XM(3) As Double
     Dim ORDC(3) As Double, ORD(3) As Double, qt As Double
     Dim t As Double, epsil As Double, sum As Double
     Dim i As Integer, j As Integer, m As Integer, nn As Integer
     Dim k As Integer, kk As Integer, p As Integer, r As Integer, index As Integer
     Dim determ As Integer, chec As Integer

     'convergence factor while triangularising the matrix

      epsil = 0.00000001
      m = n + 1
      nn = n - 1
      chec = 1

'     establishing the initial order in the column order vector

      For i = 1 To n
        ORDC(i) = i
      Next

'     segment for partial pivoting
      For p = 1 To nn
      Call PIVOT_EXPN(AM, n, ORD, ORDC, p)

'     triangularization by eliminating the variables

      kk = p + 1
      For i = kk To n
        If (Abs(AM(p, p)) < epsil) Then
          chec = 0
```

```
      Else
        qt = AM(i, p) / AM(p, p)
      End If
      For j = p To m
        AM(i, j) = AM(i, j) - qt * AM(p, j)
      Next
     Next
    Next

'   checking for the singularity of coefficient matrix

    determ = 1
    For i = 1 To n
     If ((Abs(AM(i, i)) < epsil) Or (chec = 0)) Then
      determ = 0

     MsgBox "Coefficient matrix is singular or nearly singular,No Solution exists "

     End If
    Next
    If (determ = 1 And chec = 1) Then

'   back substitution

    XM(n) = AM(n, m) / AM(n, n)
    For i = nn To 1 Step -1
      sum = 0
      k = i + 1
      For j = k To n
        sum = sum + AM(i, j) * XM(j)
          Next
      XM(i) = (AM(i, m) - sum) / AM(i, i)
       Next

'   rearranging the solution vector

    For i = 1 To n
      j = ORDC(i)
      YM(j) = XM(i)
       Next
    End If

End Sub
```

```
'**********************************************************************
'                COMPLETE PIVOTING (Loop in the calling program)
'**********************************************************************

Sub PIVOT_EXPN(A, n, ORD, ORDC, i)

   Dim t As Double, tem As Double
   Dim j As Integer, ii As Integer, jj As Integer
   Dim col As Integer, row As Integer, p As Integer, r As Integer, m As Integer

'  complete pivoting - finds the biggest value in the whole matrix

  row = i
  col = i
  For p = i To n
   For r = i To n
    If (Abs(A(row, col)) < Abs(A(p, r))) Then
     row = p
     col = r
    End If
   Next
  Next

  If (col <> i) Then
   For ii = 1 To n
    tem = A(ii, i)
    A(ii, i) = A(ii, col)
    A(ii, col) = tem
   Next
   t = ORDC(i)
   ORDC(i) = ORDC(col)
   ORDC(col) = t
  End If

  m = n + 1
  If (row <> i) Then
   For jj = 1 To m
     tem = A(i, jj)
     A(i, jj) = A(row, jj)
     A(row, jj) = tem
   Next
   t = ORD(i)
   ORD(i) = ORD(row)
   ORD(row) = t
  End If
```

```
End Sub

'*******************************************************************
'                SUBROUTINE FOR PARTIAL DERIVATIVE
'*******************************************************************

Sub PAR_EXPN(X, n, A)

      Dim i As Integer, j As Integer

'     compute partial derivatives and store it in the array A

      For j = 1 To n
         For i = 1 To n
         Call DER_EXPN(X, A)
         Next
      Next

End Sub
```

```
'*********************************************************************
'               SUBROUTINE FOR FUNCTION EVALUATION
'*********************************************************************

Sub FUN_EXPN(XX, F)

    Dim x1 As Double, x2 As Double, x3 As Double, x4 As Double


    x1 = XX(1)    'Xr1
    x2 = XX(2)    'Xr2
    x3 = XX(3)    'a02


    'Listing the four functions

    F(1) = Rho01 * (Xi1 + x1 - 1) ^ G5 * A1 * G5 * a01 * (Xi1 - x1) +
            Rho02 * (Xi2 + x2 - 1) ^ G5 * A2 * G5 * x3 * (Xi2 - x2) 'F(1) = 0


    F(2) = ((G5 * a01 * (Xi1 - x1)) ^ 2 + G5 * a01 ^ 2 * (Xi1 + x1 - 1) ^ 2)
            - ((G5 * x3 * (Xi2 - x2)) ^ 2 + G5 * x3 ^ 2 * (Xi2 + x2 - 1) ^ 2) 'F(2) = 0



    F(3) = p0 * A2 * ((Xi1 + x1 - 1) ^ G7 - (Xi2 + x2 - 1) ^ G7) +
            (Rho01 * (Xi1 + x1 - 1) ^ G5 * A1 * G5 * a01 * (Xi1 - x1)) * (G5 * a01 *
                                        (Xi1 - x1) + G5 * x3 * (Xi2 - x2)) 'F(3) = 0


End Sub
```

```
'*********************************************************************
'              SUBROUTINE FOR EVALUATION OF PARTIAL DERIVATIVES
'*********************************************************************
Sub DER_EXPN(XX, DFDX)

    Dim x1 As Double, x2 As Double, x3 As Double, x4 As Double

    x1 = XX(1)    'Xr1
    x2 = XX(2)    'Xr2
    x3 = XX(3)    'a02


    DFDX(1, 1) = Rho01 * A1 * G5 * a01 * (-(Xi1 + x1 - 1) ^ G5 +
                 (Xi1 - x1) * (G5 * (Xi1 + x1 - 1) ^ (G5 - 1))) 'dF1/dX1

    DFDX(1, 2) = Rho02 * A2 * G5 * x3 * (-(Xi2 + x2 - 1) ^ G5 +
                 (Xi2 - x2) * (G5 * (Xi2 + x2 - 1) ^ (G5 - 1))) 'dF1/dX2

    DFDX(1, 3) = Rho02 * A2 * G5 * (Xi2 + x2 - 1) ^ G5 * (Xi2 - x2) 'dF1/dX3


    DFDX(2, 1) = (-2 * G5 ^ 2 * a01 ^ 2 * (Xi1 - x1) +
                                  2 * G5 * a01 ^ 2 * (Xi1 + x1 - 1)) 'dF2/dX1

    DFDX(2, 2) = -(-2 * G5 ^ 2 * x3 ^ 2 * (Xi2 - x2) +
                                  2 * G5 * x3 ^ 2 * (Xi2 + x2 - 1)) 'dF2/dX2

    DFDX(2, 3) = -(2 * G5 ^ 2 * x3 * (Xi2 - x2) ^ 2 +
                                  2 * G5 * x3 * (Xi2 + x2 - 1) ^ 2) 'dF2/dX3

    DFDX(3, 1) = (p0 * A2 * G7 * (Xi1 + x1 - 1) ^ (G7 - 1)) _
           + (Rho01 * A1 * G5 ^ 2 * a01 ^ 2 * (-2 * (Xi1 + x1 - 1) ^ G5 * (Xi1 - x1) +
             G5 * (Xi1 - x1) ^ 2 * (Xi1 + x1 - 1) ^ (G5 - 1))) _
           + (Rho01 * A1 * G5 ^ 2 * a01 * x3 * (Xi2 - x2) * (-(Xi1 + x1 - 1) ^ G5 + G5
                                  * (Xi1 - x1) * (Xi1 + x1 - 1) ^ (G5 - 1)))

                                                                     'dF3/dX1

    DFDX(3, 2) = (-p0 * A2 * G7 * (Xi2 + x2 - 1) ^ (G7 - 1)) –
                 (Rho01 * A1 * G5 ^ 2 * a01 * x3 * (Xi1 + x1 - 1) ^ G5 * (Xi1 - x1))
                                                                     'dF3/dX2

    DFDX(3, 3) = Rho01 * A1 * G5 ^ 2 * a01 * (Xi1 + x1 - 1) ^ G5 * (Xi1 - x1) * (Xi2 -
x2) 'dF3/dX3

End Sub
```

```
'*********************************************************************
'           NEWTON-RAPHSON WITH GAUSS-ELIMINATION
'           CONTRACTION IN PIPE AREA - EQNS 2.11.7, 2.11.8 PP 217-220
'           PROGRAM FOR SOLUTION OF SYSTEM OF NONLINEAR
'           EQUATIONS USING NEWTON-RAPHSON ITERATIVE METHOD
'*********************************************************************

'Ref Source :   Gordon.P.Blair, Design and simulation of four stroke engines-ch.2

'A subroutine for sudden expansion in pipe area is developed. This routine solves the equations
'denoted by the equation numbers above. A newton - raphson methodology is used. initial guesses are
'obtained by benson's method as in pp207 - 208 of ch2.

'The equations are

'CONTINUITY

'(Xi1+Xr1-1)^G5 A1(Xi1-Xr1) + (Xi2+Xr2-1)^G5 A2(Xi2-Xr2) = 0

'FIRST LAW OF THERMODYNAMICS

'[G5(Xi1-Xr1)^2 + (Xi1+Xr1-1)^2 ]- [G5 (Xi2-Xr2)^2 + (Xi2+Xr2-1)^2 ] = 0



'The unknowns are Xr1, Xr2

'Initial guesses by Bensons solution eq 2.9.7 and 2.9.8 is as follows

'Xr1 = [(1-Ar)Xi1 + 2Xi2 Ar ]/[1+Ar]

'Xr2 = [2Xi1-Xi2(1-Ar)]/[1+Ar]


'Sonic velocity case

'If the Mach number at station1 (Ms2) i.e the smaller pipe exceeds unity it is brought back to unity
'and this directly gives Xr2 and then this value is fed into the next iteration
```

```
Sub sudden_contraction()


Xi1 = XR1(n1 + n2)
Xi2 = XL(n1 + n2 + 1)


A1 = Area(n1 + n2)
A2 = Area(n1 + n2 + 1)



    'The input variables are Xi1,Xi2.
    'The output variables are Xref1,Xref2.



    'Declaration of variables


    Dim Ar As Double, Ms2 As Double

    Dim n As Integer, m As Integer, i As Integer, j As Integer
    Dim maxi As Integer, p As Integer, k As Integer
    Dim AM(2, 3) As Double, XM(2) As Double, HM(2) As Double, DX(2) As Double
    Dim xmtemp(2) As Double

    'Number of equations

    n = 2

    'Initialising the variables

'Calculation of  initial guesses using benson's approximations


     Ar = A2 / A1

     'XM(1) = Xr1
     'XM(2) = Xr2
```

```
'*********************************************************************
            'Initial guesses by Bensons solution eq 2.9.7 and 2.9.8 is as follows
'*********************************************************************

    XM(1) = ((1 - Ar) * Xi1 + 2 * Xi2 * Ar) / (1 + Ar) 'Xr1
    XM(2) = (2 * Xi1 - Xi2 * (1 - Ar)) / (1 + Ar) 'Xr2


'Number of iterations

    maxi = 100
    m = n + 1
    i = 0

'*********************************************************************
'HERE THE MACH NUMBER CHECK AT STATION 1 NEEDS TO BE
PERFORMED AND XM(1) NEEDS TO BE MODIFIED IF REQD
'*********************************************************************

'Mach number check at station 2 - eqn 2.11.9


    Ms2 = G5 * (Xi2 - XM(2)) / (Xi2 + XM(2) - 1)

    If Ms2 >= 1 Then

      XM(2) = (1 + G4 * Xi2) / G6 'Xr1

    End If


'Providing the label for the goto statement that follows later

For i = 1 To maxi


'    function evaluation

     Call FUN_CONTRN(XM, HM)

'    compute partial derivatives

     Call PAR_CONTRN(XM, n, AM)
```

```
'    form the matrix by augmenting function values

     For k = 1 To n
     AM(k, m) = -HM(k)
     Next

'    solve the jacobian matrix

       Call GAUSS_CONTRN(AM, n, DX)

'    apply the correction to XM values

      For k = 1 To n
        XM(k) = XM(k) + (DX(k))
      Next

      i = i + 1

      'Mach number check at station 2 - eqn 2.11.9


       Ms2 = G5 * (Xi2 - XM(2)) / (Xi2 + XM(2) - 1)

       If Ms2 >= 1 Then

          XM(2) = (1 + G4 * Xi2) / G6 'Xr1

       End If

      For k = 1 To n
        If XM(k) = xmtemp(k) Then Exit For
        xmtemp(k) = XM(k)
      Next

Next i

'    error upon substituting the solution in the equations

     Call FUN_CONTRN(XM, HM)

     Xref1 = XM(1) 'Xr1
     Xref2 = XM(2) 'Xr2



End Sub
```

```
'*********************************************************************
'                          GAUSS ELIMINATION
'*********************************************************************

Sub GAUSS_CONTRN(AM, n, YM)

    Dim XM(2) As Double
    Dim ORDC(2) As Double, ORD(2) As Double, qt As Double
    Dim t As Double, epsil As Double, sum As Double

    Dim i As Integer, j As Integer, m As Integer, nn As Integer
    Dim k As Integer, kk As Integer, p As Integer, r As Integer, index As Integer
    Dim determ As Integer, chec As Integer

    'convergence factor while triangularising the matrix

     epsil = 0.00000001
     m = n + 1
     nn = n - 1
     chec = 1

'    establishing the initial order in the column order vector

     For i = 1 To n
       ORDC(i) = i
     Next

'    segment for partial pivoting

     For p = 1 To nn
     Call PIVOT_CONTRN(AM, n, ORD, ORDC, p)

'    triangularization by eliminating the variables

     kk = p + 1
     For i = kk To n
       If (Abs(AM(p, p)) < epsil) Then
         chec = 0
       Else
         qt = AM(i, p) / AM(p, p)
       End If
       For j = p To m
         AM(i, j) = AM(i, j) - qt * AM(p, j)
       Next
      Next
```

```
      Next

'     checking for the singularity of coefficient matrix

      determ = 1
      For i = 1 To n
        If ((Abs(AM(i, i)) < epsil) Or (chec = 0)) Then
         determ = 0

        MsgBox "Coefficient matrix is singular or nearly singular,No Solution exists "

        End If
      Next
      If (determ = 1 And chec = 1) Then

'     back substitution

      XM(n) = AM(n, m) / AM(n, n)
      For i = nn To 1 Step -1
         sum = 0
         k = i + 1
         For j = k To n
           sum = sum + AM(i, j) * XM(j)
             Next
         XM(i) = (AM(i, m) - sum) / AM(i, i)
          Next

'     rearranging the solution vector

      For i = 1 To n
         j = ORDC(i)
         YM(j) = XM(i)
          Next
      End If

End Sub
```

```
'*******************************************************************
'                COMPLETE PIVOTING (Loop in the calling program)
'*******************************************************************
Sub PIVOT_CONTRN(A, n, ORD, ORDC, i)

   Dim t As Double, tem As Double
   Dim j As Integer, ii As Integer, jj As Integer
   Dim col As Integer, row As Integer, p As Integer, r As Integer, m As Integer

'  complete pivoting - finds the biggest value in the whole matrix

  row = i
  col = i
  For p = i To n
   For r = i To n
    If (Abs(A(row, col)) < Abs(A(p, r))) Then
     row = p
     col = r
    End If
   Next
  Next

  If (col <> i) Then
   For ii = 1 To n
    tem = A(ii, i)
    A(ii, i) = A(ii, col)
    A(ii, col) = tem
   Next
   t = ORDC(i)
   ORDC(i) = ORDC(col)
   ORDC(col) = t
  End If

  m = n + 1
  If (row <> i) Then
   For jj = 1 To m
     tem = A(i, jj)
     A(i, jj) = A(row, jj)
     A(row, jj) = tem
   Next
   t = ORD(i)
   ORD(i) = ORD(row)
   ORD(row) = t
  End If

End Sub
```

```
'*********************************************************************
'                    SUBROUTINE FOR PARTIAL DERIVATIVE
'*********************************************************************

Sub PAR_CONTRN(X, n, A)

    Dim i As Integer, j As Integer

'   compute partial derivatives and store it in the array A

    For j = 1 To n
      For i = 1 To n
      Call DER_CONTRN(X, A)
      Next
    Next

End Sub

'*********************************************************************
'                    SUBROUTINE FOR FUNCTION EVALUATION
'*********************************************************************

Sub FUN_CONTRN(XX, F)

   Dim x1 As Double, x2 As Double


   x1 = XX(1)    'Xr1
   x2 = XX(2)    'Xr2



   'Listing the four functions

   F(1) = (Xi1 + x1 - 1) ^ G5 * A1 * (Xi1 - x1) + (Xi2 + x2 - 1) ^ G5 * A2 * (Xi2 - x2)
'F(1) = 0


   F(2) = (G5 * (Xi1 - x1) ^ 2 + (Xi1 + x1 - 1) ^ 2) - (G5 * (Xi2 - x2) ^ 2 + (Xi2 + x2 -
1) ^ 2) 'F(2) = 0



End Sub
```

```
'*******************************************************************
'              SUBROUTINE FOR EVALUATION OF PARTIAL DERIVATIVES
'*******************************************************************

Sub DER_CONTRN(XX, DFDX)

    Dim x1 As Double, x2 As Double

    x1 = XX(1)    'Xr1
    x2 = XX(2)    'Xr2


    DFDX(1, 1) = A1 * (-(Xi1 + x1 - 1) ^ G5 + (Xi1 - x1) * G5 * (Xi1 + x1 - 1) ^ (G5 -
1)) 'dF1/dX1

    DFDX(1, 2) = A2 * (-(Xi2 + x2 - 1) ^ G5 + (Xi2 - x2) * G5 * (Xi2 + x2 - 1) ^ (G5 -
1)) 'dF1/dX2


