Diploma thesis

1-D Simulation of the intake manifold of a single-cylinder Engine

Georgios V. Varsos

Supervisors: **Alex Taylor** (Imperial College, London, UK)

Yannis Hardalupas (Imperial College, London, UK)

Evaggelos Giakoumis (National Technical University of Athens, Greece)

March – September 2010

Email: george.varsos10@imperial.ac.uk

Structure of Diploma Thesis

Figure Contents

Chapter 1

Introduction

The aim of this diploma thesis is to effectively simulate a single-cylinder engine's intake manifold by understanding how the waves produced from running the engine affect its performance. For this purpose a pulse wave method will be used and more specifically, the one dimensional *method of characteristics* described by Benson [1] and Shapiro [2]. The method of characteristics will be explained and exampled by using appropriate codes in **MATLAB**. Using the codes developed, the user will be able to adjust individually the parameters in order to simulate the exact occasion of interest. As this code is not a black-box type, the user can alter these values and see immediately the result while being able to understand why and how certain results are being produced. Summarizing, the most important aim of this thesis is to provide a user friendly tool for simulating the intake manifold of an engine via the method of characteristics.

All the main calculations have been done supposing an isentropic flow across the engine and thus the codes provided are based on this theory. Moreover, the theory of a non-isentropic flow is analysed and some codes are provided in order to act as a basis for future studies with that kind of flow.

At the same time, the same simulation will be attempted to be done by using the simulation program **GT-Power**, where also some comparison results will be shown for a natural aspirated, a turbocharged and a turbocharged with added supercharger single-cylinder engine.

Furthermore, certain suggestions are being made for improving the engine's efficiency using the method of the characteristics. These suggestions can act as a start for future diploma thesis or experiments.

All the variables used in this report are listed and explained below [6]:

Variable name	**Variable name**	**Suffix**
A = non dimensional speed of sound	$\dot{M}$ = non dimensional mass flow rate	in =characteristic towards boundary
A = jacobian matrix	$\dot{Q}$ = heat transfer	e = exhaust
a = velocity of sound	$\dot{W}$ = work rate	cr = critical
β= Riemann variable	U_B= non dimensional velocity	c = cylinder
BDC = bottom dead center	ρ = density	s = grid space index
C = measure of entropy change	ψ = valve area ratio	0 = stagnation
D = diameter of duct	t = time	I = inlet
e_0 = specific internal energy stagnation	V = volume	p = pipe
$\vec{C}$ = vector of source terms	S=stroke	t = throat
E_{cv} = intrinsic internal energy	q = heat transfer per unit mass	r = grid time index
EVO = exhaust valve opening	TDC = top dead center	ref = reference
EVC = exhaust valve closing	IVO = inlet valve opening	s = grid space index
f = wall friction co-efficient	π = pressure function	son = sonic
F = area	p = pressure	sub = subsonic
ϕ= nozzle area ratio	S = sonic boundary	
$\vec{F}$ = Vector in x-direction	s = Entropy	
h_0 = specific stagnation enthalpy	x = distance	out= characteristic leaving boundary
k = ratio of specific heats	u = velocity (x-direction)	
λ= Riemann variable	U = non dimensional speed	
$\dot{m}$ = mass flow rate	T = temperature	
M = mesh number	X = non dimensional distance	
	Z = non dimensional time	

Chapter 2

Brief History of the Engine Intake Manifold

[Theory of Engine Manifold Design, Wave Action Methods for IC Engines, DE Winterbone and RJ Pearson]

The design of intake manifolds for internal combustion engines was given a significant importance once the desire to increase the engine's overall output took place at 1940s. Intake manifolds are typical devices were there are unsteady fluid dynamics; thus different to steady flow models should be implemented in order to understand and calculate the unsteady flow of the compressible gases flowing through the engine.

The equations describing unsteady flow were first introduced at the 19^{th} century and a method for solving these equations was proposed by Riemann. What Riemann did is that he showed that *the hyperbolic partial differential equations of unsteady flow could be reduced to ordinary differential equations if they were solved along particular lines in the space-time field*. Those are the 'characteristics' lines and that is why the method that will be showcased here is called the 'Method of Characteristics'. The method at first was graphical and painstaking but since the computer methods were introduced, the method of characteristics became the dominant approach for calculating engine performance.

A phenomenon of waves of fluids that is worth mentioning is called waterhammer and occurs when a valve in a pipeline is closed rapidly, just as the intake valve of the internal combustion engine. Due to the rapid closure of the valve the waterhammer effect can generate pressures that are high enough to destroy the pipe, with catastrophic results for the whole system. Using the method of characteristics one can manipulate and control the travelling of those waves, and even use this effect and its pressure waves to pump water from a low reservoir to a higher one.

Another area where waves and unsteady incompressible flow can have an effect is in fuel injection equipment. Unsteady wave flows can affect the pressure profile of the pump-pipe-nozzle system used in diesel engines. Due to the compression and mainly here the expansion waves, the pressure of the fluid might drop below the saturated vapour pressure, possibly leading to cavitation.

Another showcase of unsteady flow is the blood flow, where there can also be pressure waves in the incompressible fluid. Despite the great variety of the problems mentioned, all of them can be described by the basic generic equations and be attempted to be solved with the method of characteristics.

Chapter 3

An insight to Manifolds [3] (J. Lumley)

In this chapter a physical explanation of the situations that take place in an intake manifold during an engine cycle will be given. The fluid (air) in the system is sloshing back and forth due to its inertia, bouncing against the resilience of the compressed gas in the resonant cavities; there are compression and expansion waves travelling through the gas, reflecting from closed and open ends, and from changes in cross-section.

The gas in the intake branch from the manifold to the valve cannot be considered just as a mass that must be moved; it is compressible. *We may think of it like a string of tiny masses connected by tiny springs*, so that waves can travel back and forth along the branch.

These waves can be analysed by the soon to be described method of characteristics, considering that if the incident wave sees a larger cross section, the reflected wave is of the other type (compression wave is reflected as expansion, and expansion wave as compression) and if the incident wave sees a smaller cross-section, the reflected wave is of the same type. That is a generalization of the theory mentioned above, concerning open and closed ends.

There is a fine introduction to wave modelling of a single cylinder with one intake pipe system engine in [3].

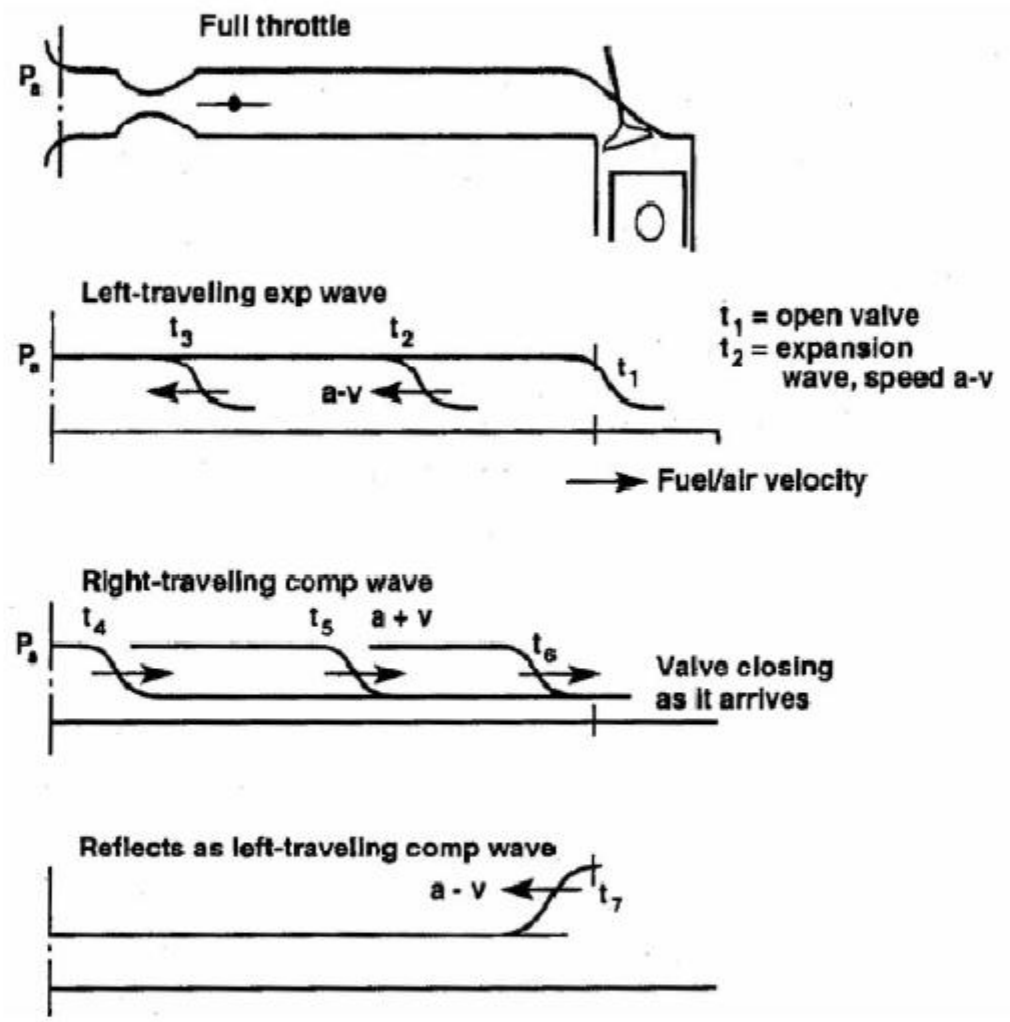


Figure1: The figure illustrates a schematic representation of wave action in the inlet manifold [Lumley, 1999]

As can be seen in the above *figure 1* at time t1 –while the piston is starting to drop- the inlet valve opens and the air rushing in the cylinder pushes further the piston down. Consequently, the pressure in the cylinder drops by that movement-relative to the pressure in the manifold- and that leads to the creation of an expansion wave as a reduction of pressure is being displayed. This expansion wave of relative velocity a-v travels towards the open end of the manifold, which at time t4 is reflected back as a compression wave of relative velocity a+v, as at open ends the incident waves are converted.

At time t6 the compression waves arrives just as the inlet valve is closing, being considered as a closed end boundary (approximately). As a result, we may have a reflection of the wave but the transmitted compression wave will enter the cylinder with increased pressure, thus acting as a mild supercharging, which results in increased volumetric efficiency. ***The potential of this effect is obvious and can be fully realised if it is known exactly how much time is needed for the compression wave to reach the valve, thus the cylinder.***

Besides the engine speed, which of course has an immediate effect to the valve opening, the focal point is on the length of the inlet pipe which defines the distance the wave has to travel in order to reach the valve in open mode.

The general idea is that **for lower engine speed the length of the inlet pipe must be longer and as the speed increases the length should be shortened.** That is exactly what systems of variable length intake manifold do in order to optimise power and torque across the range of engine speed operation, as well as to help provide better fuel efficiency. This effect is often achieved by having two separate intake ports, each controlled by a valve, that open two different manifolds - one with a short path that operates at full engine load, and another with significantly longer path that operates at lower load.

For references purposes there are in total four common implementations of variable intake manifold [8]. First, two discrete intake runners with different length are employed, and a butterfly valve can close the short path. Second the intake runners can be bent around a common plenum, and a sliding valve separates them from the plenum with a variable length. Straight high-speed runners can receive plugs, which contain small long runner extensions. The plenum of a 6 or 8 cylinder engine can be parted into halves, with the even firing cylinders in one half and the odd firing cylinders in the other part. Both sub-plenums and the air intake are connected to an Y (sort of main plenum). The air oscillates between both sub-plenums, with a large pressure oscillation there, but a constant pressure at the main plenum. Each runner from a sub plenum to the main plenum can be changed in length. For V engines this can be implemented by parting a single large plenum at high engine speed by means of sliding valves into it when speed is reduced.

Chapter 4

Introduction to the Method of Characteristics

The Method of Characteristics is a method for modelling the waves of 1D gas flow motion through various areas such as inlets, ducts, reservoirs, valves etc. The reason for negotiating one dimensional flow is because the length to diameter ratios of the pipes is considered to be large enough for the flow to be a fully developed turbulent flow. However, one of the defects of this method is that it best describes small pressure waves but is not suitable for large waves or supersonic flow and shock waves; for our engine application though the method is adequate.
Although it has now been replaced by modern finite difference techniques the Method of Characteristics remains a powerful approach to formulating boundary conditions in many modern engine simulation codes, while also giving a physical insight into the propagation of waves in fluids.

The essential differences between simulating engine manifold's flow and calculating the flow in other applications is [4]:

- The length/diameter ratio in engine manifolds is small compared to pipe work problems
- The engine case has periodic flow and will reach a steady periodic state; on contrast many pipe work cases have a single, sudden blowdown.
- The fluid in engine manifolds may well be considered to behave as an ideal gas which is not always necessary the case for the other type of problems.

The most important thing is to understand the ways in which the fluid flow interacts with the encountered boundaries.

The method is using the wave theory to describe the flow during unsteady motion. More specifically, special lines on the surface are used for this purpose, called '**physical characteristics**'. These lines are of constant properties, showing the path along which the perturbation properties and disturbances are propagated, and are analogous to the Mach waves (characteristic curves) of a steady, two dimensional flow. So, the characteristics are lines in space along which information can travel, in the same way that roads are lines along which vehicles can travel.

There are two families of parallel straight lines, designated respectively by the symbols I and II, according to their direction in comparison to the motion of fluid. Type I lines are right-forward and type II are left-forward (opposite to flow motion). The plane on which these lines –commonly spoken of as waves- appear is a x-t plane, called the **physical plane**.
For a certain type of line the perturbation velocity and sound velocity are constant along the line, as is the slope of each line, depending on the speed of sound relative to the fluid. More specifically:

$$(\frac{dx}{dt})_{I,II} = \bar{u} \pm \bar{c}$$: propagation speed of the line-wave

Where $\bar{u}$ is the velocity of the fluid and $\bar{c}$ is the speed of sound. The upper sign refers to the symbol I and the lower to the symbol II. The slope is also the propagation speed of the wave and is constant along a wave of a given family.

The procedure is to create a net of finite number of these lines, covering the x-t plane. The approximation is to consider that *a fluid particle is assumed to undergo changes in its velocity, speed of sound or pressure only when it is crossed by either a left moving or right moving pressure wave* (lines of type II and I respectively). So, in each of the quadrilaterals formed by the net of I and II lines the perturbation properties are assumed to be constant (and different for each quadrilateral).

Flow patterns may be synthesized in terms of four elementary types of pressure pulses, depending on whether they are of family I or II and whether they are compressions or rarefactions. In a rarefaction (expansion) wave the fluid over which the wave passes undergoes a reduction in pressure, the opposite happens in a compression (condensation) wave.

The reason of forming this net of lines is because we want to produce and image the *path line*, which is the time-space trajectory of a fluid particle of fixed identity, defined by the relation

$$(\frac{dx}{dt})_{path.line} = u$$

Corresponding to the main velocity of the fluid. The path line represents a disturbance propagating with the local fluid velocity, which transports by advection the bulk gas energy level and composition. The slope of the path lines represents the temperature propagation velocities and these disturbances as can be seen in the above equation do not propagate at the same speed as the pressure waves, actually they arrive later thus there is a phase difference. These path lines are analogous to the streamlines of a steady flow and their direction can be seen by the crossing points of two lines I and II in the position diagram. All in all, a path line represents a string of a particle's locations, its transport history of movement.

In order to see the corresponding changes in fluid properties another plane is used, called the **u-c state plane**. In this plane lines of I and II have now respectively the following slopes:

$$(\frac{dc}{du})_{I} = -\frac{k-1}{2} \qquad (\frac{dc}{du})_{II} = \frac{k-1}{2}$$

The waves on the position diagram can be transferred on to the state diagram, when the state of the gas is altered by the disturbance passing through it. Each point in the physical plane can be shown in the state plane, and by using a horizontal line crossing the c-axis we can find its sound velocity; by using a vertical line crossing the u-axis we can find its velocity.

Subsequently we are able to compute the values of pressure, density and temperature by the corresponding values of the sound velocity.

In general, the particle velocity and the speed of propagation of a particle in space, as a finite amplitude wave passes through it, are non-linear functions of the gas pressure; the *propagation* speed of points in a wave increases as the pressure increases. The result will be in a distorting shape of a wave as the high pressure regions travel faster than the low-pressure ones. [3]

The method of characteristics is a graphical method which will be transformed to a numerical method, in order to make calculations for waves of finite amplitude, for which analytical methods are usually not available.

This method uses two main variables, called the Riemann variables: λ for right-forward and β for left-forward waves, used to determine the change of the properties of the flow as a wave propagates.

According to Benson, these variables are set as follow:

$$\lambda = c + u \cdot \frac{k-1}{2}$$

$$\beta = c - u \cdot \frac{k-1}{2}$$

Where c is the velocity of sound, u the flow velocity and k the ratio of specific heats.

By dividing those equations with the reference speed of sound, cref, we transform them in a non-dimensional form:

$$\lambda = A + U \cdot \frac{k-1}{2} \qquad \text{a}$$

$$\beta = A - U \cdot \frac{k-1}{2} \qquad \text{b}$$

As a summary, the non-dimensional variables are:

$$A = \frac{c}{cref} = \left(\frac{p}{pref}\right)^{\frac{k-1}{2k}}$$ pref a reference pressure

$$U = \frac{u}{cref}$$

$$X = \frac{x}{Lref}$$ Lref a reference length

$$Z = t\frac{cref}{Lref}$$

Along each wave, right-forward or left-forward, the corresponding Riemann variables are constant, which is a base for the numerical method of characteristics.

Determining flow properties using the Riemann variables

To understand how the Riemann variables are used to determine the flow properties, a x - t plane will be considered.

Figure 2, shows a grid divided into meshes of size DX and DZ. The rightward travelling characteristic starts at point (r, s-1) with velocity U + A, and arrives at point P. Recall that a Riemann variable is constant along a characteristic, therefore

$$\lambda_r^{s-1} = \lambda_p^{s}$$

Similarly the leftward travelling characteristic departs at point (r, s+1) with velocity U-A, arriving at point P.

$$\lambda_r^{s+1} = \lambda_p^{s}$$

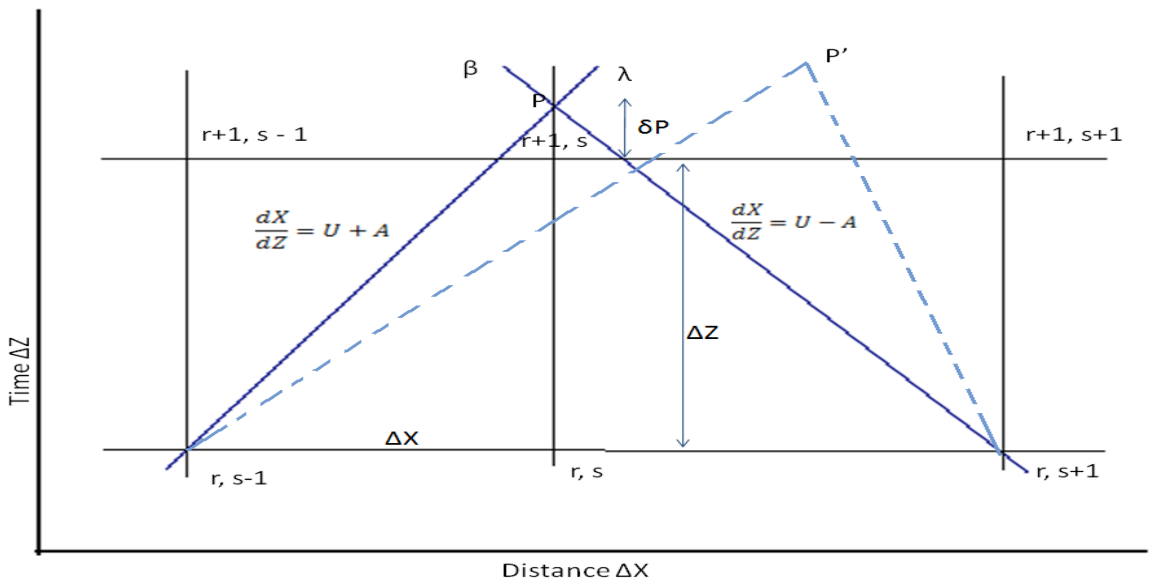


Figure2: Characteristics shown on position diagram

At point P the two travelling characteristics cross each other and therefore, all the properties at point P can be determined by simple simultaneous equations. This means that by knowing the values of A and U at two points s-1 and s+1 in the flow at time r, it is possible to

determine the values of A and U at point s and time P. This is a powerful tool as it enables the determination of the properties of the gas everywhere along the flow that is surrounded by two other points.
This is not the case for boundaries as only the values from one characteristic are known. In the next chapter it will be explained how to deal with boundaries.
Point P is at a distance δP from point (r+1, s). The Riemann variables at point (r+1, s) are obtained by simple geometry. The slopes of the lines (r, s-1)-P and (r, s+1)-P are known:

$$\left(\frac{dX}{dZ}\right)_\lambda = \left\{\frac{k+1}{2(k-1)}\right\}\lambda - \left\{\frac{3-k}{2(k-1)}\right\}\beta$$

And

$$\left(\frac{dX}{dZ}\right)_\beta = \left\{\frac{3-k}{2(k-1)}\right\}\lambda - \left\{\frac{k+1}{2(k-1)}\right\}\beta$$

Furthermore the length between points (r, s-1) and (r, s) is ΔX therefore the length between points (r, s) and P is:

$$\Delta Z + \delta P = \frac{\Delta X}{(\frac{dX}{dZ})\lambda} = \frac{\Delta X}{(\frac{dX}{dZ})\beta}$$

Therefore the value of $\lambda_{r+1,s}$ can be related to $\lambda_{r,s}$ and $\lambda_{r,s-1}$ by linear interpolation, since $(\lambda_{r+1,s} - \lambda_{r,s})*(\Delta Z + \delta P) = \Delta Z*(\lambda_{r,s-1} - \lambda_{r,s})$

And similarly $\beta_{r+1,s}$ can be related to $\beta_{r,s}$ and $\lambda_{r,s+1}$, by linear interpolation, since

$(\beta r+1,s - \beta_{r,s})*(\Delta Z + \delta P) = \Delta Z*(\beta r,s+1 - \beta\ r,s)$

Now the value of (ΔZ + δP) is known and by substituting it in the above equations the following relationships are obtained [1]:

$$\lambda(r+1,s) = \lambda(r,s) + \frac{\Delta Z}{\Delta X}[bb*\lambda(r,s-1) - \alpha\alpha*\beta(r,s-1)]*[\lambda(r,s-1) - \lambda(r,s)]$$

$$\beta(r+1,s) = \beta(r,s) + \frac{\Delta Z}{\Delta X}[bb*\beta(r,s+1) - \alpha\alpha*\lambda(r,s+1)]*[\beta(r,s-1) - \beta(r,s)]$$

In order to evaluate the Riemann variables at all the mesh points, given a set of boundary conditions,

the required ratio of $\frac{\Delta Z}{\Delta X}$ must be determined. The triangle created by the two characteristic lines in *figure 2,* represents the zone of dependence of point P, similarly the triangle created by the two dashed lines represents the zone of dependence of P'. However point (r+1,s) lies in the zone of dependence of P but not the zone of dependence of P'. The zone of dependence is that region where all the points within it are strictly dependant on the values at the vertices of the triangle, thus it must be ensured that point (r+1,s) lies within the zone of dependence of (r,s-1) and (r,s+1). In order to do this:

$$\frac{\Delta Z}{\Delta X} < \frac{1}{A + |U|}$$

This is the ***stability criterion*** that λ and β must satisfy.
Generally ΔX is kept constant whilst ΔZ is re-determined at each new time level, by independently calculating A and U.

Boundaries [6]

In this chapter the ways in which the fluid flow interacts with the boundaries will be examined. Each manifold has two boundaries, and the best way to understand how the waves interact with these boundaries is via "the Method of Characteristics" as it models the waves behaviour along a characteristic line.
There are many boundaries that can describe the fluid flow in the engine manifolds, in this chapter we will describe the most common ones, and are going to be based on the derivations obtained by [1].

The most important aspect of the boundary theory is the fact that *in each and every boundary the flow is considered to be* ***quasi-steady****.*

The assumption of quasi-steady flow implies two things:

1) The physical size of the boundary can be neglected when compared with the length of the pipe connected to it and that the equations of steady flow can be applied locally to this infinitesimally small region.
2) It also implies that at the boundary the rate of change of properties with distance is very much greater than the rate of change with time:

$$\frac{\partial i}{\partial x} \gg \frac{\partial i}{\partial t}$$

Where 'i' is any property such as pressure, velocity etc.

The basic idea is that at a boundary there will be an arriving characteristic which will be called λin and a departing characteristic that will be called λout. Therefore the objective will always be to see how the two characteristics relate to each other at the boundaries. If the boundary is at the right hand side of the manifold then as it can be seen from *figure* 2, the arriving characteristic will be λ and the departing characteristic will be β so in this case λin = λ and λout = β. If the boundary is at the left hand side the opposite will happen and then λin = β and λout = λ.

The main types of boundaries are inflow from an infinite reservoir (constant principles), open end (exit from a duct), closed or partially closed end.

- **Closed end**

At a closed end the velocity of the gas is zero, thus by setting U = 0 in equation

$$U = \frac{\lambda - \beta}{k - 1}$$

We obtain:

$$\lambda_{in} = \lambda_{out}$$

A wave reflected by interaction with a closed end is the same sense as the incident one, meaning also that a pressure wave is reflected as a pressure wave of equal magnitude. This occurs because there can be no flow at the closed end and hence the incident wave must be sent back into the pipe. For example, if there is a compression wave heading towards a closed end, the reflection of it results in an increase in the pressure acting on the closed end; this is caused by the reversal of the momentum of the wave.

- **Open end**

The main feature at an open end is that the pressure will be the same as the ambient pressure at the exit of the pipe, therefore $A = \frac{c}{cref} = (\frac{p}{pref})^{\frac{k-1}{2k}}$ is equal to 1 and from equation

$$A = \frac{\lambda + \beta}{2}$$

one obtains:

$$\lambda_{out} = 2 - \lambda_{in}$$

Physically, the effect of such a boundary is to invert an incident wave; thus an incident compression wave is reflected as a rarefaction wave and vice versa. It must not be neglected that this inversion results in an increase in the velocity at the end of the pipe.

- **Flow through a valve**

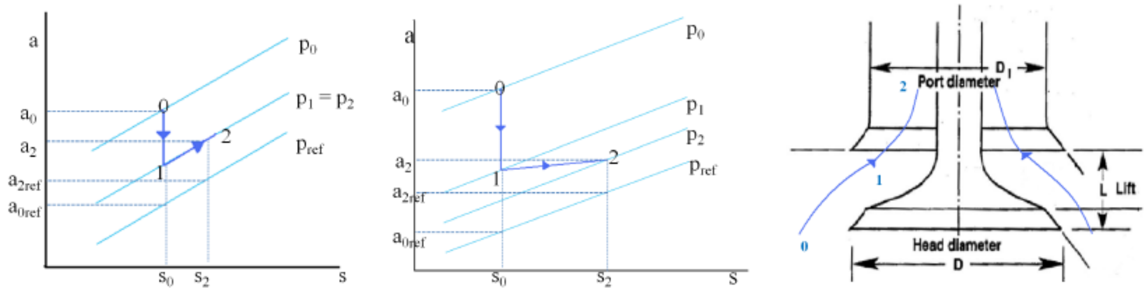


Figure 3: a-s diagram for subsonic and sonic flow and valve schematic [1,3]

There are two possible flows through the valve, inflow and outflow, ideally for the inlet and the exhaust valve respectively.

The flow through the valve plays a crucial role in the design of an internal combustion engine. In [3] there is a qualitative discussion on how the design of the inlet and exhaust valves affect the volumetric efficiency of the engine. Valves' primary role is to transport gases in and out of the cylinder, and possible ways to maximize this is either by increasing the valve area, achieved by increasing the valve lift, or by increasing the time the valves are open.
There are however limitations to this and a fine explanation is given in [3]. Put briefly, above a certain level there is no point in lifting the valve any more, as a free jet will form. The maximum lift is determined by the valve port area and is approximately 1/4 of the valve diameter. Moreover, there is a limit of how long the valves should stay open, as back flow into the inlet manifold can occur during valve overlap at TDC, especially at partial throttle. Possibly one of the biggest limitations of the performance of an internal combustion engine is sonic flow through the inlet valve, which occurs when the gas velocity entering the cylinder approaches the speed of sound.
The inflow and outflow through a valve will be analysed, in conjunction to the previously given *figure 3*.

We assume that the gas:

-is expanding *isentropically* through the valve seat and
-is expanding *adiabatically*, *irreversibly* and *at constant pressure* through the valve port into the exhaust manifold.

In detail, *figure* 3 shows the variation of entropy at the different points through the valve:

-Point 0 represents the flow conditions in the cylinder,
-Point 1 represents the flow conditions just after the valve and point 2 the flow conditions at the pipe inlet.

The gas exhibits isentropic expansion between points 0 and 1, both for subsonic and sonic flow. Point 2 is located at the pipe inlet, and will have the same pressure as point 1 for subsonic flow, whilst it will be at a lower pressure for sonic flow, as the flow will be choked.

The aim is to develop a relationship between points 0, 1 and 2 for both subsonic and sonic flow and afterwards a model for the flow through a valve will be developed here using the derivations obtained by [1].

Inflow through a valve

The flow conditions ahead of the valve and after the valve can be related using the continuity and energy equations.
Continuity across the valve, divided by reference conditions:

$$\frac{\rho}{\rho_{ref}}\frac{u}{u_{ref}} = \frac{\rho_t}{\rho_{ref}}\frac{u_t}{u_{ref}}\phi$$

Where φ is the ratio of ratio of the port to valve effective area. Taking as a fact that the flow is isentropic it can be obtained:

$$UA^{\frac{2}{k-1}} = U_t A_t^{\frac{2}{k-1}}\phi$$

Where we substituted:

$$\frac{\rho}{\rho_{ref}} = A^{\frac{2}{k-1}} \qquad \frac{\rho_t}{\rho_{ref}} = A_t^{\frac{2}{k-1}}$$

Furthermore the steady state energy equation across the valve is:

$$A_0^2 = A^2 + \frac{k-1}{2}U^2 = A_t^2 + \frac{k-1}{2}U_t^2$$

And so by combination of the mentioned equations we can obtain the following equation:

$$U^2 = \frac{\frac{2}{k-1}(A^2 - A_t^2)}{\frac{1}{\phi^2}\left(\frac{A}{A_t}\right)^{4/(k-1)} - 1}$$

Considering this equation we can examine the possible situations for a flow through a valve: subsonic and sonic (chocked) flow. **The supersonic scenario cannot be described by the method of characteristics and that is one of the disadvantages of this method.**

- For subsonic flow the pressure at the throat of the valve is the same as the reference pressure, therefore Av = 1 and the following simplified equation is produced:

$$U^2 = \frac{\frac{2}{k-1}(A^2 - 1)}{\frac{A^{4/(k-1)}}{\phi^2} - 1}$$

- For sonic flow the speed is equal to the speed of sound and that translates to:

$$\frac{U}{A_t} = \phi \left(\frac{A_t}{A}\right)^{\frac{2}{k-1}}$$

Or

$$\frac{U}{A} = \phi \left(\frac{A_t}{A}\right)^{\frac{k+1}{k-1}}$$

Finally the equation is produced:

$$\phi^2 = \left(\frac{k+1}{k-1} - \frac{2}{k-1}\left(\frac{A}{A_t}\right)^2\right)\left(\frac{A}{A_t}\right)^{\frac{4}{k-1}}$$

Which will be solved via the numerical method.

In order to take into consideration the pressure of the cylinder the following parameters will be used:

$$A_v = A\left(\frac{p_{ref}}{p_c}\right)^{\frac{k-1}{2k}}$$

$$U_v = U\left(\frac{p_{ref}}{p_c}\right)^{\frac{k-1}{2k}}$$

Outflow through the valve

The two categories in which the flow can be set are subsonic and sonic. The basic idea is that we always want to know what kind of flow there is through the valve and thus use the corresponding equations that model that flow.

Subsonic flow

The basic fact of the subsonic flow through a valve is that the pressure between points 1 and 2 on the diagram is constant: $p_2=p_1$.

The complete theory behind the subsonic flow is given in a nice way at [1] and will be not analysed here. Thus, using the continuity equation and according to [1], the basic equation describing the flow is:

$$\frac{\psi}{\pi}\left[\frac{2}{k-1}\left(\frac{1}{\pi^2}-1\right)\right]^{\frac{1}{2}}=\left(\frac{U}{1-\frac{k-1}{2}U^2}\right)$$

Where $\pi=\left(\frac{p_2}{p_0}\right)^{\frac{k-1}{2k}}$ and $U=\frac{u_2}{a_0}$.

