Mathematical and Numerical Modeling of Flow and Combustion Processes in a Spark Ignition Engine

Zografakis Georgios

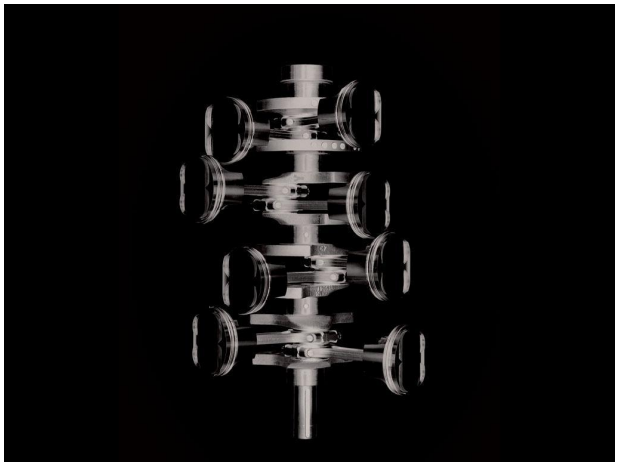
Department of Applied Mathematics

Supervisor:Assistant Professor Christakis Nikolaos

10 September 2005

Contents

Nomenclature	
A_f	Area of spherical flame front.
B	Chamber Bore. Measured in m.
B_{iv}	Intake Valve diameter. Measured in m.
C_p	Specific heat at constant pressure, $Jkg^{-1}K^{-1}$.
h	Enthalpy, Jkg^{-1}.
L	Chamber's Stroke. Measured in m.
L_{iv}	Maximum intake valve lift. Measured in m.
l_t	Characteristic length scale of turbulent flame, m.
m	mass, kgr.
m_e	mass entrained by flame front into flame zone, kgr.
p_0	Atmospheric pressure. Measured in atm.
p_r	Exhaust pressure. Measured in atm.
p_a	Ambient pressure. Measured in atm.
rpm	Engine Rotation. Measured in rpm.
R_c	Ratio of crank radius to connecting rod length.
R	Universal gas constant, $Jkg^{-1}K^{-1}$.
R^*	Individual gas constant , $Jkg^{-1}K^{-1}$.
$\dot{Q}_w$	Total heat transfer rate to cylinder wall, Watt.
S_t	Laminar flame speed, ms^{-1}.
T_0	Atmospheric temperature. Measured in C^o.
T_r	Exhaust temperature. Measured in C^o.
T_w	Temperature of chamber's wall. Measured in C^o.
U_t	Characteristic turbulent speed, ms^{-1}.
V	Instantaneous cylinder volume, m^3.
V_f	Enflame volume, m^3.
Greek letters	
α	The crankshaft radious.
γ	C_p/C_v.Dimensionless.
ϵ	Compression Ratio.Dimensionless.
θ	Crank angle, deg.
θ_0	Initial crank angle of combustion, deg.
ρ	density, kgm^{-3}.
τ_b	Characteristic reaction time to burn eddy of size l_t, s.
Subscripts	
b	burned gas
e	entrainment
f	flame
u	unburned gas



Abstract

In this report a mathematical and numerical model of flow and combustion process in a spark ignition engine is developed. A quasi-dimensional spark ignition (SI) engine cycle model has been used.In this model,combustion is modeled as a turbulent flame propagation process.During the combustion,it is assumed that the cylinder charge consists of unburned and burned gas zone.A computer code was developed for the mathematical cycle model presented.By using this code, parameters that characterize the combustion cycle and engine performance can be computed practically.

1 Introduction

Perhaps the best-known engine in the world, nowadays, is the reciprocating internal combustion engine. Virtually every person who has driven an automobile or pushed a power lawnmower has used one. By far the most widely used internal combustion engine is the spark-ignition gasoline engine, which takes us to school, work or on pleasure jaunts.

Practical heat engines have served mankind for over two and a half centuries. For the first 150 years, water, raised to steam was interposed between the combustion gases produsd by burning the fuel and the work-producing piston in-cylinder expander. It was Etienne Lenoir (1822-1900) who developed an early form of the reciprocating internal combustion engine. Gas and air were drawn into the cylinder during the first half of the piston stroke. The charge was then ignited with a spark, the pressure increased, and the burned gases then delivered power to the piston for the second half of the cycle.