    DFDX(2, 1) = -2 * G5 * (Xi1 - x1) + 2 * (Xi1 + x1 - 1) 'dF2/dX1

    DFDX(2, 2) = -(-2 * G5 * (Xi2 - x2) + 2 * (Xi2 + x2 - 1)) 'dF2/dX2


End Sub
```

APPENDIX C

Detailed Results

Input parameters for Virtual 4 Stroke® software test case

The input parameters for the Virtual 4 Stroke® software test case discussed in section 5.2 are tabulated below.

Engine details

Overview

Engine.OriginalID	PROD00.00000020
Engine.Description	EXPMODEL2
Engine.Ambients.Ambient(1).Name	Int1
Engine.Ambients.Ambient(2).Name	Exh1
Engine.Connections.Connection(1).LeftSideComponentCode	1
Engine.Connections.Connection(1).LeftSideComponentName	Int1
Engine.Connections.Connection(1).RightSideComponentCode	2
Engine.Connections.Connection(1).RightSideComponentName	Int1
Engine.Connections.Connection(1).X1	864
Engine.Connections.Connection(1).Y1	576
Engine.Connections.Connection(1).X2	2016
Engine.Connections.Connection(1).Y2	576
Engine.Connections.Connection(2).LeftSideComponentCode	2
Engine.Connections.Connection(2).LeftSideComponentName	Int1
Engine.Connections.Connection(2).RightSideComponentCode	13
Engine.Connections.Connection(2).RightSideComponentName	Inv1
Engine.Connections.Connection(2).X1	2016
Engine.Connections.Connection(2).Y1	576
Engine.Connections.Connection(2).X2	3168
Engine.Connections.Connection(2).Y2	576
Engine.Connections.Connection(3).LeftSideComponentCode	13
Engine.Connections.Connection(3).LeftSideComponentName	Inv1
Engine.Connections.Connection(3).RightSideComponentCode	6
Engine.Connections.Connection(3).RightSideComponentName	TEST
Engine.Connections.Connection(3).X1	3168
Engine.Connections.Connection(3).Y1	576
Engine.Connections.Connection(3).X2	4032
Engine.Connections.Connection(3).Y2	1152
Engine.Connections.Connection(4).LeftSideComponentCode	13
Engine.Connections.Connection(4).LeftSideComponentName	Exv1
Engine.Connections.Connection(4).RightSideComponentCode	2
Engine.Connections.Connection(4).RightSideComponentName	Exh1
Engine.Connections.Connection(4).X1	5184
Engine.Connections.Connection(4).Y1	576
Engine.Connections.Connection(4).X2	6336
Engine.Connections.Connection(4).Y2	576

Overview (cont'd)

Engine.Connections.Connection(5).LeftSideComponentCode	6
Engine.Connections.Connection(5).LeftSideComponentName	TEST
Engine.Connections.Connection(5).RightSideComponentCode	13
Engine.Connections.Connection(5).RightSideComponentName	Exv1
Engine.Connections.Connection(5).X1	4032
Engine.Connections.Connection(5).Y1	1152
Engine.Connections.Connection(5).X2	5184
Engine.Connections.Connection(5).Y2	576
Engine.Connections.Connection(6).LeftSideComponentCode	2
Engine.Connections.Connection(6).LeftSideComponentName	Exh1
Engine.Connections.Connection(6).RightSideComponentCode	1
Engine.Connections.Connection(6).RightSideComponentName	Exh1
Engine.Connections.Connection(6).X1	6336
Engine.Connections.Connection(6).Y1	576
Engine.Connections.Connection(6).X2	7488
Engine.Connections.Connection(6).Y2	576
Engine.Connections.Connection(7).LeftSideComponentCode	1
Engine.Connections.Connection(7).LeftSideComponentName	Exh1
Engine.Connections.Connection(7).RightSideComponentCode	0
Engine.Connections.Connection(7).RightSideComponentName	0
Engine.Connections.Connection(7).X1	7488
Engine.Connections.Connection(7).Y1	576
Engine.Connections.Connection(7).X2	0
Engine.Connections.Connection(7).Y2	0

Table C-1. Overview of Engine input parameters for software test case

Cylinder details

Engine.Cylinders.Cylinder(1).Name	TEST
Engine.Cylinders.TEST.ClosedCycle.CombustionEfficiency	0.85
Engine.Cylinders.TEST.ClosedCycle.IgnitionDelay	13
Engine.Cylinders.TEST.ClosedCycle.IgnitionDuration	44
Engine.Cylinders.TEST.ClosedCycle.IgnitionTiming	-27
Engine.Cylinders.TEST.ClosedCycle.TrappedAirFuelRatio	12
Engine.Cylinders.TEST.ClosedCycle.WiebeA	6.02
Engine.Cylinders.TEST.ClosedCycle.WiebeM	1.64
Engine.Cylinders.TEST.ClosedCycle.IsBurnByUser	0
Engine.Cylinders.TEST.ClosedCycle.IsDebug	0
Engine.Cylinders.TEST.ClosedCycle.IsSynch	0
Engine.Cylinders.TEST.ConnectingRod.Diameter	0
Engine.Cylinders.TEST.ConnectingRod.Length	0.148
Engine.Cylinders.TEST.Piston.CompressionHeight	0.04
Engine.Cylinders.TEST.Piston.Height	0.08
Engine.Cylinders.TEST.Piston.InitialTemperature	300
Engine.Cylinders.TEST.Bore	0.088
Engine.Cylinders.TEST.Stroke	0.082
Engine.Cylinders.TEST.FrictionFactor	350
Engine.Cylinders.TEST.FrictionConstant	100000
Engine.Cylinders.TEST.SquishClearance	0.00125
Engine.Cylinders.TEST.ClearanceVolume	0.00003788
Engine.Cylinders.TEST.HeadSurfaceFactor	1.5
Engine.Cylinders.TEST.InitialGasPresFactor	4
Engine.Cylinders.TEST.InitialGasTemp	927
Engine.Cylinders.TEST.WallTemp	150
Engine.Cylinders.TEST.HeadTemp	300
Engine.Cylinders.TEST.HeadType	4 Stroke 2 Valve
Engine.Model.Model	Unknown
Engine.Model.Year	0
Engine.Model.Manufacturer	Unknown

Table C-2. Cylinder input parameters for software test case

Manifold details

Inlet piping

Engine.Pipes.Pipe(1).Name	Int1
Engine.Pipes.Int1.Sections.Section(1).EntranceDiameter	0.05
Engine.Pipes.Int1.Sections.Section(1).ExitDiameter	0.0381
Engine.Pipes.Int1.Sections.Section(1).RestrictionDiameter	0
Engine.Pipes.Int1.Sections.Section(1).Length	0.04
Engine.Pipes.Int1.Sections.Section(1).ForcesContinuity	-1
Engine.Pipes.Int1.Sections.Section(2).EntranceDiameter	0.0381
Engine.Pipes.Int1.Sections.Section(2).ExitDiameter	0.0381
Engine.Pipes.Int1.Sections.Section(2).RestrictionDiameter	0
Engine.Pipes.Int1.Sections.Section(2).Length	0.175
Engine.Pipes.Int1.Sections.Section(2).ForcesContinuity	-1
Engine.Pipes.Int1.Sections.Section(3).EntranceDiameter	0.0381
Engine.Pipes.Int1.Sections.Section(3).ExitDiameter	0.0381
Engine.Pipes.Int1.Sections.Section(3).RestrictionDiameter	0
Engine.Pipes.Int1.Sections.Section(3).Length	0.1
Engine.Pipes.Int1.Sections.Section(3).ForcesContinuity	-1
Engine.Pipes.Int1.Thickness	0.002
Engine.Pipes.Int1.InitialPurity	1
Engine.Pipes.Int1.WallTemp	30
Engine.Pipes.Int1.GasTemp	25

Exhaust piping

Engine.Pipes.Pipe(2).Name	Exh1
Engine.Pipes.Exh1.Sections.Section(1).EntranceDiameter	0.0069
Engine.Pipes.Exh1.Sections.Section(1).ExitDiameter	0.0069
Engine.Pipes.Exh1.Sections.Section(1).RestrictionDiameter	0
Engine.Pipes.Exh1.Sections.Section(1).Length	0.08
Engine.Pipes.Exh1.Sections.Section(1).ForcesContinuity	-1
Engine.Pipes.Exh1.Sections.Section(2).EntranceDiameter	0.0069
Engine.Pipes.Exh1.Sections.Section(2).ExitDiameter	0.0157
Engine.Pipes.Exh1.Sections.Section(2).RestrictionDiameter	0
Engine.Pipes.Exh1.Sections.Section(2).Length	0.001
Engine.Pipes.Exh1.Sections.Section(2).ForcesContinuity	-1