Sonic flow

In sonic flow the main fact is that the velocity of the fluid is equal to the speed of sound, $u_1=a_1$. Similarly to previous, an equation is derived by Benson [1]:

$$\psi=\left(\frac{2}{k-1}\right)^{\frac{1}{2}}\left(\frac{U}{1-\frac{k-1}{2}U^2}\right)$$

There might be sonic flow through the valve port or in the manifold to the valve port.

On the first occasion, sonic flow through the valve port, the pressure at point 2 will be less than the pressure at point 1 and thus an equation will be shown describing the chocked flow and the loss of pressure. According to Benson [1]:

$$\pi=\left(\frac{p_2}{p_0}\right)^{\frac{k-1}{2k}}=\left[\psi\left(\frac{2}{k+1}\right)^{\left(\frac{k+1}{2(k-1)}\right)}\left(\frac{1-\frac{k-1}{2}U^2}{U}\right)\right]^{\left(\frac{k-1}{2k}\right)}$$

On the second occasion, sonic flow through the manifold, will occur when the flow in the manifold is equal to the speed of sound in the manifold, thus $u_2 = a_2$, and by taking the steady state energy equation between point 0 and point 1 a simple relationship can be obtained:

$$U=\frac{u_2}{u_0}=\left(\frac{2}{k+1}\right)^{\frac{1}{2}}$$

Chapter 5

Analysis of the Simulation program

In this chapter the MATLAB code concerning the isentropic flow in one cylinder engine will be analysed. The number of meshes used have an effect on the accuracy of the code till a point when afterwards more CPU time is needed. For this code, 30 meshes were used and the ratio of specific heats (k) is taken as 1.4.

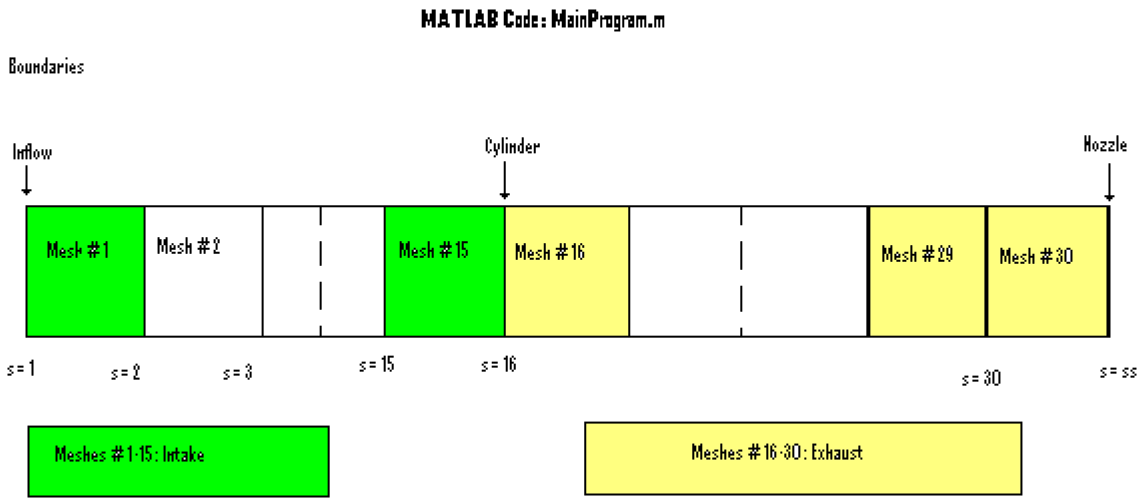


Figure 4: Schematic distribution of meshes

More specifically, the main parts of a 1 cylinder model are:

1) *Inlet Manifold* which has been modelled at first as an infinite reservoir with constant properties.
2) *Inlet duct* connecting the manifold to the inlet valve of the cylinder
3) *Inlet valve* of the cylinder
4) The *cylinder* itself
5) *Exhaust valve* from the cylinder
6) *Exhaust manifold* which has been modelled as a nozzle.

In this simulation, based on the methodology provided in [1], one complete engine cycle is analysed at a constant speed of 33.3 rev/s (~2000 rpm).

Analysis of one complete engine cycle

The part of the engine cycle that is of interest for this simulation starts from 100 degrees from Top Dead Center (TDC). All degrees mentioned here will be from TDC. At **136 degrees** the exhaust valve is opening (EVO) and so the exhaust stroke starts. The pressure in the cylinder is decreasing from the maximum value of 3.6 bar and at **261 degrees** the exhaust valve is starting to close. Due to the inertia of the valve, this movement cannot be instant, taking some time to be done. At this time now the piston is moving upwards and that results in a slight increase of pressure as also the exhaust gas is moving out of the cylinder. At **346 degrees** the inlet valve opens (IVC) and we can see the pressure rising momentarily at the end of the exhaust stroke due to the piston approaching scavenge TDC. During this period of time there is overlap of the two valves and so some pressure is lost due to fresh air escaping through the exhaust valve. At **367 degrees** the exhaust valve closes (EVC) and so the cylinder is being filled with air from the inlet valve. However, due to the piston movement – which has now overcome the TDC and is moving downwards to the BDC- we can see the pressure yet decreasing till 390 degrees approximately as the piston movement is sucking the air. After this point the pressure in the cylinder increases and at **474 degrees** the inlet valve starts to close, effectively closing completely at **582 degrees**. At that time the piston is moving upwards, compressing the air inside the cylinder. Each point mentioned is being showcased in the following diagram:

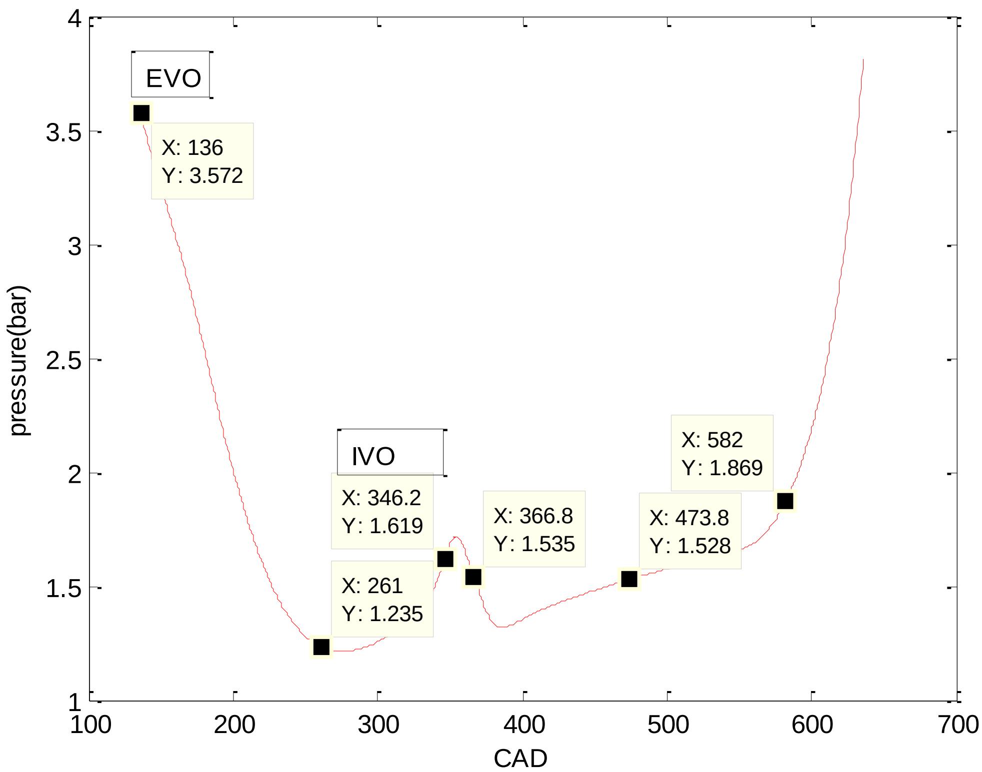


Figure 5: Analysis of an engine's cycle

The cylinder is part of a four stroke engine, has a diameter of 0.125m, stroke of 0.13m and a connecting rod of 0.273m. The nominal compression ratio based on swept volume is 14.

For this part of the code, we consider as given facts:

1) The cylinder pressure and temperature at exhaust valve opening, Pcr = 3.593 bar and Tc = 762 K respectively.
2) The exhaust valve opens (EVO) at 136 degrees from top dead center (TDC), whereas the inlet valve opening (IVO) is at 346 degrees from TDC.

The code consists of a main part where the basic data is provided, concerning the initial state of the fluid, the valve timing and area opening. Moreover, based on the fact that this method works by setting left and right boundaries, the λ and β characteristics are calculated and based on the values inserted in the boundaries. For this specification of the simulation code we will be using an *inflow* boundary for the inlet manifold, a *cylinder* boundary and a *nozzle* boundary for the exhaust outflow.

More specifically, the boundaries were set via subroutines which can be seen in the code provided in the appendix section. The *inflow* subroutine simulates the flow from a reservoir to a pipe, whereas the *nozzle* subroutine describes the outflow from the cylinder to the environment.
The *cylinder* subroutine is more complex, calculating the inflow and outflow from the valves according to the crank angle position. Another subroutine inside the cylinder subroutine is used called *valve*, calculating whether there is inflow or outflow from a specific valve, or whether there is subsonic or sonic flow. Finally, the *massrec* subroutine is used for calculating the air mass coming into the cylinder.

According to theory provided by Benson [1], we should be using left and right boundaries for simulating the flow into our engine, using arriving and departure characteristics. That means that we should be using an even number of boundaries. The basic boundaries would be the inflow boundary, the cylinder and the nozzle boundary for the exhaust. Based on the theory we should also be using another boundary and that is why at first we have opted for an inflow-nozzle-cylinder-nozzle specification. However, having an extra nozzle between the inflow and the cylinder boundary meant that despite being according to Benson's theory the results were not so good.

More specifically, the major problem was witnessed in the velocity diagram. After the inlet valve closed, the velocity at that point appeared to remain exactly the same and constant at its highest value as was the case when the valve was fully open. This means that there was flow towards the inlet valve, 'entering' a closed wall. That is not correct according to continuity equation and moreover there seemed to be no fluctuation in the velocity diagram, which indicated that there was not any pulse wave effect at all. Same problem was showcased in the pressure diagram, where the pressure after the inlet valve closed appeared to be constant, without any fluctuation.

In order to overcome these kinds of problems, we improvised, removing the first nozzle boundary and having only the inflow-cylinder-nozzle specification. We used two left boundaries, inflow and cylinder, and one right boundary, the nozzle. As has been explained earlier in this thesis, the objective is always to see how the two characteristics (the arriving characteristic called λin and the departing characteristic called λout) relate to each other at the boundaries. In our case there is a departing characteristic from the inflow boundary and another departing characteristic from the cylinder boundary, leaving the nozzle to have the only arriving characteristic. Whereas theoretically that is not correct, however the results are very promising and physically well perceived and understood, as will be showcased in the results section.

Simplifying assumptions

We have made a simplification in order to analyse the pulses and that is the consideration of air as the only fluid. So, there is no fuel injected, just air all the way through the engine cycle.

Another assumption made is that we consider as a fact that the engine cycle starts from a fixed point from a previous cycle, with fixed pressure and temperature at the EVO point. These values can be changed by the program user, but in order to alter them, the engine speed should be altered too.

The flow is considered as geometrically one-dimensional, implying that all fluid properties are uniform over each cross section of the passage, and that changes in cross-sectional area take place very slowly. The viscosity and thermal conductivity of the gas will be neglected and that means the flow is considered as isentropic.

We also consider the equation of state to be that of a perfect gas, the gravity effects are negligible and the fluid will be treated as a continuum. The governing equations are briefly given for reference:

Continuity

$$\frac{\partial \rho}{\partial t} + u\frac{\partial \rho}{\partial x} + \rho\frac{\partial u}{\partial x} + \frac{\rho u}{F}\frac{dF}{dx} = 0$$

Momentum

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + \frac{1}{\rho}\frac{\partial p}{\partial x} + G = 0$$

Energy

$$\frac{\partial p}{\partial t} + u\frac{\partial p}{\partial x} - a^2\left(\frac{\partial \rho}{\partial t} + u\frac{\partial \rho}{\partial x}\right) - (k-1)\rho(q + uG) = 0$$

$$G = \frac{1}{2}u|u|f\frac{4}{D}$$

An initial first demonstration of the code will be done in order to see the difference in the cylinder pressure between a naturally aspirated and a supercharged or turbocharged engine (the way in which the engine is turbocharged or supercharged will not be negotiated at this point, we just want purely the result of the increased pressure and temperature). The cycle starts during the power mode (piston heading down to BDC), while the pressure is decreasing. At 136 degrees after TDC the exhaust valve opens (EVO) and at 346 degrees from TDC the inlet valve opens (IVO) as has been already mentioned. During this period there is overlap of the two valves and thus the 'scavenge' effect is produced, where the pressure increases momentarily in the cylinder once the inlet valve opens, due to piston movement.

So, firstly we consider a NA cylinder with ambient intake pressure (1 bar) and temperature (300K) and then, by Benson, intake pressure of 1.65 bar and temperature of 360K. On purpose the peak pressure and temperature points (at the EVO point) have been left the same as a reference point. The results are given below:

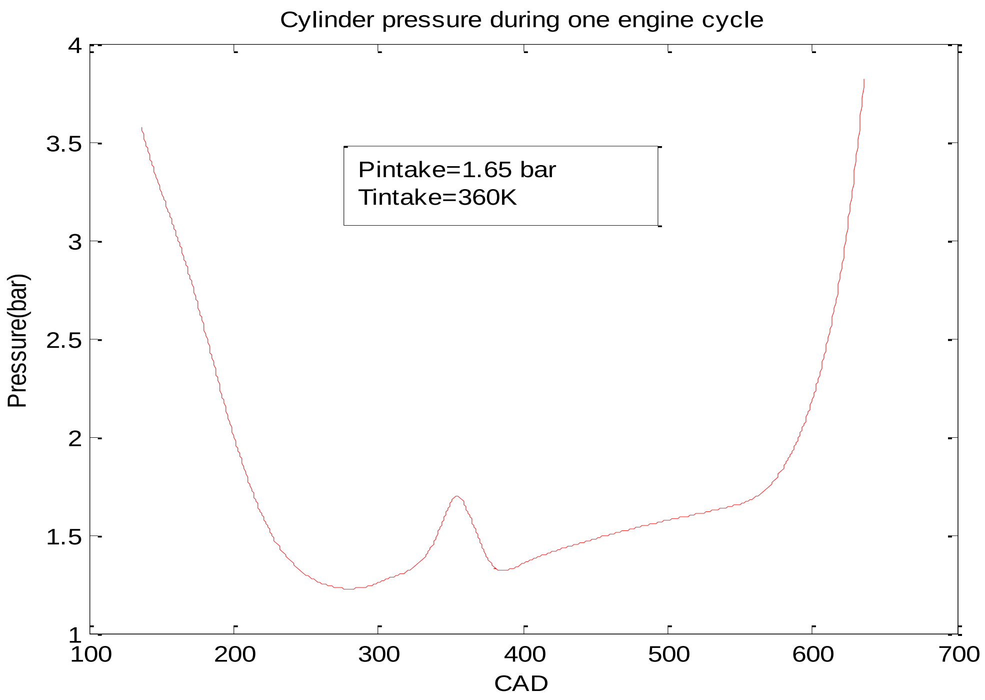


Figure 6: Cylinder pressure during engine cycle for specific pressure and temperature

And for the naturally aspirated:

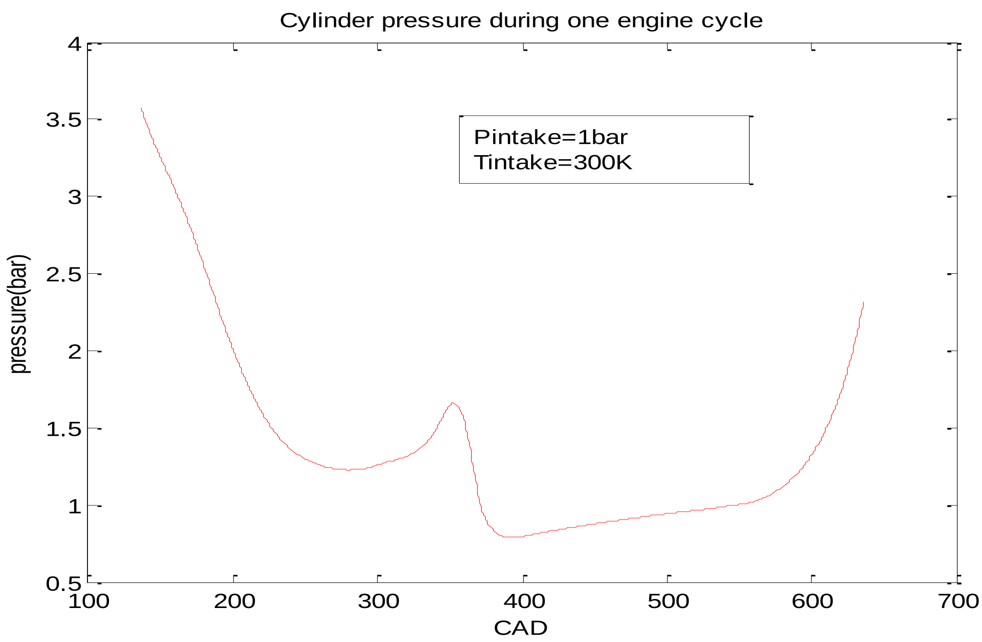


Figure 7: Cylinder pressure during engine cycle for atmospheric pressure and temperature

By comparing the two diagrams we can see on the second diagram the decreased pressure after the exhaust valve has opened. Due to the reduced intake pressure, the overall pressure is -not surprisingly- smaller than the first's diagram. That can be seen also by the end pressure of the cycle; in the second diagram we have let the peak point of pressure as it is as a start point and by inserting fresh air that point cannot longer be reached.

From this point and onwards we will be considering as intake pressure 1.65bar and temperature 360K.

Chapter 6

Code running and Results

In this chapter the results obtained by running the code will be showcased and explained. The main MATLAB code that was used is the MainProgram.m, which can be found in the appendix. The main program consists of numerous subprograms and subroutines, mainly the boundary subroutines. The basic structure consists of the following steps:

```
Step 1: Initial data
Step 2: Constants for specific flow conditions
Step 3: Reference Values
Step 4: Initial conditions at z = 0
Step 5: Determining value of delta Z using stability criteria
Step 6: Determining the Riemann variables at every mesh point using
Benson's equation
Step 7: BOUNDARY conditions
Step 8: Determine pressure at every point
Step 9: Construction of position and state diagrams
Step 10: Results and Figures
```

The code is self-explanatory, so the reader is prompted to follow the code closely in order to understand its function.

In total of 30 Meshes, we have devoted 15 Meshes to the Intake pipe, from inflow of air at the start of the pipe to the intake valve of the cylinder. The intake length value has been set at 0.609m.

Firstly, we examine the velocity of the fluid in the first three meshes of the intake pipe.

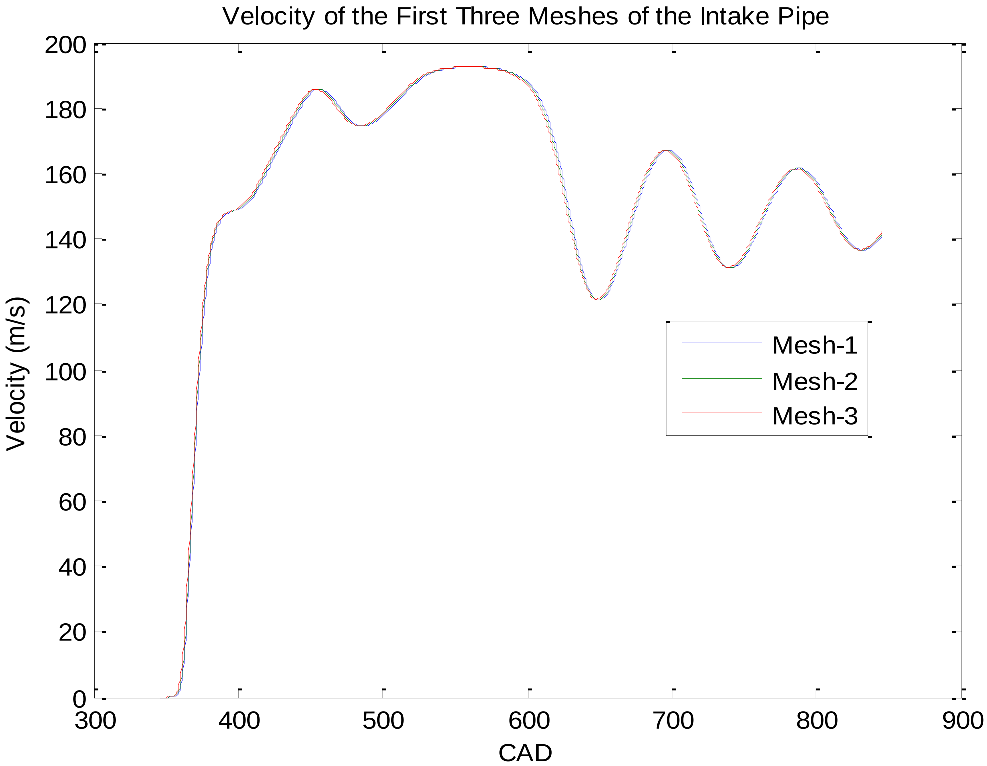


Figure 8

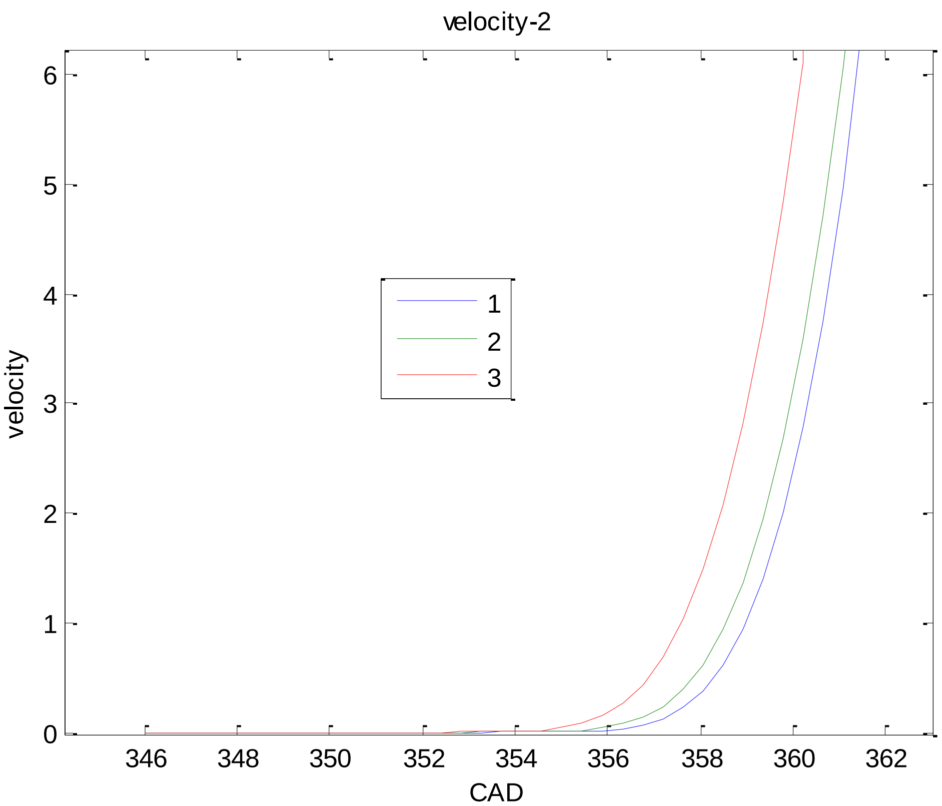


Figure 9: Detail of velocity diagram

The velocity starts to evolve after the intake valve opens at 346 CAD. In the magnification shown in the second diagram (*figure* 9) it is clear that the velocity at mesh #3 starts first to evolve, afterwards at #2 and finally at #1. This is the notion for all the meshes in the inlet pipe from the last one (before entering to the cylinder, mesh # 15) to the first one (start of inlet pipe, mesh #1). This will be shown in the next *figure* (Velocity at various Meshes of the Intake Pipe).

As the intake valve is opening at 346 degrees from TDC, a rarefaction wave is produced starting from the intake valve and heading towards the start of the intake manifold. This wave is transmitted leftwards with speed c-v, where c is the speed of sound and v the local fluid velocity. The result of the wave is a decrease in the pressure of the intake pipe (hence rarefication wave) and an increase in velocity as the fluid moves from the pipe to the cylinder through the intake valve. Because the wave is produced at the end of the inlet pipe, the last meshes located there will showcase first the increase in speed and the norm will be transported gradually (after t = $\frac{DX}{c-v}$) to the start of the inlet manifold.

Looking at the first diagram, *figure 8*, we can notice that as the pressure wave reaches the pipe entry, the velocity is increasing till approximately 450 degrees and afterwards is starting to decrease. This is due to the wave reaching the air intake being reflected from the open end as a compression wave in the opposite direction with speed c+v. This leads to an increase of pressure and a decrease of velocity till approximately 490 degrees where the compression wave now is being reflected again from the open end (the intake valve is still open at that point) as an expansion wave, resulting again in an increase in the velocity of the meshes of the pipe at 580 degrees. At that point the intake valve closes (582 degrees). The expansion wave is now being reflected in the same manner from the closed end and that results in a drop of pressure till 650 degrees in which point the wave now reaches the opposite open end of the air intake and is being reflected back in opposite manner, increasing the velocity. After that there is a repetition of the wave movement and reaction until the engine cycle finishes. These waves will still have an effect to the next engine cycles, affecting their performance. In this thesis though a simplifying assumption is made, such as waves from the previous cycle don't exist and therefore don't affect our understanding of the wave theory in the intake manifold.

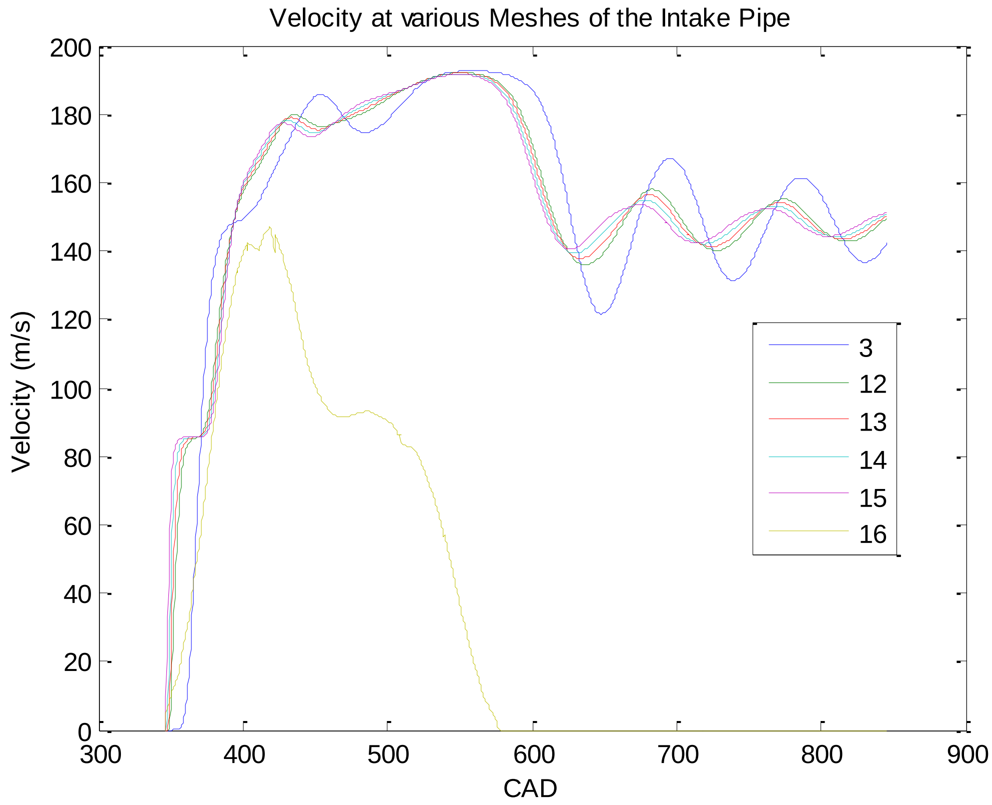


Figure 10

Same conclusions can be made from the above third diagram, *figure* 10, where we can see the evolution of speed across the meshes #3, 12, 13, 14, 15, 16. Apart from mesh #16 in which the velocity does indeed decrease gradually to zero as should be the case, the other meshes' velocity fluctuates due to expansion and compression waves. What we may see is that the velocity evolution in mesh #3 has a time deficit as it follows the same norm to the other meshes, except after time dt which is the time needed for the waves to cover the distance from the meshes near the cylinder to the air intake. Apart from this delay we can also notice that the amplitude of mesh #3 is greater than the rest, especially after the inlet valve has closed. The same behaviour can be seen across all the meshes from #15 to #12: *the amplitude after IVC at a mesh nearer the air intake is greater than the immediately following mesh towards the cylinder*. This may be explained by the presence of the open end, where inflow of air takes place.

Pressure figure for meshes # 1, 2, 3 (start of the intake manifold)

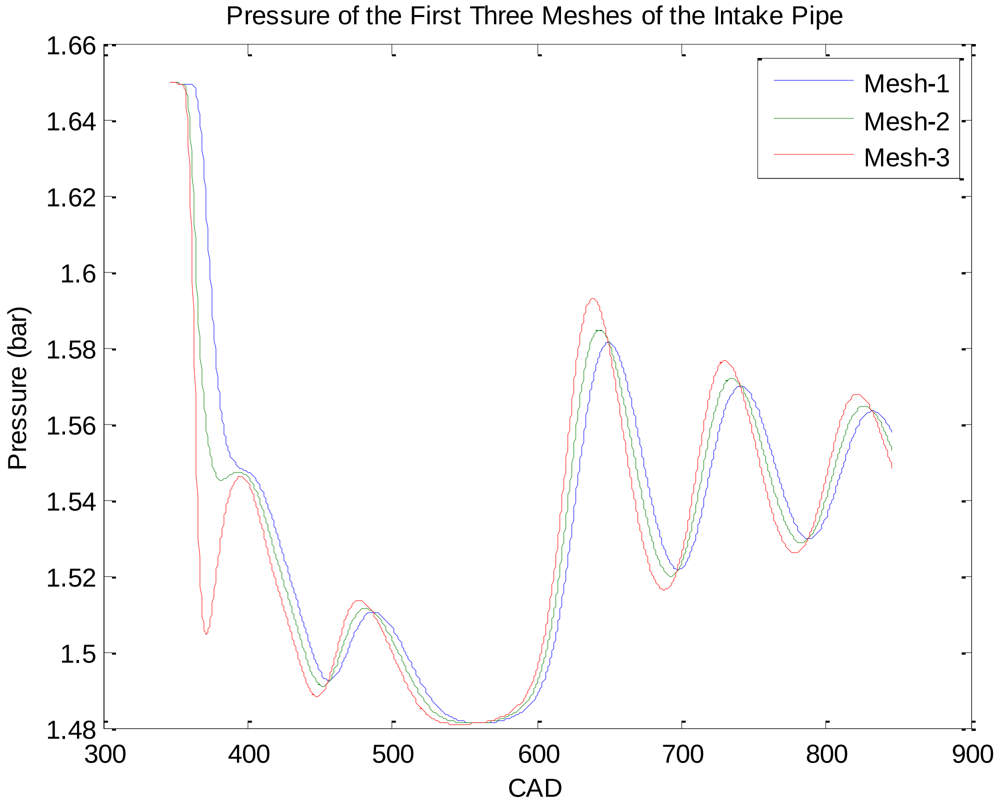


Figure 11

With the initial conditions (z=0) at the intake pipe set to be at 1.65 bar the pressure evolution starts from this point. At 346 degrees the intake valve opens and an expansion-rarefaction wave is produced heading towards the air intake. Hence, as explained previously, the pressure decreases from right to left, so from mesh #3 to mesh #1 as shown in the above *figure*. The pressure drops till the wave is reflected from the open end of the pipe as a compression wave which results into the pressure starting to increase around 450 degrees. Afterwards the compression wave reaches the open intake valve where part of it is transmitted inside the cylinder resulting in a desired increase of pressure and part of it is being reflected back as an expansion wave, resulting again to a drop of pressure along the pipe towards the air intake. At 582 degrees the same procedure takes place but now there is a change in the boundary, the inlet valve is closed and the boundary is considered as a closed end, reflecting every incident wave in the same manner and not the opposite as was the case for the open end. After that, the pressure is starting to increase as there is now again a compression wave.