A more successful development was introduced in 1876 by Nicolaus Otto (1832-1891), who is generally credited with the invention of the engine and with the statement of its theoretical cycle. It was an atmospheric engine which used the pressure rise resulting from combustion of the fuel-air charge early in the outward stroke to accelerate a free piston and rack assembly so its momentum would generate a vacuum in the cylinder. Atmospheric pressure then pushed the piston inward, with the rack through a roller clutch to the output shaft.

Another important engine is a new form of the internal combustion engine that made the name of Ruddolf Diesel (1858-1913) famous. The diesel engine, the workhorse of the heavy truck industry, is widely used in industrial power and marine applications. It replaced the reciprocating steam engine in railroad locomotives about fifty years ago and remains dominant in the role today. Diesel's concept of initiating combustion by injecting a liquid fuel into air heated solely by compression permitted a doubling of efficiency over other internal combustion engines. Much greater expansion ratios, without detonation or knock, were possible.

2 Theoretical Model

2.1 Engine Componets

A schematic diagram of a reciprocating internal combustion engine is shown in Figure 1 . The basic components of a reciprocating internal combustion engine are the cylinder and the piston. A car engine consists of more than one cylinder which are contained in the engine block. The piston both seals the cylinder and transmits the combustion-generated gas pressure to the crankshaft via the connecting rod. The crankshaft passes the rotary motion through the gearbox to the wheels. The cylinder head contains the spark plug for a spark ignition engine or fuel injector for a compression engine and the intake and exhaust valves from which an the air-fuel mixture comes inside the cylinder. Finally, an intake and exhaust manifold connected with the intake and exhaust valve respectively, complete the spark ignition engine assembly.

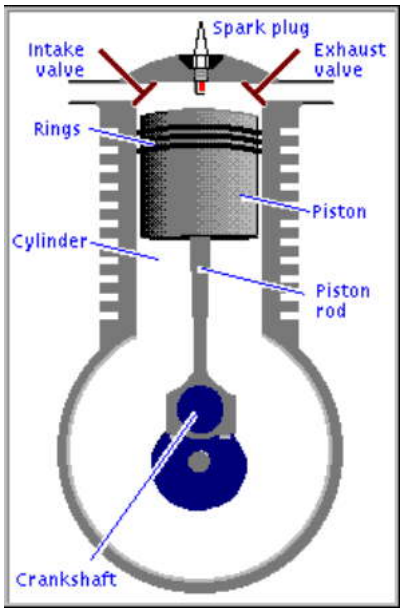


Figure 1: *Schematic diagram of a reciprocating internal combustion engine.*

The piston reciprocates in the cylinder between two fixed positions called the *top dead center* (TDC) and the *bottom dead center* (BDC). The TDC is the position of the piston when it forms the smallest volume in the cylinder and the BDC when the piston forms the largest volume in the cylinder. The distance between the TDC and the BDC is the largest distance that the piston can travel in one direction and it is called the *stroke* (L) of the engine. The stroke can be also assumed by:

$$L = 2a \tag{1}$$

where a is the radius of the crankshaft. The distance of the piston is called the *bore* (B). The air or air-fuel mixture is drawn into the cylinder through the intake valve and the combustion products are expelled from the cylinder through the exhaust valve.

The minimum volume formed in the cylinder when the piston is at TDC is called the clearance volume. The volume displaced by the piston as it moves between TDC and BDC is called the displacement volume. The ratio of the maximum volume formed in the cylinder to the minimum volume is called the *compression ratio* r of the engine:

$$r = \frac{V_{max}}{V_{min}} = \frac{V_{BDC}}{V_{TDC}} \tag{2}$$

2.2 Engine Operating Cycles

Spark ignition engines, diesel engines and most power producing engines operate on cycles.In all these engines energy is provided by burning a fuel within the system boundaries. That is the reason why they are called internal combustion engines. Even though the piston in internal combustion engines returns to its starting position at the end of each revolution, the working fluid does not undergo a complete thermodynamic cycle. It is thrown out of the engine at some point in the cycle as exhaust gases instead of being returning to the initial

state like in the ideal cycles. Working on an open cycle is the characteristic of all internal combustion engines. A typical cycle of internal combustion engine is shown in Figure 2 .

In the following sections, Lenoir, Otto and Diesel cycles which are some of the most important ideal cycles for reciprocating engines are briefly discussed. Finally, the ideal cycle of an internal combustion reciprocating engine is also presented.