Engine.Pipes.Exh1.Sections.Section(3).EntranceDiameter	0.0157
Engine.Pipes.Exh1.Sections.Section(3).ExitDiameter	0.0157
Engine.Pipes.Exh1.Sections.Section(3).RestrictionDiameter	0
Engine.Pipes.Exh1.Sections.Section(3).Length	0.135
Engine.Pipes.Exh1.Sections.Section(3).ForcesContinuity	-1
Engine.Pipes.Exh1.Sections.Section(4).EntranceDiameter	0.0157
Engine.Pipes.Exh1.Sections.Section(4).ExitDiameter	0.0069
Engine.Pipes.Exh1.Sections.Section(4).RestrictionDiameter	0
Engine.Pipes.Exh1.Sections.Section(4).Length	0.001
Engine.Pipes.Exh1.Sections.Section(4).ForcesContinuity	-1
Engine.Pipes.Exh1.Sections.Section(5).EntranceDiameter	0.0069
Engine.Pipes.Exh1.Sections.Section(5).ExitDiameter	0.0069
Engine.Pipes.Exh1.Sections.Section(5).RestrictionDiameter	0
Engine.Pipes.Exh1.Sections.Section(5).Length	0.08
Engine.Pipes.Exh1.Sections.Section(5).ForcesContinuity	-1
Engine.Pipes.Exh1.Thickness	0.002
Engine.Pipes.Exh1.InitialPurity	1
Engine.Pipes.Exh1.WallTemp	350
Engine.Pipes.Exh1.GasTemp	600

Table C-3. Manifold input parameters for software test case

Valve details

Intake valve details

Engine.PoppetValveSystems.PoppetValveSystem(1).Name	Inv1
Engine.PoppetValveSystems.Inv1.PoppetValves.PoppetValve(1).OuterSeatDiameter	0.0502
Engine.PoppetValveSystems.Inv1.PoppetValves.PoppetValve(1).InnerSeatDiameter	0.0482
Engine.PoppetValveSystems.Inv1.PoppetValves.PoppetValve(1).SeatAngle	45
Engine.PoppetValveSystems.Inv1.PoppetValves.PoppetValve(1).StemDiameter	0.0079
Engine.PoppetValveSystems.Inv1.PoppetValves.PoppetValve(1).PortDiameter	0.0472
Engine.PoppetValveSystems.Inv1.PoppetValves.PoppetValve(1).ManifoldDiameter	0.0381
Engine.PoppetValveSystems.Inv1.PoppetValves.PoppetValve(1).ValveOpen	305
Engine.PoppetValveSystems.Inv1.PoppetValves.PoppetValve(1).ValveClose	616
Engine.PoppetValveSystems.Inv1.PoppetValves.PoppetValve(1).RampUpPeriod	40
Engine.PoppetValveSystems.Inv1.PoppetValves.PoppetValve(1).RampUpRatio	0.2
Engine.PoppetValveSystems.Inv1.PoppetValves.PoppetValve(1).RampDownPeriod	40
Engine.PoppetValveSystems.Inv1.PoppetValves.PoppetValve(1).RampDownRatio	0.2
Engine.PoppetValveSystems.Inv1.PoppetValves.PoppetValve(1).MaxLift	0.012
Engine.PoppetValveSystems.Inv1.PoppetValves.PoppetValve(1).Count	1
Engine.PoppetValveSystems.Inv1.PoppetValves.PoppetValve(1).CDMapName	Masked

Exhaust valve details

Engine.PoppetValveSystems.PoppetValveSystem(2).Name	Exv1
Engine.PoppetValveSystems.Exv1.PoppetValves.PoppetValve(1).OuterSeatDiameter	0.0413
Engine.PoppetValveSystems.Exv1.PoppetValves.PoppetValve(1).InnerSeatDiameter	0.0393
Engine.PoppetValveSystems.Exv1.PoppetValves.PoppetValve(1).SeatAngle	45
Engine.PoppetValveSystems.Exv1.PoppetValves.PoppetValve(1).StemDiameter	0.0111
Engine.PoppetValveSystems.Exv1.PoppetValves.PoppetValve(1).PortDiameter	0.0385
Engine.PoppetValveSystems.Exv1.PoppetValves.PoppetValve(1).ManifoldDiameter	0.0413
Engine.PoppetValveSystems.Exv1.PoppetValves.PoppetValve(1).ValveOpen	195
Engine.PoppetValveSystems.Exv1.PoppetValves.PoppetValve(1).ValveClose	355
Engine.PoppetValveSystems.Exv1.PoppetValves.PoppetValve(1).RampUpPeriod	40
Engine.PoppetValveSystems.Exv1.PoppetValves.PoppetValve(1).RampUpRatio	0.2
Engine.PoppetValveSystems.Exv1.PoppetValves.PoppetValve(1).RampDownPeriod	40
Engine.PoppetValveSystems.Exv1.PoppetValves.PoppetValve(1).RampDownRatio	0.2
Engine.PoppetValveSystems.Exv1.PoppetValves.PoppetValve(1).MaxLift	0.01
Engine.PoppetValveSystems.Exv1.PoppetValves.PoppetValve(1).Count	1
Engine.PoppetValveSystems.Exv1.PoppetValves.PoppetValve(1).InterValveClearance	0.005
Engine.PoppetValveSystems.Exv1.PoppetValves.PoppetValve(1).CDMapName	Masked

Table C-4. Valve input parameters for software test case

Units

Engine.UnitProfile.LengthUnitName	mm
Engine.UnitProfile.AreaUnitName	sq mm
Engine.UnitProfile.VolumeUnitName	cc
Engine.UnitProfile.PressureUnitName	atm
Engine.UnitProfile.TemperatureUnitName	deg C
Engine.UnitProfile.DensityUnitName	kg/cu m
Engine.UnitProfile.MassUnitName	kg
Engine.UnitProfile.ForceUnitName	N
Engine.UnitProfile.PowerUnitName	kW
Engine.UnitProfile.EnergyUnitName	kWh
Engine.UnitProfile.ConsumptionUnitName	g/kWh
Engine.UnitProfile.VelocityUnitName	m/s
Engine.UnitProfile.AccelerationUnitName	m/s/s
Engine.UnitProfile.MassFlowUnitName	g/s
Engine.UnitProfile.AngularVelocityUnitName	RPM

Table C-5. Units used for software test case

Detailed results for Virtual 4 Stroke® software test cases

Pressure and velocity plots for frequencies 60 Hz and 225 Hz are shown below

Pulse frequency 60 Hz (1550 RPM)

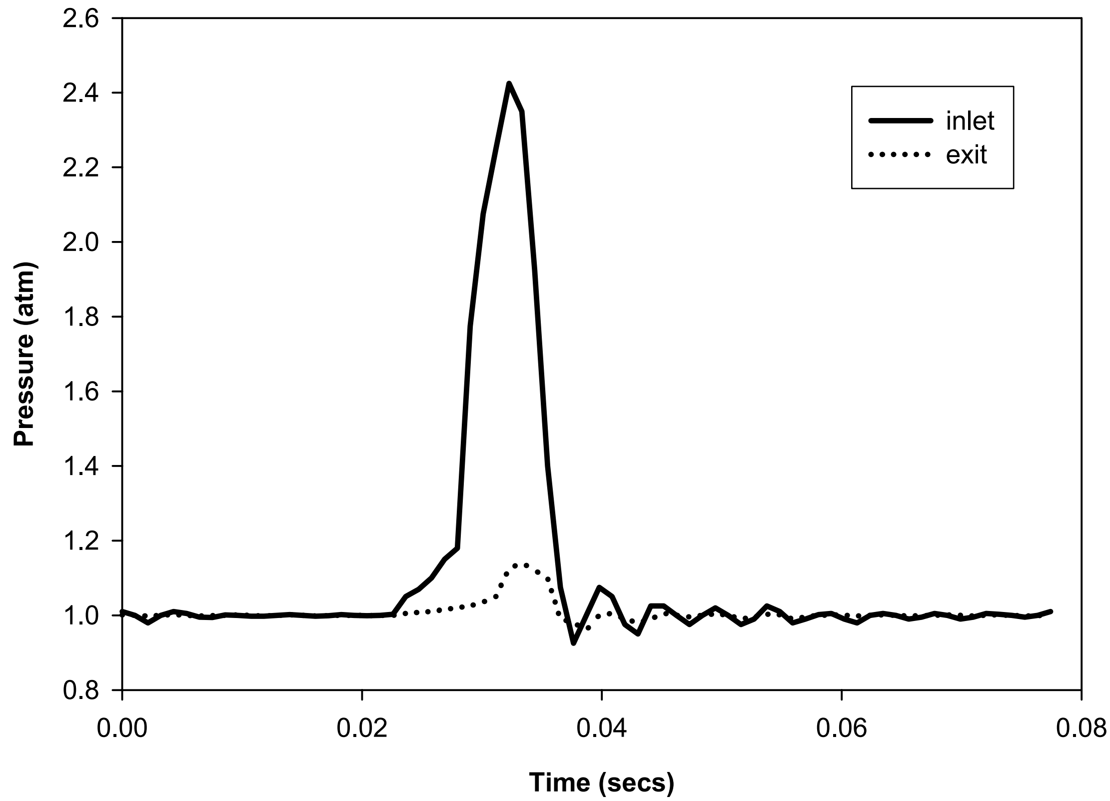


Figure C.1 Pressure plot for software test case – 60 Hz

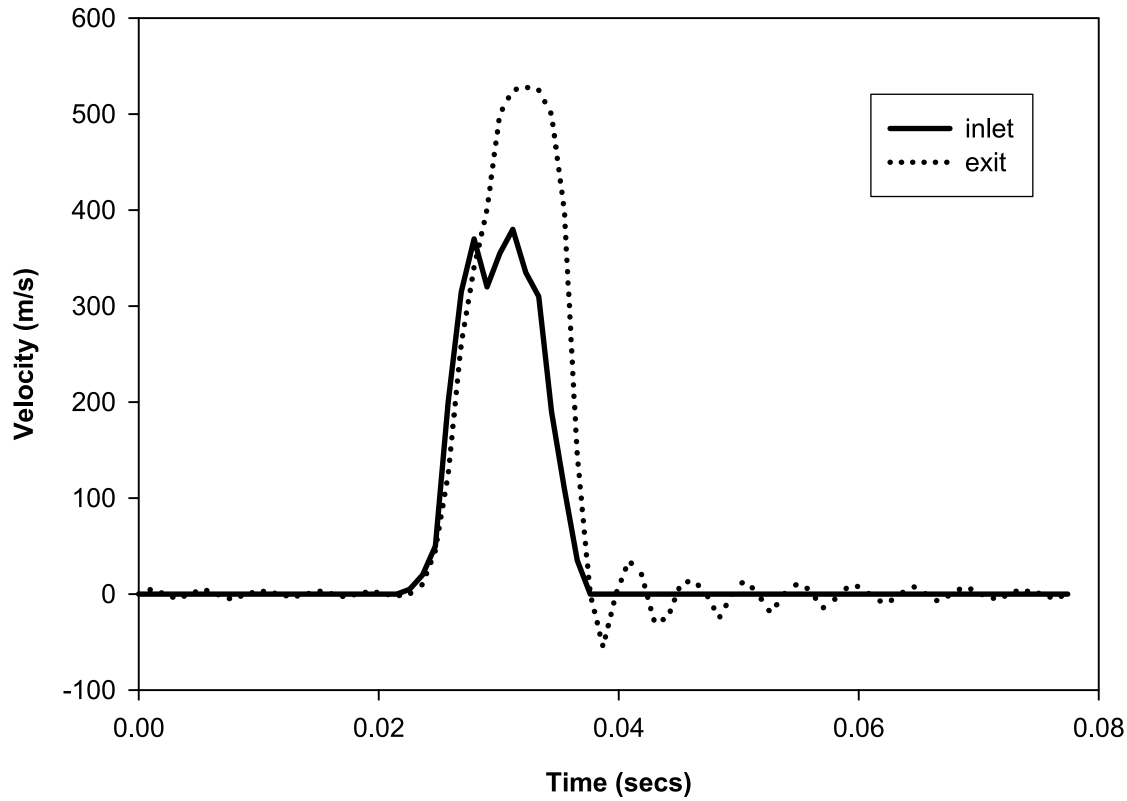


Figure C.2 Velocity plot for software test case – 60 Hz

Pulse frequency 225 Hz (6000 RPM)

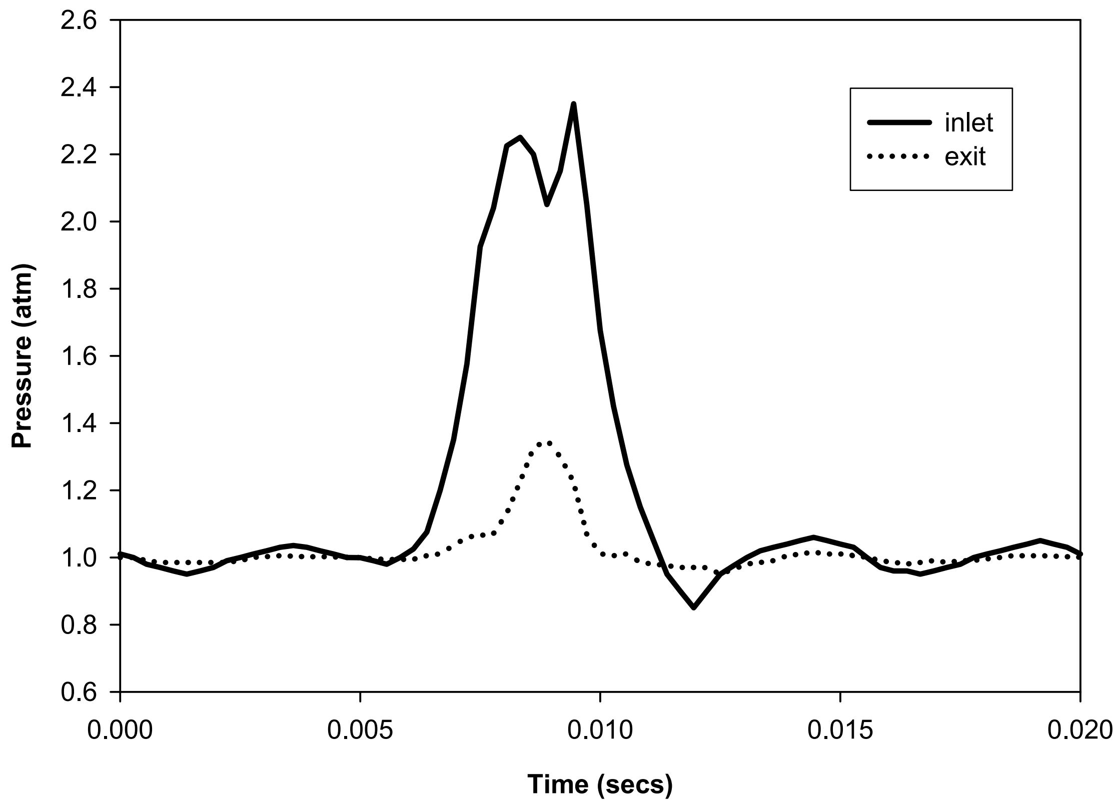


Figure C.3 Pressure plot for software test case – 225 Hz

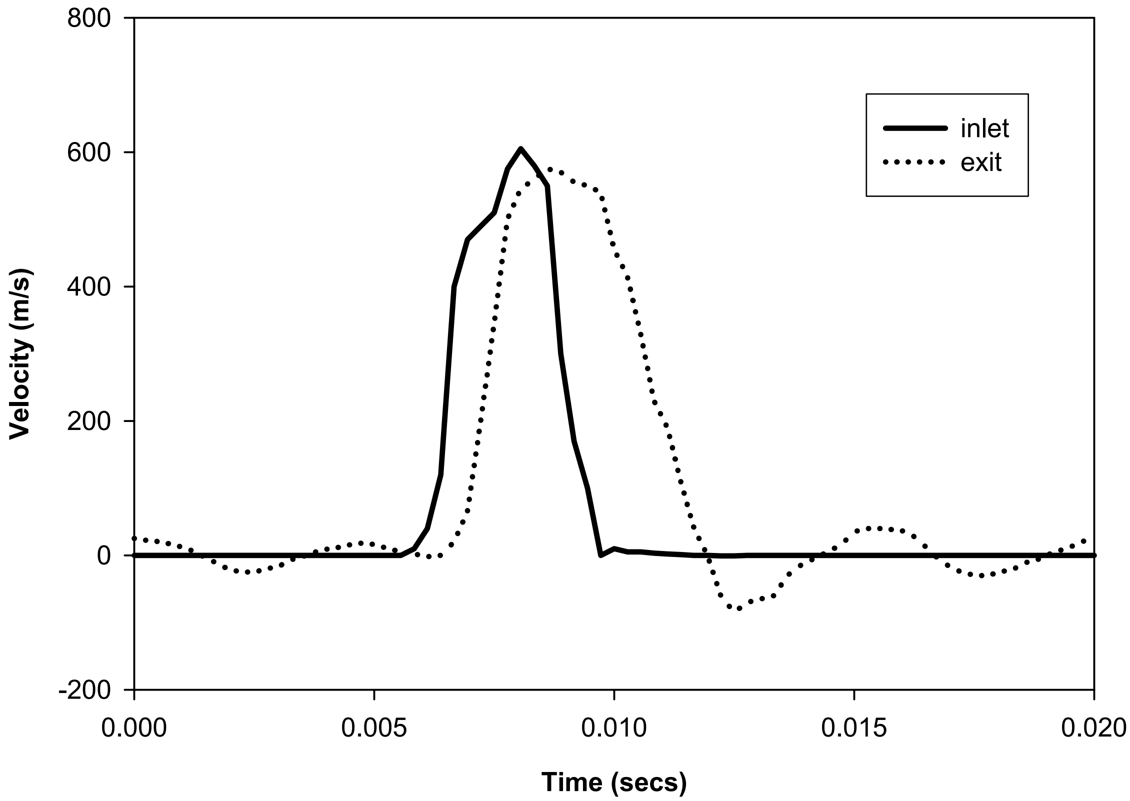


Figure C.4 Velocity plot for software test case – 225 Hz

Detailed results for flow simulation test case

Plots of parameters not shown in sections 5.3.1 and 5.3.2 are given below.

Test case 1 – section 5.3.1

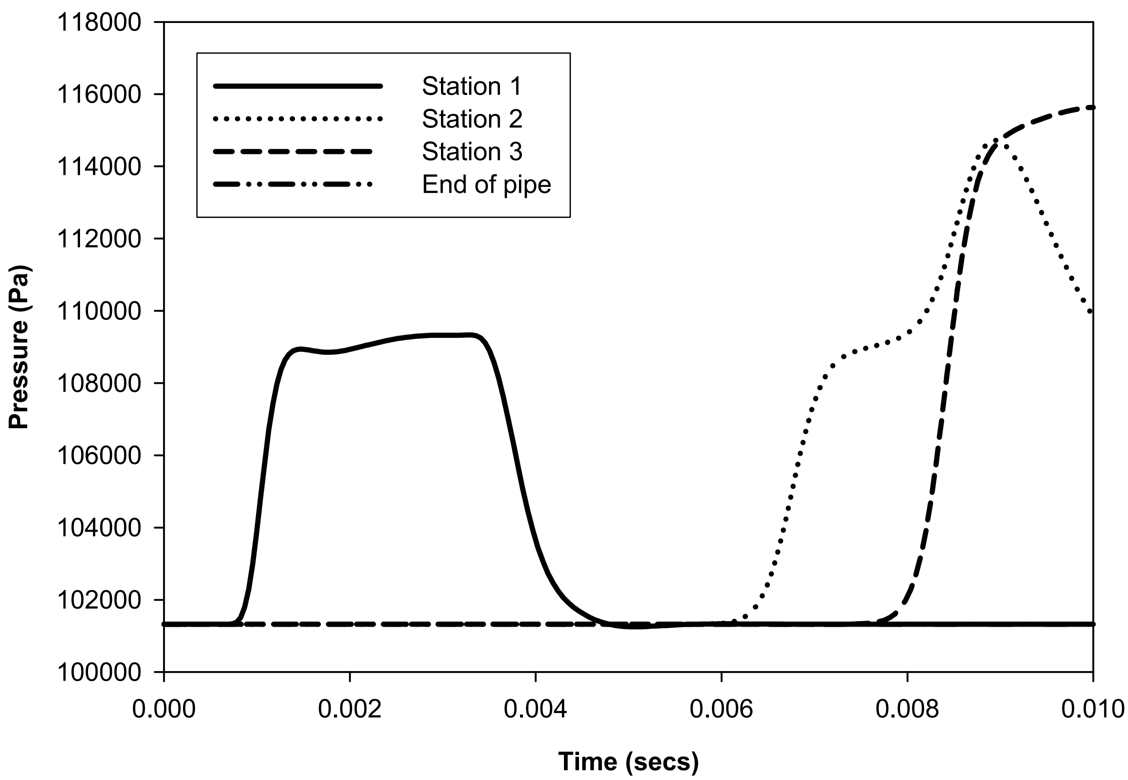


Figure C.5 Pressure at transducer locations

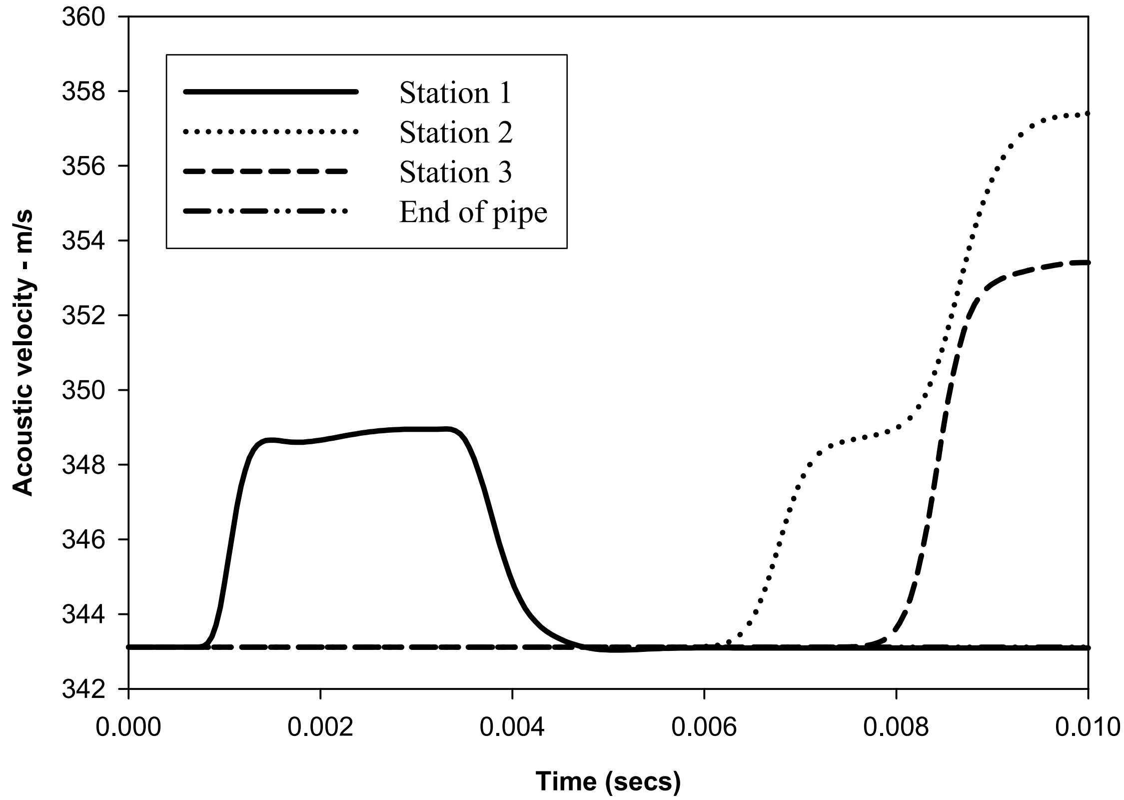


Figure C.6 Acoustic velocity at transducer locations

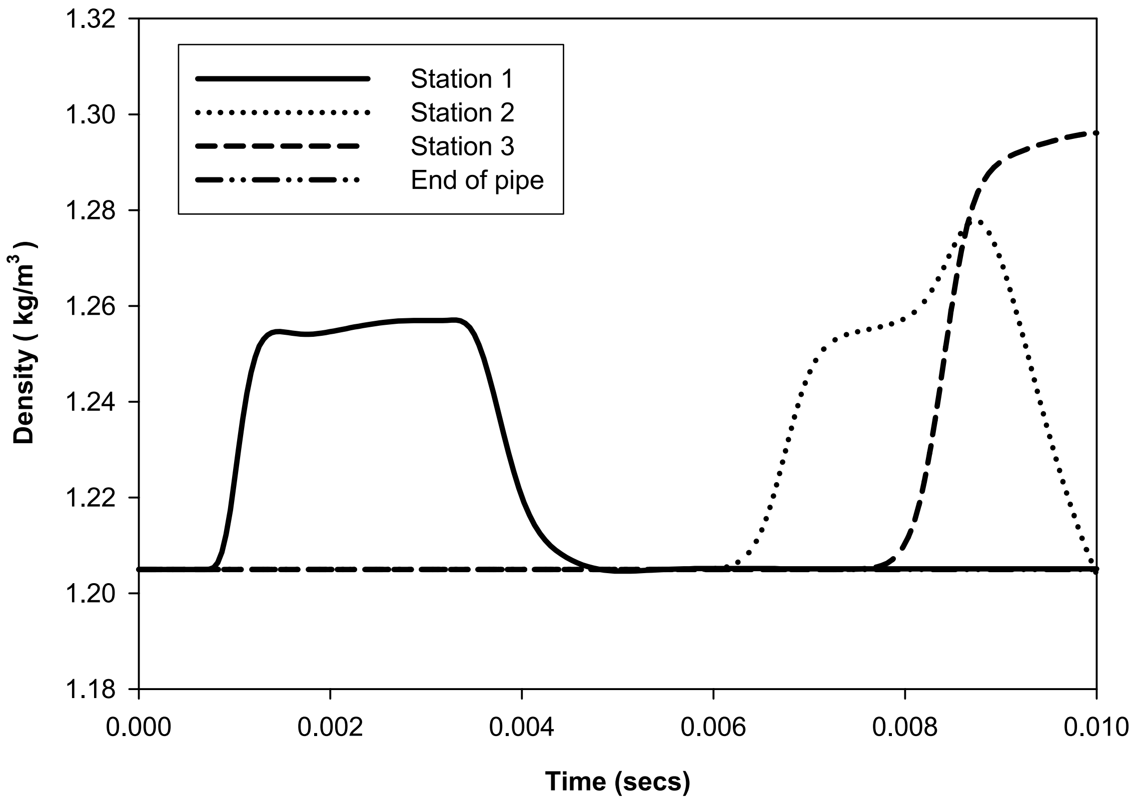


Figure C.7 Fluid density at transducer locations

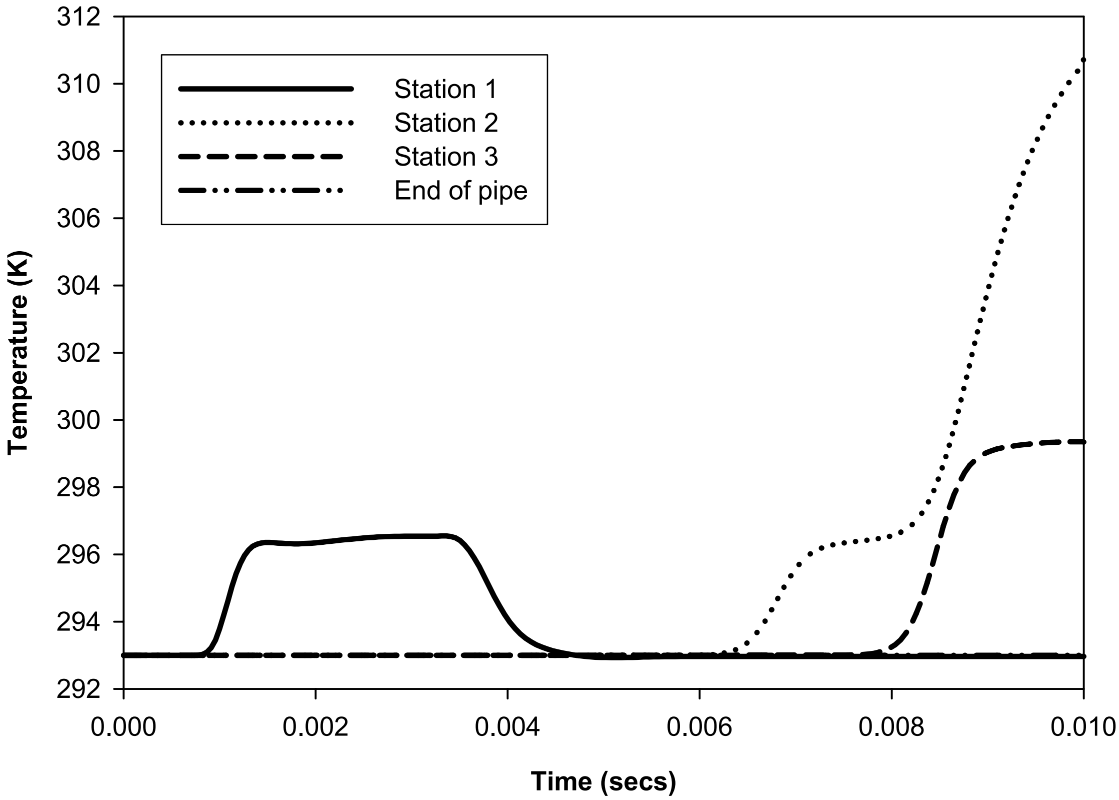


Figure C.8 Reference temperature at transducer locations

Test case 2 – section 5.3.2

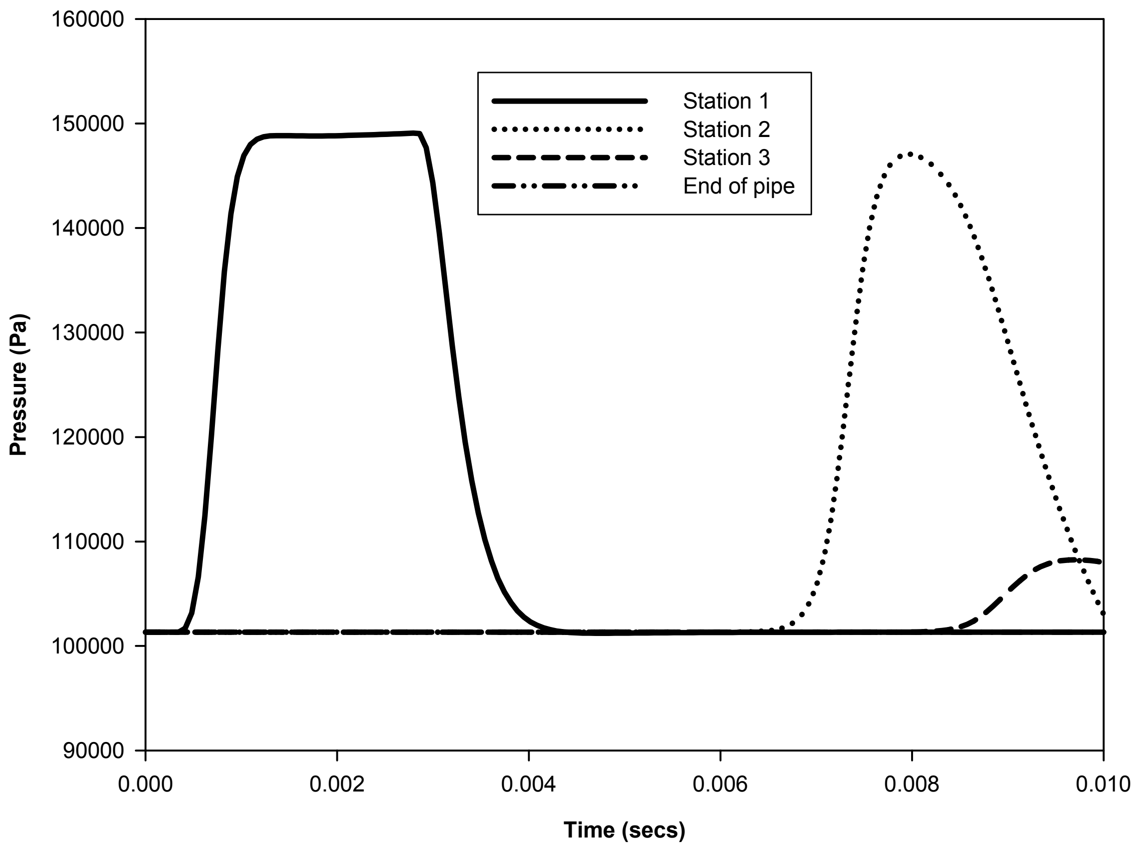


Figure C.9 Pressure at transducer locations

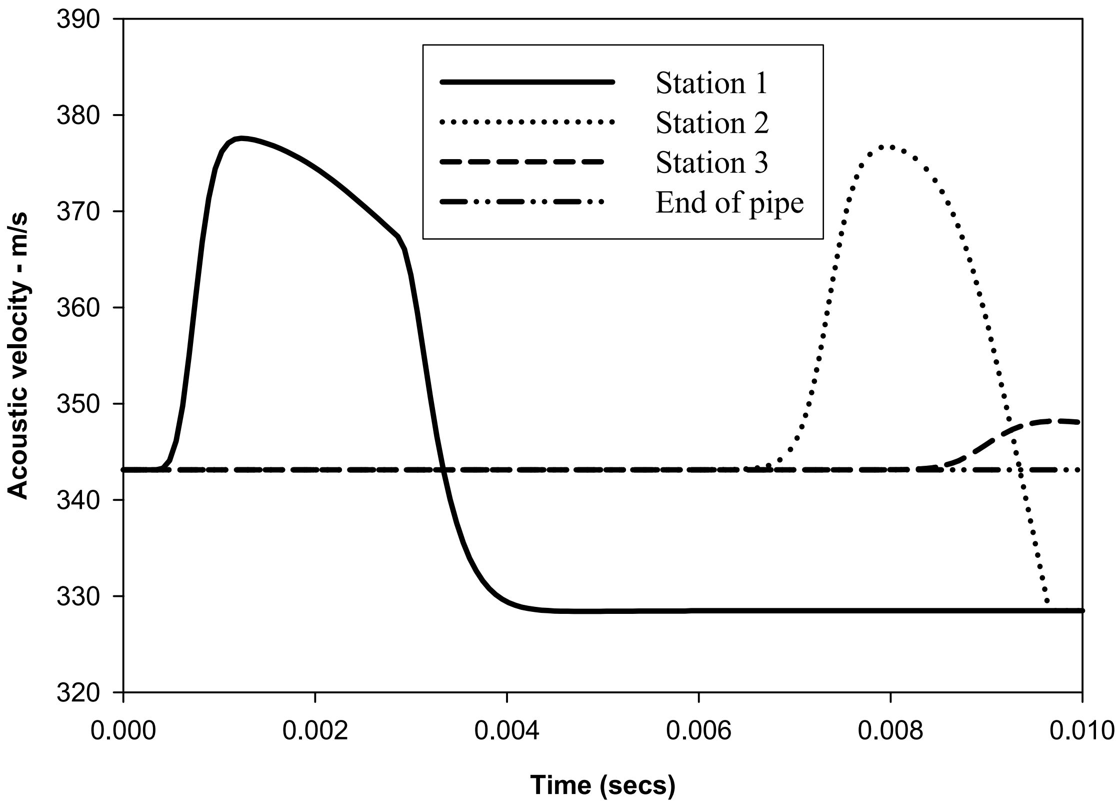


Figure C.10 Acoustic velocity at transducer locations

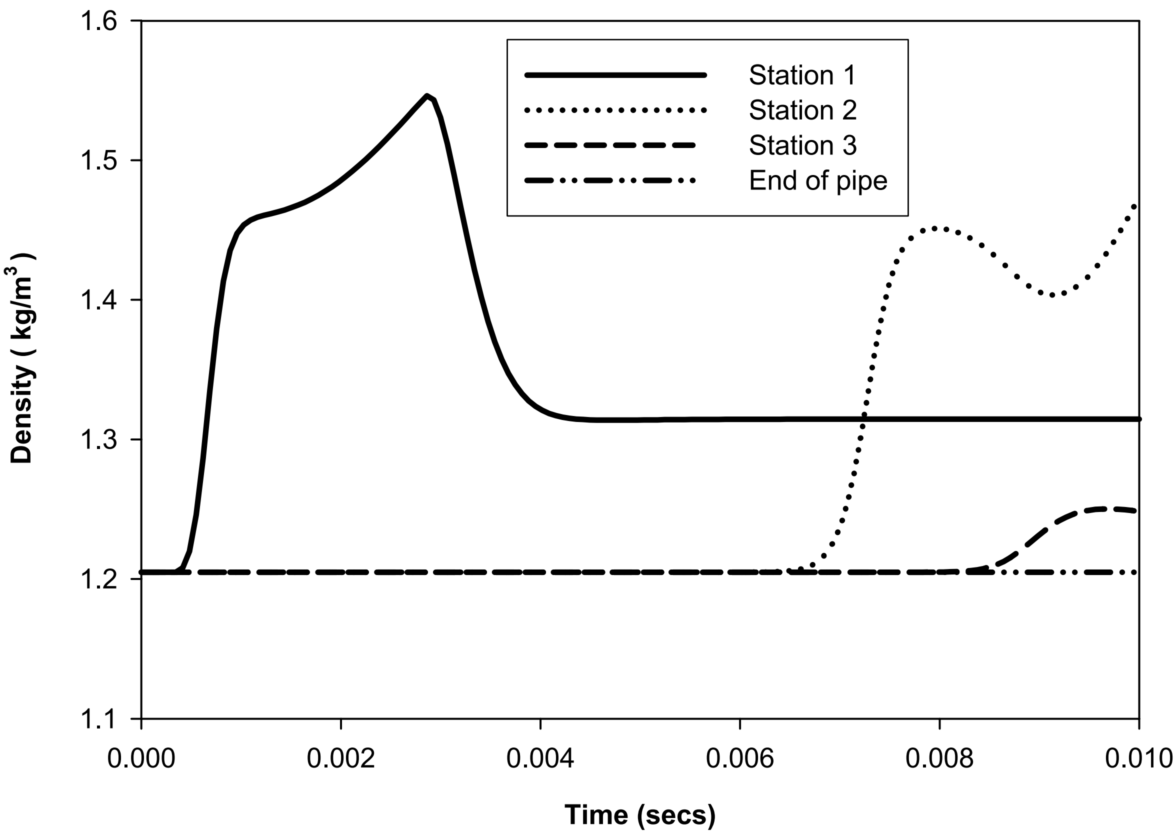


Figure C.11 Fluid density at transducer locations

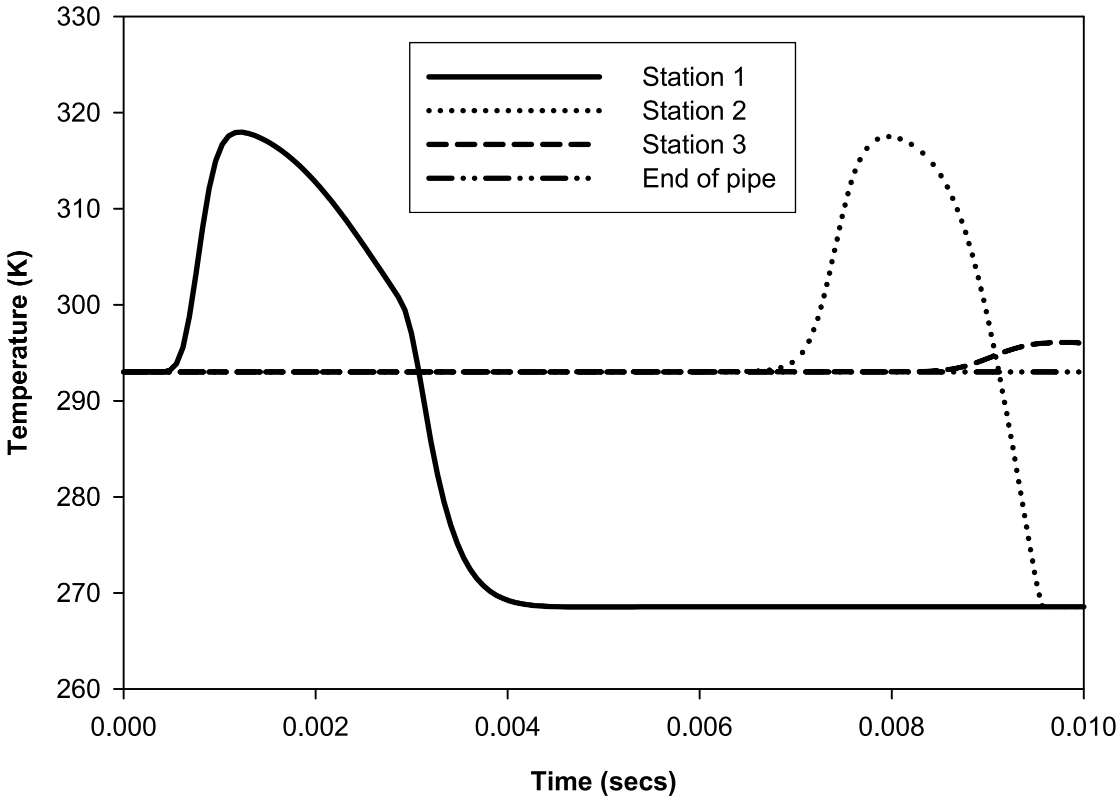


Figure C.12 Reference temperature at transducer locations

Effect of mesh size

Detailed results for the test case for assessing the effect of mesh size – section 5.4.1

Station 1

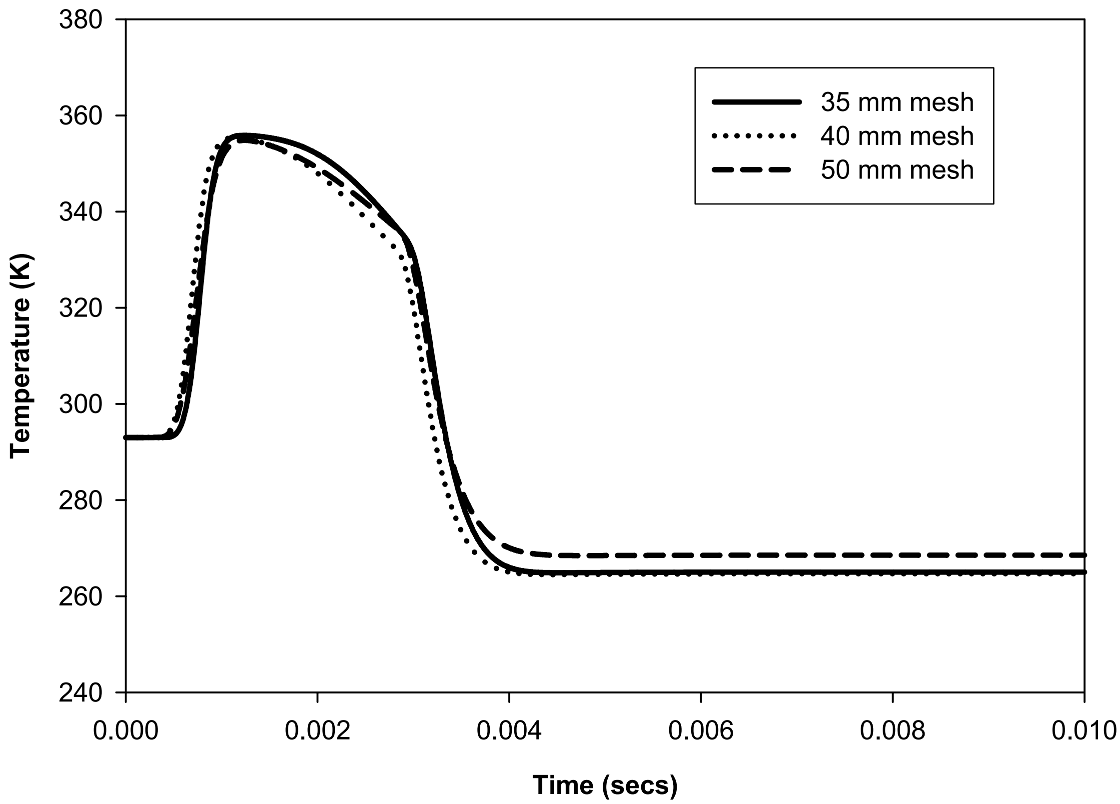


Figure C.13 Temperature plots for three mesh sizes, station 1 - comparison

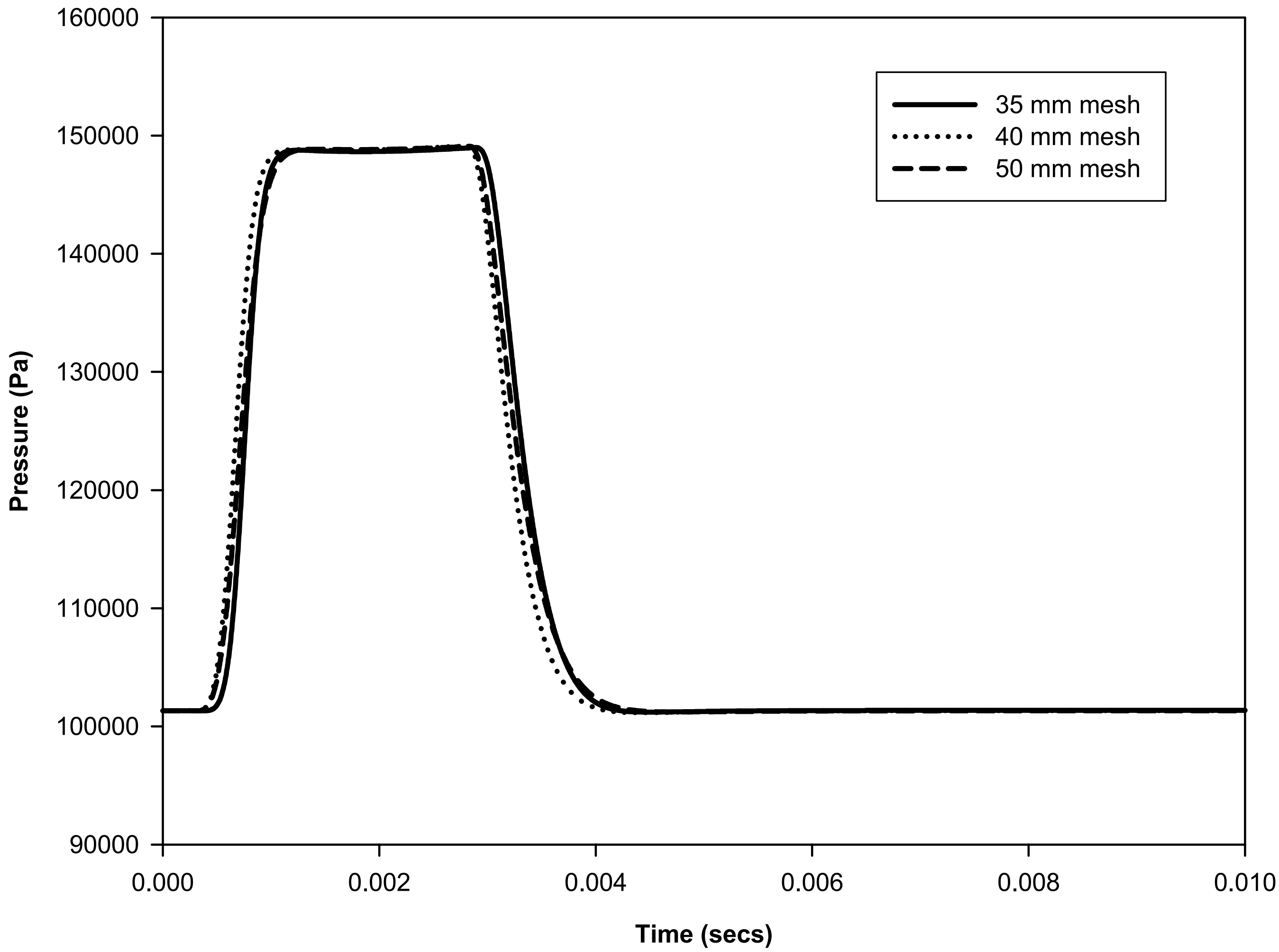


Figure C.14 Pressure plots for three mesh sizes, station 1 - comparison

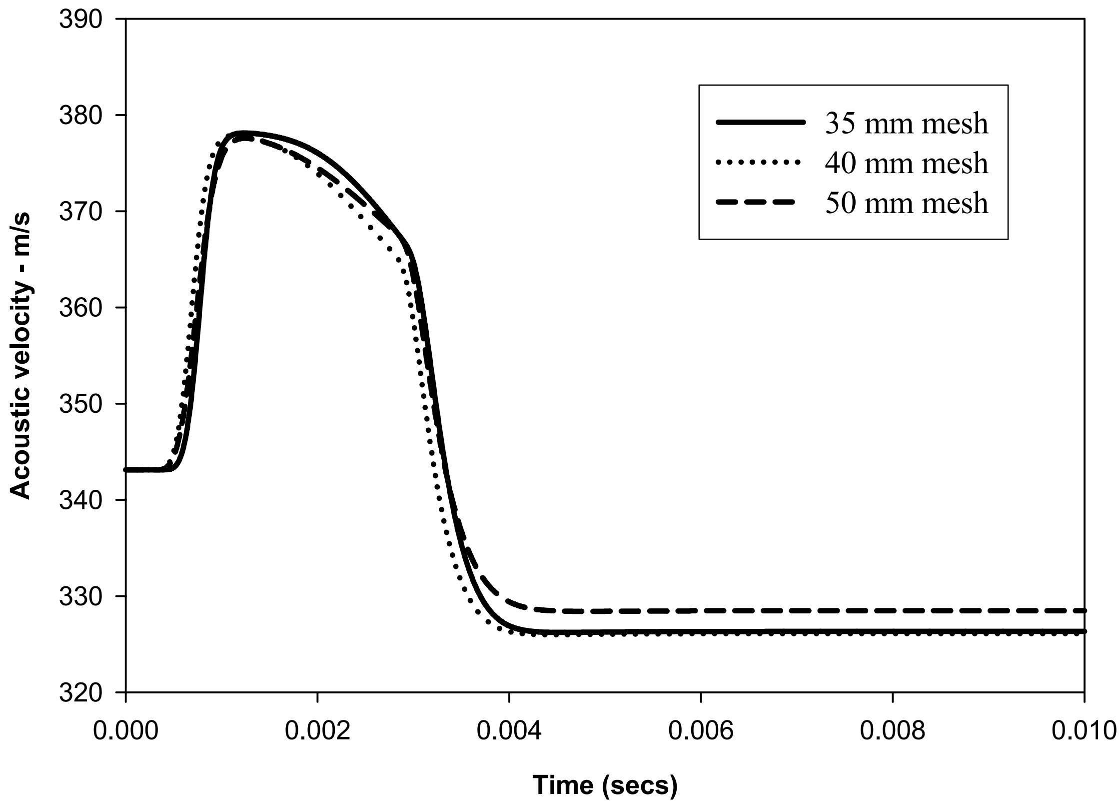


Figure C.15 Acoustic velocity plots for three mesh sizes, station 1 - comparison

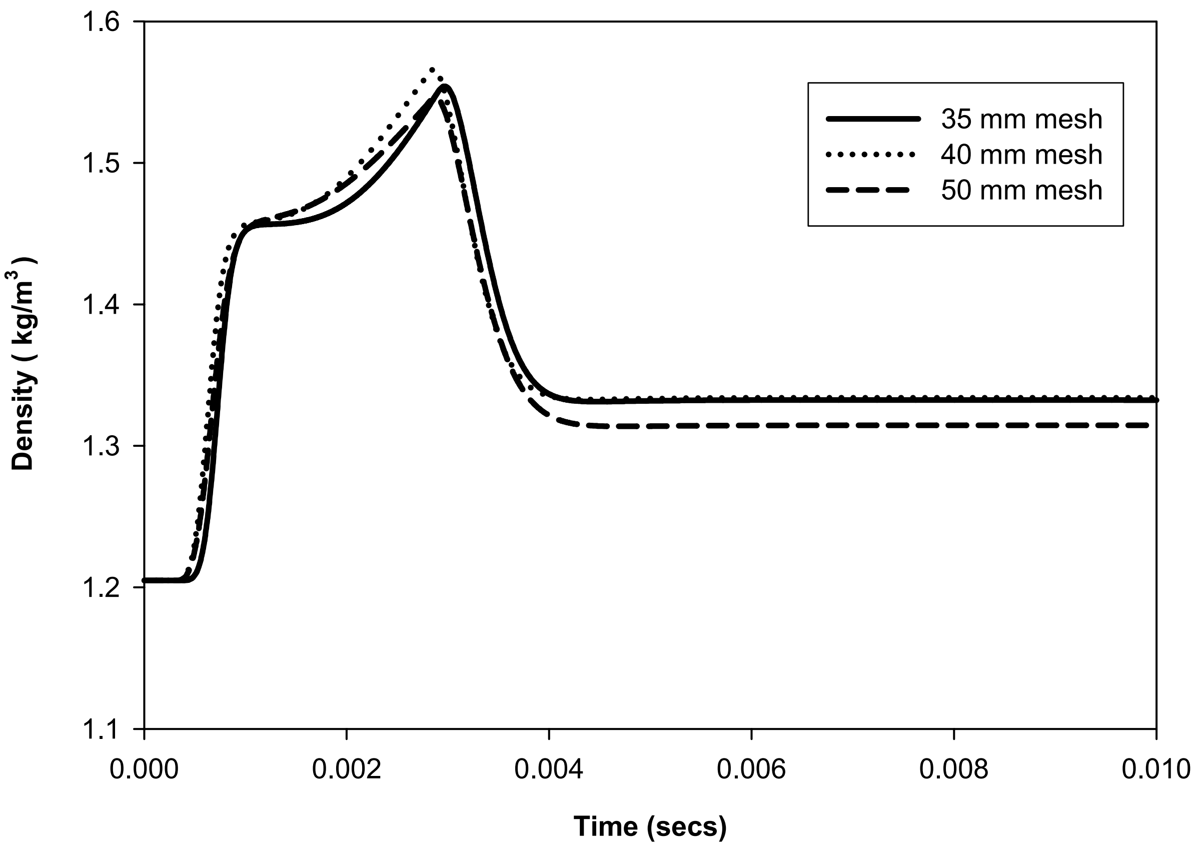


Figure C.16 Fluid density plots for three mesh sizes, station 1 - comparison

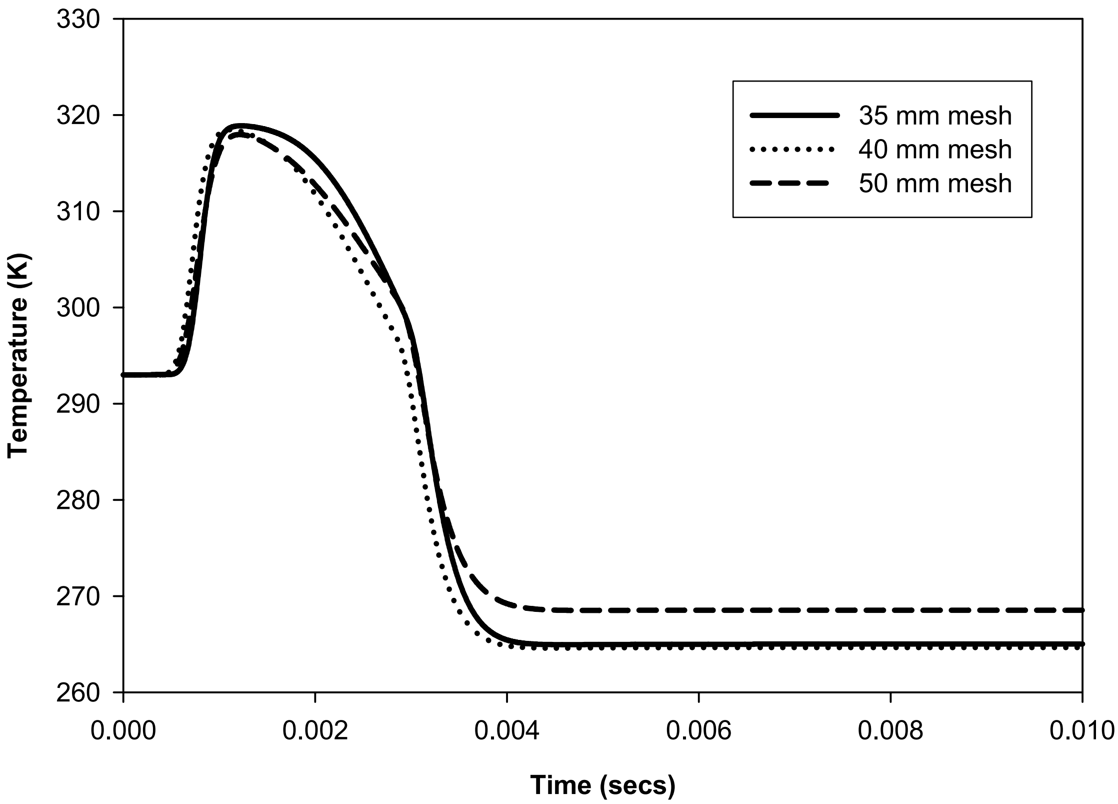