However, we cannot benefit from that increase of pressure as the valve needs to stay closed in order for the compression stroke of the engine cycle to take place.

Two-pipe/two-valve system

An idea in order to overcome this fact would be to have two air ports for the same valve: the one port (no.1) to be as usual and the other port (no.2) to be like a cave, being connected to the end of the port-1 and the inlet of the valve. The idea is to achieve a supercharging effect inside the port-2 leading to a closed end. Once the air is trapped there through a second small valve (as to not having an effect to the rest of the system), and due to the captivated compression waves (the timing will be an important and crucial factor as we would need to captivate a compression wave that has been reflected from the closed end of the main valve and after that the small valve to close as to trap it inside the small tube, port-2) the pressure would rise inside the tube as the compression waves will be reflected in the same manner, as now there are two **closed end boundaries** [*figure 12*]. This procedure needs to take place as long as the main valve is closed, in order for the compression waves to be created. Actually, it could take place just after the intake valve is closed, during which period we don't exploit the pressure increase due to the reflected compression waves. It will be crucial though the second valve to be open at a point when compression waves are heading inside the closed end tube, and closed when the same reflected waves are coming back to the valve [*figure 13*]. This means that the length of the pipe-2 will share an important role here too, as does in the variable intake length systems. Probably on first hand there should be another way of 'blocking' the waves -reflected from the closed main valve- from entering the intake manifold and thus losing the compression effect as was seen on the previous diagram by not exploiting the increase of pressure once the valve is closed. The second valve –with the now increased air pressure- should open exactly when the main intake valve opens. Of course, in order not to have backflow from the increased pressure of the pipe to the intake manifold the port that connects them should still be closed (that port should only be open when we need to fill up the pipe). In that way there will be two flows towards the intake valve, the normal flow through the intake manifold and the second flow increased pressure from the second pipe. We may see an advantage as we exploit more the compression waves and thus we increase the pressure of air coming into the cylinder and hence the volumetric efficiency. However, the idea for this system was based on our constant rpm model outlined in this thesis and in order to work efficiently we would need to have a variable length for the second pipe depending on the valve timing and engine rpm. Even though the system is complex and adds to the difficulty of manufacturing, as we need to introduce that around the intake valves, the technology for controlling the lengths of the intake pipes exists and could apply here too. Another crucial factor needing attention is the potential pressure loses from the ports-second valve. Of course it will add to the cost but there may be some good gains by exploiting the increase in pressure by the pure compression waves; that is why this system is proposed and not the introducing of a valve at the left boundary of the intake manifold in order to be transformed to a closed end. In the second occasion there would be a mix of compression and expansion waves, hence reducing the potential. A future thesis or research may be done for this system to investigate whether there is going to be static waves inside the second pipe and if the gains –if any- are adequate in order to use this system or not.

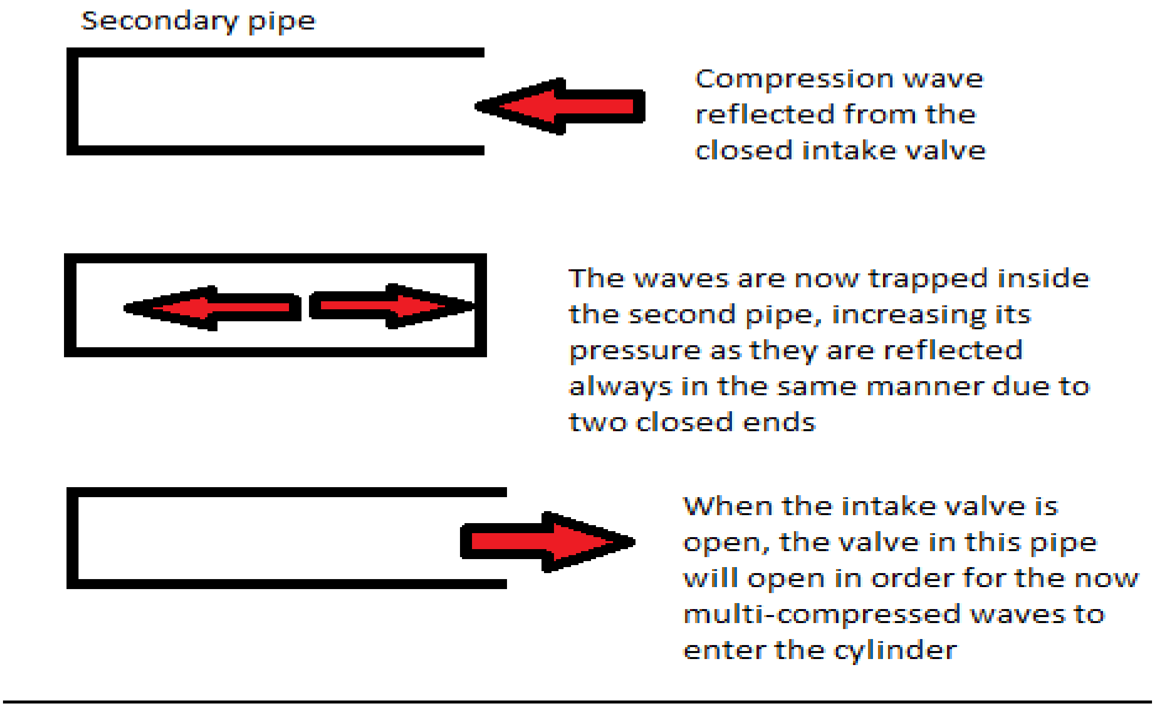


Figure 12: basic idea behind the two valve/two pipe system: secondary pipe

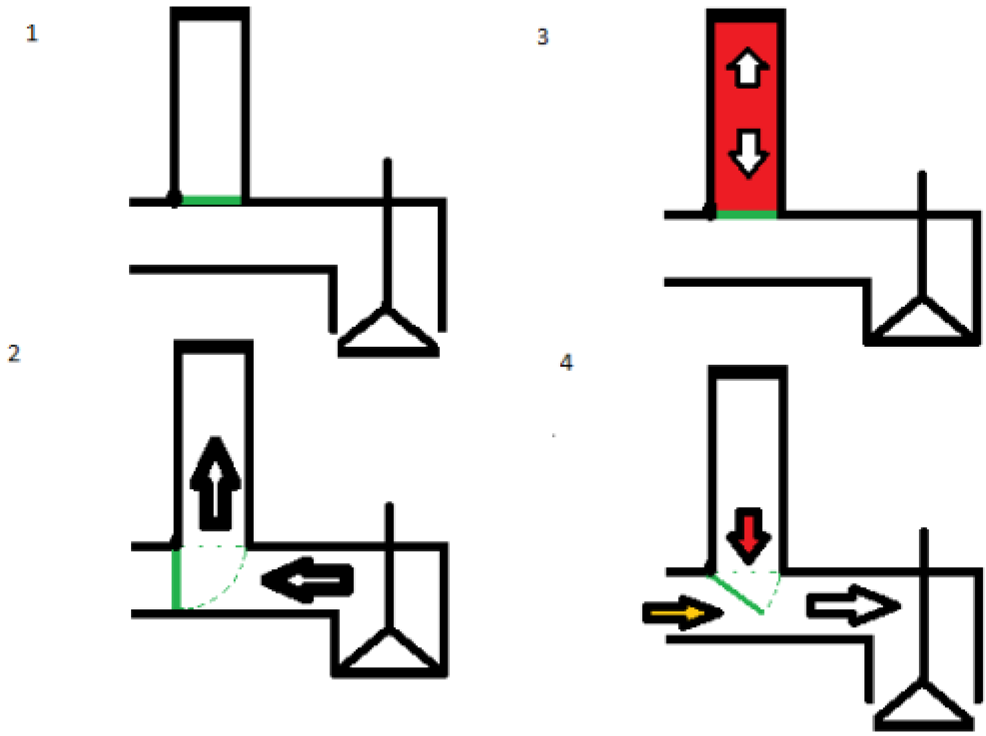


Figure 13: Function of the two pipe/two valve system

Phase 1: First production of expansion waves due to valve opening, the port is closed.

Phase 2: The port has opened in order to trap the reflected compression waves from the closed valve.

Phase 3: After sufficient time and before the compression waves are being reflected back to the main pipe the valve #2 closes and traps those waves where we can exploit the two closed boundaries effect.

Phase 4: The second valve opens when the main intake valve opens and so we can exploit the possibly increased pressure of the trapped waves inside the secondary pipe.

Complete pressure figure

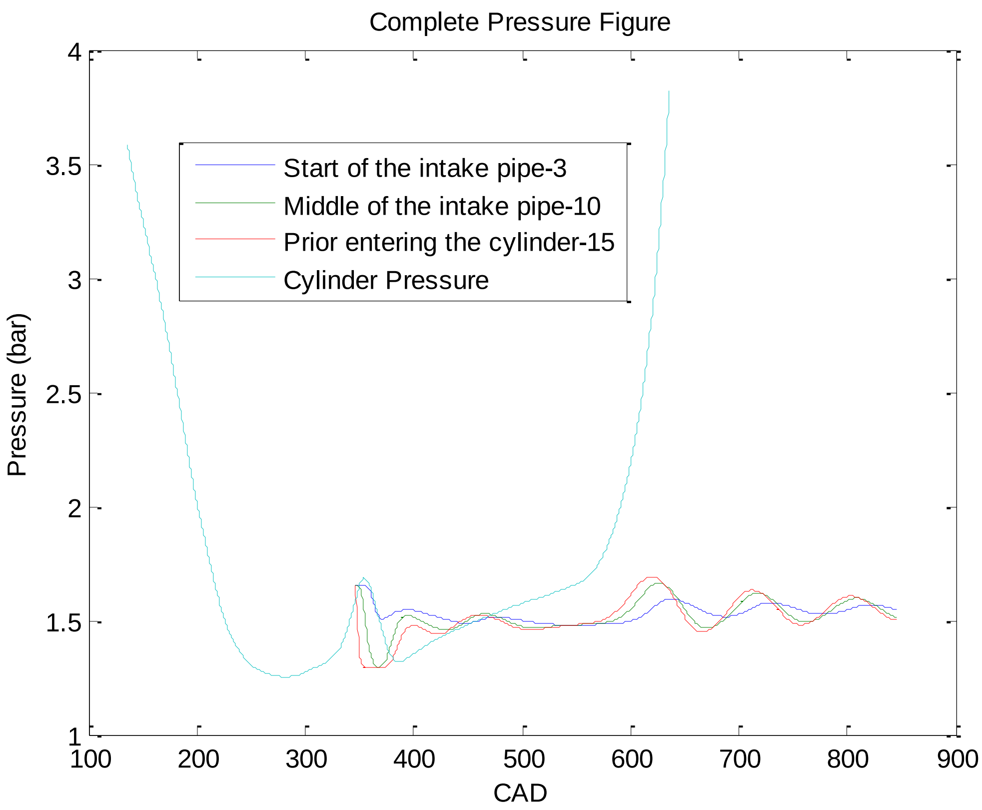


Figure 14

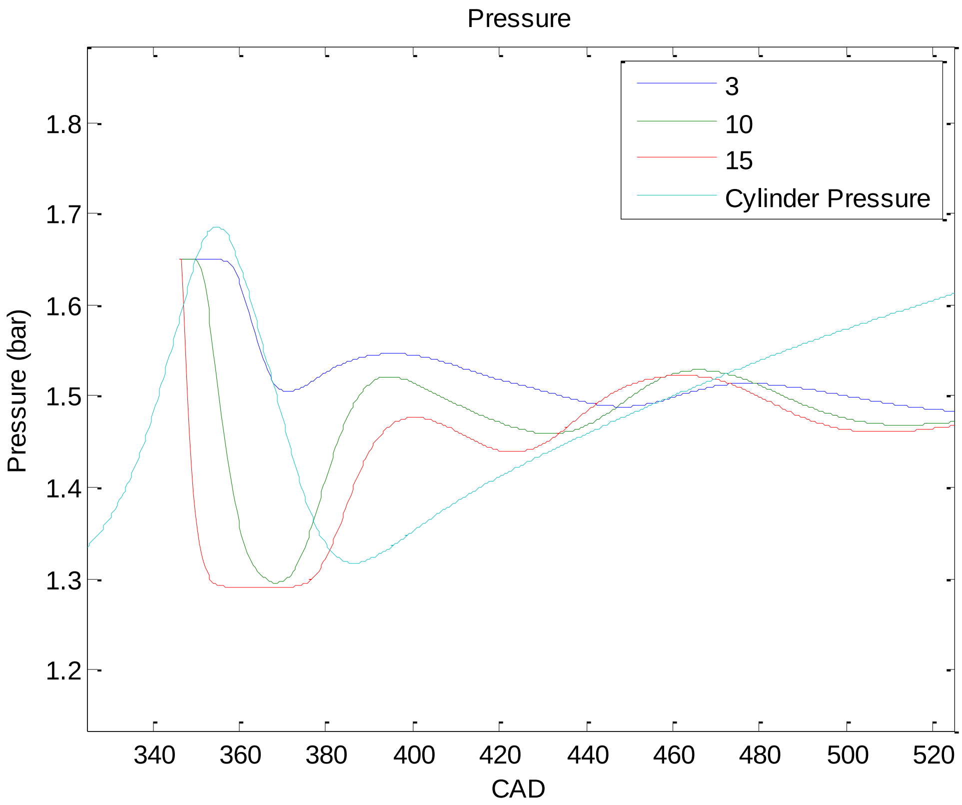


Figure 15: Detail of the complete pressure figure

In both of these *figure*s we can see the pressure inside the cylinder for one complete engine cycle and at the same period of time the pressure evolution for a mesh at the start of the air intake (mesh #3), for a mesh at the middle (mesh #10) and for a mesh at the cylinder inlet (mesh #15).

When the intake valve opens the pressure at every mesh drops from right to left-from the cylinder inlet to the air intake at the manifold. As explained above, due to the expansion wave going upstream the air flow, pressure at the mesh #15 drops instantly (we have set for the whole pipe the initial pressure to be 1.65 bar) , followed afterwards by mesh #10 and #3. The pressure rises momentarily inside the cylinder due also to piston approaching scavenge TDC. The expansion wave produced is being reflected at the open end of the air intake and is being reflected as a compression wave towards the valve. This the focal point of the theory of variable intake length pipes. The optimum strategy is to use this reflected compression wave in a way that we fill up the cylinder with even more air, a mild supercharging effect. As it can be seen, the compression wave arrives as the inlet valve is still open, as the peak of the wave at mesh #15 is inside the open valve time period (346 to 582 degrees). That means our intake pipe length at 0.609 meters provides the cylinder with one big compression wave and a smaller one at 460 degrees, raising the pressure further. We can see this result in the increase of pressure inside the cylinder most notably at 400 to 500 degrees.

The aim is to vary the length of the intake pipe in order for as many compression waves as possible to reach the valve while it is open. In higher engine speed we would like to have an as sort as possible pipe length, in order for the expansion waves to be reflected at the air intake and return as compression waves towards the cylinder. However, as the valve is starting to close the boundary is transformed to a closed end, which means that all the waves are reflected in the opposite manner; so in a case where a compression wave is reaching the closing valve (a large decrease in a cross-sectional area can be considered as a closed end, it doesn't have to be necessarily a boundary wall) it will be transmitted inside the cylinder as a compression wave –which is beneficial- but will be reflected too as a compression wave heading towards the open end at air intake. If the valve has not been closed yet the now reflected expansion wave will be heading towards the valve, decreasing the pressure along its way to the cylinder and thus having the opposite effect to what we want to achieve inside the cylinder.

As a consequence, if we have a constant short pipe length in order to benefit from the compression waves we need to have a valve that closes almost instantly in order to 'block' the -reflected from the closing valve- expansion waves and not letting them get inside the cylinder. However, the inertia of the valve and its moving mechanism above other factors make that impossible.

The notion is to change the intake length accordingly to the engine speed but in this thesis we have made an assumption of constant engine speed.

A second idea to get the optimum, besides the **two-pipe/two-valve system** described above, would be to have a variable intake length in order to dismiss the closing valve waves effect; a very short pipe length when the valve is fully open (as for the waves to have time to be reflected and reach the valve as compression ones) and that length to be increased when the valve is starting to close and starting to be considered as a closed end. However, it is unclear as to in which point exactly the boundary is 'transformed' from an open end to a closed end. Future experimental research could be made in order to have a clearer idea as at which closing point of the valve the waves are no longer reflected in the opposite manner, but in the same.

Point of concern

From 360 to 375 degrees the pressure at mesh #15 is constant and the curve is flat. Of course it is logical for the pressure to drop significantly when the intake valve opens and afterwards to increase when the –now- reflected compression wave reaches the mesh.

3-D Carpets diagrams

Pressure Figure showing the evolution of pressure across the intake pipe with time as CAD

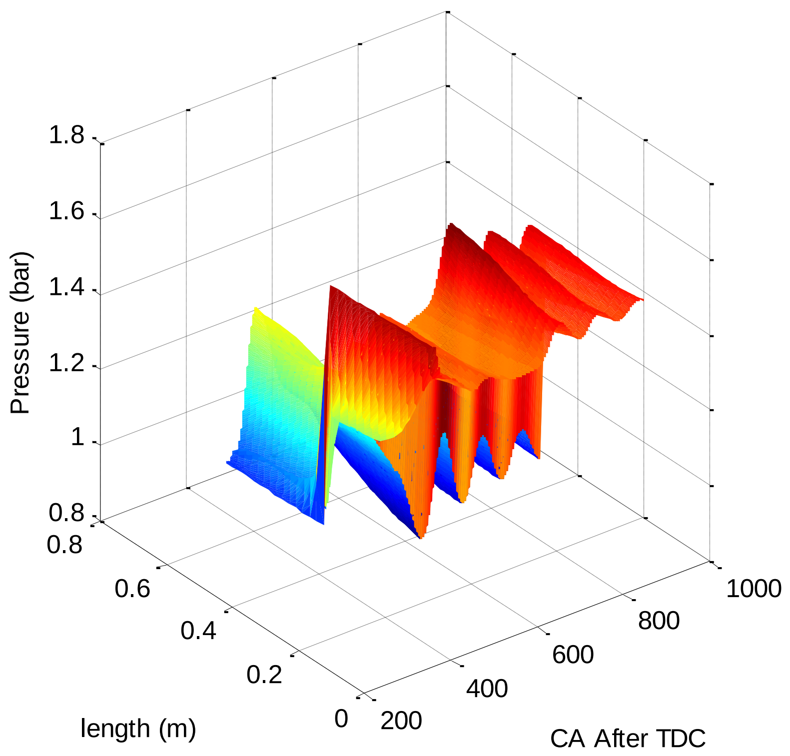


Figure 16 [Areas coloured in red are the areas of highest pressure value, as seen above]

Velocity figure showing the evolution of velocity across the intake pipe with time as CAD

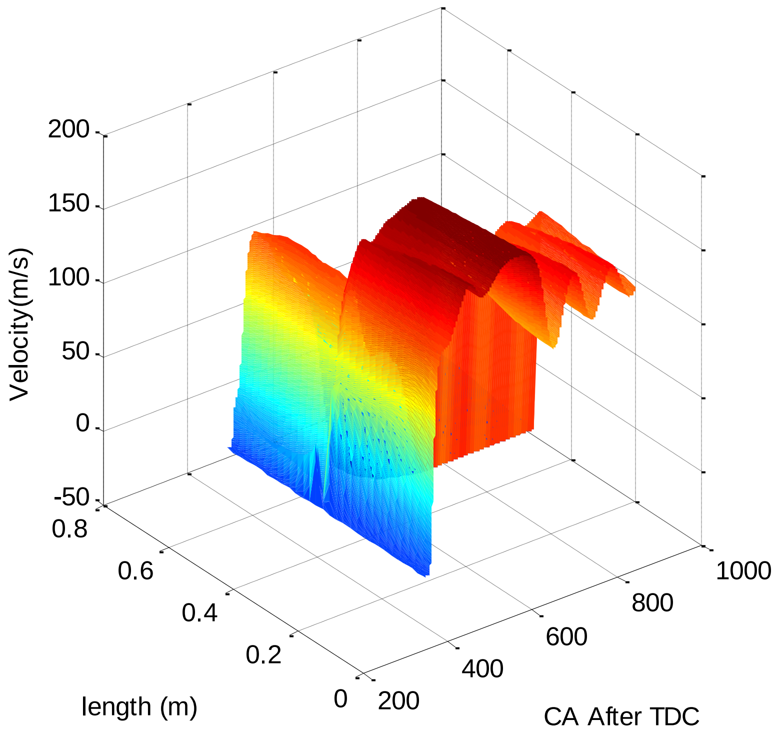


Figure 17 [Areas coloured in red are the areas of highest velocity value, as seen above]

Valve Diagrams

In this section the area-time diagrams for the inlet and exhaust valves for one complete engine cycle are presented.

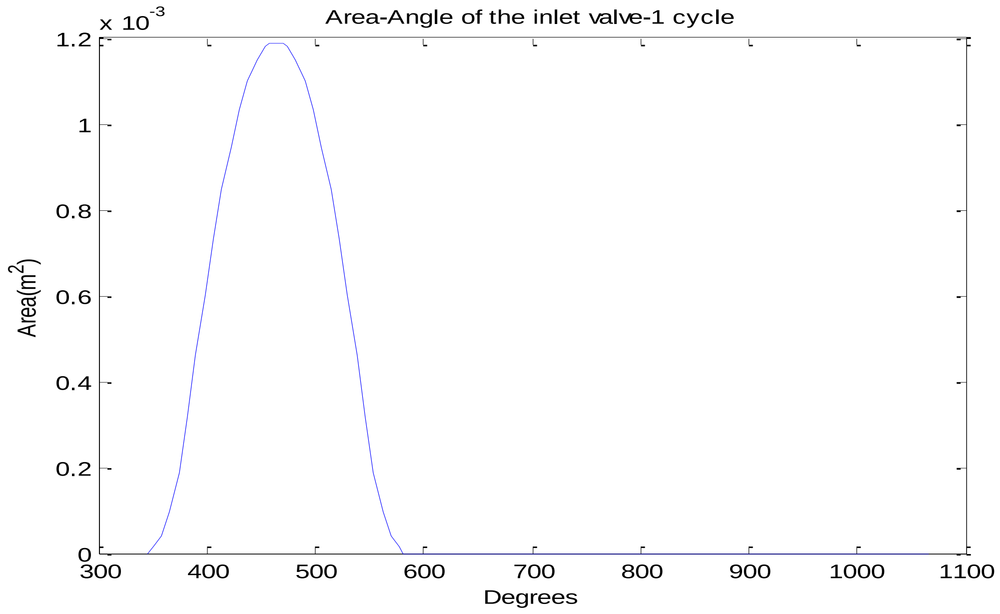


Figure 18

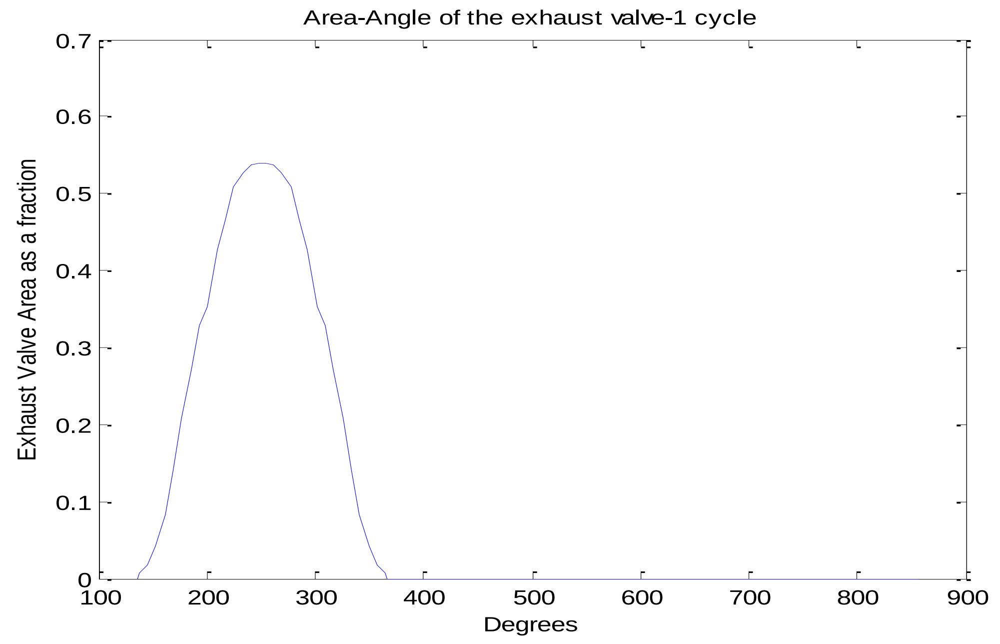


Figure 19

Where in the second figure, *figure 19*, the area is showed as a fraction of minimum (throat) to effective area of the valve.

Exhaust Manifold Pressure Figure

Two versions of the exhaust sub-model

Two versions for the exhaust model were created. In the first specification of the model, the exhaust was modelled with a nozzle connecting the exhaust valve to the environment. That produced good results, showing the fluctuation in the exhaust area due to compression and expansion waves, as can be seen in the *figure* below for the exhaust manifold inlet and outlet. The drawbacks though in this occasion are:

1) The cylinder and exhaust pipe are taken altogether as a package. That means that there is no strict distinction as to how many meshes are used individually for the cylinder or the exhaust pipe. For the purpose of this thesis though, we only consider them both as boundaries; we only need to know the end result of the boundary cylinder which is the outlet pressure.
2) Despite the results being correct according to the wave theory, physically the model is not as accurate as at the end of the exhaust pipe there is an open end to the environment.

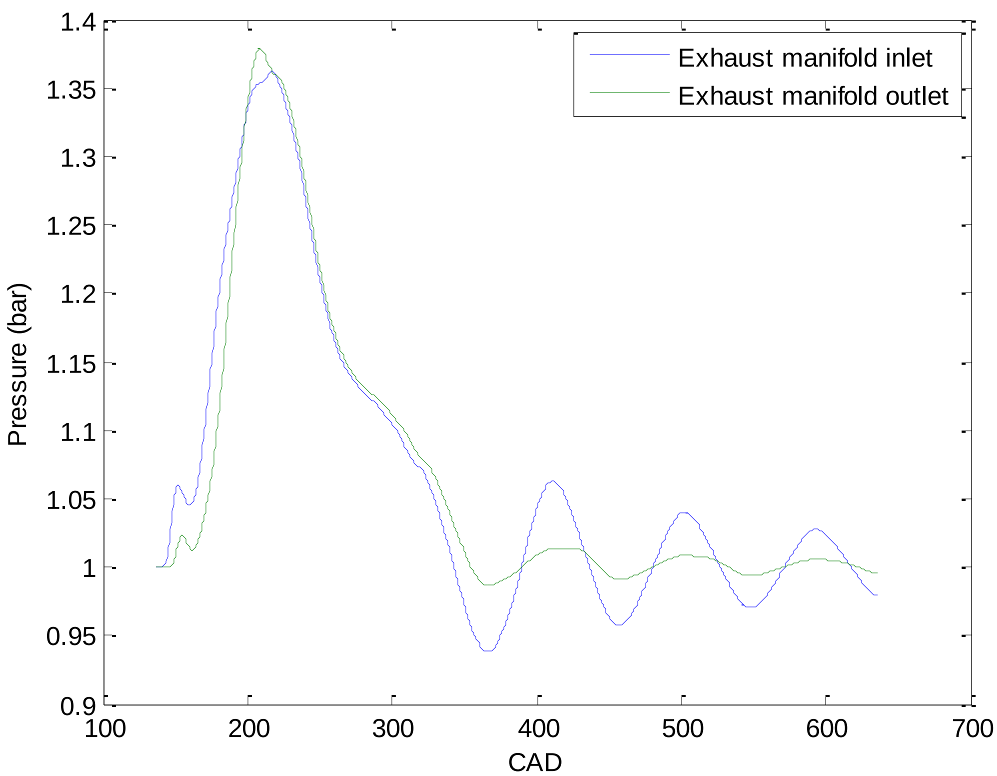


Figure 20: Pressure of the exhaust manifold-1st model

Following this analysis, another model was created, specifically for the exhaust system that is more physically accurate.

For this model we maintained the basic structure of the intake manifold and the cylinder, changing only the exhaust manifold boundaries. Instead of using only a nozzle, we opted to use an open end boundary to the environment, where the pressure is atmospheric, meaning that the pressure at the exhaust end is constant and equal to 1 bar; which is a simplification close to reality as exhaust gases have different pressure and temperature to the environment but the optimum situation is to drop the gas pressure as close to the environmental pressure as possible. However, with this simplification there are no waves produced at the exhaust end as the pressure at the final mesh is constant to 1 bar. On the other hand, the pressure at a point of the exhaust pipe does have fluctuation due to expansion and compression waves caused by the interaction of the gases to open and closed ends; this fluctuation however is much smoother with a mean value of 1 bar, very small amplitudes and is not necessarily fully correct according to wave theory.

Following the main logic of the method of characteristics- which is using left and right boundaries- the only way we can model this efficiently is by having two nozzles: one exiting the cylinder as the right boundary and afterwards, another one acting as a left boundary, followed by a right boundary towards the environment, an open end. The results are displayed in the following diagram.

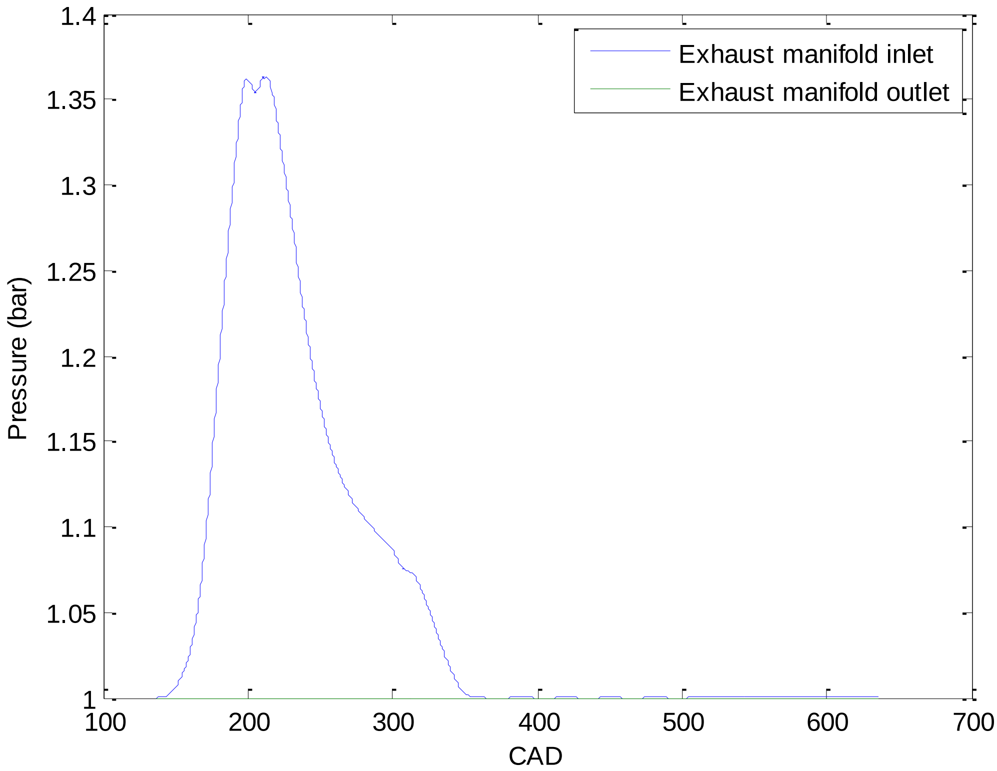


Figure 21: Pressure of the exhaust manifold-2nd model

It must be noted though that with either of the exhaust specifications of the code, all the other results are the same and do not display even a single difference. So, in all other parts of this thesis the first model will be used as we want to showcase the pulse wave fluctuation. The reader may also opt to use the second one, producing the exact results apart from the exhaust area, by inserting either of the following lines in the boundaries' section of the code, under the cylinder boundary. The two configurations are:

1st exhaust model:

```
beta(r+1,ss) = nozzle(lambda(r+1,ss),PHI,k); % exit from cylinder to a
nozzle
```

2nd exhaust model:

```
beta(r+1,17) = nozzle(lambda(r+1,17),PHI,k); % exit from cylinder to a
nozzle

lambda(r+1,21) = nozzle(beta(r+1,21),PHI,k); % from nozzle to open end is
the outlet pipe (exhaust)

beta(r+1,ss) = open(lambda(r+1,ss)); % exit to environment
```

In the following *figure* we can see again the pressure evolution with time as degrees of the engine cycle for the chosen model:

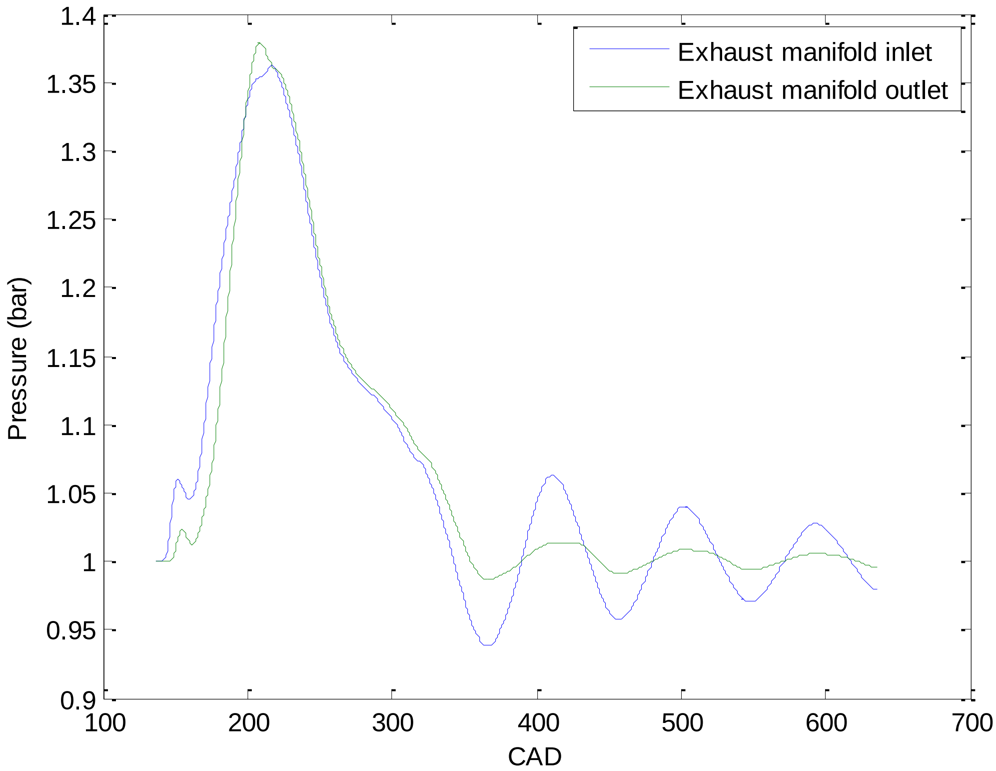


Figure 22: Final pressure exhaust model

The exhaust valve opens at 136 degrees from TDC; there is an increase in pressure in the intake manifold as the gases are now leaving the cylinder. Initial value of pressure in the exhaust pipe has been atmospheric. It can be observed from *figure 22* that there is delay in time of the increase of pressure at the exhaust outlet and that is due to the distance the compression wave has to cover in order to reach that point. The compression wave from the exhaust valve reaches the open end (end of the exhaust manifold, leading to the environment) increasing the pressure of meshes along its way and is reflected back as an expansion wave, hence the decrease of pressure at 220 degrees. The exhaust valve closes at 367 degrees and from this point and afterwards the pressure fluctuates due to the reflection of the compression and expansion waves, till reaching approximately the atmospheric pressure of 1 bar.