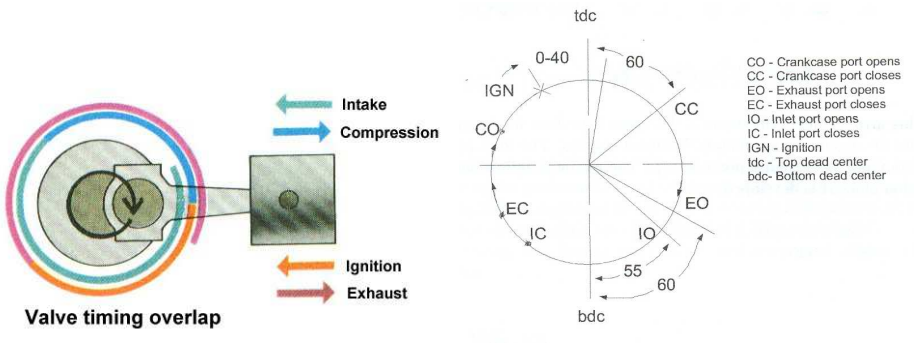


Figure 2: *A typical cycle of internal combustion engine.*

2.2.1 The Lenoir Cycle

An early form of reciprocating internal combustion engine is created by J.J.E Lenoir. His engine used a crank-slider-piston-cylinder arrangement in which a combustible mixture confined between the piston and the cylinder is ignited after TDC. The resulting combustion gas pressure forces acting on the piston deliver work by way of the connecting rod to the rotating crank. When the piston is at BDC, combustion gases are allowed to escape. The rotational momentum of the crank system drives the piston toward to TDC, expelling additional gases as it goes. A fresh combustible mixture is again admitted to the combustion chamber and the cycle is repeated.

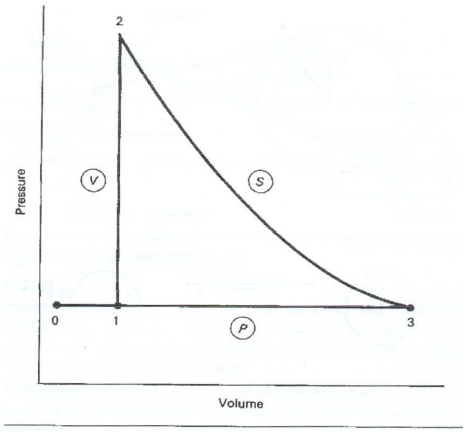


Figure 3: *The theoretical diagram of Lenoir cycle.*

The theoretical Lenoir cycle (Figure 3) consists of the intake of the working fluid from state 0 to state 1, a constant-volume temperature and pressure rise from state 1 to state 2, approximating the combustion process, an isentropic expansion of the combustion gases to state 3, and a constant-pressure expulsion of the residual gases back to state 0. A portion of the piston displacement from state 0 to state 1, is used to take in the combustible mixture and does not participate in the expansion stroke from state 2 to state 3. The power delivered is due to the extremely rapid combustion pressure rise or explosion of the mixture in the confined space of the cylinder.

2.2.2 The Otto Cycle

In 1876 Niklaus Otto demonstrated an engine cycle which is considered as the theoretical cycle commonly used to represent the processes in the spark ignition (SI) internal combustion engine. It is assumed that a fixed mass of working fluid (fuel and air) is confined in the cylinder by a piston that moves from BDC to TDC and back.

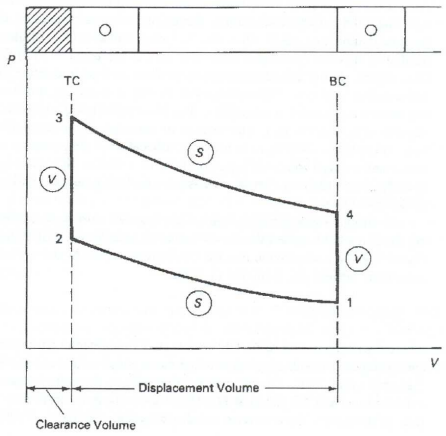


Figure 4: *The theoretical diagram of Otto cycle.*

The cycle consists of four internally reversible processes (Figure 4): An isentropic compression of an air-fuel mixture from state 1 to state 2, which occurs when the piston moves from BDC to TDC; a constant-volume heat addition process to state 3, where the mixture is ignited; an isentropic expansion of the combustion gases to state 4, where no heat is added and finally a constant-volume heat rejection back to state 1. The constant-volume heat rejection is a simple means of closing the cycle. It obviates the need to represent the complex expansion and outflow of combustion gases from the cylinder at the end of the cycle. The Otto cycle which involves only two strokes (intake and exhaust stroke) is not discussed in this report.