Figure C.17 Reference temperature plots for three mesh sizes, station 1 - comparison

Station 2

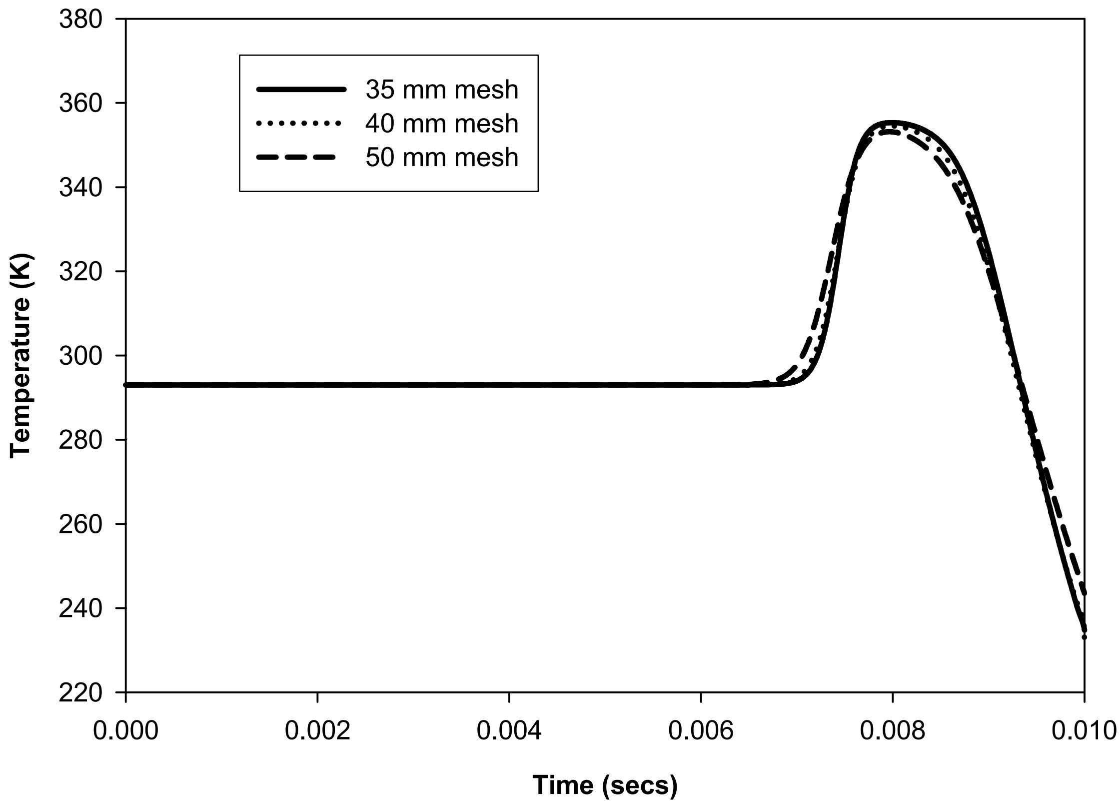


Figure C.18 Temperature plots for three mesh sizes, station 2 - comparison

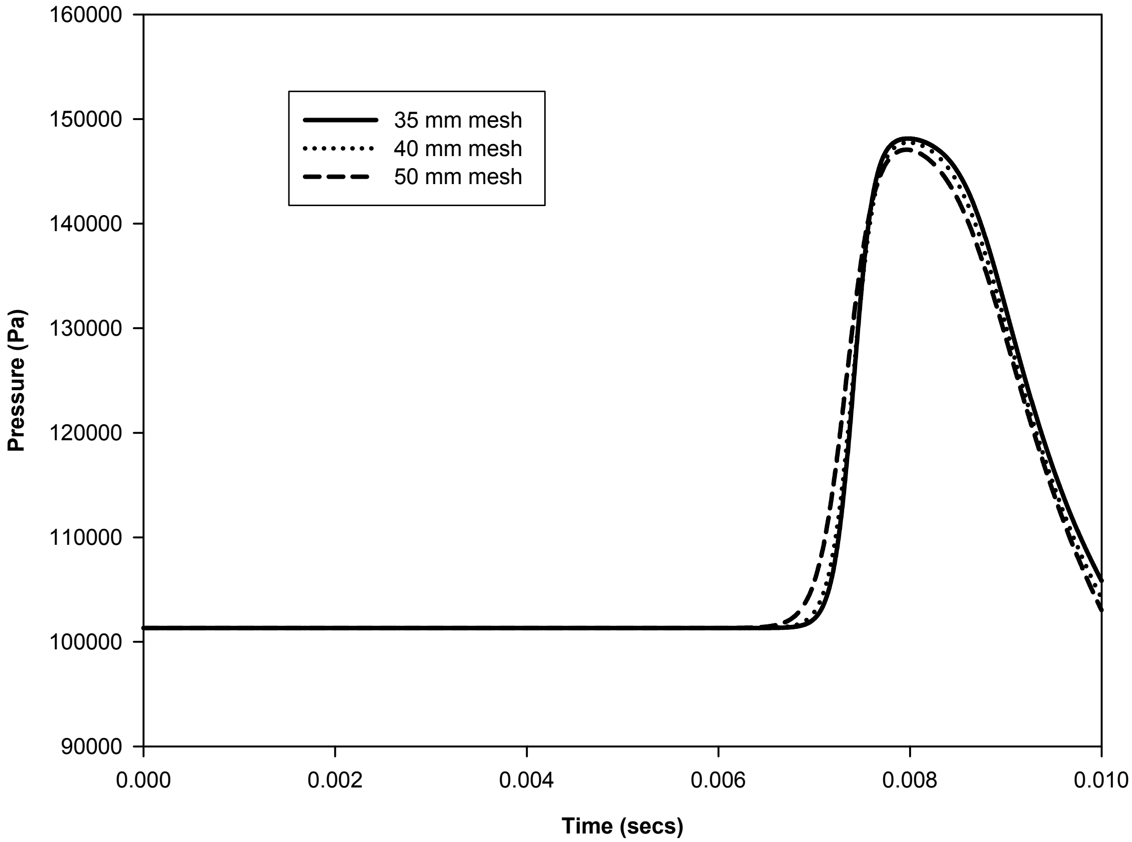


Figure C.19 Pressure plots for three mesh sizes, station 2 - comparison

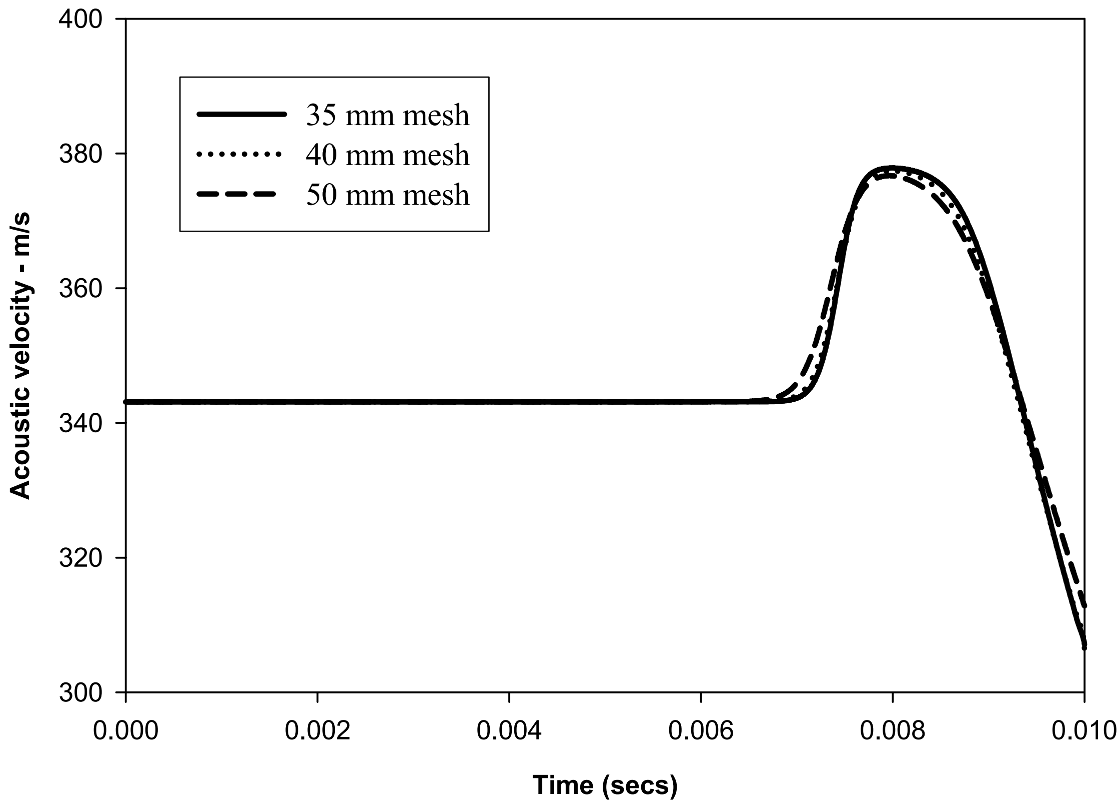


Figure C.20 Acoustic velocity plots for three mesh sizes, station 2 - comparison

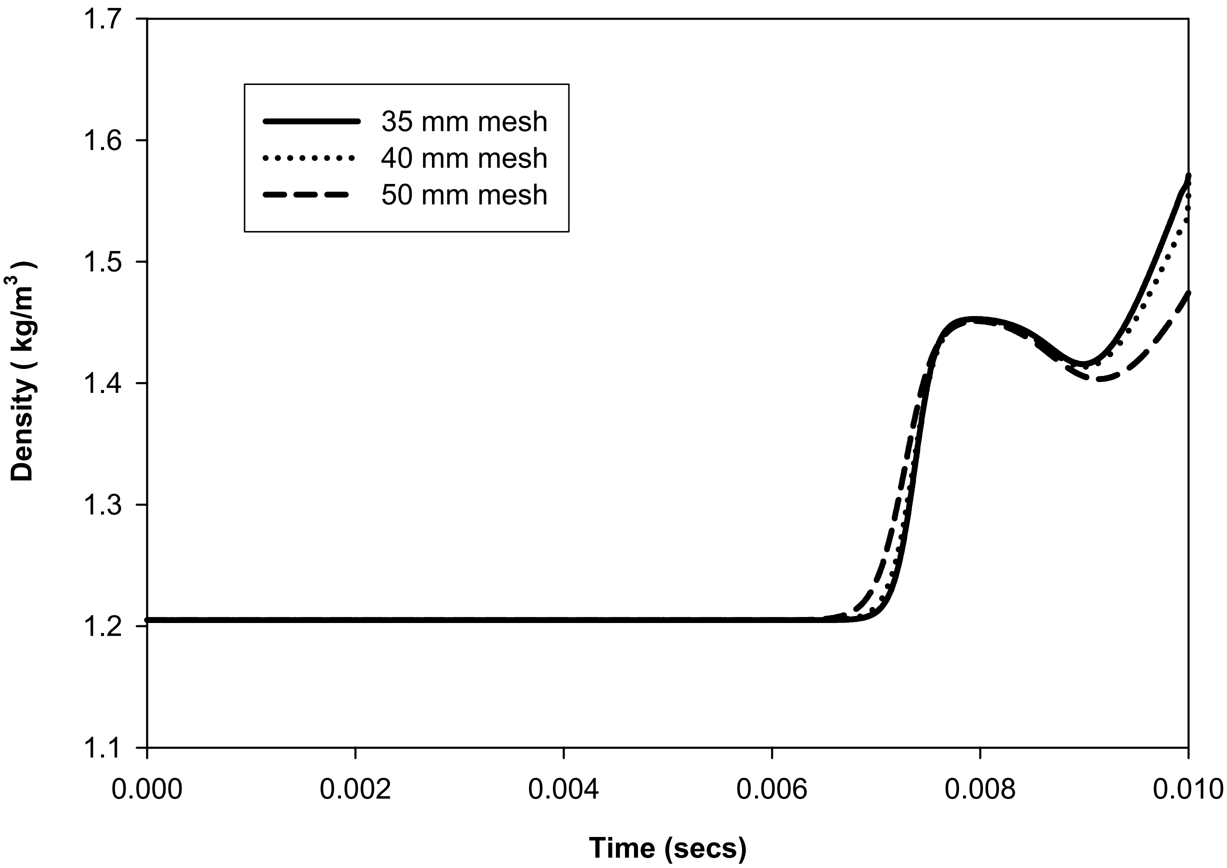


Figure C.21 Fluid density plots for three mesh sizes, station 2 - comparison

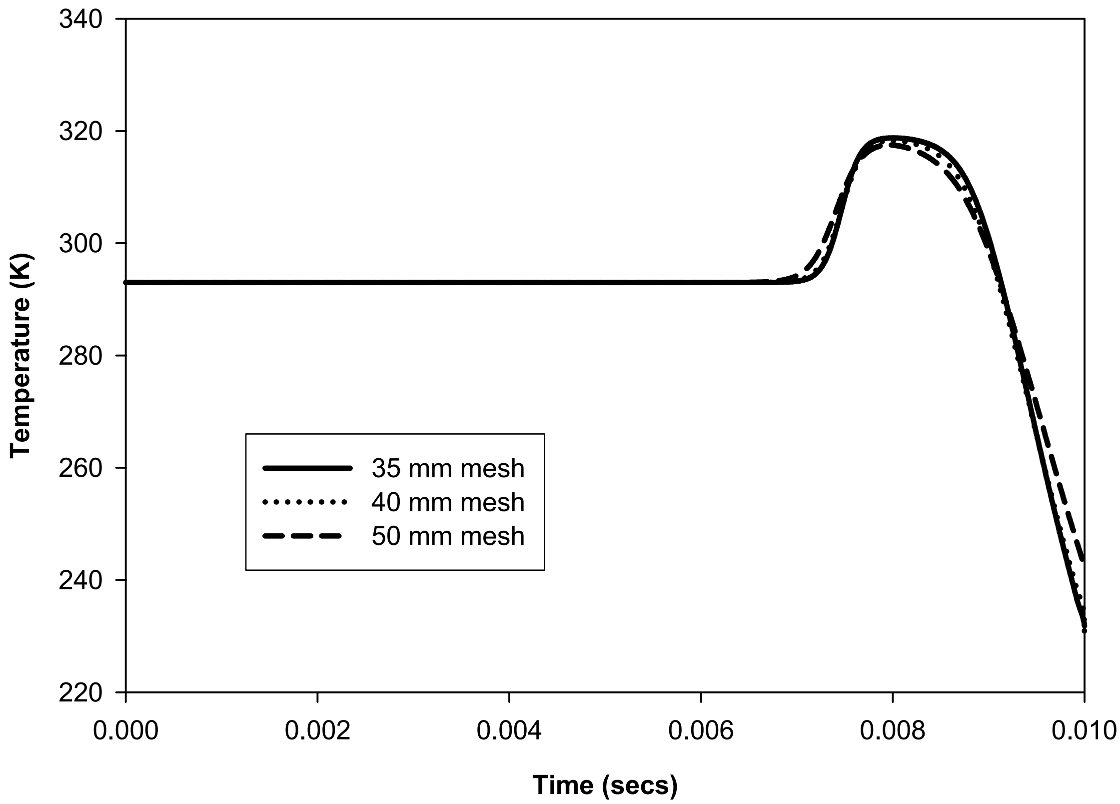


Figure C.22 Reference temperature plots for three mesh sizes, station 2 - comparison

Station 3

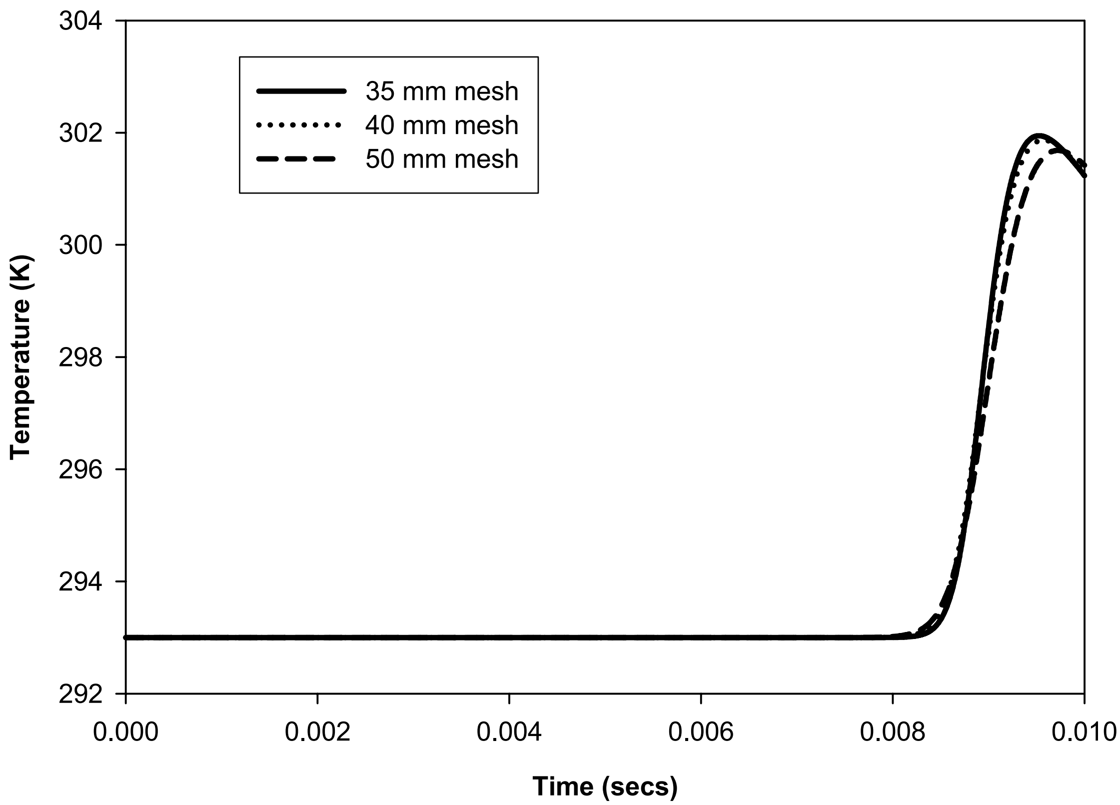


Figure C.23 Temperature plots for three mesh sizes, station 3 - comparison

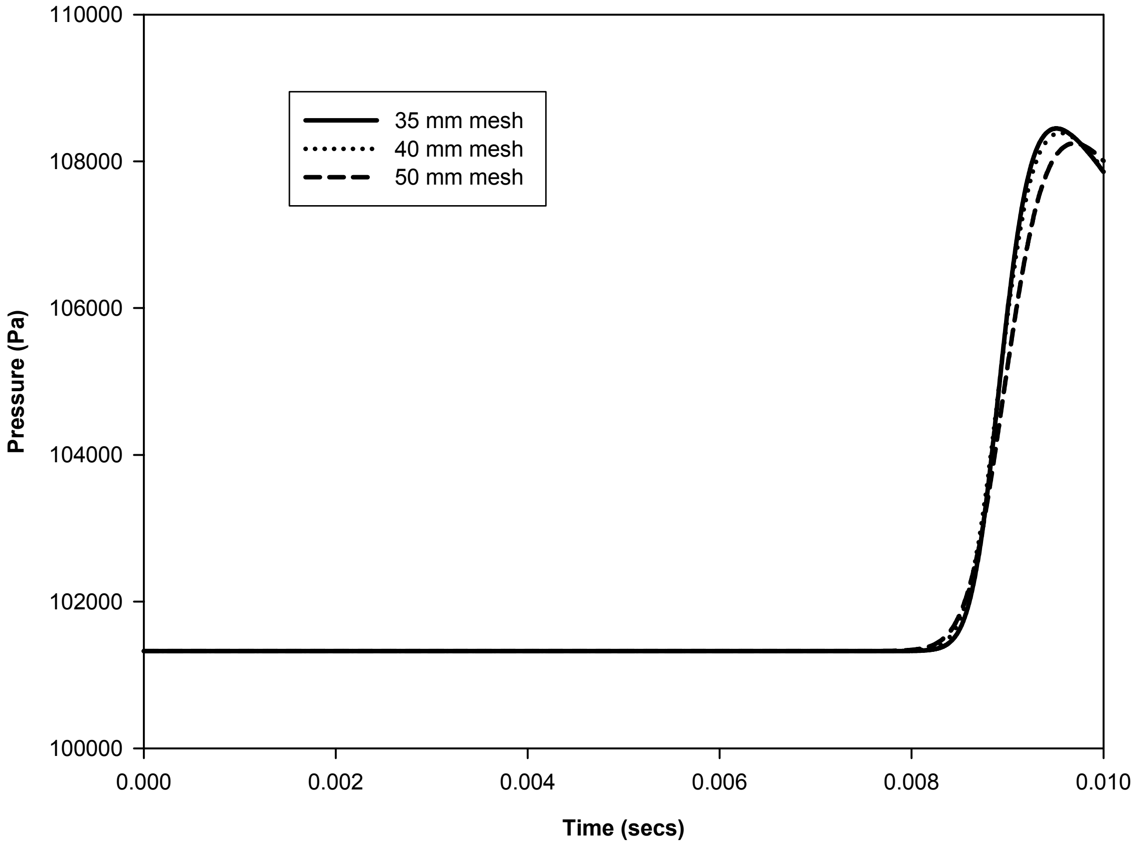


Figure C.24 Pressure plots for three mesh sizes, station 3 - comparison

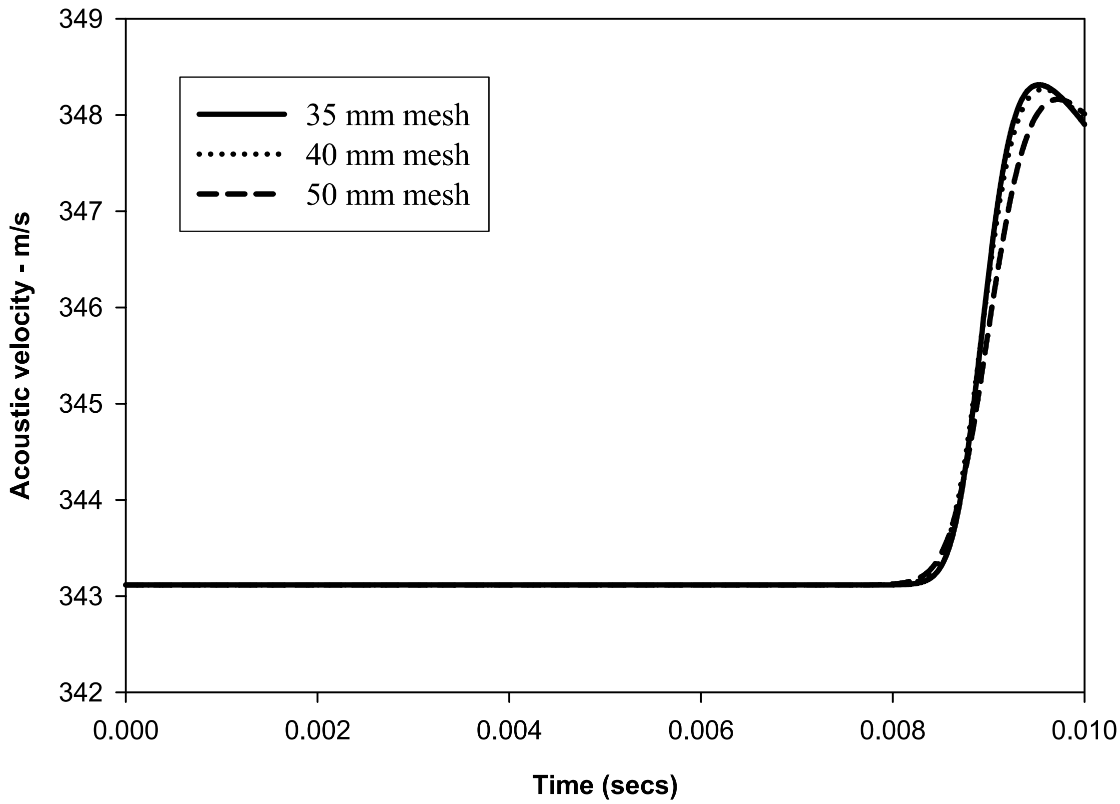


Figure C.25 Acoustic velocity plots for three mesh sizes, station 3 - comparison

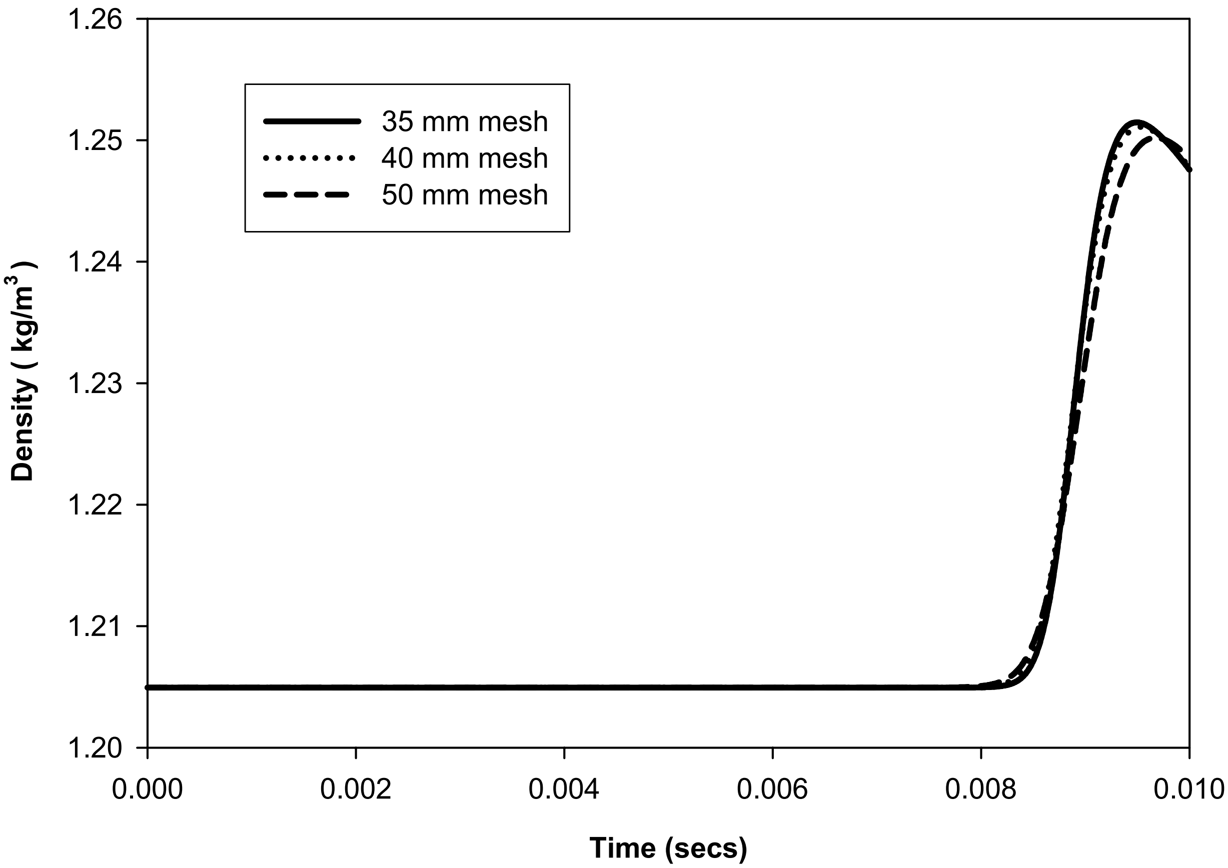


Figure C.26 Fluid density plots for three mesh sizes, station 3 - comparison

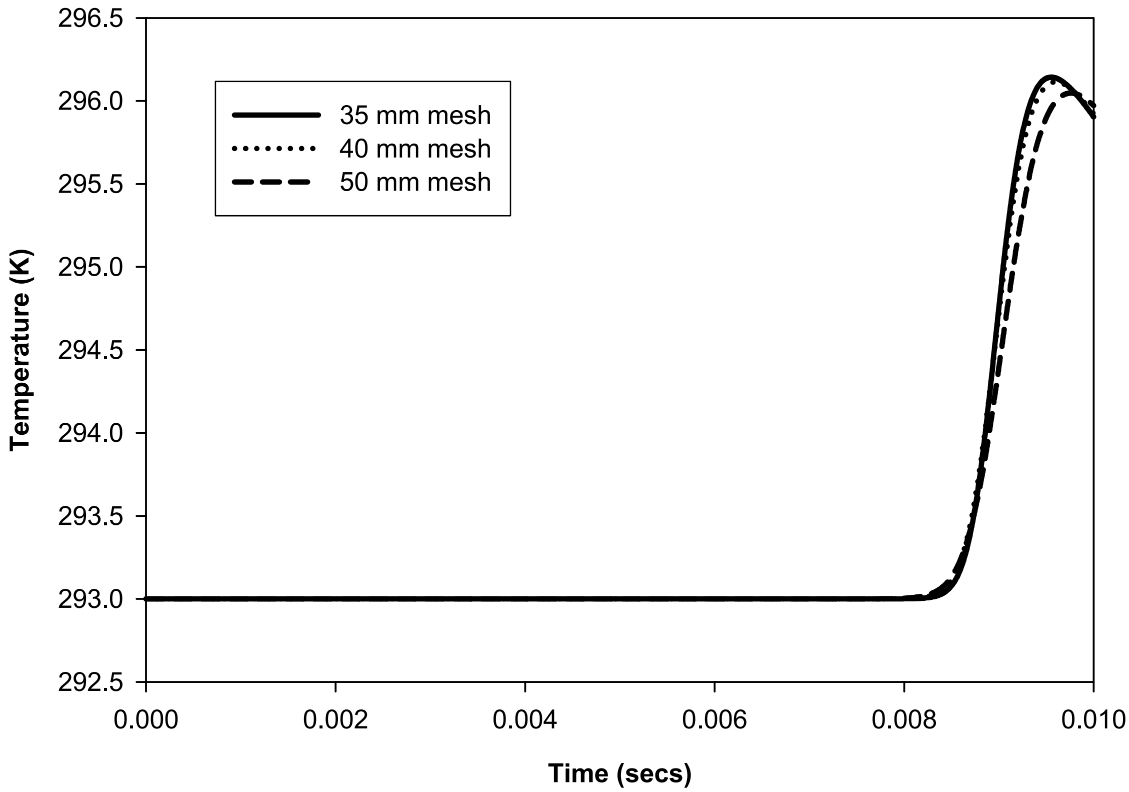


Figure C.27 Reference temperature plots for three mesh sizes, station 3 – comparison

Effect of time step

Detailed results for the test case for assessing the effect of time step – section 5.4.3

Station 1

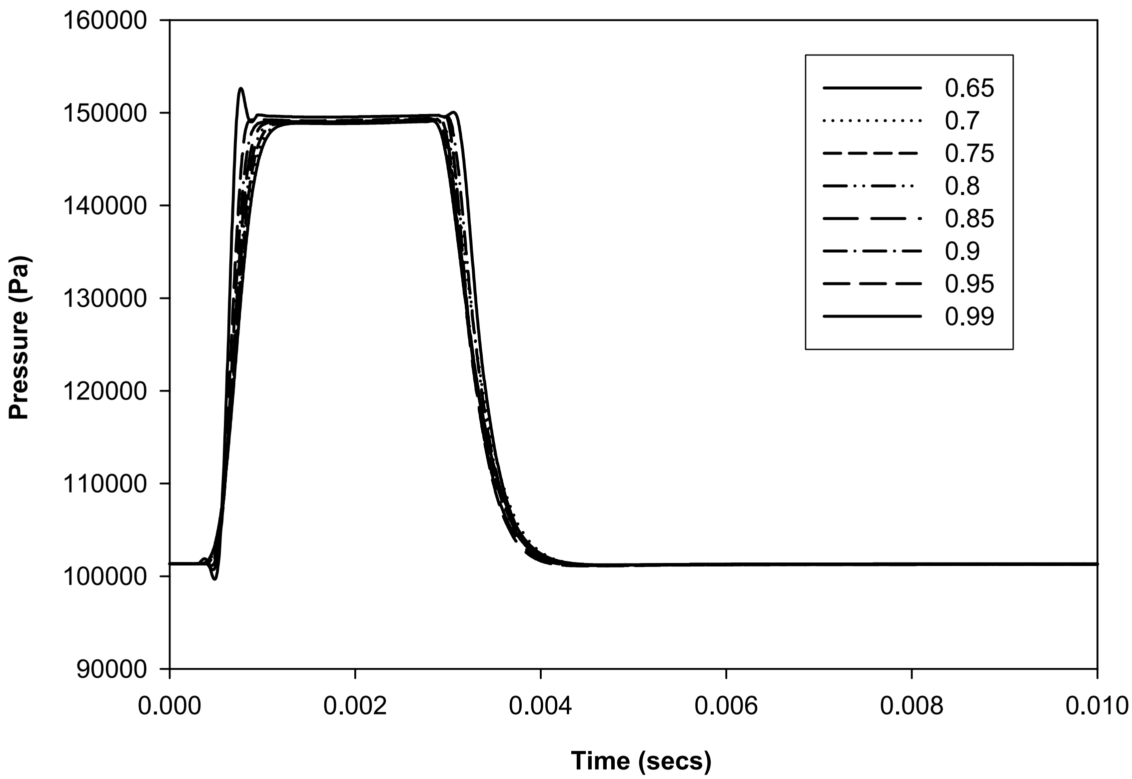


Figure C.28 Pressure plots for station 1 with varying under-relaxation factors

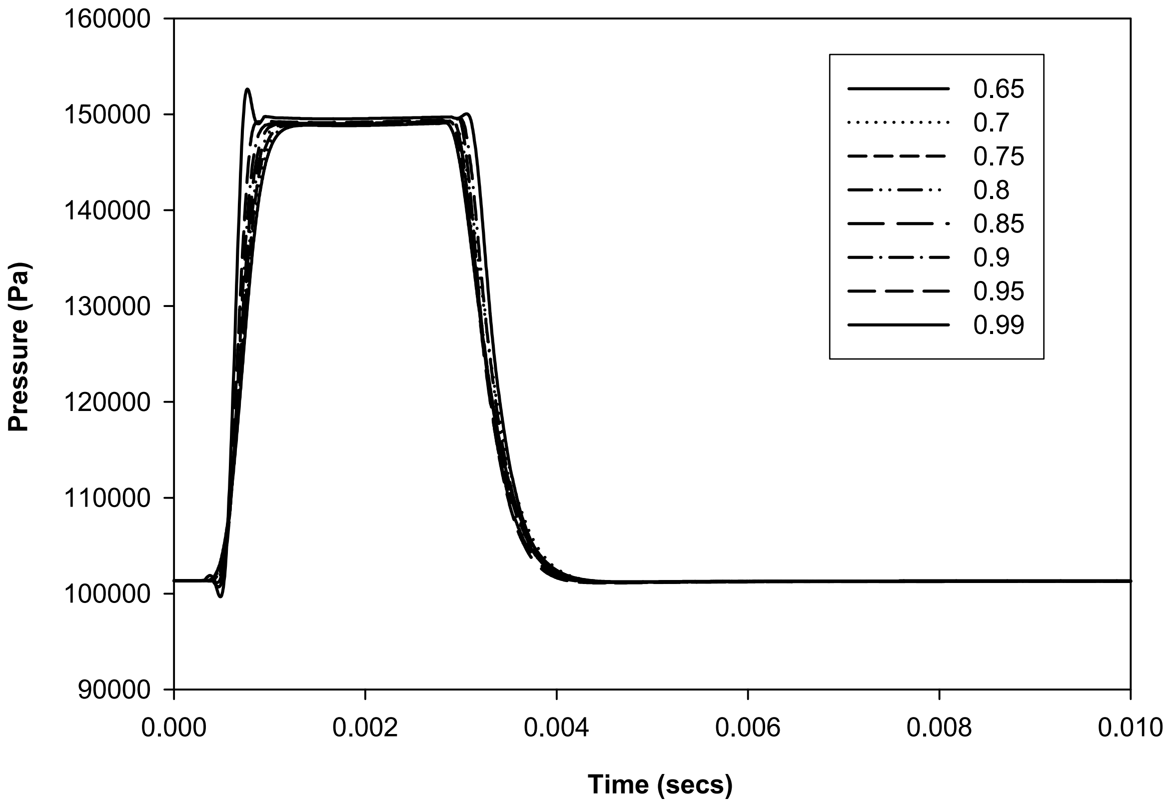


Figure C.29 Acoustic velocity plots for station 1 with varying under-relaxation factors

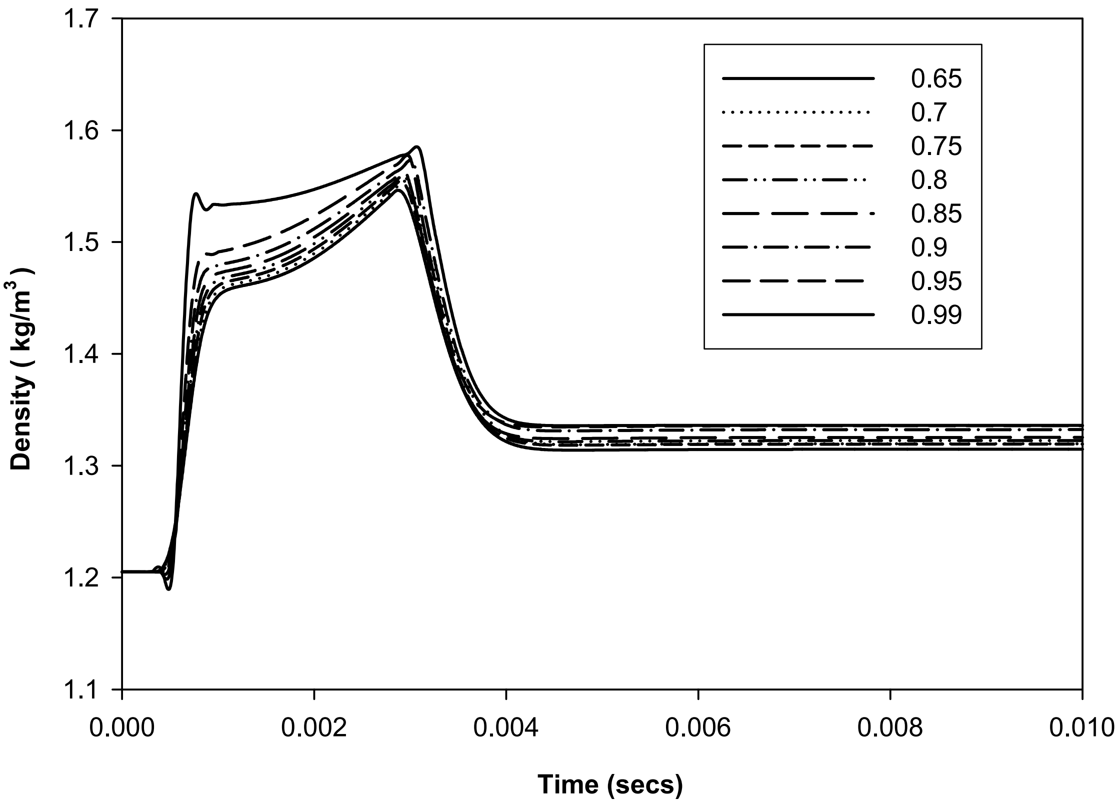


Figure C.30 Fluid density plot for station 1 with varying under-relaxation factors

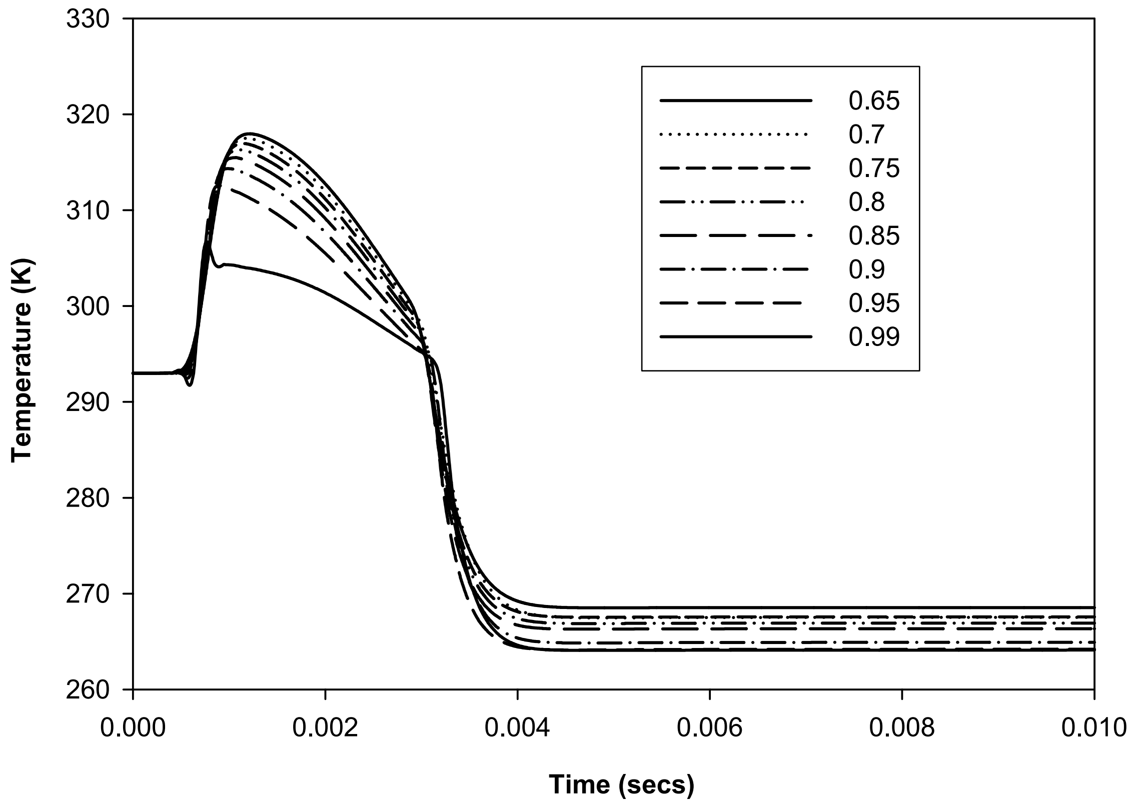


Figure C.31 Reference temperature plot for station 1 with varying under-relaxation factors

Station 2

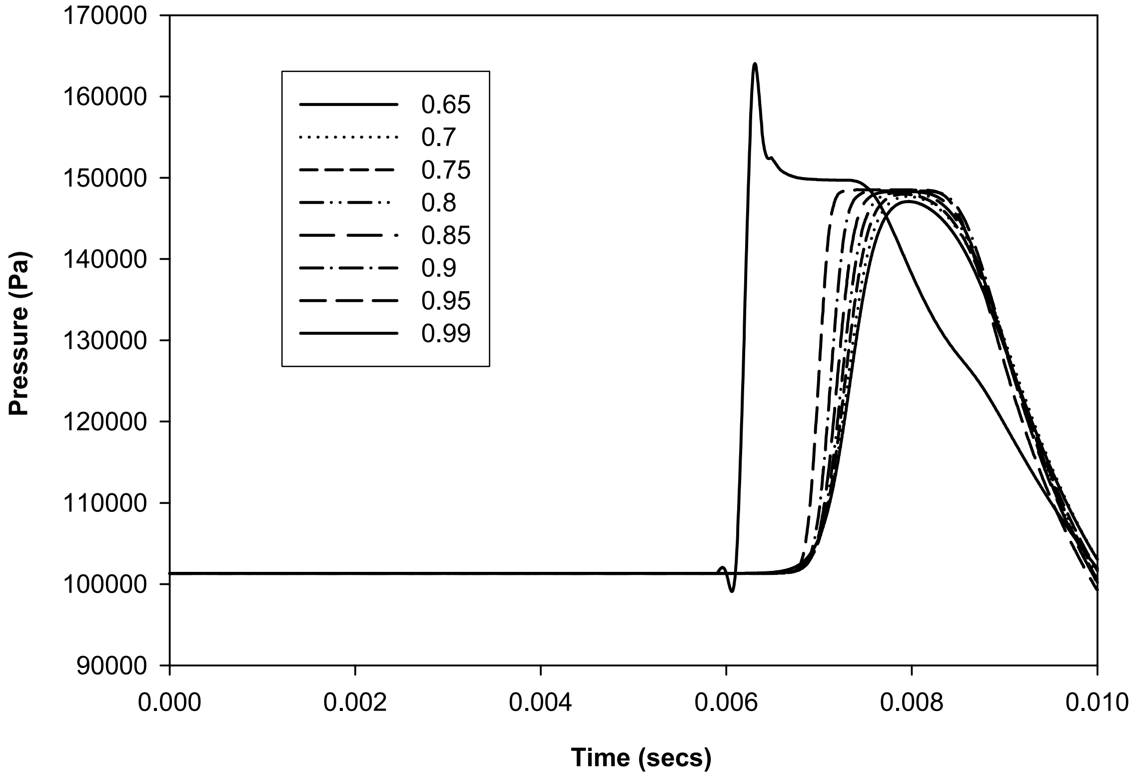


Figure C.32 Pressure plot for station 2 with varying under-relaxation factors

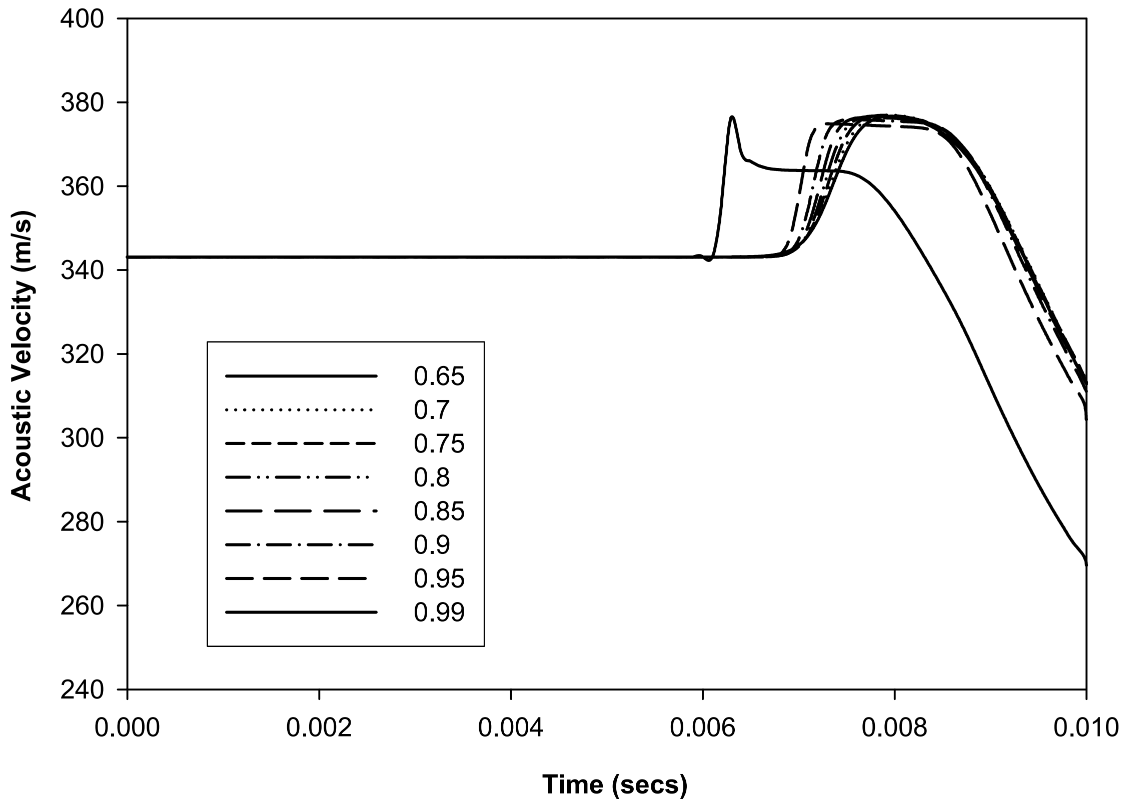


Figure C.33 Acoustic velocity plots for station 2 with varying under-relaxation factors

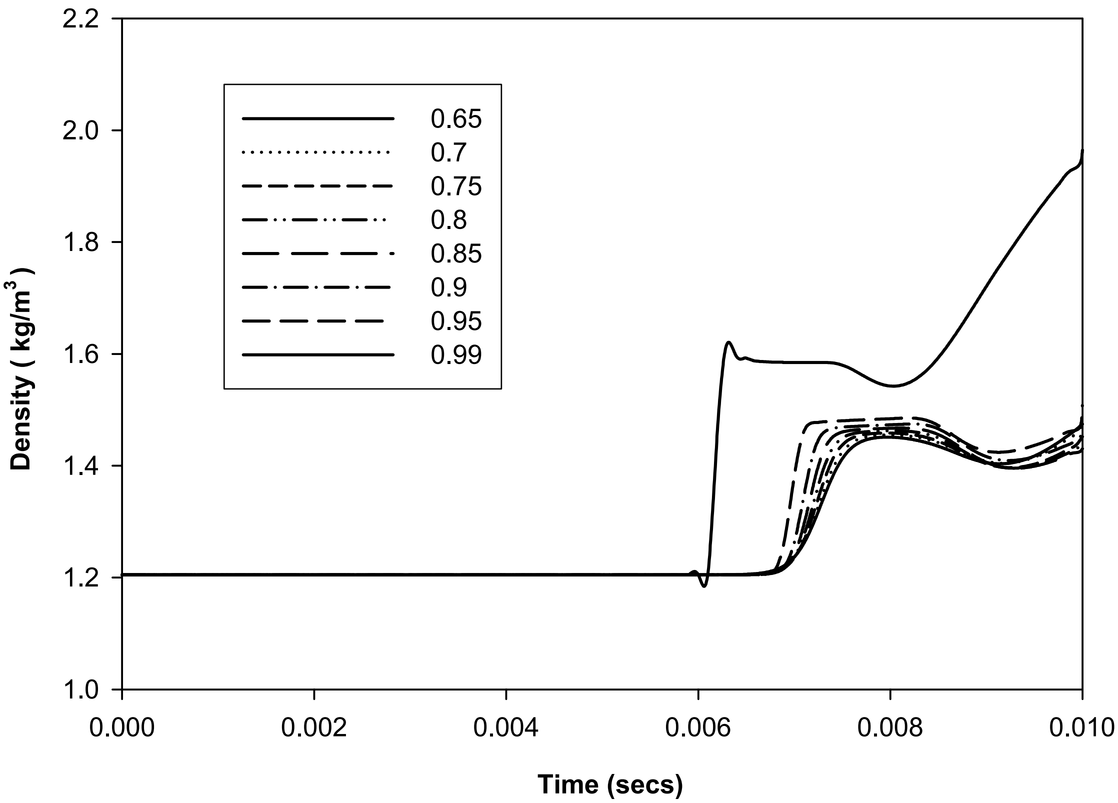


Figure C.34 Fluid density plot for station 2 with varying under-relaxation factors

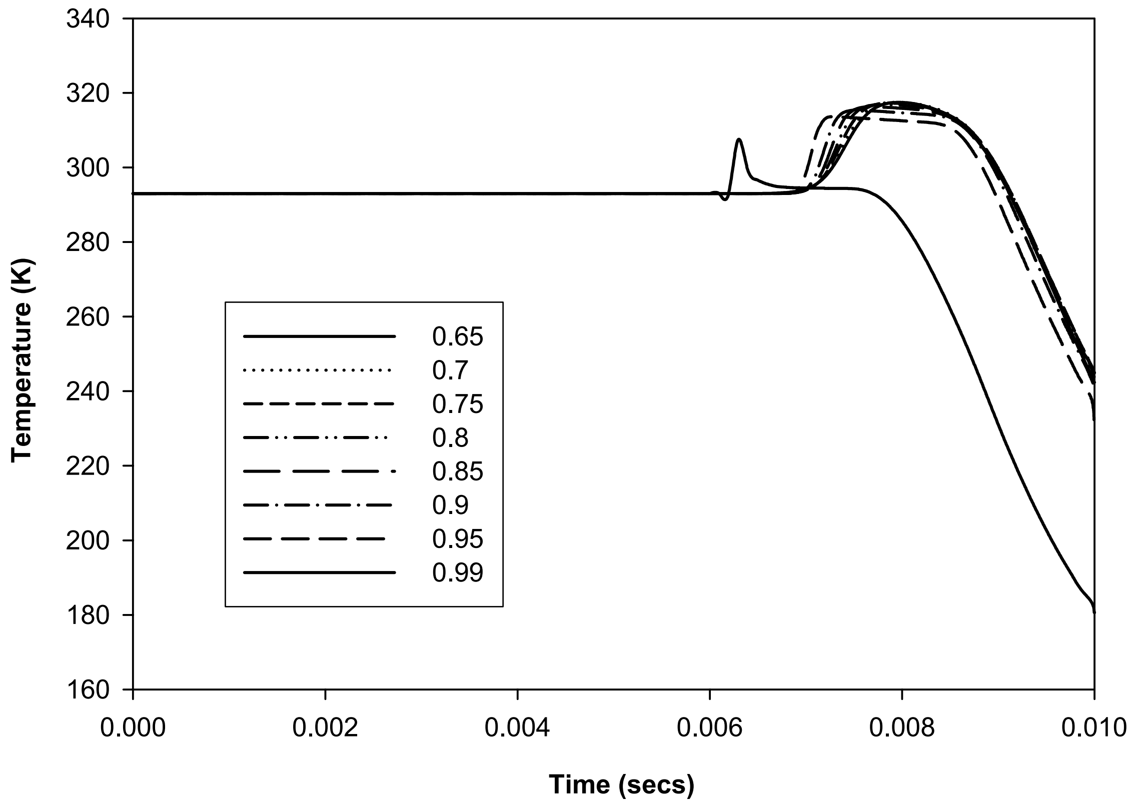


Figure C.35 Reference temperature plot for station 2 with varying under-relaxation factors

Station 3

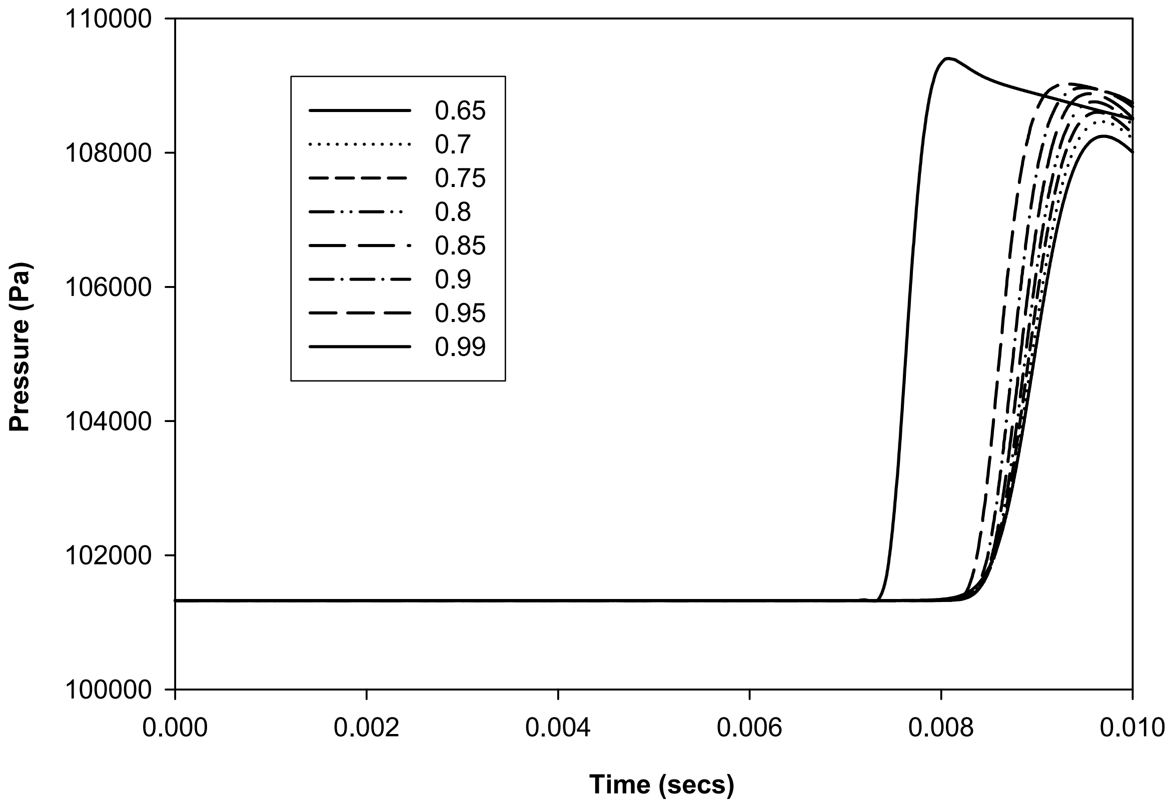


Figure C.36 Pressure plots for station 3 with varying under-relaxation factors

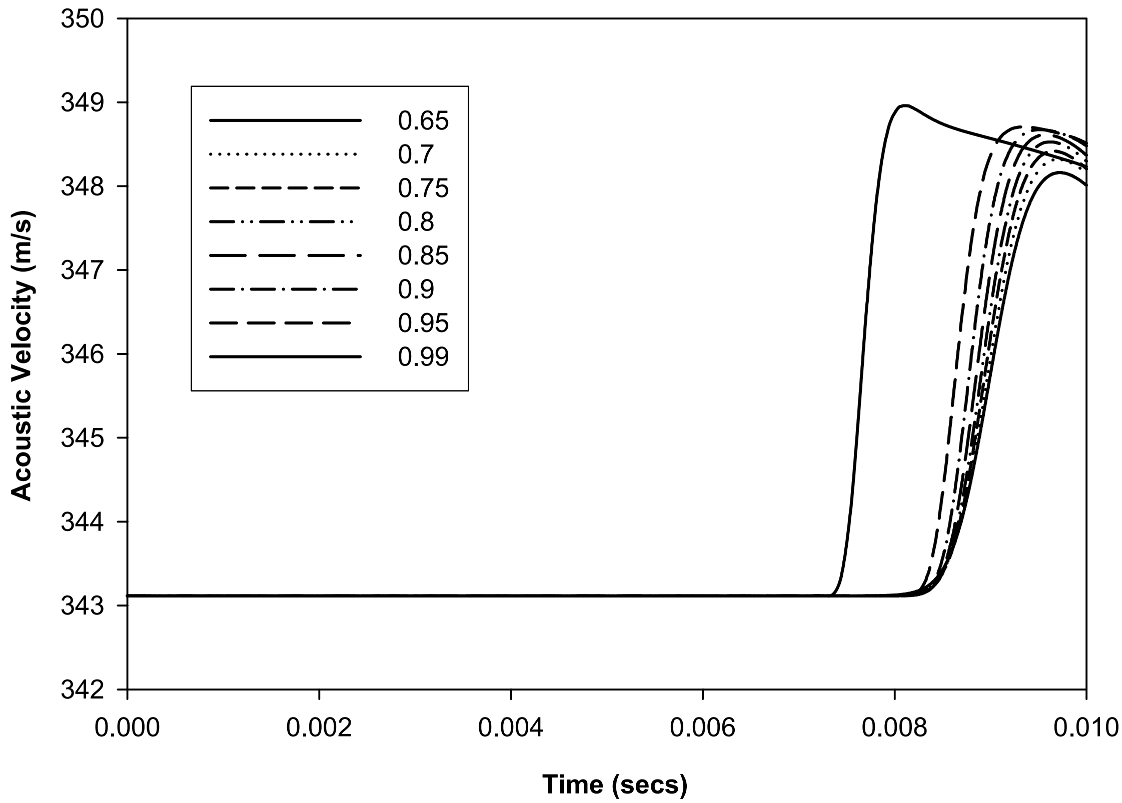


Figure C.37 Acoustic velocity plots for station 3 with varying under-relaxation factors

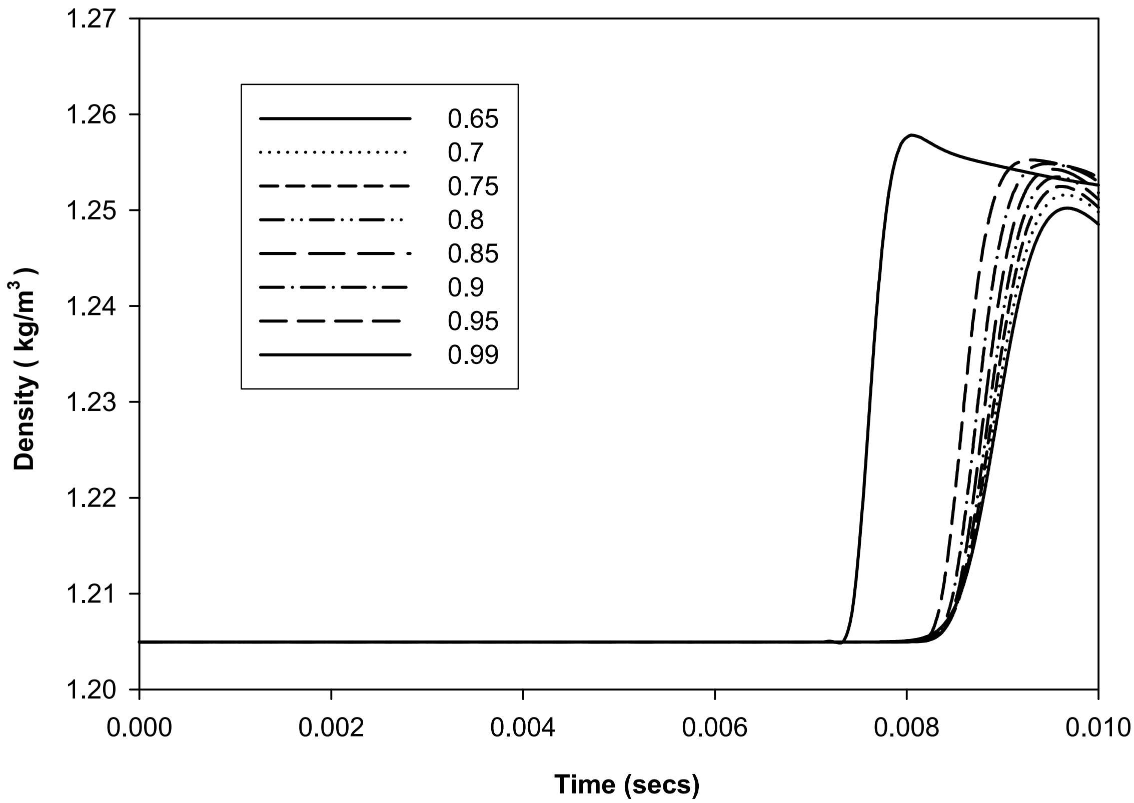


Figure C.38 Fluid density plot for station 3 with varying under-relaxation factors

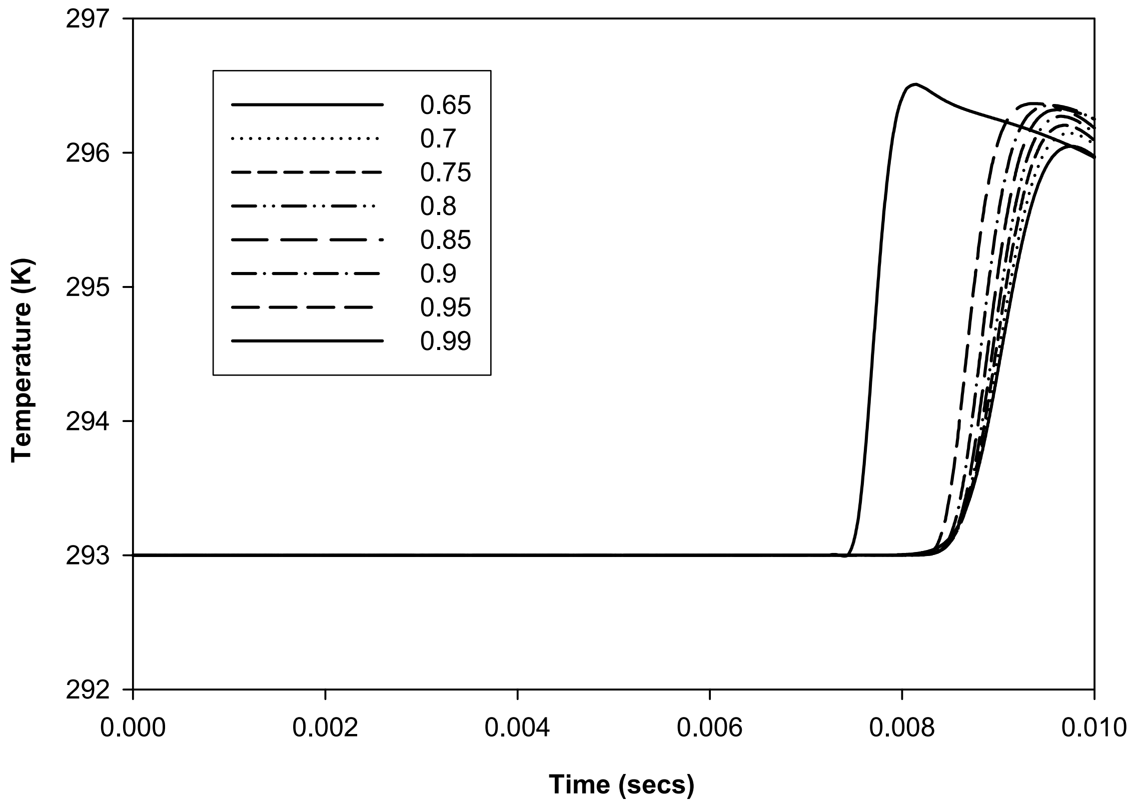


Figure C.39 Reference temperature plot for station 3 with varying under-relaxation factors

Comparison to analytical solution (section 5.6) – Table of results

Analytical solution

p0	101325
T0	293
R	287
Rho0	1.204945
a0	343.1143
Pm	150000
s	0.2
f	100

gamma	1.4
G5	5
G17	0.142857

t=0					dt	0	0.002	0.004	0.006	0.008	0.01
step	time	p	X	Rho	a	u	x	x	x	x	x
0	0	150000	1.057643	1.594642	362.8925	98.89125	0.923568	1.847135	2.770703	3.69427	4.617838
1	0.0001	151884.47	1.059531	1.608927	363.5403	102.1303	1.854909	3.709818	5.564727	7.419635	9.274544
2	0.0002	153761.50	1.061392	1.623104	364.1788	105.3226	2.793912	5.587823	8.381735	11.17565	13.96956
3	0.0003	155623.68	1.063219	1.637121	364.8056	108.4567	3.740436	7.480872	11.22131	14.96174	18.70218
4	0.0004	157463.64	1.065006	1.650923	365.4187	111.522	4.694317	9.388635	14.08295	18.77727	23.47159
5	0.0005	159274.12	1.066747	1.66446	366.0159	114.5084	5.655366	11.31073	16.9661	22.62146	28.27683
6	0.0006	161047.97	1.068436	1.677679	366.5955	117.4063	6.62337	13.24674	19.87011	26.49348	33.11685
7	0.0007	162778.18	1.070068	1.690534	367.1556	120.2066	7.598094	15.19619	22.79428	30.39238	37.99047
8	0.0008	164457.93	1.071639	1.702977	367.6945	122.901	8.579285	17.15857	25.73785	34.31714	42.89642
9	0.0009	166080.57	1.073143	1.714962	368.2106	125.4814	9.566669	19.13334	28.70001	38.26668	47.83334
10	0.001	167639.69	1.074576	1.726446	368.7024	127.9406	10.55995	21.11991	31.67986	42.23982	52.79977
11	0.0011	169129.15	1.075935	1.737389	369.1686	130.2717	11.55884	23.11767	34.67651	46.23534	57.79418
12	0.0012	170543.05	1.077215	1.747751	369.6079	132.4682	12.56299	25.12598	37.68896	50.25195	62.81494
13	0.0013	171875.81	1.078414	1.757496	370.0192	134.5245	13.57208	27.14415	40.71623	54.2883	67.86038
14	0.0014	173122.17	1.079528	1.76659	370.4013	136.4352	14.58575	29.1715	43.75724	58.34299	72.92874
15	0.0015	174277.20	1.080554	1.775001	370.7533	138.1953	15.60365	31.20729	46.81094	62.41458	78.01823
16	0.0016	175336.34	1.081489	1.782699	371.0744	139.8005	16.6254	33.25079	49.87619	66.50158	83.12698
17	0.0017	176295.41	1.082332	1.789659	371.3637	141.247	17.65062	35.30123	52.95185	70.60247	88.25308
18	0.0018	177150.62	1.083081	1.795856	371.6205	142.5311	18.67892	37.35784	56.03676	74.71568	93.3946
19	0.0019	177898.60	1.083733	1.801269	371.8442	143.6498	19.70991	39.41982	59.12972	78.83963	98.54954
20	0.002	178536.38	1.084287	1.805879	372.0344	144.6006	20.74318	41.48636	62.22953	82.97271	103.7159
21	0.0021	179061.45	1.084742	1.809671	372.1905	145.3811	21.77832	43.55664	65.33496	87.11328	108.8916
22	0.0022	179471.74	1.085097	1.812632	372.3122	145.9897	22.81492	45.62985	68.44477	91.2597	114.0746
23	0.0023	179765.62	1.08535	1.814751	372.3992	146.4248	23.85257	47.70515	71.55772	95.41029	119.2629
24	0.0024	179941.94	1.085502	1.816023	372.4514	146.6856	24.89085	49.78169	74.67254	99.56339	124.4542
25	0.0025	179999.99	1.085553	1.816441	372.4686	146.7715	25.92933	51.85865	77.78798	103.7173	129.6466
26	0.0026	179939.56	1.0855	1.816005	372.4507	146.6821	26.96759	53.93519	80.90278	107.8704	134.838
27	0.0027	179760.87	1.085346	1.814717	372.3978	146.4178	28.00522	56.01045	84.01567	112.0209	140.0261
28	0.0028	179464.63	1.085091	1.81258	372.3101	145.9791	29.0418	58.0836	87.12541	116.1672	145.209
29	0.0029	179052.02	1.084734	1.809603	372.1877	145.3671	30.07691	60.15382	90.23074	120.3076	150.3846
30	0.003	178524.65	1.084277	1.805794	372.0309	144.5831	31.11014	62.22028	93.33042	124.4406	155.5507
31	0.0031	177884.63	1.083721	1.801167	371.8401	143.629	32.14108	64.28216	96.42323	128.5643	160.7054