Position diagrams

In our simulation program there is a double-wave flow, meaning that there are both right-travelling and left-travelling waves. Those waves are physically reproduced as right and left characteristics and along these straight lines all the properties are constant. An area of the position diagram –or else physical plane- maps as an area in the state diagram as will be seen in the corresponding section.

The following position diagrams are time and space diagrams showcasing the lines-waves of constant properties, showing the path along which the perturbation properties and disturbances are propagated, and are analogous to the Mach waves (characteristic curves) of a steady, two dimensional flow.

As has been already mentioned there are two families of parallel straight lines, designated respectively by the symbols I and II, according to their direction in comparison to the motion of fluid. Type I lines are right-forward and type II are left-forward (opposite to flow motion). For a certain type of line the perturbation velocity and sound velocity are constant along the line, as is the slope of each line, depending on the speed of sound relative to the fluid. More specifically:

$$\left(\frac{dx}{dt}\right)_{I,II} = \bar{u} \pm \bar{c}$$

Where $\bar{u}$ is the velocity of the fluid and $\bar{c}$ is the speed of sound. The upper sign refers to the symbol I and the lower to the symbol II. The slope is also the propagation speed of the wave and is constant along a wave of a given family.

The time in these diagrams is in degrees of the engine cycle and the space in meters, displaying the length of the intake manifold. The difference in the two diagrams lies in the time axis. On the first diagram we consider the motion to start from moment zero as is always described in this way in the bibliography. In the second diagram, we consider the zero point to be at 346 degrees which is exactly when the intake valve is opening and thus when we actually have flow and waves. It must not be neglected that we have considered our engine cycle to be the ‘first’, meaning that we don’t actually take into consideration the effects of the waves produced in the intake manifold from the previous cycles. We have chosen to do this in order to analyse in full how the waves are indeed created and how they interact with the corresponding waves of the *same* engine cycle. Of course, in order to have a broader picture for the efficiency of an engine concerning its intake manifold we should include multiple engine cycles, however this will not be analysed in this diploma thesis, but a proposed way to do this is to add the next generated waves from the second cycle to our position diagram showcased here (expanding the time axe) and see the interaction between those two.

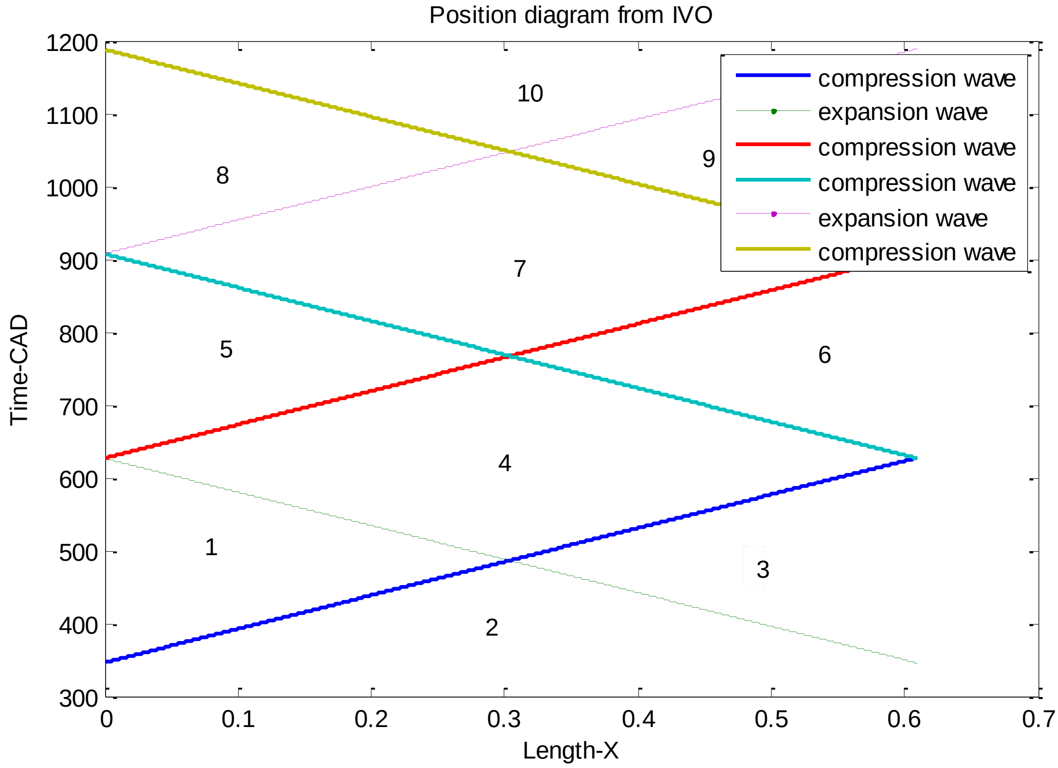


Figure 23: Position diagram

Here we will discuss and analyse the position diagram starting from the moment the inlet valve opens (above diagram). As we consider a steady situation in the intake pipe prior the IVO point, we can see the first generation of compression and expansion waves in the pipe, travelling rightwards and leftwards (blue and green line) respectively. After this point the waves are being reflected from the open and closed boundaries. The boundaries are located in the positions ‘0’ in the diagram, which is always an open end, thus always reverses the waves in the opposite wave, and ‘0.609’ which is the end of the intake pipe. This end includes the valve, which means that depending on the valve timing sometimes it acts as an open end, whereas other times as a closed end, from 582 degrees and afterwards, thus reversing the waves in the same manner.

It must be noted that the fluid properties are constant (pressure, velocity) between the lines, in all the numbered quadrilaterals. The properties change once the fluid gets past these lines and hence in the next quadrilateral we have fluid pressure and velocity that have been dictated by the propagated disturbances.

Thereby, from *figure 23* we have:

<u>Pressure Change due to wave propagation</u>

p2	**>**	**p1**
p2	**>**	**p3**
p4	**>**	**p3**
p4	**<**	**p1**
p4	**<**	**p5**
p4	**<**	**p6**
p7	**>**	**p5**
p7	**>**	**p6**
p7	**>**	**p8**
p7	**<**	**p9**
p10	**>**	**p8**
p10	**<**	**p9**

And correspondingly for velocity:

<u>Velocity Change due to wave propagation</u>

u2	**<**	**u1**
u2	**<**	**u3**
u4	**<**	**u3**
u4	**>**	**u1**
u4	**>**	**u5**
u4	**>**	**u6**
u7	**<**	**u5**
u7	**<**	**u6**
u7	**<**	**u8**
u7	**>**	**u9**
u10	**<**	**u8**
u10	**>**	**u9**

From the last table we can have a vague idea as to how the path line would form, as its slope is corresponding to the velocity of the fluid, thus for a greater velocity there will be a steeper curve.

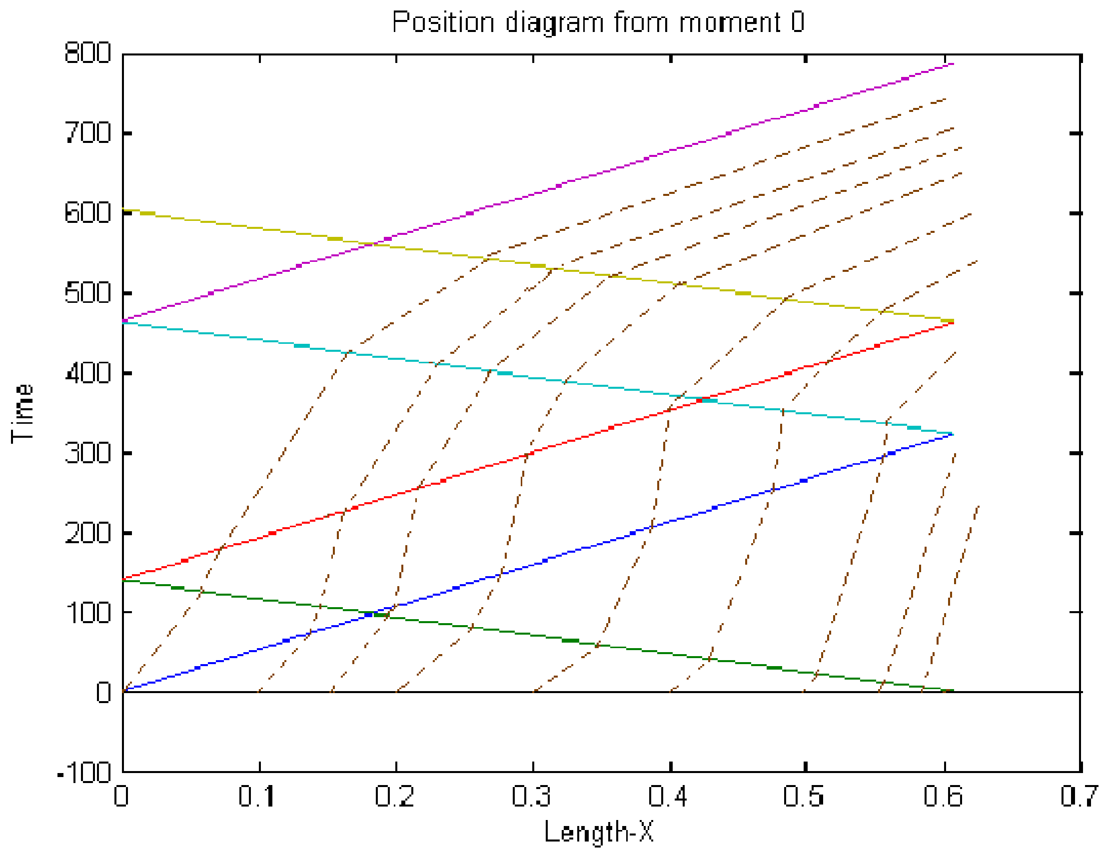


Figure 24: Position diagram from moment zero (346 degrees) with potential path lines (brown doted lines)

As has been already mentioned, our aim is to manage to have as many compression waves as possible inside the cylinder while the inlet valve is open. In the diagram that means the red line to approach the '0.609' boundary when the valve is open and hence to benefit from that compression wave. This can be achieved by optimizing the intake pipe length as will be showcased later in this diploma thesis.

It must be noted that in order these diagrams to be produced for different lengths the user should manually change the connection points of each reflected characteristics by adding or subtracting degrees in the corresponding equations (see appendix).

State diagram

The characteristics in the state diagram are two families of parallel straight lines with slopes $\mp\frac{k-1}{2}$ which are the compatibility equations that relate the speed of sound and the velocity of the fluid. Moreover, these lines are uniquely determined in the sense that they are independent of the particular example under consideration, unlike the physical characteristics that form a different network depending on the occasion. What we can see in this diagram is a map of the variation of the speed of sound in the gas as its velocity changes. For a closed end the velocity is zero, whereas for an open end the outside pressure equals the initial pressure inside, thus the speed of sound is constant. The waves in the position diagram can be transferred on to the state diagram, when the state of the gas is altered by the disturbance passing through it. By using a horizontal line crossing the c-axis we can find its sound velocity; by using a vertical line crossing the u-axis we can find its velocity. Subsequently we are able to compute the values of pressure, density and temperature by the corresponding values of the sound velocity.

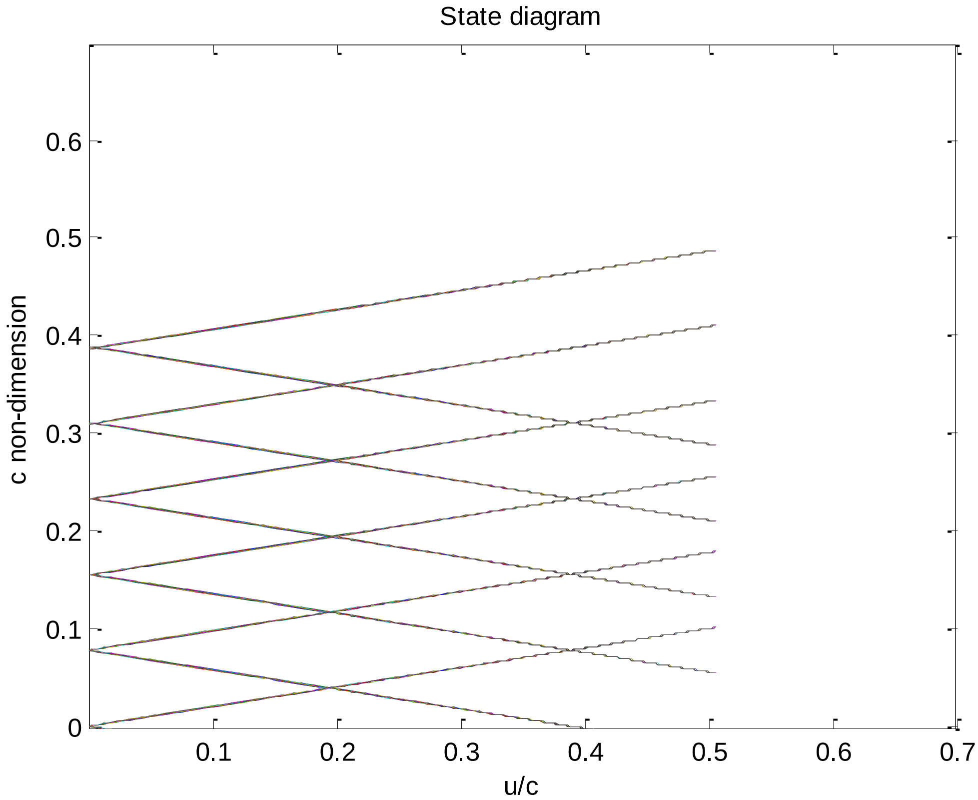


Figure 25

Simulation of various occasions

Doubling the length of the intake pipe

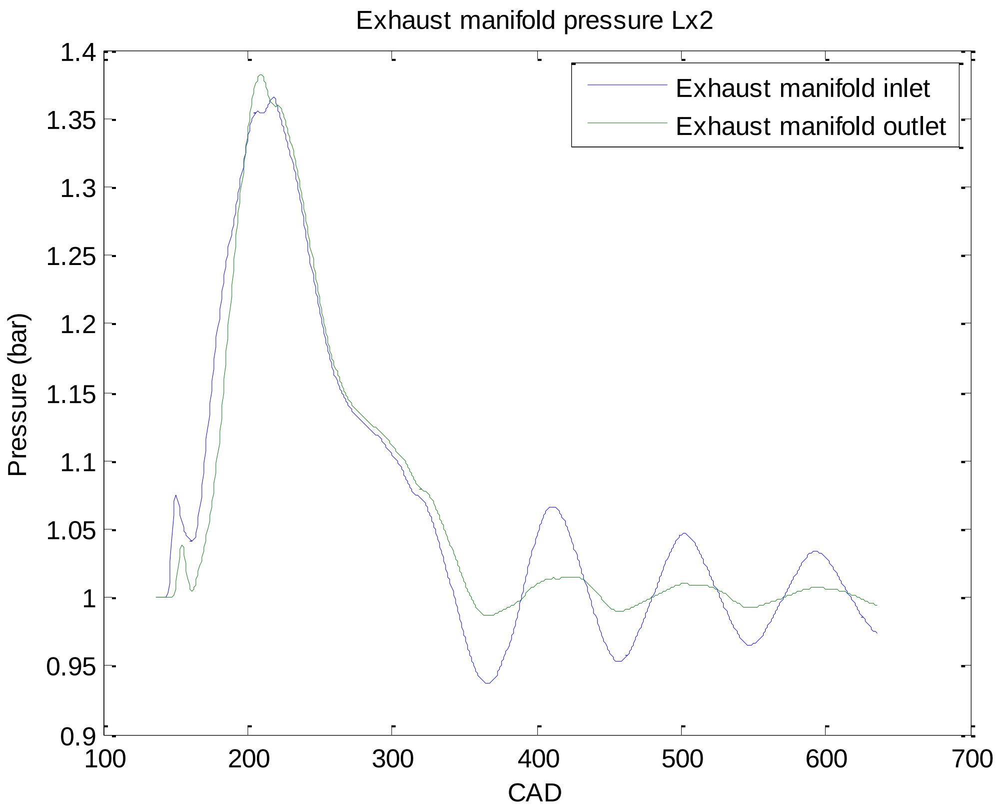


Figure 26

Comparing to the normal length of the intake pipe we can observe an increase in the pressure at both the exhaust manifold inlet and outlet approximately at 150 degrees, leading to the conclusion that there might be increased pressure inside the cylinder and so the exhaust pressure maybe be even higher too due to that reason. We will examine the rest of the diagrams in order to see if we can reach a valid conclusion. The next two *figure*s are magnifications of the exhaust pressure diagrams indicating the slight difference between the two configurations of the inlet pipe.

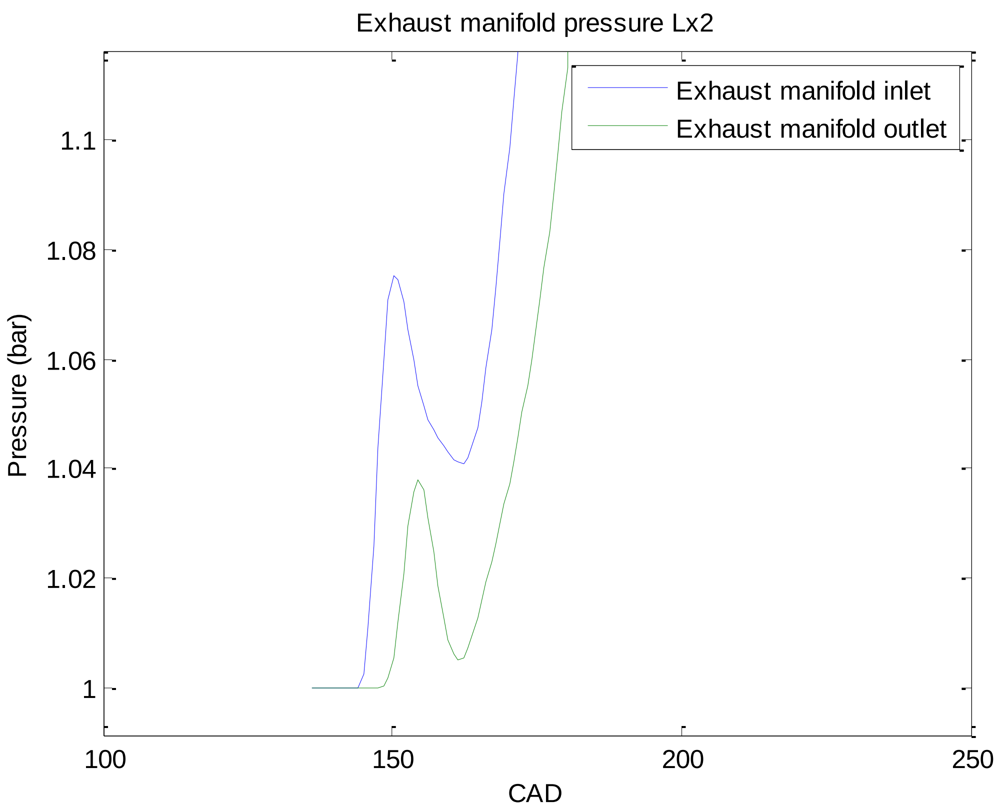


Figure 27: Detail of exhaust manifold pressure figure for double length of the intake pipe

Whereas for normal pipe length:

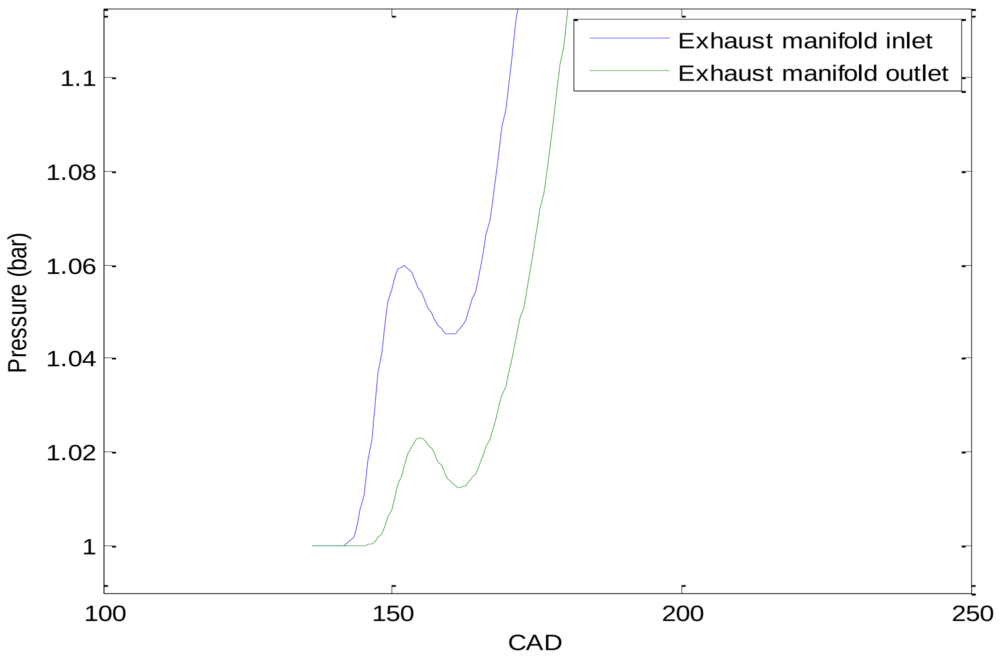


Figure 28: Detail of exhaust manifold pressure figure for normal length of the intake pipe

Now we will examine the **Complete Pressure Figure** for the following two occasions:

Normal length:

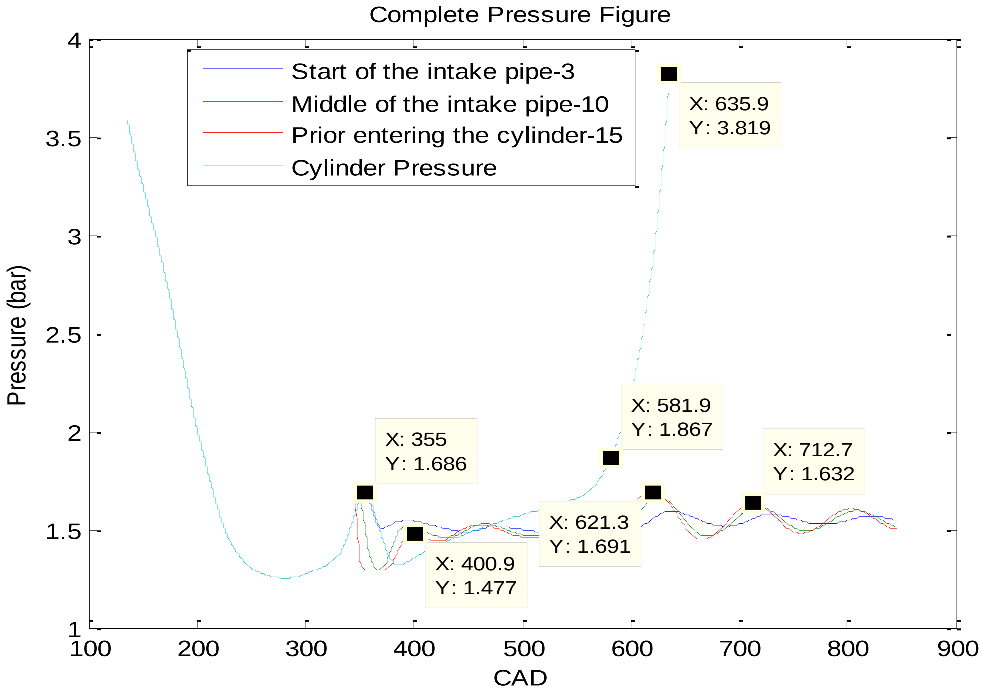


Figure 29

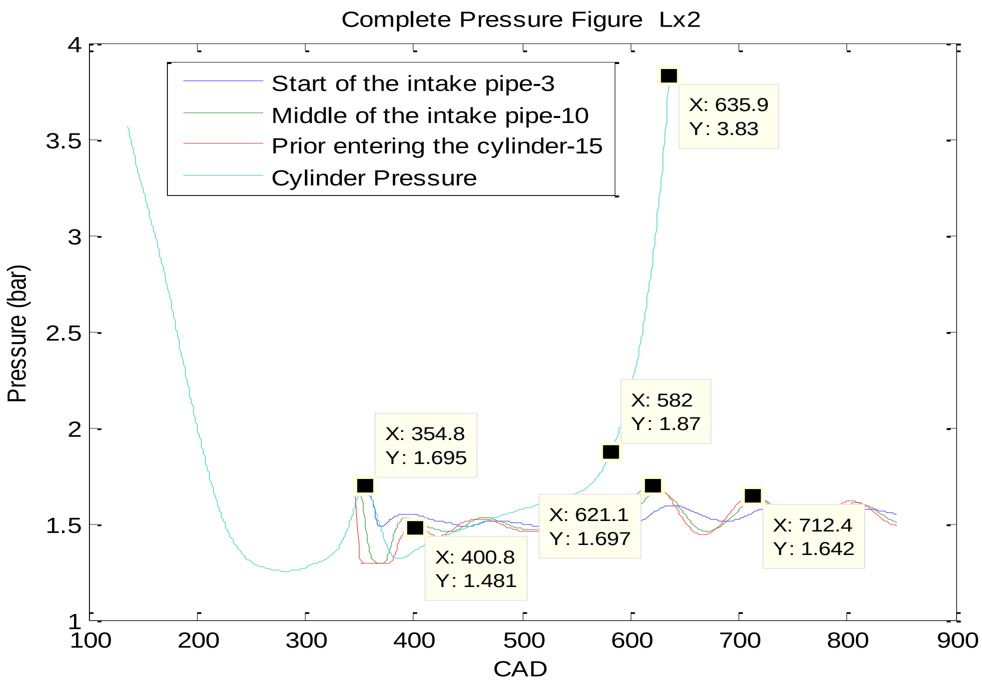


Figure 30

In both *figures* we have inserted some points of reference in strategic positions in order to compare them in detail. We are interested to know if the increase of length for this constant engine speed (~2000 rpm) has benefits for the pressure inside the cylinder, which accordingly means the potential work to be produced to be higher. We will focus on the lines of **'Cylinder Pressure'** and **'Prior Entering the Cylinder-15'**.It was difficult though to peak exactly the same points in both *figures*, so we have to consider there to be a slight error for reading the *figures*. However, following closely the lines we can make a general conclusion: the pressure has been increased –marginally- in all the occasions that interest us. In the **'Cylinder Pressure'** line the pressure at scavenge TDC has been increased to 1.695 bar from 1.686 bar and at the end of the cycle the peak pressure has been increased to 3.83 bar instead of 3.819. This proves that the increase of length acted indeed in a beneficial way for the engine, by better managing the travelling of compression and expansion waves across the intake pipe and into the intake valve. Moreover, we can see an increased pressure along the line of **'Prior Entering the Cylinder-15'** at the peak points of the fluctuation. Examining the diagrams of velocity and pressure across the intake pipe we can conclude that there are no major differences; actually, apart from a small increase in amplitude of the fluctuation of pressure and velocity it is almost impossible to see any worth noting difference, hence the *figures* are not presented here as they are identical. The major conclusion reached is that by doubling the length of the intake pipe –for this specific constant speed- we managed to increase the pressure figures inside the cylinder which is the main thing we wanted to achieve. So by ***doubling the length*** we achieved an average ***increase of pressure inside the cylinder of 0.31%.*** As a final case scenario we triple the length of pipe and we get the following results in the **Complete Pressure Figure**:

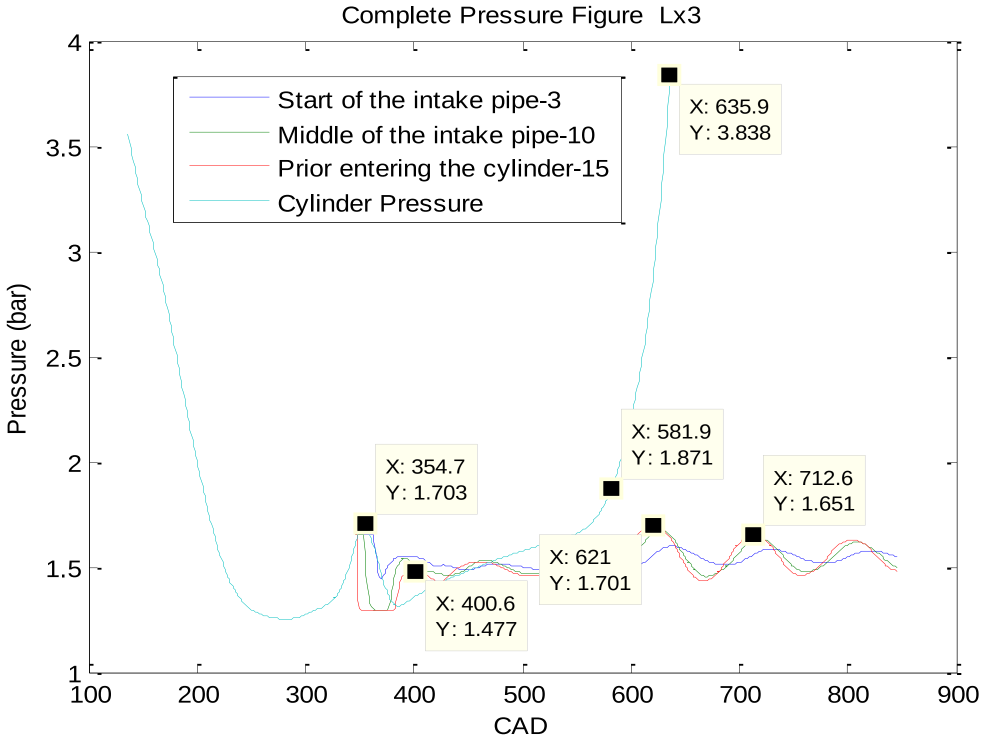


Figure 31

Performing the same analysis to the previously shown comparison between normal and double length, we focus on **'Cylinder Pressure'** line and the result is an average increase of pressure inside the cylinder for the triple length by ***0.54% to normal length*** and consequently ***0.23% to double length***.