32	0.0032	177134.47	1.083067	1.795739	371.6156	142.5069	33.16932	66.33865	99.50797	132.6773	165.8466
33	0.0033	176277.13	1.082316	1.789526	371.3582	141.2195	34.19448	68.38896	102.5834	136.7779	170.9724
34	0.0034	175316.01	1.081471	1.782551	371.0682	139.7698	35.21615	70.43231	105.6485	140.8646	176.0808
35	0.0035	174254.89	1.080534	1.774838	370.7465	138.1614	36.23397	72.46794	108.7019	144.9359	181.1699
36	0.0036	173097.98	1.079506	1.766413	370.3939	136.3982	37.24755	74.49511	111.7427	148.9902	186.2378
37	0.0037	171849.83	1.078391	1.757306	370.0112	134.4846	38.25655	76.51309	114.7696	153.0262	191.2827

38	0.0038	170515.39	1.07719	1.747548	369.5994	132.4254	39.2606	78.52119	117.7818	157.0424	196.303
39	0.0039	169099.91	1.075908	1.737174	369.1595	130.2261	40.25937	80.51873	120.7781	161.0375	201.2968
40	0.004	167609.00	1.074548	1.72622	368.6927	127.8924	41.25254	82.50507	123.7576	165.0101	206.2627
41	0.0041	166048.53	1.073113	1.714725	368.2004	125.4307	42.2398	84.4796	126.7194	168.9592	211.199
42	0.0042	164424.68	1.071608	1.702731	367.6838	122.8479	43.22086	86.44173	129.6626	172.8835	216.1043
43	0.0043	162743.85	1.070036	1.690279	367.1445	120.1513	44.19545	88.39091	132.5864	176.7818	220.9773
44	0.0044	161012.69	1.068402	1.677417	366.584	117.3489	45.16332	90.32664	135.49	180.6533	225.8166
45	0.0045	159238.03	1.066712	1.66419	366.0041	114.4492	46.12423	92.24845	138.3727	184.4969	230.6211
46	0.0046	157426.89	1.06497	1.650648	365.4065	111.4611	47.07796	94.15592	141.2339	188.3118	235.3898
47	0.0047	155586.41	1.063183	1.636841	364.7931	108.3943	48.02434	96.04867	144.073	192.0973	240.1217
48	0.0048	153723.86	1.061355	1.62282	364.166	105.2589	48.96319	97.92637	146.8896	195.8527	244.8159
49	0.0049	151846.61	1.059494	1.60864	363.5274	102.0656	49.89437	99.78874	149.6831	199.5775	249.4719
50	0.005	149962.07	1.057605	1.594354	362.8794	98.82569	50.81778	101.6356	152.4533	203.2711	254.0889
51	0.0051	148077.67	1.055696	1.580018	362.2245	95.55095	51.73333	103.4667	155.2	206.9333	258.6667
52	0.0052	146200.87	1.053774	1.565688	361.565	92.25371	52.64097	105.2819	157.9229	210.5639	263.2049
53	0.0053	144339.07	1.051847	1.55142	360.9036	88.9468	53.54067	107.0813	160.622	214.1627	267.7034
54	0.0054	142499.63	1.049921	1.537272	360.243	85.64349	54.43244	108.8649	163.2973	217.7298	272.1622
55	0.0055	140689.81	1.048006	1.523301	359.5858	82.35751	55.31633	110.6327	165.949	221.2653	276.5817
56	0.0056	138916.77	1.046109	1.509564	358.9349	79.10298	56.19241	112.3848	168.5772	224.7696	280.962
57	0.0057	137187.50	1.044239	1.496117	358.2931	75.89432	57.06078	114.1216	171.1823	228.2431	285.3039
58	0.0058	135508.84	1.042404	1.483018	357.6635	72.74623	57.9216	115.8432	173.7648	231.6864	289.608
59	0.0059	133887.42	1.040612	1.470321	357.049	69.67358	58.77505	117.5501	176.3251	235.1002	293.8752
60	0.006	132329.63	1.038874	1.458081	356.4525	66.69133	59.62133	119.2427	178.864	238.4853	298.1067
61	0.0061	130841.64	1.037197	1.446351	355.8772	63.81447	60.46072	120.9214	181.3822	241.8429	302.3036
62	0.0062	129429.32	1.03559	1.435183	355.3258	61.05784	61.29348	122.587	183.8805	245.1739	306.4674
63	0.0063	128098.25	1.034062	1.424624	354.8015	58.4361	62.11996	124.2399	186.3599	248.4798	310.5998
64	0.0064	126853.68	1.032621	1.414724	354.307	55.96353	62.9405	125.881	188.8215	251.762	314.7025
65	0.0065	125700.54	1.031275	1.405526	353.8451	53.65396	63.7555	127.511	191.2665	255.022	318.7775
66	0.0066	124643.37	1.030031	1.397072	353.4184	51.52061	64.56538	129.1308	193.6961	258.2615	322.8269
67	0.0067	123686.35	1.028898	1.389402	353.0295	49.57594	65.37059	130.7412	196.1118	261.4824	326.8529
68	0.0068	122833.26	1.027881	1.38255	352.6806	47.83156	66.17161	132.3432	198.5148	264.6864	330.8581
69	0.0069	122087.48	1.026987	1.376549	352.3739	46.29806	66.96896	133.9379	200.9069	267.8758	334.8448
70	0.007	121451.94	1.026222	1.371427	352.1112	44.9849	67.76315	135.5263	203.2894	271.0526	338.8157
71	0.0071	120929.16	1.025589	1.367208	351.8943	43.90029	68.55474	137.1095	205.6642	274.2189	342.7737
72	0.0072	120521.20	1.025094	1.363912	351.7245	43.05111	69.34429	138.6886	208.0329	277.3772	346.7214
73	0.0073	120229.67	1.02474	1.361554	351.6028	42.44278	70.13238	140.2648	210.3971	280.5295	350.6619
74	0.0074	120055.73	1.024528	1.360147	351.5301	42.07921	70.9196	141.8392	212.7588	283.6784	354.598
75	0.0075	120000.05	1.02446	1.359696	351.5068	41.96275	71.70654	143.4131	215.1196	286.8261	358.5327
76	0.0076	120062.87	1.024537	1.360205	351.5331	42.09416	72.49379	144.9876	217.4814	289.9752	362.469
77	0.0077	120243.94	1.024757	1.36167	351.6088	42.47258	73.28195	146.5639	219.8459	293.1278	366.4098
78	0.0078	120542.53	1.02512	1.364084	351.7334	43.09558	74.07161	148.1432	222.2148	296.2865	370.3581
79	0.0079	120957.47	1.025624	1.367436	351.9061	43.95914	74.86334	149.7267	224.59	299.4534	374.3167
80	0.008	121487.12	1.026264	1.371711	352.1258	45.05774	75.65771	151.3154	226.9731	302.6308	378.2886
81	0.0081	122129.39	1.027037	1.376887	352.3912	46.38444	76.45526	152.9105	229.3658	305.821	382.2763
82	0.0082	122881.74	1.027939	1.38294	352.7005	47.93095	77.25652	154.513	231.7696	309.0261	386.2826
83	0.0083	123741.19	1.028963	1.389842	353.0518	49.68773	78.062	156.124	234.186	312.248	390.31
84	0.0084	124704.37	1.030103	1.397561	353.4431	51.64413	78.87218	157.7444	236.6165	315.4887	394.3609
85	0.0085	125767.45	1.031353	1.40606	353.872	53.78848	79.6875	159.375	239.0625	318.75	398.4375
86	0.0086	126926.25	1.032705	1.415302	354.3359	56.10827	80.50839	161.0168	241.5252	322.0335	402.5419
87	0.0087	128176.18	1.034152	1.425243	354.8323	58.59024	81.33523	162.6705	244.0057	325.3409	406.6762
88	0.0088	129512.31	1.035685	1.43584	355.3584	61.22053	82.16839	164.3368	246.5052	328.6736	410.8419
89	0.0089	130929.36	1.037297	1.447044	355.9112	63.98483	83.00818	166.0164	249.0245	332.0327	415.0409
90	0.009	132421.73	1.038977	1.458806	356.488	66.86847	83.85489	167.7098	251.5647	335.4196	419.2745
91	0.0091	133983.53	1.040719	1.471075	357.0856	69.8566	84.70878	169.4176	254.1263	338.8351	423.5439
92	0.0092	135608.59	1.042513	1.483798	357.7011	72.93423	85.57005	171.1401	256.7101	342.2802	427.8502
93	0.0093	137290.50	1.04435	1.496919	358.3315	76.0864	86.43889	172.8778	259.3167	345.7555	432.1944
94	0.0094	139022.60	1.046223	1.510385	358.9739	79.29824	87.31543	174.6309	261.9463	349.2617	436.5772
95	0.0095	140798.07	1.048121	1.524138	359.6253	82.55508	88.19979	176.3996	264.5994	352.7992	440.999
96	0.0096	142609.87	1.050037	1.538122	360.2828	85.84249	89.09204	178.1841	267.2761	356.3682	445.4602