In both situations the gains have been made by managing the travel the waves have to make till reaching the open valve. The results showed that we managed to synchronize more efficiently the arrival of the compression waves and the period the intake valve is open. It shouldn't be neglected that by fact a compression wave will result in an increase in pressure and this increase will help to fill the cylinder just before the valve closes. The results are also validated in the paper [5] where an increase in the volumetric efficiency is stated with a certain intake pipe length.

Of course these gains are relatively small, but we shouldn't neglect the fact we are simulating the length for a constant RPM running engine. It is an optimization process to determine the optimum length for that constant speed. Huge gains can be made with variable intake length for each RPM of the engine, which means the optimum length is needed to be found for every running speed of the engine and afterwards the length of each intake pipe to change accordingly. As has been mentioned, the simple norm is that for lower engine speed we need a longer intake pipe and for higher RPM we need a smaller intake pipe.

J. Lumley [3] suggests that the maximum length for a constant rpm running engine must take into consideration the time the waves need to travel backwards and forth to the intake valve. More specifically:

If the pipe length is L, the total transit time for a wave to travel back to the valve after being reflected by the valve is $\Delta t_1 + \Delta t_2 = \frac{L}{c-v} + \frac{L}{c+v} = \frac{2cL}{c^2-v^2}$ where c is the local speed of sound, and v the velocity of the fluid.

This time must be comparable to the open time of the valve which is for our case 236 degrees. Lumley introduces the term k/2N for the open time of the valve, where k is the ratio of the open angle to π and N the engine speed in revolutions per second. In order the condition to be valid the length of the intake pipe must be:

$$L = \frac{k(c^2-v^2)}{4cN}$$

The velocity v is changing during the calculation and in order to get accurate results we should integrate the velocity during the outgoing and returning trips.

So, with

```
k=1.3111
c=380.3 m/s
v=170 m/s
N=33.333333rps
```

The maximum length is then: Lmax= 2.9923 meters.

The code that we run uses 30 Meshes and that made impossible to simulate such an intake length. The limit till which the code produced effectively results was the triple length of the normal intake length, as showcased earlier. In order to simulate Lmax we need to increase the number of Meshes used.

Chapter 7

Non-Isentropic Flow

In the previous chapters the method of characteristics for a 1-D, unsteady and isentropic flow was showcased. However, in reality, things are more complex and that is due –beside others- to entropy change variations. In non-isentropic flow information, such as Riemann variables, leaving one part of a pipe may not stay in a field of constant entropy throughout its path. Since the actual value of the primitive variables, such as pressure, is related to the Riemann variables through the local entropy level then it is necessary to take account of variations in entropy with the passage of the wave. [4]

The entropy is not constant across an engine system and the reason [1] is caused by:

- Irreversibilities in the flow through valves
- Temperature gradients
- Heat transfer
- Friction
- Variable areas in general means

The main thing to do is to manage introducing the terms of area change, heat transfer and friction –all of which affect entropy- to the equations from which the characteristic curves may be derived.

*The basic difference between isentropic and non-isentropic flow is that the Riemann variables λ and β are **not** constant along the characteristics.* A particle travels along a path line and thus entropy changes refer to particular particles.

Based on the one-dimensional conservation equations for one steady flow and the first law of thermodynamics Benson [1] produced the compatibility equations for three families of characteristics: λ, β characteristics and path line characteristic all containing heat transfer, entropy change and friction terms. The procedure for producing the following equations is analytically described in [1] and will not be reproduced here.

All of the following equations have been non-dimensionalized following the same procedure of the 1st chapter.

Family of λ characteristics (right-forward)

Direction condition:

$$\frac{DX}{DZ} = U + A$$

Compatibility condition:

$$d\lambda = dA + \frac{\kappa - 1}{2} dU$$

Where

$$d\lambda = -\frac{(\kappa-1)}{2}\frac{AU}{F}\frac{dF}{dX}dZ + \frac{A}{A_A}dA_A - \frac{(\kappa-1)}{2}2\frac{fxref}{D}U^2\frac{U}{|U|}\left\{1-(\kappa-1)\frac{U}{A}\right\}dZ + \frac{(\kappa-1)^2}{2}\frac{qxref}{a_{ref}^3}\frac{1}{A}dZ$$

In the previous equation giving the change of λ across a characteristic –dλ- the first term is due to area change, the second one due to entropy change, the third one due to friction and the last one due to heat transfer.

Family of β characteristics (left-forward)

Direction condition:

$$\frac{DX}{DZ} = U - A$$

Compatibility condition:

$$d\beta = dA - \frac{\kappa - 1}{2} dU$$

Where

$$d\beta = -\frac{(\kappa-1)}{2}\frac{AU}{F}\frac{dF}{dX}dZ + \frac{A}{A_A}dA_A + \frac{(\kappa-1)}{2}2\frac{fxref}{D}U^2\frac{U}{|U|}\left\{1+(\kappa-1)\frac{U}{A}\right\}dZ + \frac{(\kappa-1)^2}{2}\frac{qxref}{a_{ref}^3}\frac{1}{A}dZ$$

Which contains the same terms and is similar to the right-forward characteristics apart from the signs.

In both families the entropy change term dA_A is given by:

$$dA_A = \frac{\kappa - 1}{2}\frac{A_A}{A^2}\left(\frac{qxref}{a_{ref}^3} + 4\frac{fxref}{D}\frac{|U^3|}{2}\right)dZ$$

Where also:

- $f = \frac{\tau_w}{\frac{1}{2}\rho u^2}$ for the wall shear stress and

- q being the rate of heat transfer per unit time per unit mass.

Path line characteristic

Frictional and heat transfer effects act on individual fluid particles and since these effects may vary arbitrarily, *the path lines are characteristic curves on which the entropy or temperature gradients may have discontinuities* [2].

Direction condition

$$\frac{DX}{DZ} = U$$

Compatibility equation

$$dA_A = \frac{\kappa - 1}{2}\frac{A_A}{A^2}\left(\frac{q\ xref}{a_{ref}^3} + 2\frac{fxref}{D}|U^3|\right)dZ$$

Where the first term in the bracket refers to heat transfer and the second one to friction.

It is important to mention that this is the change in entropy for a particular particle travelling along a path line; **<u>entropy levels are propagated along the path lines</u>**.

It can be noticed:

1) The direction conditions are the same for isentropic and non-isentropic flows.
2) All the equations contain the time variable dZ. The values of λ, β and A_A at time Z are used to determine the changes in their magnitude dλ, dβ and dA_A respectively.

In general, the numerical technique used to solve the characteristic equations in the Z-X field is the same that is used for isentropic flow, apart from the fact that we now have three variables to determine at each mesh point, instead of the two Riemann variables: λ, β and A_A.

For λ and β the stability criterion remains the same as to isentropic flow:

$$\frac{\Delta Z}{\Delta X} < \frac{1}{A + |U|}$$

And for the entropy term A_A the criterion is

$$\frac{\Delta Z}{\Delta X} < \frac{1}{|U|} \text{ [Path line criterion]}$$

It is important to note that if the stability criteria for λ and β characteristics are satisfied, the path line criterion will also be satisfied.

λ and β criteria satisfied ⇒ path line criterion satisfied

Furthermore, based on the fact that λ characteristics are defined as the right running characteristics (the nominal slope of the characteristics is in the same direction as the positive X-axis) and β as the left running ones (their slope is respectively on the opposite direction of the positive X-axis), Benson [1] produced a ***general compatibility equation*** for both type of characteristics:

$$d\lambda = -\frac{(\lambda+\beta)(\lambda-\beta)}{4}\frac{1}{F}\left(\frac{dF}{dX}\right)dz + \frac{(\lambda+\beta)}{2}\frac{dA_A}{A_A} - \frac{(\kappa-1)}{2}\frac{2f\,x_{ref}}{D}\left(\frac{\lambda-\beta}{\kappa-1}\right)^2\frac{(\lambda-\beta)}{|\lambda-\beta|}\left\{1-\frac{2(\lambda-\beta)}{(\lambda+\beta)}\right\}dZ + \frac{(\kappa-1)^2}{2}\frac{qx_{ref}}{a_{ref}^3}\frac{2}{(\lambda+\beta)}dZ$$

Or

$$d\lambda = (\delta\lambda)_{area} + (\delta\lambda)_{entropy} + (\delta\lambda)_{friction} + (\delta\lambda)_{heat\ transfer}$$

The **limitation** as to using this general definition is that the sign of $\left(\frac{dF}{dX}\right)$ is positive if the area increases in the direction of travel of the considered wave.

Along a characteristic, it is assumed that

$$\lambda(X + dx, Z + dZ) = \lambda(X, Z) + d\lambda$$

And that this variation of λ does not affect the slope of the characteristic.

It goes without saying that in each of the terms in the general compatibility equation, where λ, β are mentioned, ***refer to the previous point***.

Let us explain the sub-terms in the above equation. In particular:

Area term $(\delta\lambda)_{area}$

F is considered to be the area of the duct, and if it is assumed that the ducts are circular we obtain:

$$F = \frac{\pi}{4} D^2$$

$$\frac{dF}{dX} = \frac{\pi}{4} 2D \frac{dD}{dX} \quad \rightarrow \frac{1}{F}\frac{dF}{dX} = \frac{2}{D}\frac{dD}{dX}$$

And as though:

$$(\delta\lambda)_{area} = -\frac{(\lambda+\beta)(\lambda-\beta)}{2}\frac{1}{D}\left(\frac{dD}{dX}\right) dZ$$

Heat Transfer Term $(\delta\lambda)_{heat\ transfer}$

$$\frac{(\kappa-1)^2}{2}\frac{q x_{ref}}{a_{ref}^3}\frac{2}{(\lambda+\beta)} dZ$$

Generally this term is associated with combustion or normal heat transfer. We will consider in this point only the heat transfer in the duct. It is assumed that *the heat transfer under non-steady flow is the same as under steady flow*.

Taking as a fact the wall temperature to be constant, the Reynolds' analogy is used for the heat transfer coefficient:

$$h = \frac{f}{2} C_P u \rho$$

and the heat transfer rate given by

$$q = \frac{2 f u C_p (T_w - T_g)}{D}$$

Where T_w is the wall temperature, T_g the gas temperature, f the friction coefficient and C_p:

$C_P = \frac{\kappa}{\kappa-1} R$: for an ***ideal gas*** with constant specific heats

So it can be seen that as the gas travels along the pipe with velocity u, its entropy level is changed by heat transfer to or from the pipe walls, and by the resistance to its motion generated by the pipe wall friction.

By taking these thoughts into consideration the heat transfer term becomes now:

$$(\delta\lambda)_{\text{heat transfer}} = 2\kappa\left(\frac{|\lambda-\beta|}{\lambda+\beta}\right)\frac{f\,\Delta X}{D}\left(\frac{x_{ref}}{\Delta X}\right)\left(\frac{R}{a_{ref}^{2}}\right)\left(T_w - T_g\right)\Delta Z$$

One <u>important assumption</u> that was made is that we assumed of ***heat transfer to be taking place only when there is flow in the pipe*** (thus speed), otherwise we don't have conduction between the gas and the wall.

Again it shouldn't be neglected that we are discussing over differences and all the λ and β symbols refer to the initial state-point at time Z.

Numerical Solution

As it has been mentioned, the same procedure will be followed as for the isentropic flow, but we need to evaluate three variables instead of two. At time Z the values of λ, β, A are known for all the mesh points. We need to evaluate these values again after a time step ΔZ: Z+ΔZ, calculating the variation of λ and β along a characteristic.

- Calculation for the next point after time step ΔZ
- Iterative procedure for calculating λ at the boundary

The calculation of Riemann variables at the boundaries is similar to the one used in isentropic flow, by using the notation of **λ_{in}, λ_{out}**. However, in non-isentropic flow there is a variation in λ along a characteristic in the time interval ΔZ; this could be caused by inflow through the boundary at a different entropy level [1].

If λ_{inn} is the value of the Riemann variable approaching the boundary <u>when there is no flow into the pipe</u>, A_{An} is the corresponding value of A_A and if λ_{inc}, λ_{outc}, A_{Ac} are the values of λ_{in}, λ_{out} and A_A when account is taken of the inflow entropy gradient, then

$$\lambda_{inc} = \lambda in_{n} + (\delta\lambda)_{entropy}$$

Or by equation 7.97 [1]:

$$\lambda_{inc} = \lambda_{inn} + \left(\frac{\lambda_{inc}+\lambda_{outc}}{2}\right)\left(\frac{A_{Ac}-A_{An}}{A_{Ac}}\right)$$

To solve this equation we need to use an **iterative method**:

1) The values of λ_{inn} and A_{An} are maintained fixed throughout the calculation.
2) We need to insert an initial value for λ_{inc}, λ_{outc} . For the first one we make a first assumption: $\lambda_{inc} = \lambda_{inn}$ and the initial value of λ_{outc} is taken as the value of λ_{out} at the previous time, Z.
3) The entropy level A_{Ac} is estimated from the boundary conditions.

We insert these values at the equation given above and thus we obtain a first estimate of $\lambda_{inc.}$

This calculated value is *inserted* into the boundary equations as to evaluate new values for λ_{outc} and A_{Ac}.

Afterwards steps 1-3 are repeated until successive values of λ_{in} are within the required tolerance.

Introducing the 'starred' Riemann variable λ*

The 'starred' Riemann variable λ* is used for examining complex flows with variable entropy, by including the change in entropy directly into this variable:

$$\lambda_{in}^{*} = \frac{\lambda_{in}}{(A_A)_{in}}$$

and

$$\lambda_{out}^{*} = \frac{\lambda_{out}}{(A_A)_{out}}$$

According to [1], for flow into a variable, but uniform, back pressure p_b , λ* becomes:

$$\lambda_{in}^{*} = \frac{\lambda_{in}}{(A_A)_{in}} (\frac{p_{ref}}{p_b})^{\frac{k-1}{2k}}$$

$$\lambda_{out}^{*} = \frac{\lambda_{out}}{(A_A)_{out}} (\frac{p_{ref}}{p_b})^{\frac{k-1}{2k}}$$

For the entering and leaving characteristic respectively.

This will help us describe the flow through a valve or port where there is variable pressure p_b.

Changes in the boundary equations in order to include entropy changes

The previously introduced methodology is general and does not include the implementation of boundaries along the string of meshes. So, for specific boundaries such as inflow to a pipe, intake and exit valves of the cylinder and the nozzle certain changes need to take place. The following modifications will be made to the corresponding subroutines of the MATLAB code in order to include the entropy changes across them. For each part of a code that has been changed, an 'e' letter has been added to the name of the code showing that the change in entropy has not been neglected, for example instead of '*inflow*' subroutine used in isentropic flow, now we use '*inflowe*'.

Closed end

Same as for isentropic flow:

$$\lambda_{out} = \lambda_{in}$$

Open end

If the reference pressure is set equal to the back pressure then we use again the isentropic flow equation, thus:

$$\lambda_{out} = 2 - \lambda_{in}$$

Else we use the starred Riemann variable to include the back pressure:

$$\lambda *_{out} = 2 - \lambda *_{in}$$

Inflow

As the entropy of the inflowing gas may be different from the entropy of the gas in the pipe we have to use the iterative method mentioned above for calculating λ_{in}. In brief the variables used are:

$$\lambda in_c = new\ value\ of\ \lambda in$$

$$\lambda in_n = previous\ value\ of\ \lambda in$$

$$\lambda out_c = previous\ value\ of\ \lambda out$$

$$A_{Ac} = new\ value\ of\ A_A = Ao(\frac{p_{ref}}{po})^{\frac{k-1}{2k}}$$

Where $$Ao = (\frac{To}{Tref})^{\frac{1}{2}}$$

T_0: the stagnation temperature

$$A_{An} = previous\ value\ of\ A_A$$

After we have calculated the new λin_c we use the following equation by Benson [1] to calculate λ_{out}:

$$\lambda_{out} = \left(\frac{3-\kappa}{\kappa+1}\right)\lambda_{in} + \frac{2}{\kappa+1}\sqrt{\{(\kappa^2-1)A_o^2 + 2(1-\kappa)\lambda_{in}^2\}}$$

This value is set equal to $\lambda_{out}c$ and again we start the iterative procedure until we have the desired accuracy.

If we have reverse flow (open end) we use the starred equation:

$$\lambda *_{out} = 2 - \lambda *_{in}$$

The corresponding code is the subroutine '*Inflowe*' which in turn calls the iterative subroutine '*IterativeInflow*'. Both can be found in the Appendix section.

Flow through a partially open end- Nozzle

The same basic procedure as in isentropic flow is followed in order to calculate the Riemann variable at the exit of the nozzle.

The basic difference is first of all the introduction of the starred Riemann variable instead of the simple one. Thus:

$$\lambda_{in}^{*} = \frac{\lambda_{in}}{(A_A)_{in}} (\frac{p_{ref}}{p_b})^{\frac{k-1}{2k}}$$

Moreover, we need to examine the conditions at the throat of the nozzle. Where a parameter related to the throat conditions is mentioned, it will have the suffix 't'.

A parameter that we need to introduce is the fraction of the speed of sound at throat to the reference speed of sound:

$$A_t = \frac{a_t}{a_{ref}}$$

However, a_{ref} is not necessarily on the same isentropic line as the general speed of sound, a , and the reference pressure is not accordingly p_t, the throat pressure for subsonic flow. To reduce the complexity of the problem, the same value as previously will be used for a_{ref}, without any major difference in the accuracy of the solution taking place.

Another parameter to be used is the fraction:

$$A^* = \frac{A}{A_t}$$

Where $A = \frac{a}{a_{ref}}$

And $A_{cr}^{*} = (\frac{A}{A_t})_{cr}$

Firstly we check whether there is normal or reverse flow condition in the nozzle, by examining the value of the Riemann variable compared to unity.

Afterwards, as explained in the code part, we check whether there is subsonic or sonic flow on the throat, in which case it would mean that the flow is chocked.

The corresponding code is the subroutine '*nozzlee*', which can be found in the Appendix section.

Outflow from a cylinder to a pipe through a valve

-Constant pressure mode

The major difference for a non-isentropic flow through a valve is that the entropy change across the valve for flow into the pipe changes the value of the Riemann variable λ_{in} as it enters the boundary calculation [Benson [1], chapter 7.5]. Better explained, the entropy in the pipe immediately before the gas enters the pipe is different to the entropy in the same position immediately after the valve has opened. Hence, that increase in entropy affects the entering Riemann variable λ_{in} which needs to be modified. For this purpose, an iterative method will be used in order for a known λ_{in} and a computed λ_{out} to have estimated values of λ_{in}, A_A which must converge to give the computed values of λ_{in}, λ_{out}, A_A.

The iterative method to be followed is similar to the one presented for the inflow and the corrected Riemann variable λ_{in} due to entropy change may be given by

$$\lambda_{inc} = \lambda_{inn} + \left(\frac{\lambda_{inc} + \lambda_{outc}}{2}\right)(\frac{A_{Ac} - A_{An}}{A_{Ac}})$$

Where $\boldsymbol{A_{An}}$ is the entropy level in the pipe *immediately before* the gas enters the pipe from the exhaust valve and $\boldsymbol{A_{Ac}}$ the entropy level in the pipe *immediately after* the valve has opened and the gas enters the pipe. Finally, λ_{outc} is the corrected Riemann variable λ_{out}.

$$\lambda_{out} = \left(\frac{3-\kappa}{\kappa+1}\right)\lambda_{in} + \frac{2}{\kappa+1}\sqrt{\{(\kappa^2-1)A_c^2 + 2(1-\kappa)\lambda_{in}^2\}}$$

So, the iterative subroutine at the exhaust valve '*Iterativevalvee2*' and the subroutine '*valvee2*' accordingly, should return the values λ_{inc}, λ_{outc}, A_{Ac} , based on the firstly estimated variables λ_{inn}, A_{An}.

The main program –*MainProgramEntropy*- calls the cylinder boundary function –*cylindere*-, which calls the valve function –*valvee2*- to determine inflow or outflow from the valve. Furthermore, the valve function calls the iterative function, *Iterativevalvee2*.

Inflow from a pipe through a valve

The same boundary conditions are used as in the isentropic case. The only thing that changes is the introduction of the entropy level in the boundary for checking the flow direction: outflow, inflow or no flow. Practically, as has been showcased before, the starred Riemann variable is introduced in place of the simple Riemann variable:

$$\lambda_{in}^* = \frac{\lambda_{in}}{(A_A)_{in}}(\frac{p_{ref}}{p_b})^{\frac{k-1}{2k}}$$

This can be best seen in the code located at the appendix section which is self-explanatory.

Cylinder Boundary Conditions

The procedure for calculating the pressure changes in the cylinder is the same as for isentropic flow, except for we need to add a term for heat transfer.

The heat transfer term is usually based on empirical relationships, for the moment though we will consider the expression suggested by Annand [9]:

$$\frac{dQ}{dt} = \frac{a\ Re^b}{D_c} k_q F(T_w - T_C)$$

Where α and b are constants, the first one's value is dependent on the engine type [1] and we will be considering a value of 0.5, whereas b equals to 0.7.

F is the exposed surface area of cylinder walls plus piston and cylinder head area.

Dc is the cylinder bore, T_w is the wall temperature of the cylinder and T_C is the gas temperature.

Re is the Reynolds number:

$$Re = \frac{\rho D_C V_P}{\mu}$$

Where:

ρ is the gas density, μ the gas viscosity, Vp the average piston velocity and Kq the conductivity: $k_q = \frac{C_p \mu}{0.7}$ for Prandtl number 0.7, where Cp is the specific heat of gas at constant pressure.

The average piston velocity is given by the expression:

$$Vp = 2 * stroke * RPM/60 \text{ (m/s)}$$

The mean piston speed is the average speed of the piston in a reciprocating engine. It is a function of stroke and RPM. There is a factor of 2 in the equation to account for one stroke to occur in 1/2 of a crank revolution (or alternatively: two strokes per one crank revolution) and a '60' to convert seconds from minutes in the RPM term.

The gas density is given by the expression:

$$\rho = \frac{\kappa}{A^2}\frac{p_{ref}}{a_{ref}^2}\left(\frac{\lambda_{in} + \lambda_{out}}{2A_A}\right)^{\frac{2\kappa}{\kappa-1}}$$

As we can see, the Riemann variables are introduced and define the value of density.

The variation of pressure inside the cylinder can be calculated by the expression proposed by [1]:

$$\frac{dp_c}{dt} = \frac{1}{V_c}\left[a_{ao}^2\left(\frac{dm}{dt}\right)_a - a_c^2\left(\frac{dm}{dt}\right)_e - \kappa p_c\frac{dV_c}{dt} + (\kappa - 1)\frac{dQ}{dt}\right]$$

By introducing these expressions into the cylinder code we convert the function from isentropic to non-isentropic (‘*cylindere*’ subroutine). Furthermore, certain arbitrary values were given concerning the wall temperature after considering some default values given in GT-Power program.

<u>Expected results</u>

Benson [1] showed that the initial exhaust pipe temperature has a big influence on the pressure drop in the cylinder and the pressure development in the exhaust pipe. This is due to wave reflections at the temperature discontinuity in the exhaust pipe and their interaction with the cylinder conditions. This is where the non-isentropic flow plays a significant role and must be analysed with the theory given above, as isentropic calculations do not allow for temperature discontinuities in the pipe.

For a single-cylinder single pipe system there are no major differences for the calculation of the pressure in the intake manifold as is the case for this diploma thesis. Non-isentropic calculations are useful for more accurate results, especially when there are major area differences, heat transfer, friction and temperature discontinuities. All these thermal effects are most noted in the exhaust pipe and not in the area of the intake manifold. So, as a future research, a code can be developed using the existing code parts to be given later in the appendix section for calculating the exact figures for the non-isentropic flow, oriented though towards the exhaust manifold.

Chapter 8

GT-POWER analysis

GT-power is a strong simulation tool for testing and optimizing engine performance, providing almost unlimited configuration options. However, it is a black box program meaning that the user cannot alter the program and is only responsible for creating a model from existing parts while providing the input depending on experience and problem to be tackled.

GT-Power simulation program will be used mainly for two reasons:

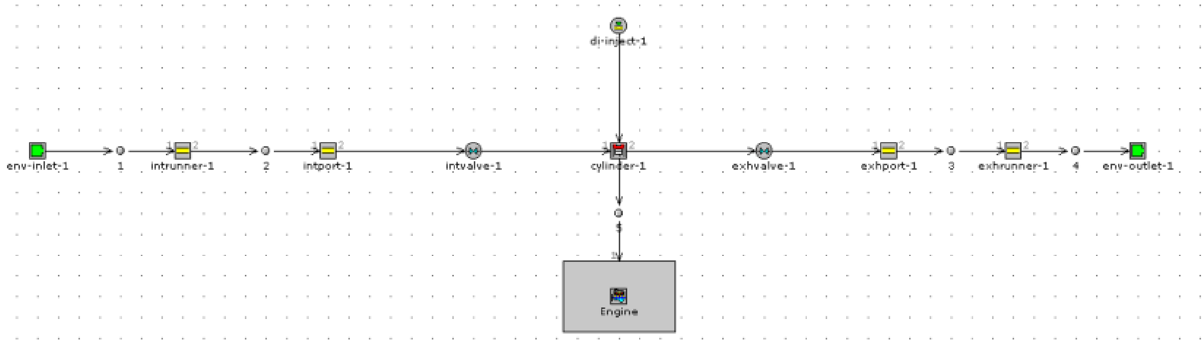
- To see if we can duplicate the results produced by the MATLAB code and to give us a first idea as to how some variables are changing during the cycle
- Secondly, a simulation was made for comparing the performance of a natural aspired one cylinder engine (like the one used in MATLAB code) to a turbocharged engine with same specifications and to an engine with both a turbocharger and a mechanically driven compressor (roots).

MATLAB and GT-power programs are being run at the same time. A model of a 1-cylinder engine was created, similar to the model that has been created in MATLAB code. However, it was not possible to emulate exactly the MATLAB model as we need to have fuel injection in the GT-Power program, whereas the code in MATLAB does not include any burning models, which was a simplification made on purpose, as we just wanted to understand the pulse waves in the intake pipe and how they affect the efficiency of the engine. For this reason the results couldn’t be confirmed or rejected using the GT-Power program.

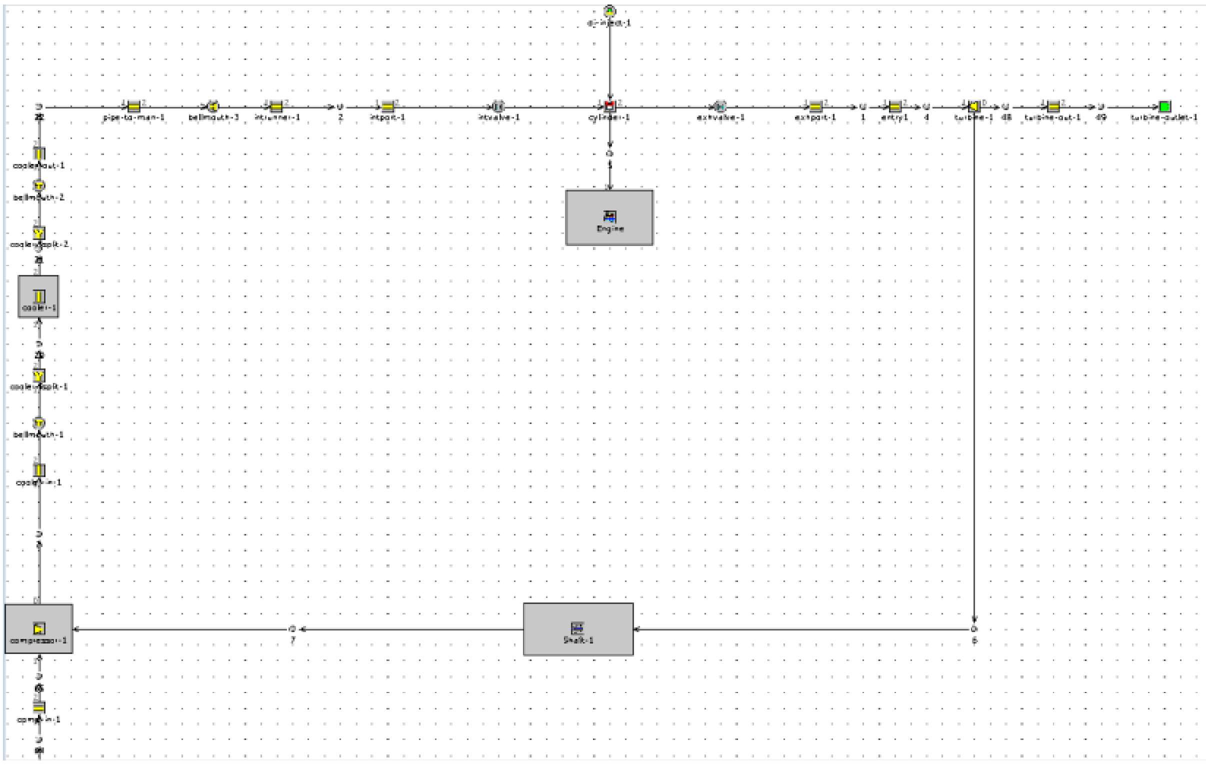
Comparison of different types of engine aspiration

In this extra section we will showcase the differences in performance between a normally aspirated one cylinder diesel engine and its configuration in turbocharged specification with exactly the same basic design.

Normally Aspirated -NA- single cylinder model



Turbocharged single cylinder model



The aim will be just to see the differences between the two configurations and present the simulation capabilities of GT-Power. However, being a black box program we are not able to see exactly the way it is working and thus have the ability to modify it. What we do is to create models in a graphic environment with inputs of engine displacement, environment conditions and models concerning the burning flame and heat transfer. The basic difference to the MATLAB code is that in GT-Power we need to have fuel in order for the program to work, however in this way we cannot see how the differences in the length of the intake pipe can affect the efficiency of the engine, in the way the MATLAB code did.