97	0.0097	144450.87	1.051963	1.552278	360.9435	89.1464	89.99222	179.9844	269.9767	359.9689	449.9611
98	0.0098	146313.78	1.053891	1.566552	361.6049	92.45311	90.90034	181.8007	272.701	363.6013	454.5017
99	0.0099	148191.25	1.055812	1.580884	362.2641	95.74935	91.81636	183.6327	275.4491	367.2655	459.0818
100	0.01	150075.87	1.05772	1.595218	362.9187	99.02232	92.74025	185.4805	278.2207	370.961	463.7012

Table C-6. Analytical solution – table of results

Numerical solution

p0	101325
T0	293
R	287
Rho0	1.204944643
a0	343.1142667
Pm	150000
s	0.2
f	100

gamma	1.4
G5	5
G17	0.142857

t=0					dt	0	0.002	0.004	0.006	0.008	0.01
step	time	p	X	Rho	a	u	x	x	x	x	x
0	0	131984.49	1.038487	1.455364	356.3196	66.02651	0.844692	1.689384	2.534076	3.378769	4.223461
1	7.7166E-05	160248.38	1.067676	1.609338	373.3681	93.37558	1.77818	3.556359	5.334539	7.112718	8.890898
2	0.00015242	169639.75	1.076398	1.667312	377.4152	118.6349	2.77028	5.540559	8.310839	11.08112	13.8514
3	0.00022708	172652.39	1.079109	1.708125	376.1755	133.2748	3.78918	7.578361	11.36754	15.15672	18.9459
4	0.000301465	174406.70	1.080668	1.744031	374.1696	138.618	4.814755	9.629511	14.44427	19.25902	24.07378
5	0.000375361	175894.29	1.08198	1.774809	372.4896	141.4317	5.842598	11.6852	17.52779	23.37039	29.21299
6	0.000449133	177326.88	1.083235	1.801681	371.2038	143.7264	6.872458	13.74492	20.61738	27.48983	34.36229
7	0.000522577	178758.28	1.08448	1.825682	370.241	145.9098	7.90476	15.80952	23.71428	31.61904	39.5238
8	0.000595952	180182.69	1.08571	1.847465	369.5153	148.0232	8.939837	17.87967	26.81951	35.75935	44.69919
9	0.000669104	181596.08	1.086922	1.867544	368.9622	150.0938	9.977949	19.9559	29.93385	39.9118	49.88975
10	0.000742214	182992.95	1.088113	1.886308	368.5318	152.1126	11.01924	22.03848	33.05771	44.07695	55.09619
11	0.000815163	184367.79	1.089277	1.90401	368.1899	154.0792	12.06378	24.12755	36.19133	48.2551	60.31888
12	0.000888059	185715.78	1.090411	1.92084	367.9111	155.9863	13.11157	26.22314	39.33471	52.44628	65.55785
13	0.000960855	187033.21	1.091513	1.936924	367.6775	157.833	14.16259	28.32518	42.48778	56.65037	70.81296
14	0.001033586	188316.21	1.092579	1.952345	367.4765	159.6143	15.21677	30.43355	45.65032	60.86709	76.08387
15	0.001106258	189561.81	1.093609	1.967157	367.2991	161.3283	16.27403	32.54806	48.82208	65.09611	81.37014
16	0.001178855	190766.86	1.094599	1.981389	367.1391	162.9724	17.33425	34.6685	52.00275	69.33701	86.67126
17	0.00125139	191928.59	1.095549	1.995054	366.9919	164.5439	18.39732	36.79465	55.19197	73.58929	91.98661
18	0.00132387	193044.60	1.096457	2.008156	366.8548	166.0409	19.46311	38.92623	58.38934	77.85246	97.31557
19	0.001396283	194112.47	1.097321	2.020689	366.7254	167.4618	20.53149	41.06298	61.59447	82.12595	102.6574
20	0.00146863	195129.97	1.098141	2.032645	366.6023	168.8044	21.6023	43.2046	64.80691	86.40921	108.0115
21	0.001540919	196095.13	1.098916	2.044009	366.4848	170.0675	22.67541	45.35081	68.02622	90.70163	113.377
22	0.00161309	197005.71	1.099643	2.054769	366.3717	171.2497	23.75065	47.5013	71.25195	95.0026	118.7532
23	0.00168515	197859.76	1.100323	2.064905	366.2627	172.3489	24.82787	49.65575	74.48362	99.31149	124.1394
24	0.001757111	198655.93	1.100954	2.074402	366.1578	173.3643	25.90692	51.81383	77.72075	103.6277	129.5346
25	0.001828983	199392.97	1.101537	2.083246	366.0569	174.2953	26.98762	53.97524	80.96286	107.9505	134.9381
26	0.001900776	200069.74	1.10207	2.091424	365.9601	175.1415	28.06982	56.13965	84.20947	112.2793	140.3491
27	0.001972499	200685.18	1.102554	2.098922	365.8672	175.9023	29.15336	58.30673	87.46009	116.6135	145.7668
28	0.002044165	201238.36	1.102988	2.10573	365.7784	176.5774	30.23808	60.47615	90.71423	120.9523	151.1904
29	0.002115782	201728.40	1.103371	2.111836	365.6937	177.1663	31.3238	62.64759	93.97139	125.2952	156.619
30	0.00218736	202154.55	1.103703	2.117232	365.613	177.6688	32.41036	64.82072	97.23108	129.6414	162.0518
31	0.002258911	202516.15	1.103985	2.121908	365.5364	178.0847	33.4976	66.9952	100.4928	133.9904	167.488
32	0.002330444	202812.62	1.104216	2.125858	365.4638	178.4136	34.58536	69.17071	103.7561	138.3414	172.9268
33	0.002401968	203043.47	1.104396	2.129075	365.3954	178.6555	35.67346	71.34692	107.0204	142.6938	178.3673
34	0.002473496	203208.33	1.104524	2.131555	365.331	178.8102	36.76174	73.52348	110.2852	147.047	183.8087
35	0.002545035	203306.89	1.1046	2.133294	365.2706	178.8777	37.85004	75.70007	113.5501	151.4001	189.2502
36	0.002616597	203338.97	1.104625	2.134289	365.2142	178.8578	38.93818	77.87636	116.8145	155.7527	194.6909
37	0.002688192	203304.46	1.104598	2.13454	365.1618	178.7506	40.02601	80.05201	120.078	160.104	200.13

38	0.002759813	203203.35	1.10452	2.134047	365.1131	178.5561	41.11334	82.22669	123.34	164.4534	205.5667
39	0.00283147	203035.77	1.10439	2.132812	365.0682	178.2743	42.20003	84.40006	126.6001	168.8001	211.0001
40	0.002903149	202801.99	1.104208	2.13084	365.0267	177.9056	43.28589	86.57179	129.8577	173.1436	216.4295
41	0.002974823	202502.49	1.103975	2.128141	364.9884	177.4501	44.37077	88.74154	133.1123	177.4831	221.8539
42	0.003046498	202137.95	1.103691	2.124717	364.9534	176.9086	45.45449	90.90899	136.3635	181.818	227.2725
43	0.003118172	201709.11	1.103356	2.120579	364.9215	176.2815	46.5369	93.0738	139.6107	186.1476	232.6845
44	0.00318985	201216.85	1.102971	2.115735	364.893	175.5696	47.61783	95.23565	142.8535	190.4713	238.0891
45	0.003261533	200662.12	1.102536	2.110195	364.8677	174.7736	48.69711	97.39422	146.0913	194.7884	243.4855
46	0.003333221	200046.05	1.102052	2.103972	364.8455	173.8944	49.77459	99.54918	149.3238	199.0984	248.8729
47	0.003404917	199369.84	1.101519	2.09708	364.8264	172.9329	50.85011	101.7002	152.5503	203.4004	254.2505
48	0.003476623	198634.81	1.100938	2.089533	364.8103	171.8902	51.92351	103.847	155.7705	207.694	259.6175
49	0.003548337	197842.44	1.100309	2.081348	364.7971	170.7674	52.99464	105.9893	158.9839	211.9785	264.9732
50	0.00362006	196994.28	1.099634	2.072543	364.7867	169.5658	54.06334	108.1267	162.19	216.2534	270.3167
51	0.003691795	196092.01	1.098913	2.063138	364.779	168.2867	55.12947	110.2589	165.3884	220.5179	275.6474
52	0.003763541	195137.39	1.098147	2.053151	364.774	166.9316	56.19288	112.3858	168.5787	224.7715	280.9644
53	0.003835297	194132.32	1.097337	2.042605	364.7713	165.5021	57.25343	114.5069	171.7603	229.0137	286.2672
54	0.003907065	193078.81	1.096485	2.031523	364.7711	163.9999	58.31097	116.6219	174.9329	233.2439	291.5549
55	0.003978844	191978.92	1.09559	2.019928	364.7731	162.4269	59.36537	118.7307	178.0961	237.4615	296.8269
56	0.004050636	190834.84	1.094655	2.007845	364.7773	160.785	60.4165	120.833	181.2495	241.666	302.0825
57	0.00412244	189648.82	1.093681	1.995299	364.7834	159.0762	61.46422	122.9284	184.3927	245.8569	307.3211
58	0.004194256	188423.23	1.092668	1.982317	364.7914	157.3028	62.50841	125.0168	187.5252	250.0336	312.542
59	0.004266085	187160.51	1.091619	1.968927	364.8012	155.4673	63.54894	127.0979	190.6468	254.1958	317.7447
60	0.004337926	185863.17	1.090535	1.955157	364.8126	153.572	64.58571	129.1714	193.7571	258.3428	322.9286
61	0.00440978	184533.78	1.089417	1.941035	364.8256	151.6197	65.6186	131.2372	196.8558	262.4744	328.093
62	0.004481648	183174.98	1.088267	1.926591	364.8399	149.6131	66.64751	133.295	199.9425	266.59	333.2375
63	0.004553529	181789.50	1.087088	1.911856	364.8554	147.5551	67.67233	135.3447	203.017	270.6893	338.3616
64	0.004625422	180380.12	1.08588	1.89686	364.8721	145.449	68.69297	137.3859	206.0789	274.7719	343.4649
65	0.004697329	178949.66	1.084645	1.881635	364.8898	143.2979	69.70935	139.4187	209.128	278.8374	348.5467
66	0.004769248	177500.98	1.083387	1.866211	364.9085	141.1053	70.72137	141.4427	212.1641	282.8855	353.6069
67	0.004841182	176036.99	1.082106	1.850621	364.928	138.8747	71.72898	143.458	215.1869	286.9159	358.6449
68	0.004913129	174560.63	1.080804	1.834897	364.9482	136.6098	72.7321	145.4642	218.1963	290.9284	363.6605
69	0.00498509	173074.88	1.079485	1.819071	364.9692	134.3146	73.73066	147.4613	221.192	294.9227	368.6533
70	0.005057063	171582.78	1.078151	1.803176	364.9907	131.9931	74.72463	149.4493	224.1739	298.8985	373.6232
71	0.005129049	170087.34	1.076804	1.787243	365.0129	129.6496	75.71396	151.4279	227.1419	302.8558	378.5698
72	0.005201049	168591.58	1.075446	1.771305	365.0356	127.2884	76.6986	153.3972	230.0958	306.7944	383.493
73	0.005273063	167098.55	1.07408	1.755394	365.059	124.914	77.67855	155.3571	233.0357	310.7142	388.3928
74	0.005345091	165611.29	1.072709	1.739542	365.0829	122.5311	78.65378	157.3076	235.9613	314.6151	393.2689
75	0.005417132	164132.86	1.071336	1.72378	365.1076	120.1446	79.62428	159.2486	238.8728	318.4971	398.1214
76	0.005489185	162666.31	1.069963	1.70814	365.133	117.7593	80.59007	161.1801	241.7702	322.3603	402.9503
77	0.005561252	161214.67	1.068594	1.692652	365.1594	115.3805	81.55115	163.1023	244.6534	326.2046	407.7557
78	0.005633333	159780.91	1.067231	1.677346	365.1868	113.0132	82.50755	165.0151	247.5226	330.0302	412.5377
79	0.005705428	158368.02	1.065877	1.662253	365.2155	110.6629	83.4593	166.9186	250.3779	333.8372	417.2965
80	0.005777537	156978.93	1.064537	1.6474	365.2458	108.3349	84.40647	168.8129	253.2194	337.6259	422.0323
81	0.005849659	155616.56	1.063212	1.632816	365.2778	106.0348	85.34909	170.6982	256.0473	341.3964	426.7455
82	0.005921794	154283.78	1.061906	1.618529	365.312	103.7682	86.28725	172.5745	258.8618	345.149	431.4363
83	0.005993942	152983.39	1.060623	1.604565	365.3486	101.5408	87.22103	174.4421	261.6631	348.8841	436.1052
84	0.006066103	151718.14	1.059366	1.59095	365.3882	99.35816	88.15052	176.301	264.4516	352.6021	440.7526
85	0.006138279	150490.72	1.058137	1.577709	365.431	97.22607	89.07584	178.1517	267.2275	356.3033	445.3792
86	0.006210469	149303.75	1.056941	1.564866	365.4776	95.15021	89.99709	179.9942	269.9913	359.9884	449.9855
87	0.006282672	148159.79	1.05578	1.552444	365.5285	93.13628	90.91442	181.8288	272.7433	363.6577	454.5721
88	0.006354887	147061.31	1.054658	1.540464	365.5842	91.18997	91.82797	183.6559	275.4839	367.3119	459.1399
89	0.006427116	146010.70	1.053578	1.528948	365.6453	89.31685	92.7379	185.4758	278.2137	370.9516	463.6895
90	0.006499358	145010.25	1.052544	1.517916	365.7123	87.52241	93.64436	187.2887	280.9331	374.5775	468.2218
91	0.006571614	144062.16	1.051558	1.507386	365.7858	85.81202	94.54756	189.0951	283.6427	378.1902	472.7378
92	0.006643884	143168.50	1.050624	1.497376	365.8662	84.19089	95.44767	190.8953	286.343	381.7907	477.2384
93	0.006716167	142331.30	1.049744	1.487904	365.9542	82.66408	96.34491	192.6898	289.0347	385.3796	481.7246
94	0.006788182	141553.97	1.048923	1.479	366.0505	81.23649	97.23949	194.479	291.7185	388.9579	486.1974
95	0.0068545	140866.98	1.048194	1.470942	366.1599	79.91786	98.13164	196.2633	294.3949	392.5266	490.6582
96	0.006920182	140263.33	1.047551	1.463648	366.2838	78.80226	99.02181	198.0436	297.0654	396.0873	495.1091
97	0.00698676	139710.55	1.046961	1.45686	366.412	77.78914	99.91022	199.8204	299.7306	399.6409	499.5511
98	0.00705461	139203.49	1.046417	1.450534	366.5431	76.8579	100.797	201.594	302.3911	403.1881	503.9851
99	0.007123782	138744.71	1.045924	1.444686	366.6785	76.01127	101.6824	203.3648	305.0472	406.7296	508.412
100	0.007194152	138338.84	1.045486	1.439351	366.8198	75.25772	102.5666	205.1331	307.6997	410.2662	512.8328

101	0.007264905	137992.6044	1.045112	1.434581	366.969	74.60751	103.4497	206.8994	310.3491	413.7988	517.2485
102	0.007336088	137710.0565	1.044806	1.43041	367.1271	74.07509	104.3321	208.6642	312.9963	417.3284	521.6605
103	0.007407523	137493.2199	1.044571	1.42686	367.2942	73.66384	105.214	210.4281	315.6421	420.8561	526.0701
104	0.00747927	137343.578	1.044408	1.423946	367.4696	73.37819	106.0957	212.1914	318.2872	424.3829	530.4786
105	0.007551258	137262.3576	1.04432	1.421684	367.6531	73.22044	106.9775	213.9549	320.9324	427.9099	534.8873
106	0.007623489	137250.5477	1.044307	1.420088	367.8438	73.19299	107.8595	215.7191	323.5786	431.4382	539.2977
107	0.007695984	137308.9229	1.04437	1.419173	368.0406	73.29741	108.7422	217.4844	326.2267	434.9689	543.7111
108	0.007768683	137438.1266	1.044511	1.41895	368.2426	73.53494	109.6258	219.2515	328.8773	438.5031	548.1289
109	0.007841637	137638.5285	1.044728	1.419433	368.4483	73.90605	110.5105	221.021	331.5314	442.0419	552.5524
110	0.007914823	137910.4641	1.045023	1.420633	368.6563	74.4111	111.3966	222.7932	334.1898	445.5865	556.9831
111	0.007988233	138253.9682	1.045394	1.42256	368.865	75.04967	112.2844	224.5689	336.8533	449.1378	561.4222
112	0.008061908	138669.0132	1.045842	1.425226	369.0726	75.82086	113.1742	226.3485	339.5227	452.6969	565.8712
113	0.008135831	139155.4584	1.046365	1.42864	369.2774	76.72379	114.0662	228.1325	342.1987	456.2649	570.3312
114	0.008209996	139712.821	1.046963	1.432809	369.4775	77.75657	114.9607	229.9214	344.8821	459.8428	574.8035
115	0.008284434	140340.5841	1.047634	1.437741	369.6709	78.9169	115.8579	231.7158	347.5736	463.4315	579.2894
116	0.008359164	141038.2196	1.048376	1.443443	369.8559	80.20254	116.758	233.516	350.274	467.032	583.79
117	0.008434153	141804.7769	1.049188	1.449917	370.0308	81.61077	117.6613	235.3226	352.9838	470.6451	588.3064