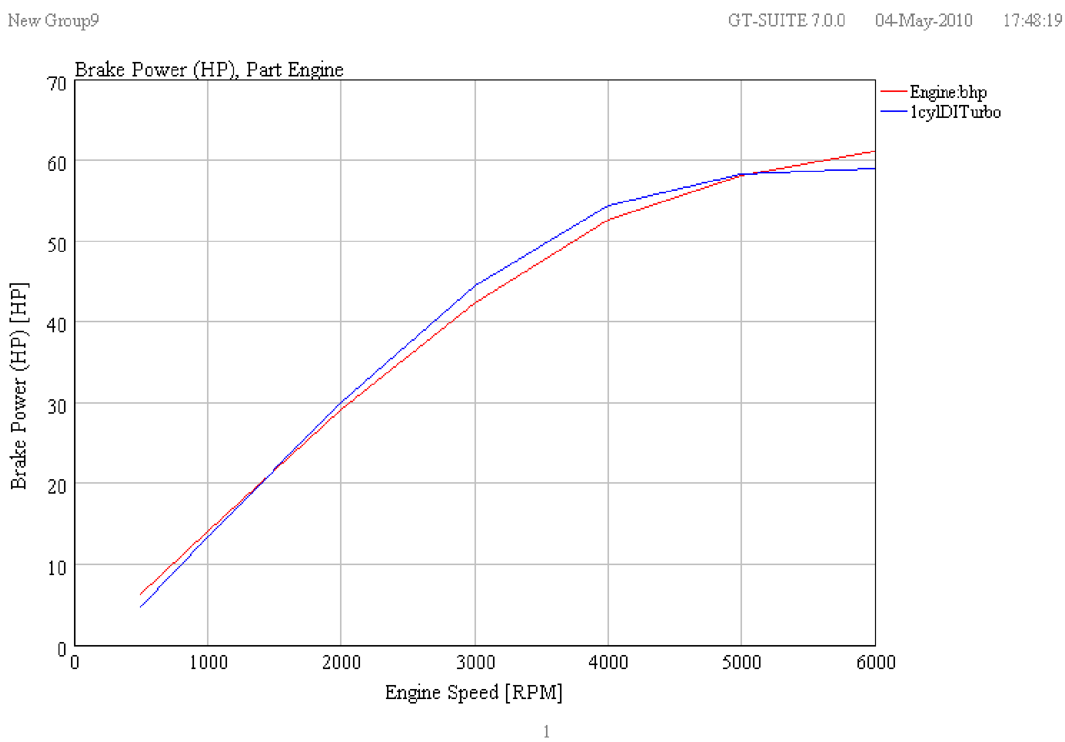


Figure 32: Comparison of Brake Power between NA and a Turbo engine

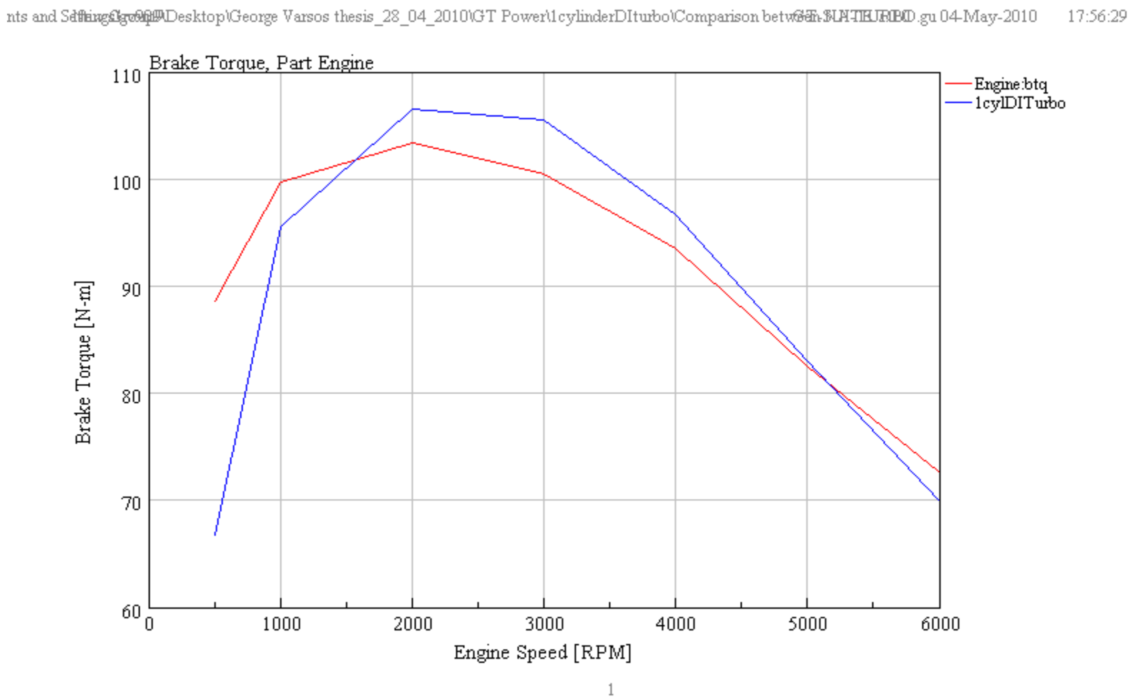


Figure 33: Comparison of Brake Torque between NA and a Turbo engine

Turbo-lag can be seen here, where, below 1500 RPM the turbocharged engine behaves as a naturally aspirated one and doesn't quite have the performance of the naturally aspirated engine, albeit it is a small difference; after 1500 RPM the turbine provides more power to the compressor which accordingly provides more air mass to the cylinder and thus we can see the boost in performance –power and torque- proving the benefits of turbo charging.

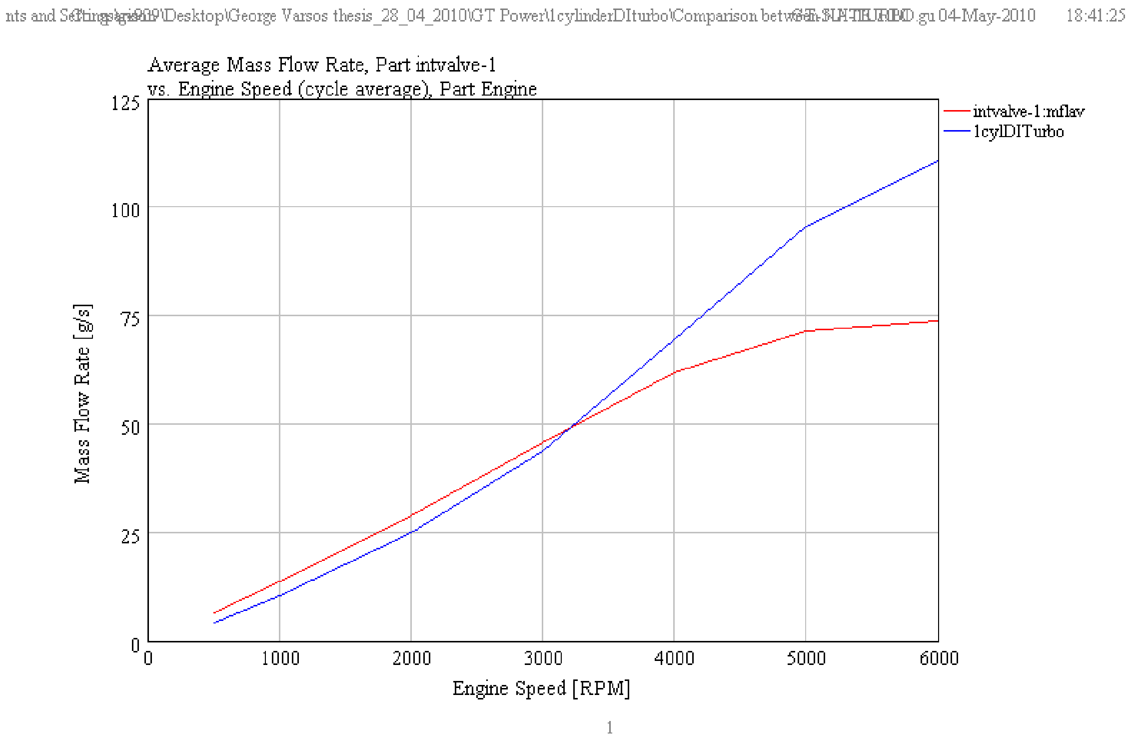
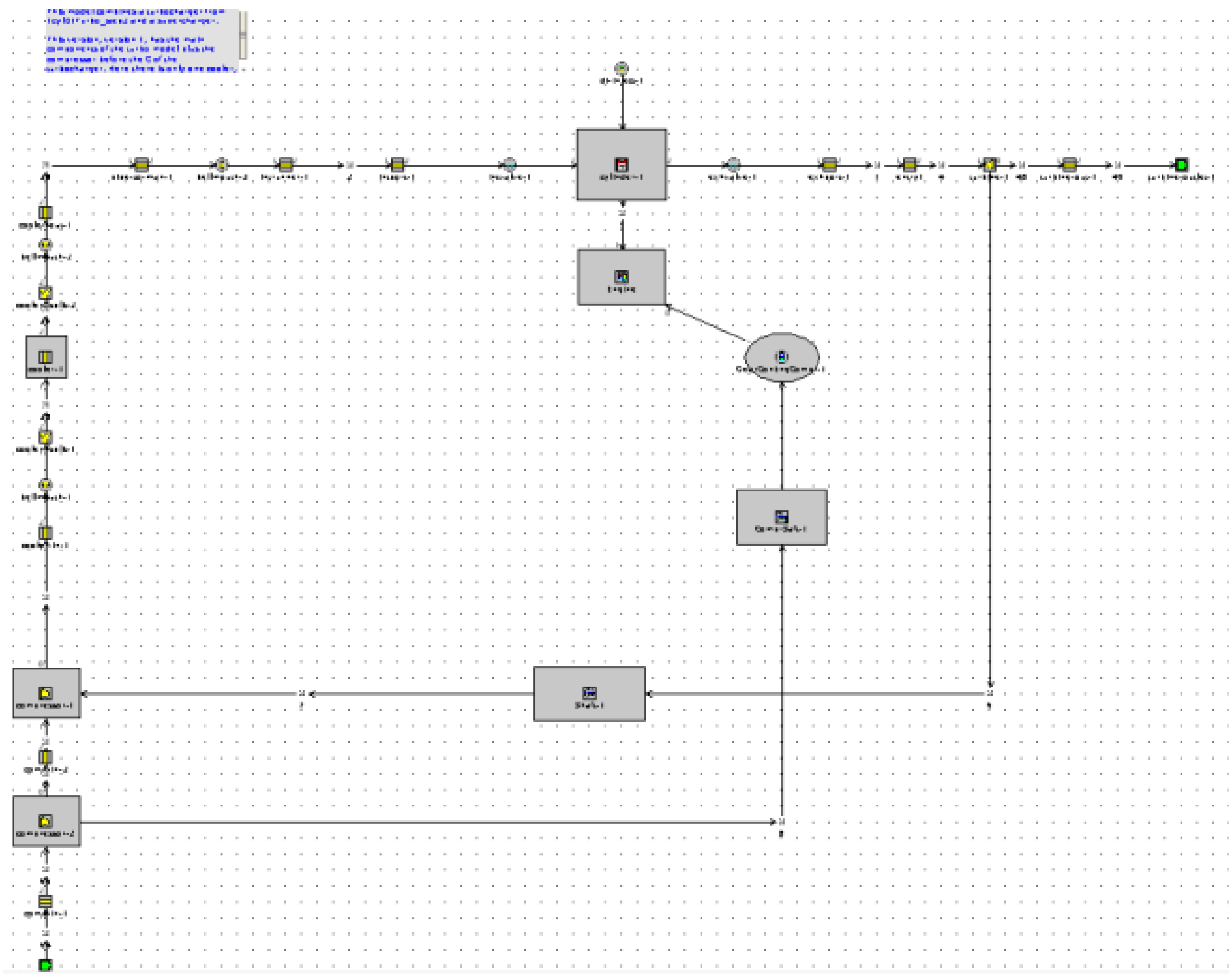


Figure 34: Average mass flow rate for a NA and a Turbo engine

In the second comparison section we will compare the outputs of another configuration of the same engine: with a combination of a turbocharger and a supercharger (Roots). This means that we are having two compressors in line, one being driven by the turbine and the other to be mechanically driven by the engine through a shaft, as has been done by VW in the TSI Otto engine. Here we will be showcasing some initial results for a diesel engine, a configuration that has not been used in production, apart from some marine engines by MAN.

Single cylinder model with a combination of a turbocharger and a mechanically driven compressor



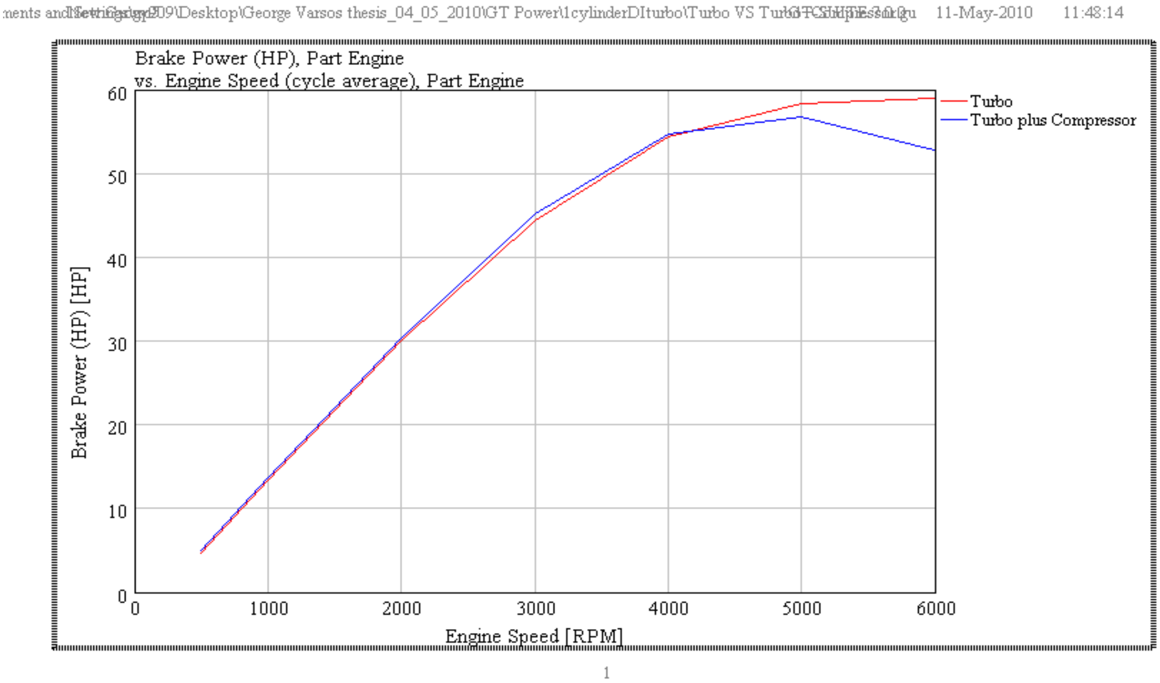


Figure 35: Comparison of Brake Power between a Turbo and a combined turbocharged engine

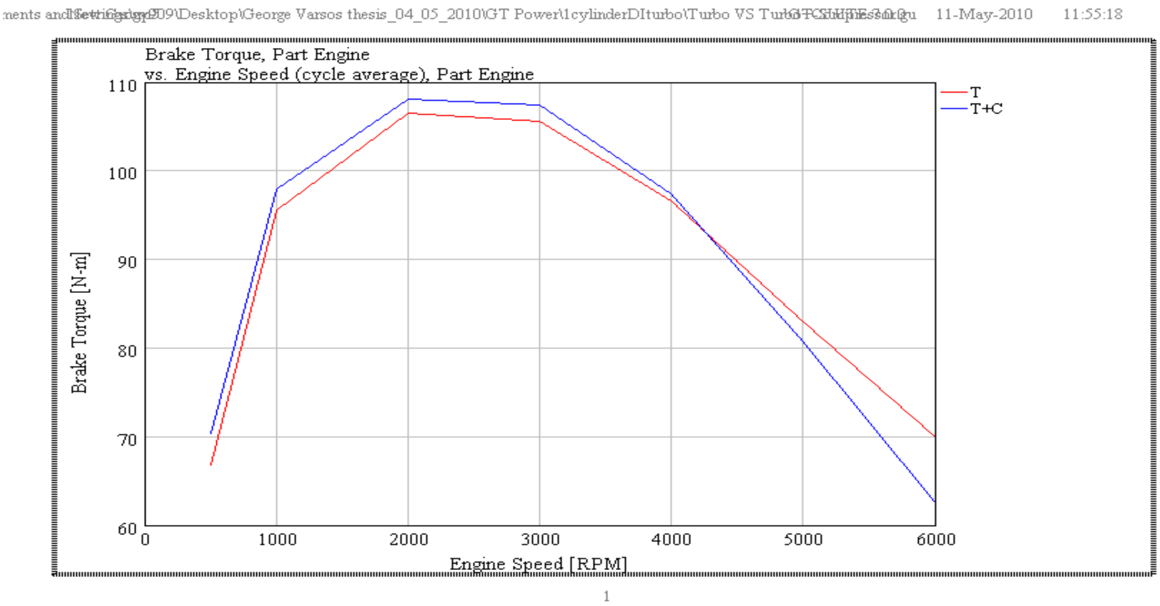


Figure 36: Comparison of Brake Torque between a Turbo and a combined turbocharged engine

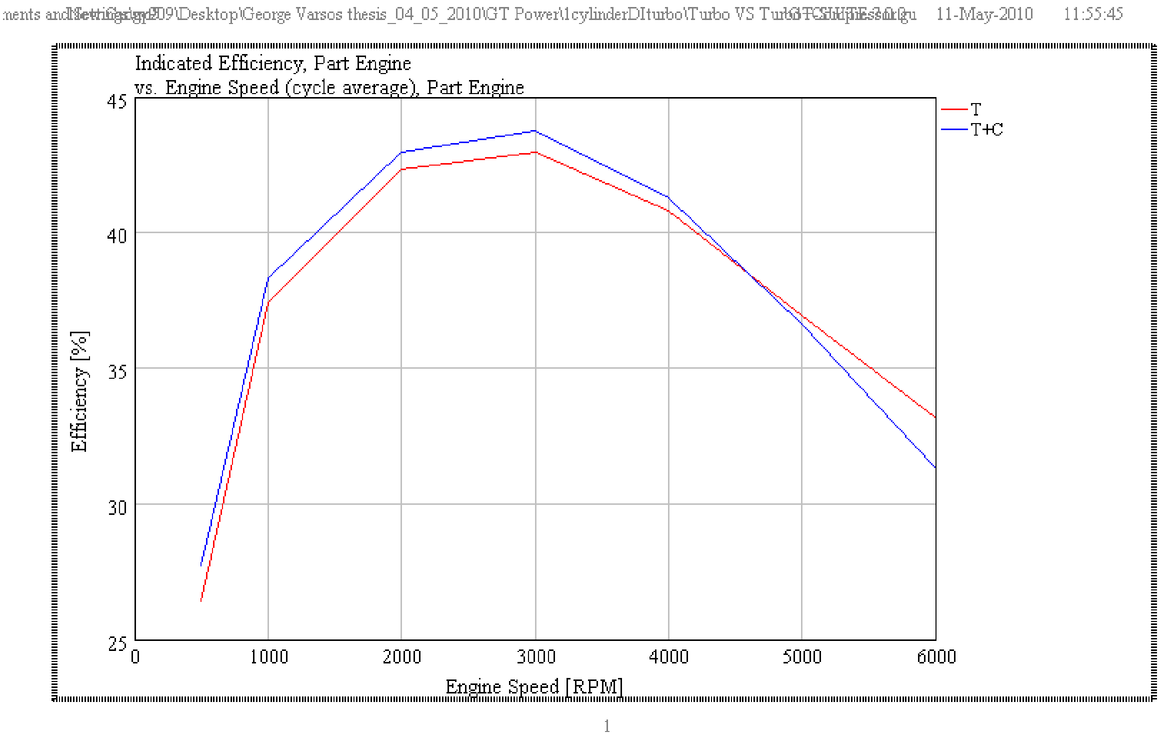


Figure 37: Comparison of Efficiency between a Turbo and a combined turbocharged engine

Conclusions:

1) Similar to slightly improved results for *brake horsepower* for the T+C model up until 4000RPM. After that the turbocharger is ahead having passed any turbo lag. The T+C cannot match –in this version at least- due to the energy taken from the engine to drive the compressor. That becomes notable in higher engine speed, that is why the compressor must be disengaged with a clutch at approximately 4000 RPM.
2) There is a vast improvement in torque right from the beginning for the T+C, showing clearly its benefits. Again after ~4000RPM the turbocharger moves ahead.
3) The overall efficiency is improved for the T+C model reaching 43,75% compared to 42,95% of the T. Again in higher RPM the energy consumption of the supercharger from the engine gives the advantage to the single T.

   Note that is only a comparative simulation. Improved accuracy would mean more accurate results, but on this level we need to compare the two models under exactly the same circumstances.

Chapter 9

Diploma Thesis Conclusions

A MATLAB code was developed for effectively simulating the function of a single cylinder engine, showcasing the use of the method of characteristics and its pulse waves. The results proved that this method is an effective tool for understanding the waves inside the intake manifold and their reaction with the boundaries they encounter.

After understanding the basic theory behind the method of characteristics the reader can use this code and easily alter it according to what simulation needs to be achieved. That may be the environment conditions, the length of the intake and exhaust pipes, the number of meshes or even the basic geometric figures of the engine to name a few. With that in mind, this code can be a very useful tool as for experimenting with certain data and see immediately the results, understanding exactly how these results are produced as this code is not a black box as some major commercial tools. It may not be as accurate as these tools but can provide a very good first picture as to how this engine can be modelled and simulated; hence it is also a fine first complex application of the method of characteristics for undergraduate students.

One of the basic conclusions reached is that with even constant engine speed there can be gains in engine efficiency by optimizing the length of the intake pipe as to better manage the movement of the compression and expansion waves. More specifically, it was showcased that for our application there can be an average increase of pressure inside the cylinder by tripling the intake length by ***0.54% to normal length*** and consequently ***0.23% to double length***.

Besides the results that can be seen in the corresponding section, two ideas were suggested for improving the efficiency of the engine by better exploiting the waves' behaviour inside the intake manifold. The ideas of the *'two-pipe-two valve system'* and the variable intake length according to the changing boundary of the closing valve have been described and maybe can provide a starting reference point for experiments.

Furthermore, a comparison was made between same engines with different types of aspiration, using the commercially available black box simulation program, GT-Power. It should be noted that it is not possible to simulate exactly the same conditions with our MATLAB code and GT-Power as in the second one it is obligatory to insert fuel facts whereas our code's results do not take account fuel burn as we are interested more in the waves produced and how they change the performance of the engine.

Finally, some code parts were developed for non-isentropic flow as a future reference for simulating the engine with variable entropy levels which, as has been stated, is indeed more accurate but is intended more on the exhaust manifold and not on the intake manifold with which this diploma thesis is concerned.

Future thesis could be based on this diploma thesis for adding more cylinders and thus simulate in this way a more complex and larger engine. The main point of concern will be to connect all the inlet pipes to the plenum, and the exhaust pipes too, while taking into account the valve timing which means different kind of waves are produced each time. The management of these waves will be the key for improving the efficiency of the engine and the method of characteristics showcased here can help as a simulation tool towards achieving this goal.

Chapter 10

Future Developments in Modelling Unsteady Flows in Engine Manifolds

[Winterbone and Pearson, 2000]

Using the method showcased in this diploma thesis most of the basic phenomena in intake and exhaust manifolds can be analysed with a high degree of accuracy. The next step in improving this method, and correspondingly its accuracy is by using multidimensional turbulence models. However, the difficulties in implementing these models are derived from (i) the general complexity in order to create a net-grid for describing the geometry of a whole engine system (ii) the time necessary to produce a realistic simulation of the engine over the number of cycles required to achieve cyclic conditions and (iii) limitations in the modelling of unsteady flow in which Mach number is the dominant parameter and the flow-field is traversed by pressure waves.

As a first step, the model should now include variation in the composition and properties of the unsteady one-dimensional gas flow. Winterbone and Pearson have developed a system of equations to allow for changes in temperature and composition of the gas as it flows through the engine's compartments, thus a non-perfect gas is being considered. The second basic difference for non-perfect gases compared to perfect ones is that the internal energy in no longer a linear function of temperature.

In general, the linear characteristics of a perfect gas are replaced by a series of curves when the gas is non-perfect. Benson [1] suggested that if a fine characteristic mesh is used the curves can be replaced by straight lines with their gradients being determined by a new compatibility equation:

$$da \pm \frac{(k-1)}{2}\left[\frac{1}{1-\left(\frac{a}{2k}\right)\left(\frac{dk}{da}\right)}\right]du = 0$$

There are various methods to describe the variation in the composition of the gas; they will not be showcased here, however it must be stated that by using these methods we can introduce fuel in the flow with all the benefits for the accuracy of the simulation that may mean.

Finally, a second step would be to have a two-dimensional flow simulation of wave propagation, thus further increasing the accuracy and the complexity of the problem. The notion with which a one-dimensional model can be converted in order to cope with more complex geometries is to include this kind of 2-D model as a local multi-dimensional region within a one-dimensional wave action simulation flow in engine manifolds.

Chapter 11

APPENDIX

Main Program [6]

```
%% Main Program for simulation of the flow in a 1 cylinder engine
% Diploma Thesis- George V. Varsos March-October 2010

% Boundary conditions: reservoir -> cylinder -> nozzle

%Intake pipe consists of M=15 meshes



global aair aref avo aza aze conrat CR cycle delz dPcdt...
     evo Fair Fcyl Fexh Fp k mc NI NE Pa Pc Pcr PREF rev neng...
     stroke theta thetarad Vc xref zcalc zrev Ta ac DC Tc

 %% Step 1:Initial data

M=30; ss = M+1; % Total Number of Meshes
k = 1.4; % Ratio of Specific Heat
q =(k-1)/(2*k); q1 = 1/q; aa = (3-k)/(2*(k-1)); bb =(k+1)/(2*(k-
1));%constants
zmax = 500;%Timing

 %% Step 2: Constants for specific flow conditions

 %% Manifolds

 PRI = 1;
 Pa= 1.65; % We consider the air pressure to be compressed by some way that
does not affect our simulation
 Ta = 360;  % air temperature

 % Lengths and diameter

XPIPE1=0.609; % length of intake pipe [the limit for this code is 3 times
the given length]
XPIPE2 = 0.609; % length of exhaust pipe
DP = 0.0508; % diameter of intake and exhaust manifold

% Exhaust Valve Opening

evo = 136; %exhaust valve opening from TDC in CA deg

%% Inlet Valve
```

```
aair = sqrt(k*287*Ta); % speed of sound for inlet manifold
avo = 346; %Inlet Valve opening from TDC in CA deg.
NI = 33; %Number of steps
aza =
[0,4,12,20,28,36,44,52,60,68,76,84,92,100,108,112,124,128,136,144,152,160,1
68,176,184,192,200,208,216,224,232,236,720]; %angle from opening
Fair
=[0,0.00001554,0.00004036,0.00009555,0.000186,0.00031737,0.00046301,0.00060
249,0.00073072,0.00084639,0.0009471,0.00103171,0.0011001,0.00115018,0.00118
184,...

0.00118862,0.00118862,0.00118184,0.00115018,0.0011001,0.00103171,0.0009471,
0.00084639,0.00073072,0.00060249,0.00046301,0.00031737,0.000186,0.00009555,
...
    0.00004036,0.00001554,0,0]; %Inlet Valve area at individual angles

%% Exhaust Valve

NE = 33; %Number of steps
aze = [0,
1,9,17,25,33,41,49,57,65,73,81,89,98,105,113,117,125,133,141,149,157,165,17
3,181,189,197,205,213,221,229,231,720]; %angle from opening
Fexh =
[0,0.00001408,0.00003658,0.00008664,0.00016884,0.00028851,0.00042153,0.0005
4926,0.00066694,0.00071524,0.00086606,0.00094408,0.00103227,0.00106908,...

0.001089,0.00109185,0.00109185,0.001089,0.00106908,0.00103227,0.00094408,0.
00086606,0.00071524,0.00066694,0.00054926,0.00042153,0.00028851,0.00016884,
...
    0.00008664,0.00003658,0.00001408,0,0];%Exhaust Valve area at individual
angles
Fp = pi()*0.25*(DP)^2; %exhaust pipe area
Fexh = Fexh./(Fp); %as a ratio
PHI = 0.5;%Exhaust manifold area ratio (Nozzle function)


%% Cylinder

Pcr = 3.593 ; % cylinder pressure at evo
Tc = 762; % cylinder temperature at evo
ac = sqrt(k*287*Tc); % speed of sound at cylinder
cycle = 4; % four stroke engine
neng = 33.333333; % engine speed in rev/s
DC = 0.125 ; % cylinder diameter
stroke = 0.13;
conrod = 0.273; %connecting rod m
CR = 14; % nominal compression ratio based on swept volume
conrat = (2*conrod)/(stroke) ;


%% Step 3: Reference Values

PREF = 1; Pc = Pcr; Tref = Tc*(PREF/Pc)^(2*q) ;
aref = (k*287*Tref)^(1/2);
xmesh1 = XPIPE1/M;
xmesh2 = XPIPE2/M;
xref = 0.058; % an arbitrary value, based on testing with various values
```

```
%% Specific boundary constants

P0 =  1.65; %for inflow B.C. pressure of reservoir
A0 = (P0/PREF)^q;


%% Step 4: Initial conditions at z = 0

theta = evo; zout(1) = theta; %starting angle is Exhaust Valve Opening
zcalc = 0;zrev = 0;rev =1;
vout(1,1:M) =0;%gas velocity at Z=0

% Pressure at Z=0

% We consider for the INTAKE pipe -Meshes 1 to 15- at Z=0 the pressure to
% be 1.65 bar

% We consider for the EXHAUST pipe -Meshes 16 to 30- at Z=0 the pressure to
% be 1.0 bar

Pout(1, 1:15) = real(Pa); % Intake
Pout(1, 16:M) = real(PRI); % Exhaust

% Wherever there is lambda and beta we have SS

% Initial Values for Riemann Variables, lambda and beta

% Intake

lambda(1,1:16) = (P0/PREF)^q;
beta(1,1:16) = (P0/PREF)^q;

% Exhaust

lambda(1,17:ss) = (PRI/PREF)^q;
beta(1,17:ss) = (PRI/PREF)^q;


%% Initial values for dPdt, Vc and dVdt Eq 3.33, 3.34, 3.35 (Benson)

thetarad = (theta*pi())/(180); % transforming theta into rad/s
fnn = sqrt(conrat^(2)-sin(thetarad)^2);
x = 0.5*stroke*(1 + conrat - fnn - cos(thetarad));
Fcyl = 0.25*pi()*DC^2; % cylinder cross sectional area
Vc = Fcyl*(x + (stroke)/(CR-1));%Eq3.34
mc = (Pc*Vc*10^5)/(287*Tc); %Initial Mass in cylinder
dxdt = 0.5*stroke*sin(thetarad)*(1 + cos(thetarad)/fnn);
dthdt = 2*pi()*neng;
dVcdt = Fcyl*dxdt*dthdt;%Eq. 3.35
dPcdt = -1*(k*Pc*dVcdt)/(Vc);%Eq. 3.33

%% Initiation of the Computational Core of the Main Program

delx1 = xmesh1/xref; % Intake
delx2 = xmesh2/xref; % Exhaust
z(1) = 0;r = 1;kk=1;AA = 0;
```

```
%% Step 5: Determining value of delta Z using stability criteria

while zcalc<=zmax
    for s = 1: 16
        A(r,s) = (lambda(r,s) + beta(r,s))/2;%Equation 2.18 (Benson)
        U(r,s) = (lambda(r,s) - beta(r,s))/(k-1);
        delz1(r,s) = delx1/(A(r,s) + abs(U(r,s)));%Stability Criteria for
intake length
    end
    for s = 17: ss
        A(r,s) = (lambda(r,s) + beta(r,s))/2;%Equation 2.18 (Benson)
        U(r,s) = (lambda(r,s) - beta(r,s))/(k-1);
        delz2(r,s) = delx2/(A(r,s) + abs(U(r,s)));%Stability Criteria for
exhaust length
    end

 %% Based on the two stability criteria, we now consider a common delz for
all the meshes

    delz1m= min(abs(delz1(r,:)));
    delz2m= min(abs(delz2(r,:)));
    delz= max(delz1m,delz2m); % We take the common delz as the maximum of
the two to avoid the situation
    %of either delz1 or delz2 to be zero


%% Step 6: We are using equations (2.22) and (2.23) by Benson to determine
Riemann variables at every mesh point

% lambda

    for s = 2:ss %% Equation (2.22)
        lambda(r+1,s) = lambda(r,s) + (bb*lambda(r,s-1)-aa*beta(r,s-1))*...
            (lambda(r,s-1)-lambda(r,s))*delz;
    end

    % We start from s=2 because at s=1 we have a boundary condition

% beta

    for s = 1:ss-1 %% Equation (2.23)
        beta(r+1,s) = beta(r,s) + (bb*beta(r,s+1)-aa*lambda(r,s+1))*...
            (beta(r,s+1)-beta(r,s))*delz;
    end

    % We end at s=ss-1 because at s=ss we have a boundary condition

%% Step 7: BOUNDARY conditions (** SEE EXPLANATION AT DIPLOMA THESIS REPORT
**)

lambda(r+1,1) = Inflow(beta(r+1,1),A0,k); % reservoir as an inlet


[cc,AA] = cylinder(beta(r+1,16),kk,AA); % cylinder
lambda(r+1,16) = cc;

beta(r+1,ss) = nozzle(lambda(r+1,ss),PHI,k); % exit from cylinder to a
nozzle
```

```
%% Second configuration of the exhaust model

%beta(r+1,17) = nozzle(lambda(r+1,17),PHI,k); % exit from cylinder to a
nozzle

%lambda(r+1,21) = nozzle(beta(r+1,21),PHI,k); % from nozzle to open end is
the outlet pipe (exhaust)

%beta(r+1,ss) = open(lambda(r+1,ss)); % exit to environment


kk = kk+1;
%% Update Z
    z(r+1) = z(r) + delz;
%% Step 8: Determine pressure at every point for given r
    for s = 1:M
        Pout(r+1,s) = ((((lambda(r+1,s)+ beta(r+1,s))/2)^q1))*PREF;
        zout(r+1) = theta;
        vout(r+1,s) = ((lambda(r+1,s)- beta(r+1,s))/(k - 1))*aref;%Equation
2.18 (Benson)

    end

    for s=1:14

      %% Step 9 :Construction of position and state diagrams

   %Position diagram
       dxdtlambda=(vout(r+1,s)/aref)+aref;
        dxdtbeta=(vout(r+1,s)/aref)-aref;
   %State diagram

   % slopes:
        dcdulambda=-(k-1)/2;
        dcdubeta=(k-1)/2;

    dcl(s+1)=dcdulambda*(vout(r+1,s+1)-vout(r+1,s));
    dcb(s+1)= dcdubeta*(vout(r+1,s+2)-vout(r+1,s));
    cl=dcl/aair;
    cb=dcb/aair;
    uc(r+1,s)=vout(r+1,s)/aair;
    end

    r = r+1;
end

%% Step 10: Figures

dcl;
dcb;

% Animation of State and Position diagrams

yl=dcdulambda*uc;
yl2=yl+abs(29.5673/aair);
yl3=yl2+abs(29.5673/aair);
yl4=yl3+abs(29.5673/aair);
```

```
yl5=yl4+abs(29.5673/aair);
yl6=yl5+abs(29.5673/aair);
yb=dcdubeta*uc;
yb2=yb+abs(29.5673/aair);
yb3=yb2+abs(29.2445/aair);
yb4=yb3+abs(29.5673/aair);
yb5=yb4+abs(29.2445/aair);
yb6=yb5+abs(29.2445/aair);
figure;plot(uc,yl,uc,yl2,uc,yl3,uc,yl4,uc,yl5,uc,yl6,uc,yb,uc,yb2,uc,yb3,uc
,yb4,uc,yb5,uc,yb6);
title('State diagram')
xlabel('u/c');ylabel('c non-dimension')



x=(0:.0001:XPIPE1);

y=dxdtlambda*x;
y2=25.21+25+(dxdtbeta*(x-0.5)); % User must change the added numbers
manually in order to create the position diagram, according to the
continuity of the characteristic lines.
y3=y+280.5; % User must change the added numbers manually in order to
create the position diagram, according to the continuity of the
characteristic lines.
y4=y2+280.9; % User must change the added numbers manually in order to
create the position diagram, according to the continuity of the
characteristic lines.
y5=y+561.4; % User must change the added numbers manually in order to
create the position diagram, according to the continuity of the
characteristic lines.
y6=y2+561.4; % User must change the added numbers manually in order to
create the position diagram, according to the continuity of the
characteristic lines.
figure; plot(x,y,x,y2,x,y3,x,y4,x,y5,x,y6);
title('Position diagram from moment 0')
xlabel('Length-X');ylabel('Time');
figure; plot(x,y+346,x,y2+346,x,y3+346,x,y4+346,x,y5+346,x,y6+346);
title('Position diagram from IVO')
xlabel('Length-X');ylabel('Time-CAD');
legend('compression wave','expansion wave','compression wave','compression
wave','expansion wave','compression wave');

%% State diagram
% In the state plane the characteristics for a gas with k=1.4 are two
% families of straight lines with slopes +-0.2


figure;plot (zout(1:end-1),AA,'r-'); xlabel('CAD');ylabel('Pressure(bar)');
title('Cylinder pressure during one engine cycle');
figure; plot(zout(1:end-1),  Pout(1:end-1,M-4),zout(1:end-1),Pout(1:end-
1,M));xlabel('CAD');ylabel('Pressure (bar)')
legend('Exhaust manifold inlet','Exhaust manifold outlet')
title('Exhaust manifold pressure')
```

```
%% Pressure diagrams

figure; plot(zout(1:end-1)+210,Pout(1:end-1,3),zout(1:end-1)+210,Pout(1:end-1,10),zout(1:end-1)+210,Pout(1:end-1,15),zout(1:end-1),AA);
legend('Start of the intake pipe-3','Middle of the intake pipe-10','Prior entering the cylinder-15','Cylinder Pressure');xlabel('CAD');ylabel('Pressure (bar)');title('Complete Pressure Figure');

figure; plot(zout(1:end-1)+210,Pout(1:end-1,1),zout(1:end-1)+210,Pout(1:end-1,2),zout(1:end-1)+210,Pout(1:end-1,3));
legend('Mesh-1','Mesh-2','Mesh-3');xlabel('CAD');ylabel('Pressure (bar)');title('Pressure of the First Three Meshes of the Intake Pipe');

%% Velocity diagrams

figure; plot(zout(1:end-1)+210,vout(1:end-1,3),zout(1:end-1)+210,vout(1:end-1,12),zout(1:end-1)+210,vout(1:end-1,13),zout(1:end-1)+210,vout(1:end-1,14),zout(1:end-1)+210,vout(1:end-1,15),zout(1:end-1)+210,vout(1:end-1,16));
legend('3','12','13','14','15','16');xlabel('CAD');ylabel('Velocity (m/s)');title('Velocity at various Meshes of the Intake Pipe');

figure; plot(zout(1:end-1)+210,vout(1:end-1,1),zout(1:end-1)+210,vout(1:end-1,2),zout(1:end-1)+210,vout(1:end-1,3));
legend('Mesh-1','Mesh-2','Mesh-3');xlabel('CAD');ylabel('Velocity (m/s)');title('Velocity of the First Three Meshes of the Intake Pipe');

%% Carpet Diagrams (for intake manifold)

%Pressure

figure;colormap 'Jet';material shiny ;rend3d2(Pout,zout+210,'Pressure (bar)',XPIPE1);

% Velocity

figure;colormap 'Jet';material shiny ;rend3d2(vout,zout+210,'Velocity(m/s)',XPIPE1);


figure;plot(aze,Fexh);xlabel('Degrees from opening');ylabel('Exhaust Valve Area as a fraction');title('Area-Angle from opening of the exhaust valve')
figure;plot(aza,Fair);xlabel('Degrees from opening');ylabel('Area(m^2)');title('Area-Angle from opening of the inlet valve')
figure;plot(aze+136,Fexh);xlabel('Degrees');ylabel('Exhaust Valve Area as a fraction');title('Area-Angle of the exhaust valve-1 cycle')
figure;plot(aza+346,Fair);xlabel('Degrees');ylabel('Area(m^2)');title('Area-Angle of the inlet valve-1 cycle')

disp('ok')
```

Inflow [6]

```
function clout = Inflow(clin,A0,k)
%% Determine constants
A =(3-k)/(k+1);B = 4*(k-1)/(k+1);
C= ((k+1)/(3-k))*clin;%Equation 3.6(condition for chocked flow)
if (clin >= A0) %% There is reverse flow
    clout =2*A0-clin; %% Equation 3.5 is used(reverse flow)
else
     clout = (A*clin+sqrt((B*A0^2-(1-A^2)*clin^2)));%Equation 3.4 is used
     if (clout > C) %%there is chocked flow
        clout = C;
     end;
end;
```

Cylinder [6]

```
function [clout,MU] = cylinder(clin,kk,AA)

global aair aref avo aza aze conrat CR cycle delz dPcdt...
     evo Fair  Fexh  k mc  Pa Pc neng...
     stroke theta thetarad Vc xref zcalc zrev ac DC Tc
%% Step1: Time calculations
dt = delz*(xref/aref);
dthrad = 2*pi()*neng*dt; % every time step dt. theta increases by dthrad
thetarad = thetarad + dthrad;% start from theta = evo
theta = (thetarad*180)/(pi()); % retransform theta in degs
if theta > 180*cycle  % i.e. for a 4-stroke engine cycle as soon as
%theta is greater than 720 we need to start again
theta = theta - 180*cycle;
end
zcalc = zcalc + (dthrad*180)/pi();
zrev = zrev + (dthrad*180)/pi();
%start with exhaust valve
alphae = theta -evo;
%% Step 2 determine exhaust valve area
if alphae < 0;
    alphae = 0;
end
i = 2;
while alphae > aze(i) % open valve
    i = i+1;
end
PSI = Fexh(i-1)+(Fexh(i) -...%exhaust area updated
    Fexh(i-1))*((alphae-aze(i-1))/(aze(i)-aze(i-1)));
%% Step 3 Determine inlet valve area
alphaa = theta -avo;
if alphaa < 0;
    alphaa = 0;
end
i = 2;
while alphaa > aza(i) % when alphaa>0 and <aza we have flow
    i = i+1;
end
F2 = Fair(i-1)+((Fair(i)-...%Inlet area updated
Fair(i-1))*(alphaa-aza(i-1)))/( aza(i)-aza(i-1));
%% Step4: Call in boundaries and determine mass of air in cylinder
Pc = Pc + dPcdt*dt;
```

```
MU(1,kk)=Pc;
if kk >1;
    MU(1,1:kk-1) = AA;
end
vv = valve(clin,PSI); %Call in valve boundary condition
clout =vv(1,1) ;%Output the value of lambda out
dmedt = vv(1,2);%Output the value of exhaust mass <<<<<FLOW OUT>>>>>
dmadt = massrec(Pa, Pc, aair, ac,k,F2);%Determine mass <<FLOW IN>>
massrecfunc.
if dmadt < 0
    disp ('ok')
end
mc = mc + (dmadt-dmedt)*dt;%Update mass in cylinder
%% Step 5: Determine cylinder volume
fnn = sqrt(conrat^(2)-sin(thetarad)^2);
x = 0.5*stroke*(1 + conrat - fnn - cos(thetarad));
Fcyl = 0.25*pi()*DC^2; % cylinder cross sectional area
Vc = Fcyl*(x + (stroke)/(CR-1));  %Cylinder volume
dxdt = 0.5*stroke*sin(thetarad)*(1 + (cos(thetarad))/fnn);
dthdt = 2*pi()*neng;
dVcdt = Fcyl*dxdt*dthdt;%Change in cylinder volume
%% Step 6: Calculate new Tc and new dPcdt
Tc = (Pc*(10^5)*Vc)/(287*mc);
ac = sqrt(287*k*Tc);
if dmadt == 0; A = 0;%No flow
else if dmadt > 0; A = aair;%flow in
    else A = ac;%flow out
    end
end
dPcdt=(-k*dVcdt*(Pc/Vc))-(dmedt*ac^(2))/(Vc*10^5)...
+((dmadt*(A^2))/(Vc*10^5));
```

Nozzle [6]

```
function clout = nozzle(clin,O,k)
t = 0.000001; % Accuracy
A1 = 4/(k-1); A2 = (k-1)/2; A3 = 2*(k-1); A4 = 2/(k+1);
A5 = (k+1)/(k-1);A6 = 2/(k-1);
%% Step 1, Check whether there is reverse flow
if clin<1
    clout = 2-clin;  %there is reverse flow
else
%% Step 2 Calculate for subsonic flow
A = 0.5*(clin + 1);%first estimate of A
dA = 0.25*(clin - 1);%step change of A
sum = (((A^A1)-O^2)*(clin-A)^2)-A2*((A^2)-1)*O^2;
while abs(sum)>=t
if sum < 0
    A = A - dA;%Make A smaller
else if sum > t
        A = A + dA;%Make A bigger
    end
end
sum = (((A^A1)-O^2)*(clin-A)^2)-A2*((A^2)-1)*O^2;
dA = dA/2;%make step change smaller for next step
end
```

```
clout = 2*A - clin;%Calculate clout
U = (clin-clout)/(k-1);%Similarly calculate U

%% Step 3: Test for sonic or subsonic <<<<<<SONIC TEST>>>>>>>
Ut = (U/O)*A^A3;%Determine Ut
if Ut < 1 % i.e we have a subsonic flow
    clout = 2*A - clin;

else   %% Solve for sonic flow
    At = 0.5 + 0.5*sqrt(A4);%Initial guess
     dAt = (1-sqrt(A4))/4;%Step change
     Y = (At)^-1;
     sum = (O^2)-(Y^(A1))*(A5-(A6*Y^2));%Substitute Y
     while abs(sum)>=t
 if sum < 0
     At = At - dAt;%make At smaller
 else if sum > t
         At = At + dAt;%make At bigger
     end
     Y = (At)^-1;
     sum = (O^2)-(Y^(A1))*(A5-(A6*Y^2));
     dAt = dAt/2;
 end
     end
%% Step 5: Obtained clout for sonic flow
 F =  (1-(At^(A5)*O*A2))/(1+(At^(A5)*O*A2));
 clout = F*clin;
 end
end
```

Valve [6]

```
function vv = Valve(clin,PSI)
global aref Fp k Pc PREF
%% Cylinder program
if PSI<=0 % --> Closed valve, modelled as a closed end
clout = clin;
dmedt = 0; %No mass
else
 %   if PSI > 1 %IF VALVE AREA GREATER THAN PIPE AREA set PSI=1 (max)
 %   PSI=1;
 %   end

%% Constants
s = sqrt(2/(k+1));%Sonic boundary ;
A1 = (k-1)/(2*k); A4 = 2/(k+1); A5 = (k+1)/(2*(k-1));A6 = (2*k)/(k-1);
A3 = 2/(k-1);
t = 0.000001;%Accuracy

%% calculate PIA recall condition for inflow, outflow or no flow
%% with PIA we determine whether we have inflow or outflow
PIA = clin*(PREF/Pc)^(A1);
    if PIA < 1 %We have OUTFLOW
    %% Determine the value of U from
    U = (s*(sqrt(1+(k^(2)-1)*PSI^2)-1))/(PSI*(k-1));%Equation 4.8
    C = (sqrt(1-(U^2)/A3))/(s);%Calculate entropy Equation 3.32
    picr = s - (U)/(A3*C);%picr obtained from Equation 4.6
```

```
    % We need to reset the entropy as we have calculated C for Critical
    C = 1/(PIA);

    %% TEST for sonic or subsonic flow

        if (PIA > picr) % -------------------------------->>>>The flow is
SUBSONIC
        %% Start iteration to solve f(pi) for subsonic flow
        pix = (1 + s)/2; % First attempt
        dpix = (1 - s)/2; % Step increase or decrease


        B = C*(pix-PIA);
        subout = (PSI/pix)*sqrt(A3*(pix^(-2)-1))-((A3*B)/(1-A3*B^2));
        C = (sqrt(1 - A3*B^(2)))/(pix);
            while dpix >t
           % while abs(subout)>t

                if subout < 0
                pix = pix - dpix; %Make pix smaller
                else if subout > t
                    pix = pix + dpix;%Make pix bigger
                    end
                end
            %if dpix>t
            B = C*(pix-PIA);
            subout = (PSI/pix)*sqrt(A3*(pix^(-2)-1))-((A3*B)/(1-A3*B^2));
            C = (sqrt(1 - A3*B^(2)))/(pix);
            dpix = dpix/2;

            end %repeat process until abs(pix) <=t

        else if (PIA <= picr) % --------------------------->>>> The flow is
SONIC
            %% Start iteration to solve f(pi) for sonic flow
            pix = (s + PIA)/2; %First attempt
            dpix = (s - PIA)/2; % Step increase or decrease


            B = C*(pix-PIA);
            E = PSI*(A4)^(A5);
            sonout = pix^(A6)-E*((1-A3*B^2)/(A3*B));
            C = (sqrt(1 - A3*B^(2)))/(pix);

%while dpix >t
            while abs(sonout)>t

                if sonout < 0
                pix = pix + dpix;
                else if sonout > 0
                    pix = pix - dpix;
                    end
                end
            B = C*(pix-PIA);
            sonout = pix^(A6)-E*((1-A3*B^2)/(A3*B));
            C = (sqrt(1 - A3*B^(2)))/(pix);
            dpix = dpix/2;

                if dpix < 0.00000001%set a limit to dpix
```

```
                sonout = 0;
                end
            end %Repeat process until abs(pix) <=t
            end
        end
        %% Obtain outputs
        clout = 2*pix*(Pc/PREF)^(A1) - clin;
        A = 0.5*(clin + clout);
        U = (clin-clout)/(k-1);

        dmedt = -k*(PREF/aref)*Fp*(U/(A^2))*(A^((2*k)/(k-1)))*1*10^5;

    else if PIA >1  %% We have INFLOW
    % There is inflow thus nozzle conditions apply
    clinv = clin*(PREF/Pc)^(A1);
    cloutn = nozzle (clinv,PSI,k);
    clout = cloutn*(Pc/PREF)^(A1);
    A = 0.5*(clinv + clout);
    U = (clinv-clout)/(k-1);
    dmedt = -k*(PREF/aref)*Fp*(U/(A^2))*(A^((2*k)/(k-1)))*1*10^5;

    else %PIA = 1 %then we have NO FLOW
    clout = clin;
    dmedt = 0;
        end
    end
end

vv(1,1)= clout;
vv(1,2)= dmedt;
```

Massrec (for calculation of the cylinder intake air mass) [6]

```
function  dmadt = massrec(Pair, Pcyl, aair, acyl,k,F2)

rm = Pcyl/Pair;
if rm > 1; rm = 1/rm;
elseif rm < 1; rm = rm;
end
    A1=(k/(k-1)); A2=(2*k^2)/(k-1); A3=1/A1;A4=(k+1)/(2*(k-1));
    son = (2/(k+1))^A1; %Condition obtained, Eq.3.19
    if rm > son % We do not have chocked flow
        frm = sqrt(A2*(rm^(2/k))*(1-rm^(A3))); %Eq. 3.30
    else frm = k*(2/(k+1))^(A4);%Choked flow Eq. 3.31
    end
    if Pcyl > Pair  % We have outflow
        dmadt = (-1*Pcyl*F2*frm*10^5)/(acyl);
    else dmadt = (Pair*F2*frm*10^5)/(aair);
    end
if rm == 0; dmadt = 0;
end
```

Rend3d2 (for creating the two 3-D carpet diagrams) [6]

```
function rend3d2(P,Z,T,pp)
[xx,yy] = size(P);

xx = max(xx,yy);
yy =min(xx,yy);

hhx = linspace(min(min(Z)),max(max(Z)),xx);
hhy = linspace(min(min(P)),max(max(P)),yy);
hpipe= linspace(0,pp,yy);
[X,Y] =meshgrid(hhx,hpipe);


%C=['m' 'g'];% 'c' 'y' 'b' 'm' 'r' 'g' 'm' 'b' 'c'];
surf(X',Y',P);
shading interp

lightangle(-35,30)
set(gcf,'Renderer','zbuffer')
alpha('color');
alphamap('rampdown');
alpha(0.8);
%assignin('base','cl',0);
camlight left;
camzoom(1);
lighting phong
hidden off
% title(T);
xlabel('CA After TDC'), ylabel('length (m)'), zlabel(T);
hold off
```

Non-Isentropic Codes

(Proposed as a basis for future references)

MainProgramEntropy

```
%% Main Program for simulation of the flow in a 1 cylinder engine with
%% entropy change

% Diploma Thesis- George V. Varsos March-October 2010

% Boundary conditions: reservoir -> cylinder -> nozzle

%Intake pipe consists of M=15 meshes
```

```
function MainProgramEntropy

global aair aref avo aza aze conrat CR cycle delz dPcdt...
     evo Fair Fcyl Fexh Fp k mc NI NE Pa Pc Pcr PREF rev neng...
     stroke theta thetarad Vc xref zcalc zrev Ta ac DC Tc f

%% Step 1:Initial data

M=30; ss = 1+M; k = 1.4; CV=287.3/(k-1); CP=(k*287.3)/(k-1);%% Number of
meshes used, ratio of specific heats and thermodynamic constants
q =(k-1)/(2*k); q1 = 1/q; aa = (3-k)/(2*(k-1)); bb =(k+1)/(2*(k-
1));%constants
zmax = 500;%Timing
f=0.0025; % pipe wall friction factor [arbitrary]
R=287.3; % air constant
B=bb;Ae=aa; % ADDED FOR NON-HOMENTROPIC FLOW
TW1=300; % arbitrary value!
TG1=360; % gas temperature at the inlet pipe (same to intake fluid
temperature)
TW2=450; % arbitrary value from GTPOWER
TG2=762; % gas temperature at the exhaust pipe (I consider it equal to gas
temperature at EVO)


%% Step 2: Constants for specific flow conditions

 %% Manifolds

 PRI = 1;
 Pa= 1.65; % We consider the air pressure to be compressed by some way that
does not affect our simulation
 Ta = 360;  % air temperature

 % Lengths and diameter

XPIPE1=0.609; % length of intake pipe [the limit for this code is 3 times
the given length]
XPIPE2 = 0.609; % length of exhaust pipe
DP = 0.0508; % diameter of intake and exhaust manifold

% Exhaust Valve Opening

evo = 136; %exhaust valve opening from TDC in CA deg

%% Inlet Valve

aair = sqrt(k*287*Ta); % speed of sound for inlet manifold
avo = 346; %Inlet Valve opening from TDC in CA deg.
NI = 33; %Number of steps
aza =
[0,4,12,20,28,36,44,52,60,68,76,84,92,100,108,112,124,128,136,144,152,160,1
68,176,184,192,200,208,216,224,232,236,720]; %angle from opening
Fair
=[0,0.00001554,0.00004036,0.00009555,0.000186,0.00031737,0.00046301,0.00060
249,0.00073072,0.00084639,0.0009471,0.00103171,0.0011001,0.00115018,0.00118
184,...

0.00118862,0.00118862,0.00118184,0.00115018,0.0011001,0.00103171,0.0009471,
```

```
0.00084639,0.00073072,0.00060249,0.00046301,0.00031737,0.000186,0.00009555,
...
    0.00004036,0.00001554,0,0]; %Inlet Valve area at individual angles

%% Exhaust Valve

NE = 33; %Number of steps
aze = [0,
1,9,17,25,33,41,49,57,65,73,81,89,98,105,113,117,125,133,141,149,157,165,17
3,181,189,197,205,213,221,229,231,720]; %angle from opening
Fexh =
[0,0.00001408,0.00003658,0.00008664,0.00016884,0.00028851,0.00042153,0.0005
4926,0.00066694,0.00071524,0.00086606,0.00094408,0.00103227,0.00106908,...

0.001089,0.00109185,0.00109185,0.001089,0.00106908,0.00103227,0.00094408,0.
00086606,0.00071524,0.00066694,0.00054926,0.00042153,0.00028851,0.00016884,
...
    0.00008664,0.00003658,0.00001408,0,0];%Exhaust Valve area at individual
angles
Fp = pi()*0.25*(DP)^2; %exhaust pipe area
Fexh = Fexh./(Fp); %as a ratio
PHI = 0.5;%Exhaust manifold area ratio (Nozzle function)


%% Cylinder

Pcr = 3.593 ; % cylinder pressure at evo
Tc = 762; % cylinder temperature at evo
ac = sqrt(k*287*Tc); % speed of sound at cylinder
cycle = 4; % four stroke engine
neng = 33.333333; % engine speed in rev/s
DC = 0.125 ; % cylinder diameter
stroke = 0.13;
conrod = 0.273; %connecting rod m
CR = 14; % nominal compression ratio based on swept volume
conrat = (2*conrod)/(stroke) ;

%% Step 3: Reference Values

PREF = 1; Pc = Pcr; Tref = Tc*(PREF/Pc)^(2*q) ;
aref = (k*287*Tref)^(1/2);
xmesh1 = XPIPE1/M;
xmesh2 = XPIPE2/M;
xref = 0.058; % an arbitrary value, based on testing with various values


%% Specific boundary constants

P0 =  1.65; %for inflow B.C. pressure of reservoir
A0 = (P0/PREF)^q;



%% Step 4:Initial conditions at z = 0

theta = evo; zout(1) = theta; %starting angle is Exhaust valve Opening
zcalc = 0;zrev = 0;rev =1;
vout(1,1:M) =0;%gas velocity at Z=0
```

```
Pout(1, 1:M) = real(PRI); %Pressure at Z=0

%Pressure at Z=0

% We consider for the INTAKE pipe -Meshes 1 to 15- at Z=0 the pressure to
% be 1.65 bar

% We consider for the EXHAUST pipe -Meshes 16 to 30- at Z=0 the pressure to
% be 1.0 bar

Pout(1, 1:15) = real(Pa); % Intake
Pout(1, 16:M) = real(PRI); % Exhaust

% Initial Values for lambda, beta and entropy level at time Z=0

% We set all the meshes to have the same initial entropy level as a basis

% Wherever there is lambda and beta we have SS

% Intake

lambda(1,1:16) = (P0/PREF)^q;
beta(1,1:16) = (P0/PREF)^q;
Aa(1,1:16)=abs(CV*log(P0/(1.65^1.4)));

% Exhaust

lambda(1,17:ss) = (PRI/PREF)^q;
beta(1,17:ss) = (PRI/PREF)^q;
Aa(1,17:ss)=abs(CV*log(PRI/(1.65^1.4)));



%% Initial values for dPdt, Vc and dVdt Eq 3.33, 3.34, 3.35 (Benson)

thetarad = (theta*pi())/(180); % transforming theta into rad/s
fnn = sqrt(conrat^(2)-sin(thetarad)^2);
x = 0.5*stroke*(1 + conrat - fnn - cos(thetarad));
Fcyl = 0.25*pi()*DC^2; % cylinder cross sectional area
Vc = Fcyl*(x + (stroke)/(CR-1));%Eq3.34
mc = (Pc*Vc*10^5)/(287*Tc); %Initial Mass in cylinder
dxdt = 0.5*stroke*sin(thetarad)*(1 + cos(thetarad)/fnn);
dthdt = 2*pi()*neng;
dVcdt = Fcyl*dxdt*dthdt;%Eq. 3.35
dPcdt = -1*(k*Pc*dVcdt)/(Vc);%Eq. 3.33

%% Initiation of the Computational Core of the Main Program

delx1 = xmesh1/xref; % intake
delx2 = xmesh2/xref; % exhaust
z(1) = 0;r = 1;kk=1;AA = 0;

%% Step 5: Determining value of delta Z using stability criteria

%% THE STABILITY CRITERIA ARE VALID ALSO FOR THE ENTROPIC STABILITY
CRITERION
```

```
while zcalc<=zmax

    disp('a097e running')

    for s = 1: 16
        A(r,s) = (lambda(r,s) + beta(r,s))/2;%Equation 2.18 (Benson)
        U(r,s) = (lambda(r,s) - beta(r,s))/(k-1);
        delz1(r,s) = delx1/(A(r,s) + abs(U(r,s)));%Stability Criteria for
intake length
    end
    for s = 17: ss
        A(r,s) = (lambda(r,s) + beta(r,s))/2;%Equation 2.18 (Benson)
        U(r,s) = (lambda(r,s) - beta(r,s))/(k-1);
        delz2(r,s) = delx2/(A(r,s) + abs(U(r,s)));%Stability Criteria for
exhaust length
    end

 % Based on the two stability criteria, we now consider a common delz for
all the meshes

    delz1m= min(abs(delz1(r,:)));
    delz2m= min(abs(delz2(r,:)));
    delz= max(delz1m,delz2m); % % We take the common delz as the maximum of
the two to avoid the situation
    %of either delz1 or delz2 to be zero

%% Step 6: Determine RIEMANN variables at every mesh space and time point

%% Variable LAMBDA 1

% We start from s=2 because at s=1 we have a boundary condition

    for s = 2:15

        % In the following equation we have the choice either to have
        % 'delz' or 'delz1' which is the corresponding stability criteria
for the
        % intake length. However, taken as a fact the global stability
        % criteria, we opt for using 'delz'. Same choice will be made for
        % all the other equations containing 'delz'.

        dxDX = (B*lambda(r,s+1)-Ae*beta(r,s+1))/((delx1/delz)+B*...
    (lambda(r,s+1)-lambda(r,s))-Ae*(beta(r,s+1)-beta(r,s))); %7.71 (Benson)


if dxDX < 0 then

    B=-b;
    Ae=-a;
   dxDX = (B*lambda(r,s+1)-Ae*beta(r,s+1))/((delx1/delz)+B*...
       (lambda(r,s+1)-lambda(r,s))-Ae*(beta(r,s+1)-beta(r,s))); %7.71
(Benson)
end

    lambdap=lambda(r,s+1)-dxDX*(lambda(r,s+1)-lambda(r,s)); %7.84 (Benson)
    betap=beta(r,s+1)-dxDX*(beta(r,s+1)-beta(r,s)); %7.85 (Benson)
    Aap=Aa(r,s+1)-dxDX*(Aa(r,s+1)-Aa(r,s)); % 7.86 (Benson)
    Xp=(XPIPE1-1)*delx1-dxDX*delx1; %7.90 (Benson)
```

```
    %% Changes to lambda

    % Area term (no area change here)

    C=0;
    Dodd=DP;
    dDdXD=2*C/(2*Dodd+C*(delx1+Xp)); %7.89 (Benson)
    % Inserted Xr=delx1
    % Dodd is the diameter at the odd end (pipe diameter)
    dlarea=-0.5*(lambdap+betap)*dDdXD*delz; %7.87 (Benson)


    % Heat transfer term (limited by velocity)


    qheat= 0.001+ 2*f*U(r,s)*CP*(TW1-TG1)/DP; % Heat transfer rate % added
0.001!
    dAa=((k-1)/2)*(Aa(r,s)/Ae^2)*((qheat*xref/(aref)^3)+...
        (2*f*xref*abs(U(r,s)^3)/DP))*delz;

    % Entropy term

    dlentropy=0.5*(lambdap+betap)*(dAa)/Aap; %7.81 (Benson) where dAa is
Aa'-Aa from compatibility equation

    % Friction term

    dlfriction=-(k-1)*(f*delx1/DP)*(xref/delx1)*(((lambdap+0.001-betap)/...
   (k-1))^(1/2))*((lambdap+0.001-betap)/abs(lambdap+0.0001-
betap))*((3*betap-lambdap)/...
(lambdap+betap))*delz; %equation 7.82 (Benson)  % % added 0.001!

dlheatransfer=2*k*(abs(lambdap+0.001-
betap)/(lambdap+betap))*(f*delx1/DP)*... % added 0.001!
(xref/delx1)*R*(1/aref^(1/2))*(TW1-TG1)*delz;

    % Total change in Lambda along a characteristic

     dl=dlarea+dlentropy+dlheatransfer+dlfriction; % 7.79

     lambda(r+1,s) =lambdap + dl;
     Aa(r+1,s)=Aap+dAa;

    end

%% Variable LAMBDA 2

for s = 17:ss-1

        dxDX = (B*lambda(r,s+1)-Ae*beta(r,s+1))/((delx2/delz)+B*...
    (lambda(r,s+1)-lambda(r,s))-Ae*(beta(r,s+1)-beta(r,s))); %7.71


if dxDX < 0 then

    B=-b;
```

```
    Ae=-a;
   dxDX = (B*lambda(r,s+1)-Ae*beta(r,s+1))/((delx2/delz)+B*...
       (lambda(r,s+1)-lambda(r,s))-Ae*(beta(r,s+1)-beta(r,s))); %7.71
end

    lambdap=lambda(r,s+1)-dxDX*(lambda(r,s+1)-lambda(r,s)); %7.84
    betap=beta(r,s+1)-dxDX*(beta(r,s+1)-beta(r,s)); %7.85
    Aap=Aa(r,s+1)-dxDX*(Aa(r,s+1)-Aa(r,s)); % 7.86
    Xp=(XPIPE2-1)*delx2-dxDX*delx2; %7.90

    % Area Change Term

    C=0 ;
    Dodd=DP;
    dDdXD=2*C/(2*Dodd+C*(delx2+Xp)); %7.89
    dlarea=-0.5*(lambdap+betap)*dDdXD*delz; %7.87


    % Heat transfer term

    qheat= 0.001+ 2*f*U(r,s)*CP*(TW2-TG2)/DP; % Heat transfer rate % added
0.001!
    dAa=((k-1)/2)*(Aa(r,s)/Ae^2)*((qheat*xref/(aref)^3)+...
        (2*f*xref*abs(U(r,s)^3)/DP))*delz;

    % Entropy change term

    dlentropy=0.5*(lambdap+betap)*(dAa)/Aap; %7.81


    % Friction term

    dlfriction=-(k-1)*(f*delx2/DP)*(xref/delx2)*(((lambdap+0.0001-
betap)/...  % added 0.001!
  (k-1))^(1/2))*((lambdap+0.001-betap)/abs(lambdap+0.001-betap))*((3*betap-
lambdap)/...
(lambdap+betap))*delz; %equation 7.82

 dlheatransfer=2*k*(abs(lambdap+0.001-
betap)/(lambdap+betap))*(f*delx2/DP)*... % added 0.001!
(xref/delx2)*R*(1/aref^(1/2))*(TW2-TG2)*delz;

   % Total change in LAMBDA along a characteristic

  dl=dlarea+dlentropy+dlheatransfer+dlfriction; % 7.79

   lambda(r+1,s) =lambdap + dl;
    Aa(r+1,s)=Aa(r,s)+dAa;

end


%% Variable BETA 1
% some terms are recalculated as for beta we have different limits (ss)

  for s = 1:15
```

```
        dxDX = (B*lambda(r,s+1)-Ae*beta(r,s+1))/((delx1/delz)+B*...
    (lambda(r,s+1)-lambda(r,s))-Ae*(beta(r,s+1)-beta(r,s))); %7.71


if dxDX < 0 then

    B=-b;
    Ae=-a;
   dxDX = (B*lambda(r,s+1)-Ae*beta(r,s+1))/((delx1/delz)+B*...
       (lambda(r,s+1)-lambda(r,s))-Ae*(beta(r,s+1)-beta(r,s))); %7.71
end

    lambdap=lambda(r,s+1)-dxDX*(lambda(r,s+1)-lambda(r,s)); %7.84
    betap=beta(r,s+1)-dxDX*(beta(r,s+1)-beta(r,s)); %7.85
    Aap=Aa(r,s+1)-dxDX*(Aa(r,s+1)-Aa(r,s)); % 7.86
    Xp=(XPIPE1-1)*delx1-dxDX*delx1; %7.90


    C=0; %=dDdX; % FOR VALVES AND NOZZLE (here it is equal to zero, we dont
have diameter change at the inlet pipe)
    Dodd=DP;
    dDdXD=2*C/(2*Dodd+C*(delx1+Xp)); %7.89

    dlarea=-0.5*(lambdap+betap)*dDdXD*delz; %7.87
    qheat= 0.001 + 2*f*U(r,s)*CP*(TW1-TG1)/DP; % Heat transfer rate % added
0.001!
    dAa=((k-1)/2)*(Aa(r,s)/Ae^2)*((qheat*xref/(aref)^3)+...
        (2*f*xref*abs(U(r,s)^3)/DP))*delz;

    dlentropy=0.5*(lambdap+betap)*(dAa)/Aap; %7.81

dlfriction=-(k-1)*(f*delx1/DP)*(xref/delx1)*(((lambdap+0.001-betap)/...
  (k-1))^(1/2))*((lambdap+0.001-betap)/abs(lambdap+0.001-betap))*((3*betap-
lambdap)/...
(lambdap+betap))*delz; %equation 7.82 % added 0.001!

dlheatransfer=2*k*(abs(lambdap+0.001-
betap)/(lambdap+betap))*(f*delx1/DP)*...
(xref/delx1)*R*(1/aref^(1/2))*(TW1-TG1)*delz; % added 0.001!

dl=dlarea+dlentropy+dlheatransfer+dlfriction; % 7.79

beta(r+1,s) =betap + dl;
Aa(r+1,s)=Aa(r,s)+dAa;
  end

%% Variable BETA 2

 for s = 16:ss-1

        dxDX = (B*lambda(r,s+1)-Ae*beta(r,s+1))/((delx2/delz)+B*...
    (lambda(r,s+1)-lambda(r,s))-Ae*(beta(r,s+1)-beta(r,s))); %7.71



if dxDX < 0 then
```

```
    B=-b;
    Ae=-a;
   dxDX =  (B*lambda(r,s+1)-Ae*beta(r,s+1))/((delx2/delz)+B*...
        (lambda(r,s+1)-lambda(r,s))-Ae*(beta(r,s+1)-beta(r,s))); %7.71
end

 lambdap=lambda(r,s+1)-dxDX*(lambda(r,s+1)-lambda(r,s)); %7.84
    betap=beta(r,s+1)-dxDX*(beta(r,s+1)-beta(r,s)); %7.85
    Aap=Aa(r,s+1)-dxDX*(Aa(r,s+1)-Aa(r,s)); % 7.86
    Xp=(XPIPE2-1)*delx2-dxDX*delx2; %7.90



    C=0;
    Dodd=DP; %
    dDdXD=2*C/(2*Dodd+C*(delx2+Xp)); %7.89

    dlarea=-0.5*(lambdap+betap)*dDdXD*delz; %7.87
    qheat= 0.001+ 2*f*U(r,s)*CP*(TW2-TG2)/DP; % Heat transfer rate % added
0.001!
    dAa=((k-1)/2)*(Aa(r,s)/Ae^2)*((qheat*xref/(aref)^3)+...
         (2*f*xref*abs(U(r,s)^3)/DP))*delz;

    dlentropy=0.5*(lambdap+betap)*(dAa)/Aap; %7.81

dlfriction=-(k-1)*(f*delx2/DP)*(xref/delx2)*(((lambdap+0.001-betap)/...
   (k-1))^(1/2))*((lambdap+0.001-betap)/abs(lambdap+0.001-betap))*((3*betap-
lambdap)/...
(lambdap+betap))*delz; %equation 7.82 % added 0.001!

dlheatransfer=2*k*(abs(lambdap+0.001-
betap)/(lambdap+betap))*(f*delx2/DP)*...
(xref/delx2)*R*(1/aref^(1/2))*(TW2-TG2)*delz; % added 0.001!

dl=dlarea+dlentropy+dlheatransfer+dlfriction; % 7.79

beta(r+1,s) =betap + dl;
Aa(r+1,s)=Aa(r,s)+dAa;
 end

 disp('OK with variable lambda1,2, beta1,2/ now entering boundaries')
%% Step 7: Insert BOUNDARY conditions [CHANGES]

lambda(r+1,1) = Inflowe(beta(r+1,1),A0,k,Aa(r+1,1)); % reservoir as an
inlet

disp('inflowe run OK')

[cc,AA,bbb,clinfnew] =
cylindere(lambda(r+1,16),beta(r+1,16),kk,AA,Aa(r+1,16)); % cylinder
lambda(r+1,16) = cc;
Aa(r+1,16)=bbb; % newly calculated entropy level
beta(r+1,16)= clinfnew; % the lambdain changes (here is the beta because
the boundary is on the left side)

disp('cylindere run OK')

beta(r+1,ss) = nozzlee(lambda(r+1,ss),PHI,k,Aa(r+1,ss)); % exit from
cylinder to a nozzle
```

```
disp('nozzlee run OK')

%% Second configuration of the exhaust model

%beta(r+1,17) = nozzle(lambda(r+1,17),PHI,k); % exit from cylinder to a
nozzle

%lambda(r+1,21) = nozzle(beta(r+1,21),PHI,k); % from nozzle to open end is
the outlet pipe (exhaust)

%beta(r+1,ss) = open(lambda(r+1,ss)); % exit to environment

%%

kk = kk+1;
%% Update Z
    z(r+1) = z(r) + delz;
%% Step 8: Determine pressure at every point for given r

disp('finished with calculations,now determining pressure at every point')
    for s = 1:M
        Pout(r+1,s) = ((((lambda(r+1,s)+ beta(r+1,s))/2)^q1))*PREF;
        zout(r+1) = theta;
        vout(r+1,s) = ((lambda(r+1,s)- beta(r+1,s))/(k - 1))*aref;%Equation
2.18

    end

    for s=1:14


  %% Step 9 :Construction of position and state diagrams

   %for position diagram
       dxdtlambda=vout(r+1,s)+aair;
        dxdtbeta=vout(r+1,s)-aair;
   %for state diagram

   % slopes:
        dcdulambda=-(k-1)/2;
        dcdubeta=(k-1)/2;

    dcl(s+1)=dcdulambda*(vout(r+1,s+1)-vout(r+1,s));
    dcb(s+1)= dcdubeta*(vout(r+1,s+2)-vout(r+1,s));
    cl=dcl/aair;
    cb=dcb/aair;
    uc(r+1,s)=vout(r+1,s)/aair;
    end

    r = r+1;
end

%% Step 10: Figures

dcl;
dcb;
```

```
% Animation of State and Position diagrams

yl=dcdulambda*uc;
yl2=yl+abs(29.5673/aair);
yl3=yl2+abs(29.5673/aair);
yl4=yl3+abs(29.5673/aair);
yl5=yl4+abs(29.5673/aair);
yl6=yl5+abs(29.5673/aair);
yb=dcdubeta*uc;
yb2=yb+abs(29.5673/aair);
yb3=yb2+abs(29.2445/aair);
yb4=yb3+abs(29.5673/aair);
yb5=yb4+abs(29.2445/aair);
yb6=yb5+abs(29.2445/aair);
figure;plot(uc,yl,uc,yl2,uc,yl3,uc,yl4,uc,yl5,uc,yl6,uc,yb,uc,yb2,uc,yb3,uc
,yb4,uc,yb5,uc,yb6);
title('State diagram')
xlabel('u/c');ylabel('c non-dimension')


x=(0:.0001:XPIPE1);
y=dxdtlambda*x;
y2=25+(dxdtbeta*(x-0.5));
y3=y+139.8;
y4=y2+323.4;
y5=y+463.2;
y6=y2+463.2;

figure; plot(x,y,x,y2,x,y3,x,y4,x,y5,x,y6);
title('Position diagram from moment 0')
xlabel('Length-X');ylabel('Time');
figure; plot(x,y+346,x,y2+346,x,y3+346,x,y4+346,x,y5+346,x,y6+346);
title('Position diagram from IVO')
xlabel('Length-X');ylabel('Time-CAD');
legend('compression wave','expansion wave','compression wave','compression
wave','expansion wave','compression wave');

%% State diagram
% In the state plane the characteristics for a gas with k=1.4 are two
% families of straight lines with slopes +-0.2


figure;plot (zout(1:end-1),AA,'r-'); xlabel('CAD');ylabel('Pressure(bar)');
title('Cylinder pressure during one engine cycle');
figure; plot(zout(1:end-1),  Pout(1:end-1,M-4),zout(1:end-1),Pout(1:end-
1,M));xlabel('CAD');ylabel('Pressure (bar)')
legend('Exhaust manifold inlet','Exhaust manifold outlet')
title('Exhaust manifold pressure')



%% Pressure diagrams

figure; plot(zout(1:end-1)+210,Pout(1:end-1,3),zout(1:end-
1)+210,Pout(1:end-1,10),zout(1:end-1)+210,Pout(1:end-1,15),zout(1:end-
1),AA);
legend('Start of the intake pipe-3','Middle of the intake pipe-10','Prior
entering the cylinder-15','Cylinder
```

```
Pressure');xlabel('CAD');ylabel('Pressure (bar)');title('Complete Pressure Figure');

figure; plot(zout(1:end-1)+210,Pout(1:end-1,1),zout(1:end-1)+210,Pout(1:end-1,2),zout(1:end-1)+210,Pout(1:end-1,3));
legend('Mesh-1','Mesh-2','Mesh-3');xlabel('CAD');ylabel('Pressure (bar)');title('Pressure of the First Three Meshes of the Intake Pipe');

%% Velocity diagrams

figure; plot(zout(1:end-1)+210,vout(1:end-1,3),zout(1:end-1)+210,vout(1:end-1,12),zout(1:end-1)+210,vout(1:end-1,13),zout(1:end-1)+210,vout(1:end-1,14),zout(1:end-1)+210,vout(1:end-1,15),zout(1:end-1)+210,vout(1:end-1,16));
legend('3','12','13','14','15','16');xlabel('CAD');ylabel('Velocity (m/s)');title('Velocity at various Meshes of the Intake Pipe');

figure; plot(zout(1:end-1)+210,vout(1:end-1,1),zout(1:end-1)+210,vout(1:end-1,2),zout(1:end-1)+210,vout(1:end-1,3));
legend('Mesh-1','Mesh-2','Mesh-3');xlabel('CAD');ylabel('Velocity (m/s)');title('Velocity of the First Three Meshes of the Intake Pipe');

%% Carpet Diagrams (for intake manifold)

%Pressure

figure;colormap 'Jet';material shiny ;rend3d2(Pout,zout+210,'Pressure (bar)',XPIPE1);

% Velocity

figure;colormap 'Jet';material shiny ;rend3d2(vout,zout+210,'Velocity(m/s)',XPIPE1);


figure;plot(aze,Fexh);xlabel('Degrees from opening');ylabel('Exhaust Valve Area as a fraction');title('Area-Angle from opening of the exhaust valve')
figure;plot(aza,Fair);xlabel('Degrees from opening');ylabel('Area(m^2)');title('Area-Angle from opening of the inlet valve')
figure;plot(aze+136,Fexh);xlabel('Degrees');ylabel('Exhaust Valve Area as a fraction');title('Area-Angle of the exhaust valve-1 cycle')
figure;plot(aza+346,Fair);xlabel('Degrees');ylabel('Area(m^2)');title('Area-Angle of the inlet valve-1 cycle')

disp('ok')
```