118	0.008509425	142639.0818	1.050068	1.457164	370.1937	83.13747	118.5679	237.1359	355.7038	474.2718	592.8397
119	0.008585006	143540.1238	1.051013	1.465187	370.343	84.77911	119.4782	238.9564	358.4346	477.9127	597.3909
120	0.008660905	144506.6863	1.052021	1.473984	370.4773	86.53189	120.3922	240.7844	361.1766	481.5688	601.961
121	0.008737101	145537.1037	1.05309	1.483548	370.5955	88.39132	121.3102	242.6204	363.9305	485.2407	606.5509
122	0.008813614	146629.5541	1.054215	1.49387	370.6964	90.35182	122.2323	244.4645	366.6968	488.9291	611.1614
123	0.008890466	147782.3566	1.055395	1.504942	370.7793	92.40841	123.1586	246.3173	369.4759	492.6346	615.7932
124	0.008967675	148993.6898	1.056627	1.516752	370.8435	94.55588	124.0894	248.1789	372.2683	496.3578	620.4472
125	0.009045225	150261.2937	1.057906	1.529282	370.8888	96.78876	125.0248	250.0496	375.0744	500.0992	625.124
126	0.009123122	151582.6034	1.05923	1.542511	370.9152	99.10025	125.9648	251.9297	377.8945	503.8593	629.8242
127	0.009201385	152955.1773	1.060595	1.556415	370.9227	101.484	126.9096	253.8193	380.7289	507.6386	634.5482
128	0.009280031	154376.5071	1.061998	1.570971	370.9117	103.9338	127.8593	255.7187	383.578	511.4374	639.2967
129	0.009359075	155843.9293	1.063434	1.586151	370.8827	106.4432	128.814	257.628	386.442	515.256	644.0699
130	0.009438506	157354.3601	1.0649	1.601922	370.8368	109.0056	129.7737	259.5473	389.321	519.0947	648.8684
131	0.009518228	158903.4457	1.066391	1.618228	370.7754	111.6134	130.7385	261.4769	392.2154	522.9538	653.6923
132	0.009597537	160479.8349	1.067896	1.634906	370.7045	114.256	131.7084	263.4167	395.1251	526.8335	658.5419
133	0.009676443	162072.2632	1.069404	1.651862	370.6223	116.9029	132.6834	265.3668	398.0503	530.7337	663.4171
134	0.009754954	163676.574	1.07091	1.669049	370.5294	119.5465	133.6636	267.3272	400.9907	534.6543	668.3179
135	0.009833078	165288.7288	1.07241	1.686423	370.4268	122.1798	134.6488	269.2976	403.9464	538.5952	673.2439
136	0.009910825	166904.8082	1.073902	1.703937	370.3153	124.7961	135.639	271.278	406.917	542.556	678.1951
137	0.009988206	168521.0132	1.075381	1.721547	370.1959	127.3895	136.6342	273.2684	409.9025	546.5367	683.1709
138	0.01	169466.1348	1.076241	1.728457	370.4898	129.9541	137.6351	275.2701	412.9052	550.5403	688.1753

Table C-7. Numerical solution – table of results

Comparison of analytical and numerical solutions (section 5.6) – plots

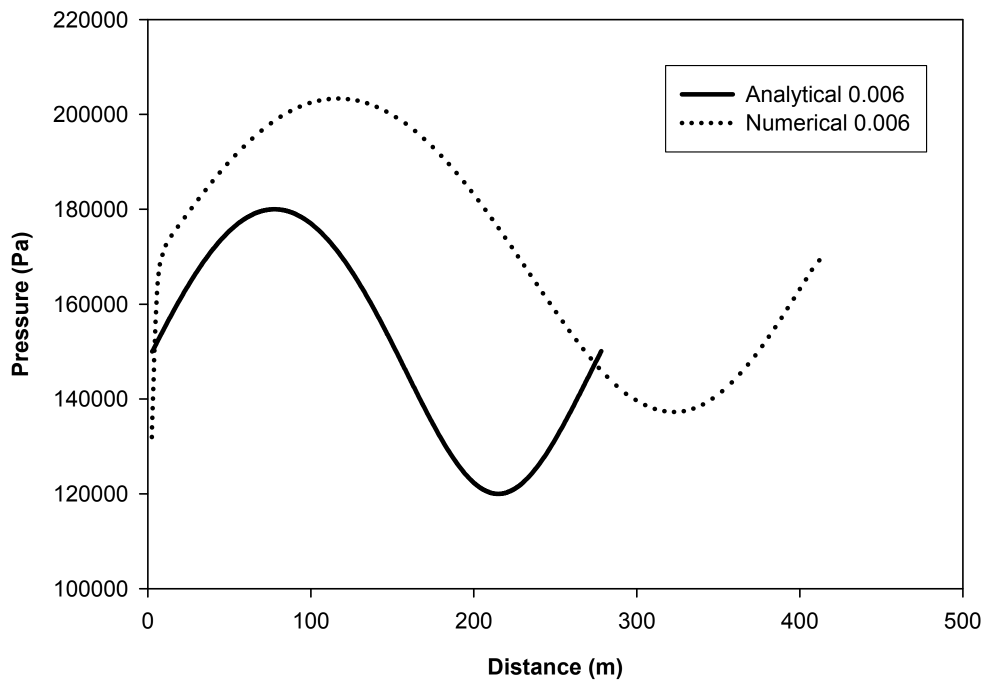


Figure C.40 Comparison of analytical and numerical solutions for time of 0.006 seconds

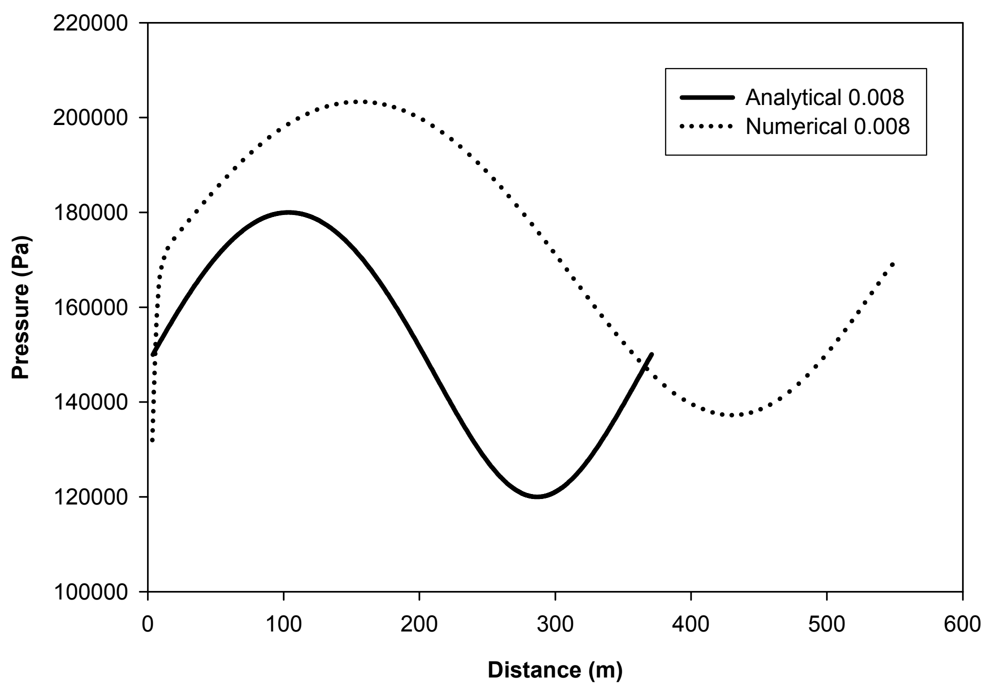


Figure C.41 Comparison of analytical and numerical solutions for time of 0.008 seconds

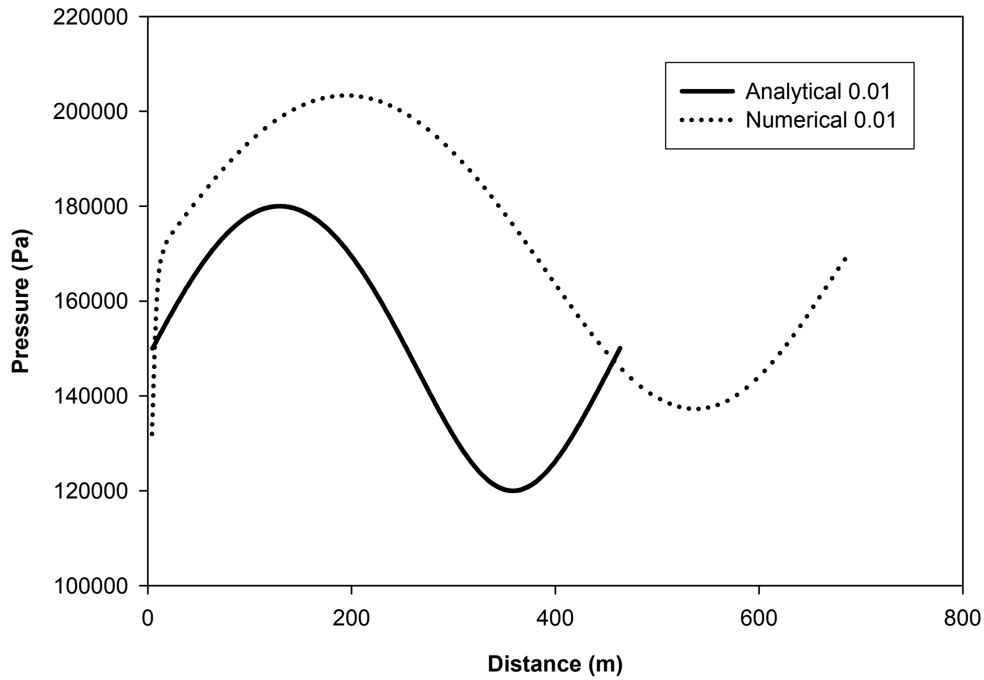


Figure C.42 Comparison of analytical and numerical solutions for time of 0.01 seconds

VITA

Kalyanasundaram Krishnan

Candidate for the Degree of

Master of Science

Thesis: ONE-DIMENSIONAL ANALYSIS TECHNIQUES FOR PULSED JET FLOW DISTRIBUTION SYSTEMS

Major Field: Mechanical Engineering

Biographical:

Personal Data: Born in Kollam, Kerala, India on May 22, 1976, the son of E.K.Radha and K.Kalyanasundaram.

Education: Received Bachelor of Technology in Mechanical Engineering from University of Kerala, India in 1997. Received the Advanced diploma in piping design and Engineering from Maharashtra Institute of Technology, Pune, India in July 2001. Completed the requirements for the Master of Science degree with a major in Mechanical Engineering at Oklahoma State University in December of 2004.

Experience: Worked as Engineer – Instrumentation for Netel Chromatographs R&D Center, Thane, India 1998 to 1999; employed as Engineer – Mechanical in Uhde India, India 1999 to 2002; employed by Oklahoma State University, Department of Mechanical Engineering as a graduate Teaching assistant 2002 to 2004; employed by Oklahoma State University, Department of Mechanical Engineering as a graduate research assistant May 2003 to August 2003.

Professional Memberships: American Society of Mechanical Engineers.

Name: Kalyanasundaram Krishnan Date of Degree: December, 2004

Institution: Oklahoma State University Location: Stillwater, Oklahoma

Title of Study: **ONE-DIMENSIONAL ANALYSIS TECHNIQUES FOR PULSED JET FLOW DISTRIBUTION SYSTEMS**

Pages in Study: 296 Candidate for the Degree of Master Science

Major Field: Mechanical Engineering

Scope and Method of Study: Active flow control offers much promise for improved aircraft performance by delaying stall, increasing lift, reducing drag, enhancing combustion, and decreasing jet noise. The enhanced performance of active control may be achieved through intelligent application of flow field modifiers with the proper characteristics at the proper times and the proper locations. Steady or pulsed blowing may be applied optimally at only those discrete locations where the control system senses it is required to maintain attached flow. The flow control performance of pulsed blowing systems has a strong dependence upon the fluid-acoustic dynamics of the flow control system. Pulsed blowing systems start with a steady flow supply and process it to generate a pulsatile flow with characteristic frequencies of the same order as the frequencies of the flow being controlled. The steady pulsatile flow of internal combustion engine exhaust systems has many similarities to pulsed jet active flow distribution systems. Pulsatile flow in internal combustion engines has been analyzed by Blair (1999). This analysis method is being extended to pulsed jet blowing systems.

Finding and Conclusion: The Virtual 4 stroke engine modeling software® of Blair is initially used for the simulation. The pressure output at the outlet to the exhaust pipe is approximated to a square pulse by varying the design parameters. A general-purpose one-dimensional code is developed to simulate the wave transmission and establish system dynamic parameters. The simulations have been carried out for two test cases found in literature. The concept of pressure wave superposition and reflection is used with the governing equations of continuity, momentum and energy to arrive at particle velocities and pressures at relevant mesh points in the flow field. A mesh method of interpolation is used. The simulation is conducted for various wave frequencies. Effect of frequency on pressure and velocity is also demonstrated. Usefulness of this one-dimensional code as an analysis tool is demonstrated.

This work has been supported by NASA EPSCOR.

ADVISOR'S APPROVAL: ______________________________