Inflowe

```
function clout = Inflowe(clin,A0,k,Aef)
% The difference to the relevant isentropic program is the new variable
% clinstar and the introduction to the iterative method 'iterativeinflowe'
%% Determine constants
PREF = 1;P0=1.65;
```

```
B = 4*(k-1)/(k+1);
C= ((k+1)/(3-k))*clin;%Equation 3.6(condition for chocked flow)
if (clin >= A0) %% we have reverse flow
    clinstar=(clin/Aef)*(1/1.65)^((k-1)/(2*k)); % 1/1.65 is the fraction of
Pref/Pback pressure
    clout =2-clinstar; %% Equation 3.5 is used(reverse flow)
else
    clinn=clin;
    AAn=A0*(PREF/P0)^((k-1)/(2*k));
    disp('inflowe running/now entering iterativeinflowe')
    clout=IterativeInflowe(clinn,AAn,Aef,A0);
    disp('OK with iterativeinflowe')
     if (clout > C) %% we have chocked flow
        clout = C;
     end;
end;
```

IterativeInflowe

```
% Iterative method - function for calculation lambda out/ Inflowe

function clouta=IterativeInflowe(clinna,AAna,Aeff,A0)

k=1.4;
q =(k-1)/(2*k);
A =(3-k)/(k+1);
T0=360;
Pc = 3.593 ;
Tc = 762;
PREF=1;
Tref= Tc*(PREF/Pc)^(2*q);
clinncx=clinna; % first-initial value for lamdba in
cloutx=2*A0-clinna;
AAcx=Aeff; % AA cannot be equal the 1st value to AAna

t=0.000001; %accuracy


Diff1=100;
Diff2=100;
Diff3=100;


    while Diff1 > t %1st iterative part to calculate lambdain within a
desired accuracy

        clinc = clinna + ((clinncx+cloutx)/2)*((AAcx-AAna)/AAcx);
              Diff1 = abs(clinc-clinncx);
        clinncx=clinc;
    end

    while Diff2 > t

   cloutx = A*clinncx+(2/(k+1))*sqrt(((k^2)-1)*(T0/Tref)+2*(1-
k)*clinncx^2);
```

```
    while Diff3 > t %2nd iterative part to calculate lambdain within a
desired accuracy

        clinc = clinna + ((clinncx+cloutx)/2)*((AAcx-AAna)/AAcx);
              Diff3 = abs(clinc-clinncx);
        clinncx=clinc;
    end

    clouta = A*clinc+(2/(k+1))*sqrt(((k^2)-1)*(T0/Tref)+2*(1-k)*clinc^2);

    Diff2 = abs(clouta-cloutx);

%AAcx=Aeff;


end

clouta = A*clinc+(2/(k+1))*sqrt(((k^2)-1)*(T0/Tref)+2*(1-k)*clinc^2);
disp('Internal iterativeinflow run OK')
    % Aeff the r+1 (next time position) level of entropy
```

Cylindere

```
function [clout,MU,Aentrvexit,clinf] = cylindere(cloute,clin,kk,AA,Aentr)
% added the level of entropy Aentr, also to the valve function

% Changes to isentropic program consists of introducing the entropy levels
% from the main program,the new subroutines for the valve,some constants
% concerning the flow inside the cylinder (for ex. Reynolds number, see at
% the end of the code) for calculating the cylinder pressure

global aair aref avo aza aze conrat CR cycle delz dPcdt...
     evo Fair  Fexh  k mc  Pa Pc neng...
     stroke theta thetarad Vc xref zcalc zrev ac DC Tc
%% Step1: Time calculations
dt = delz*(xref/aref);
dthrad = 2*pi()*neng*dt; % every time step dt. theta increases by dthrad
thetarad = thetarad + dthrad;% start from theta = evo
theta = (thetarad*180)/(pi()); % retransform theta in degs
if theta > 180*cycle  % i.e. for a 4-stroke engine cycle as soon as
%theta is greater than 720 we need to start again
theta = theta - 180*cycle;
end
zcalc = zcalc + (dthrad*180)/pi();
zrev = zrev + (dthrad*180)/pi();
%start with exhaust valve
alphae = theta -evo;
%% Step 2 determine exhaust valve area
if alphae < 0;
    alphae = 0;
end
i = 2;
while alphae > aze(i)
    i = i+1;
end
PSI = Fexh(i-1)+(Fexh(i) -...%Exhaust area updated
```

```
    Fexh(i-1))*((alphae-aze(i-1))/(aze(i)-aze(i-1)));
%% Step 3 Determine inlet valve area
alphaa = theta -avo;
if alphaa < 0;
    alphaa = 0;
end
i = 2;
while alphaa > aza(i)
    i = i+1;
end
F2 = Fair(i-1)+((Fair(i)-...%Inlet area updated
Fair(i-1))*(alphaa-aza(i-1)))/( aza(i)-aza(i-1));
%% Step4: Call in boundaries and determine mass of air in cylinder
Pc = Pc + dPcdt*dt;
MU(1,kk)=Pc;
if kk >1;
    MU(1,1:kk-1) = AA;
end

disp('cylindere running/now entering valvee2')

vv = Valvee2(clin,PSI,Aentr); %Call in valve boundary condition
disp('OK valvee2')
clout =vv(1,1) %Output the value of lambda out
dmedt = real(vv(1,2))%Output the value of exhaust mass <<<<<FLOW OUT>>>>>
Aentrvexit = vv(1,3) %Output from the valve function-iterative function,
entropy at the exit valve
clinf = vv(1,4)

dmadt = massrec(Pa, Pc, aair, ac,k,F2);%Determine mass <<FLOW IN>>
massrecfunc.
disp('massrec OK')
if dmadt < 0
    disp ('ok')
end
mc = mc + (dmadt-dmedt)*dt;%Update mass in cylinder
%% Step 5: Determine cylinder volume using EQ. 3.34 and 3.35
fnn = sqrt(conrat^(2)-sin(thetarad)^2);
x = 0.5*stroke*(1 + conrat - fnn - cos(thetarad));
Fcyl = 0.25*pi()*DC^2; % cylinder cross sectional area
Vc = Fcyl*(x + (stroke)/(CR-1));  %Cylinder volume
dxdt = 0.5*stroke*sin(thetarad)*(1 + (cos(thetarad))/fnn);
dthdt = 2*pi()*neng;
dVcdt = Fcyl*dxdt*dthdt;%Change in cylinder volume
%% Step 6: Calculate new Tc and new dPcdt
Tc = (Pc*(10^5)*Vc)/(287*mc);
ac = sqrt(287*k*Tc);
if dmadt == 0; A = 0;%No flow
else if dmadt > 0; A = aair;%flow in
    else A = ac;%flow out
    end
end
%dPcdt=(-k*dVcdt*(Pc/Vc))-(dmedt*ac^(2))/(Vc*10^5)...%Equation 3.33
%+((dmadt*(A^2))/(Vc*10^5));

disp('inside cylindere, starting to calculate dPcdt with Reynolds etc')
TW2=450; % arbitrary value from GTPOWER
TG2=762; % gas temperature at the exhaust pipe (I consider it equal to gas
temperature at EVO)
```

```
mvsc=1.73*10^(-5); % mvsc: typical air viscosity
http://www.grc.nasa.gov/WWW/K-12/airplane/airprop.html
CP=(k*287.3)/(k-1);
kq=CP*mvsc/0.7;
lambdain=cloute
lambdaout=clin
PREF=1;
dens=(PREF/(aref^2))*(k/((lambdain+lambdaout)/2)^2)*...
    ((lambdain+lambdaout)/(2*Aentr))^(2*k/(k-1));
VP= 2*stroke*neng ;% (m/s) The mean piston speed is the average speed of
the piston
%in a reciprocating engine. It is a function of stroke and RPM.
%There is a factor of 2 in the equation to account for one stroke to occur
%in 1/2 of a crank revolution (or alternatively: two strokes per one crank
revolution) and
%a '60' to convert seconds from minutes in the RPM term.

Re=dens*DC*VP/mvsc; % Reynolds number, dens: density, VP:average piston
velocity
dQdt=(0.5)*(Re^0.7)*kq*Fcyl*(TW2-TG2)/DC;
dPcdt=real((1/Vc)*(dmadt*(A^2)-(dmedt*ac^(2))-(k*Pc*dVcdt)+((k-1)*dQdt)))
```

Valvee2

```
function vv = Valvee2(clin,PSI,Aentr) % Introducing the variable lin star
global aref Fp k Pc PREF

%% Cylinder program
clinstar=(clin/Aentr)*(1/1.65)^((k-1)/(2*k));
if PSI<=0 % --> Closed valve
clout = clinstar;
dmedt = 0; %No mass
else

%% Constants
s = sqrt(2/(k+1));%Sonic boundary ;
A1 = (k-1)/(2*k); A4 = 2/(k+1); A5 = (k+1)/(2*(k-1));A6 = (2*k)/(k-1);
A3 = 2/(k-1);
t = 0.000001;%Accuracy

%% calculate PIA recall condition for inflow, outflow or no flow

PIA = clinstar*(PREF/Pc)^(A1);
    if PIA < 1 %We have OUTFLOW
    %% Determine the value of U from
    U = (s*(sqrt(1+(k^(2)-1)*PSI^2)-1))/(PSI*(k-1));%Equation 4.8
    C = (sqrt(1-(U^2)/A3))/(s);%Calculate entropy Equation 3.32
    picr = s - (U)/(A3*C);%picr obtained from Equation 4.6
    % We need to reset the entropy as we have calculated C for Critical
    C = 1/(PIA);

    %% TEST for sonic or subsonic flow

        if (PIA > picr) % -------------------------------->>>>The flow is
SUBSONIC
        %% Start iteration to solve f(pi) for subsonic flow (sel 297-
flowchart)
        pix = (1 + s)/2; % First attempt
```

```
        dpix = (1 - s)/2; % Step increase or decrease

        %% Solve equation 4.4
        B = C*(pix-PIA);
        subout = (PSI/pix)*sqrt(A3*(pix^(-2)-1))-((A3*B)/(1-A3*B^2)); %6.47
page295
        C = (sqrt(1 - A3*B^(2)))/(pix);
            while dpix >t
           % while abs(subout)>t

                if subout < 0
                pix = pix - dpix; %Make pix smaller
                else if subout > t
                    pix = pix + dpix;%Make pix bigger
                    end
                end
            %if dpix>t
            B = C*(pix-PIA);
            subout = (PSI/pix)*sqrt(A3*(pix^(-2)-1))-((A3*B)/(1-A3*B^2));
            C = (sqrt(1 - A3*B^(2)))/(pix);
            dpix = dpix/2;

            end %repeat process until abs(pix) <=t

        else if (PIA <= picr) % --------------------------->>>> The flow is
SONIC
            %% Start iteration to solve f(pi) for sonic flow
            pix = (s + PIA)/2; %First attempt
            dpix = (s - PIA)/2; % Step increase or decrease

            %% Solve equation 4.5
            B = C*(pix-PIA);
            E = PSI*(A4)^(A5);
            sonout = pix^(A6)-E*((1-A3*B^2)/(A3*B));
            C = (sqrt(1 - A3*B^(2)))/(pix);   % 6.56/pg.295

%while dpix >t
            while abs(sonout)>t

                if sonout < 0
                pix = pix + dpix;
                else if sonout > 0
                    pix = pix - dpix;
                    end
                end
            B = C*(pix-PIA);
            sonout = pix^(A6)-E*((1-A3*B^2)/(A3*B));
            C = (sqrt(1 - A3*B^(2)))/(pix);
            dpix = dpix/2;

                if dpix < 0.00000001%set a limit to dpix
                sonout = 0;
                end
            end%Repeat process until abs(pix) <=t
            end
        end
        %% Obtain outputs
        clout = 2*pix*(Pc/PREF)^(A1) - clinstar;
        A = 0.5*(clinstar + clout);
        U = (clinstar-clout)/(k-1);
```

```
        %Calculate dmedt using Eq. 4.10
        dmedt = -k*(PREF/aref)*Fp*(U/(A^2))*(A^((2*k)/(k-1)))*1*10^5;

    else if PIA >1  %% We have INFLOW
    % There is inflow thus nozzle conditions apply
    clinv = clinstar*(PREF/Pc)^(A1);
    cloutn = nozzle (clinv,PSI,k);
    clout = cloutn*(Pc/PREF)^(A1);
    A = 0.5*(clinv + clout);
    U = (clinv-clout)/(k-1);
    dmedt = -k*(PREF/aref)*Fp*(U/(A^2))*(A^((2*k)/(k-1)))*1*10^5;

    else %PIA = 1 %then we have NO FLOW
    clout = clinstar;
    dmedt = 0;
        end
    end
end

disp('valvee2 run/now entering iterativevalvee2')

[cloutf,Aaf,clinf]=Iterativevalvee2(clin,clout,Aentr,PSI);

disp('iterativevalvee2 run OK, exiting valvee2')

vv(1,1)= cloutf;
vv(1,2)= dmedt;
vv(1,3)=Aaf;
vv(1,4)=clinf;
```

IterativeValvee2

```
function [cloutf,Aaf,clinf]=Iterativevalvee2(clinna,clouta,AAna,Fexh)

 %Iterative method - function for calculation lambda out/ valve- STEPS pg
 %378/Benson

k=1.4;
q =(k-1)/(2*k);
A =(3-k)/(k+1);
T0=360;
Pc = 3.593 ;
Pcr=Pc;
Tc = 762;
PREF=1;
Tref= Tc*(PREF/Pc)^(2*q);

%% STEP 2
clinncx=clinna % 1st estimation for value of lamdba in from the main part
of the code
cloutx=clouta % 1st estimation for value of lamdba out from the main part
of the code
%AAcx=AAna;
Ap=(clinncx+cloutx)/2 % (7.152) 1st estimation for the entropy level
immediately after the valve, at the pipe
```

```
%% 1st estimations

Up=(cloutx-clinncx)/((k-1));
Ac=sqrt((Ap^2)+((k-1)/2)*(Up^2)) %(7.149)1st estimation for the entropy
level immediately before the valve, at the cylinder
%U=(cloutx-clinncx)/((k-1)*Ac);
% C=(((k-1)/2)*(U^2))/(1-((k-1)/2)*(U^2))^2;
Aais=Ac*(PREF/Pcr)^((k-1)/2*k) % Isentropic value of Aa. I set Pc =Pcr, the
pressure at the cylinder when the exhaust valve is opening

%% STEP 3

    rc=Pcr/PREF;
    if rc < 1.0
        AAEST=Aais; % estimation of Aa
        DAA=0.5; disp('case DAA=0.5')
    elseif rc > 1.0
            AAEST=Aais;
            DAA=(Ac-Aais)/2; disp('case DAA=Ac-Aais/2')
    end

%% STEP 4

t1=0.01; %accuracy

diff=100; % large first arbitrary value for the method to start


while diff > t1 % we need 'diff' to be within the desired accuracy


    clinc = (clinna + ((clinncx+cloutx)/2)*((AAEST-AAna)/AAEST)) % AAEST is
changing through the iterative method

%% STEP 5 (CHECKED OK)

% Criterion

diff2= Ac-clinc;


while diff2 < 0
     AAEST=Aais+DAA;
        DAA=0.5*DAA;
clinc = clinna + ((clinncx+cloutx)/2)*((AAEST-AAna)/AAEST);
diff2= abs(Ac-clinc);
end

%% STEP 6

% Calculation of clout

cloutc = (A*clinc+(2/(k+1))*sqrt(abs(((k^2)-1)*((Ac)^2)+2*(1-k)*...
    clinc^2)))


%% STEP 7
```

```
U=(cloutc-clinc)/(Ac*(k-1))

%% STEP 8

C=(((k-1)/2)*(U^2))/(1-((k-1)/2)*(U^2))^2;

%% STEPs 9-10


FF= (4*C/((k^2)-1))-(Fexh)^2 % Fexh is the opening fraction of the exhaust
valve

if FF > 0
    %sonic flow in the valve
    pppc=Fexh*(1-((k-1)/2)*(U^2))*((2/(k+1))^((k+1)/(2*(k-1))))*(1/U) %
equation 7.142
else % FF < = 0
    %subsonic flow in the valve
    pppc=((1/(2*C))*(Fexh*sqrt(((Fexh^2)+4*C)-Fexh^2)))^(k/(k-1)) %
equation 7.138
end

%% STEP 11

Aacalc=real(((clinc+cloutc)/2)*((1/pppc+0.0001)^((k-1)/(2*k)))*(1/rc)^((k-
1)/(2*k))) % added +0.0001 @ pppc because it is zero
AAEST
%% STEP 12

diff = abs(Aacalc-AAEST);

if diff > t1
        DAA=2*DAA; % CHANGED FROM DAA/2! 16/07
        AAEST=AAEST+DAA; % AAESTnew is the new value
        disp('case diff>t1')
    elseif diff < 0
        DAA=DAA/2;
        AAEST=AAEST-DAA;
        disp('case diff<0')
end

    diff

clinncx=clinc;
cloutx=cloutc;

end

     cloutf=cloutc;
    clinf=clinc;
    Aaf=AAEST;

    disp('iterativevalvee2 running finished')
end
```

Nozzlee

```
function clout = nozzlee(clin,O,k,Aef)

%The difference to the relevant isentropic program is the new variable
% clinstar

t = 0.000001; % determining accuracy
A1 = 4/(k-1); A2 = (k-1)/2; A3 = 2*(k-1); A4 = 2/(k+1);
A5 = (k+1)/(k-1);A6 = 2/(k-1);
clinstar=(clin/Aef)*(1/1.65)^((k-1)/(2*k)); % 1/1.65 is the fraction of
Pref/Pback pressure

%% Step 1, Check whether there is reverse flow
if clinstar<1
    clout = 2-clinstar;  %there is reverse flow
else

%% Step 2 Calculate for subsonic flow, solve Equation 4.1
A = 0.5*(clinstar + 1);%first estimate of A
Ta=360;
aair = sqrt(k*287*Ta);
PREF = 1;
Tc=762;
Pc=3.593;
q =(k-1)/(2*k);
Tref = Tc*(PREF/Pc)^(2*q) ;
aref = (k*287*Tref)^(1/2);
Astar=A/(aair/aref); % the non-dimensional speed of sound at
troat/reference
dA = 0.25*(clinstar - 1);%step change of A
sum = (((Astar^A1)-O^2)*(clinstar-Astar)^2)-A2*((Astar^2)-1)*O^2; %Equation
4.1
while abs(sum)>=t
if sum < 0
    Astar = Astar - dA;%Make A smaller
else if sum > t
        Astar = Astar + dA;%Make A bigger
    end
end
sum = (((Astar^A1)-O^2)*(clinstar-Astar)^2)-A2*((Astar^2)-1)*O^2;
dA = dA/2;%make step change smaller for next step
end

%we have now calculated through the iterative valve the value of Astar
clout = 2*Astar - clinstar;%Calculate clout using equation 2.18
U = (clinstar-clout)/(k-1);%Similarly calculate U using equation 2.18

%% Step 3: Test for sonic or subsonic <<<<<<SONIC TEST>>>>>>>
Ut = (U/O)*Astar^A3;%Determine Ut using equation 3.11
if Ut < 1 % i.e we have a subsonic flow
    clout = 2*Astar - clinstar;

else   %% Solve for sonic flow
    At = 0.5 + 0.5*sqrt(A4);%Initial guess
     dAt = (1-sqrt(A4))/4;%Step change
     Y = (At)^-1;
     sum = (O^2)-(Y^(A1))*(A5-(A6*Y^2));%Substitute Y into eq. 4.2
     while abs(sum)>=t%Repeat process until eq 4.2 = t or less
```

```
 if sum < 0
     At = At - dAt;%make At smaller
 else if sum > t
         At = At + dAt;%make At bigger
     end
     Y = (At)^-1;
     sum = (O^2)-(Y^(A1))*(A5-(A6*Y^2));
     dAt = dAt/2;
 end
     end
%% Step 5: Obtained clout for sonic flow
 F =  (1-(At^(A5)*O*A2))/(1+(At^(A5)*O*A2));%Apply equation 4.3
 clout = F*clinstar;
 end
end
```

Chapter 12

Bibliography

[1] [Rowland S. Benson] Edited by J. H. Horlock F.R.S and D.E. Winterbone (1982). The Thermodynamics and Gas Dynamics of Internal-Combustion Engines, Volume I. CLARENDON PRESS, OXFORD 1982.

[2] [Ascher H. Shapiro,1953] The dynamics and thermodynamics of compressible fluid flow, Volume II.The Ronald press company, New York.

[3] [Lumley, 1999] Engines: an introduction. Cambridge University.

[4] [Winterbone and Pearson, 2000] Theory of Engine Manifold design, Wave action methods for IC engines. Professional Engineering Publishing.

[5] [K Banisoleiman, B A French] The interaction of diesel engine turbocharging and tuned inlet manifold systems under steady state and transient operation, IMcchE 1991, Roc Instn Mccb Engra Vol205.

[6] Diploma Thesis, Mario Alberto Benjamin Franchetti (4th June 2009). Modelling Wave action in the engine manifolds of internal combustion engines, Imperial College, London.

[7] Presentation , Alexandre Avila, 4M Individual project: Final presentation . Graphical animation of the "Method of Characteristics" for solving hyperbolic PDEs, Imperial College, London.

[8] Internet sources: http://en.wikipedia.org/wiki/Manifold_(automotive) [June 2010]

[9] [Colin R. Ferguson] Internal Combustion Engines, Applied Thermosciences. John Wiley and Sons. Purdue University.