2.2.3 The Diesel Cycle

A cycle similar to the Otto cycle is the Diesel cycle, except that heat addition and rejection occur at different conditions. The Diesel cycle is the ideal cycle for compression ignition reciprocating engines.

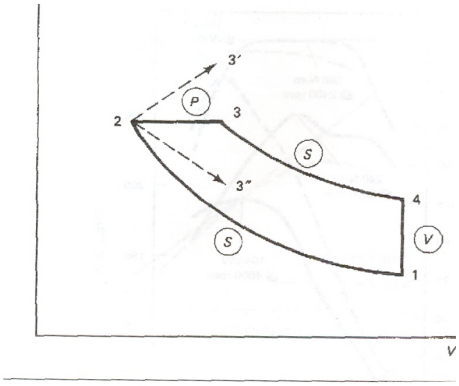


Figure 5: *The theoretical diagram of Diesel cycle.*

In a diesel engine only air is compressed and its cycle consists of four internally reversible processes (Figure 5). First, an isentropic compression of air from state 1 to state 2 occurs when the piston moves from BDC to TDC. When the air is compressed to a temperature that is above the autoignition temperature of the fuel , then combustion starts on contact as the fuel is injected in the hot air. Hence, there is a constant-pressure heat addition process to state 3. Then, an isentropic expansion of the combustion gases occurs to state 4, and the cycle is completed with a constant-volume heat rejection back to state 1. The constant-volume heat rejection is a simple means of closing the cycle.

2.2.4 The Internal Combustion Engine Cycle

The majority of gasoline fueled automobiles use engines which operate on what is known as four-stroke cycle (Figure 6). The piston executes four complete strokes within the cylinder and the crankshaft completes two

revolutions in order to complete a full power stroke. Both spark-ignited and compression-ignited engines use this cycle which comprises :

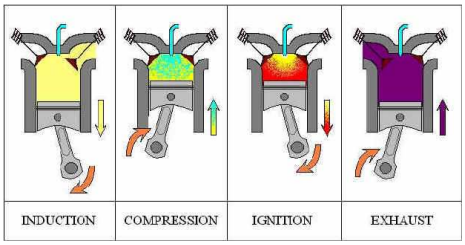


Figure 6: *The four-stroke cycle.*

1. A *compression stroke*, where both intake an exhausts valves are closed and the mixture of fuel and air is compressed to a small fraction of volume. Shortly before TDC spark plug ignites the mixture and combustion occurs, increasing the pressure and temperature of the system.

2. An *expansion stroke*, which supplies the necessary power to rotate the crankshaft. The high-temperature and high-pressure gases push the piston down to BDC creating the appropriate work of the engine. As the piston approaches BDC the first revolution of the crankshaft is completed and the cylinder is filled with combustion products. It is then that the exhaust valve opens to initiate the next stroke, the exhaust process.

3. An *exhaust stroke*, where the residual burned gases exit the combustion chamber. This happens because the pressure inside the cylinder is substantially higher than atmospheric and because of the motion of the piston towards to TDC. Shortly after TDC, the exhaust valve closes.

4. An *intake stroke*, which starts with the piston at TDC and ends with the piston at BDC, which draws fresh fuel mixture into the cylinder. To increase the mass of inducted mixture, the inlet valve opens shortly before TDC in order to take advance of the difference between the in-cylinder and atmospheric pressure and closes after TDC when the cycle starts again.

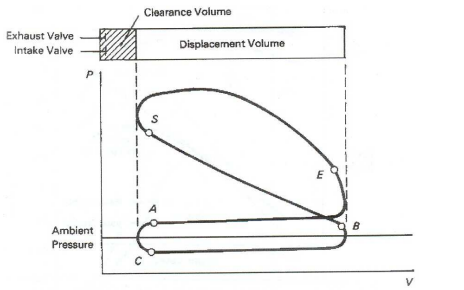


Figure 7: *The thermodynamic diagram of an actual four-stroke cycle for an internal combustion engine.*

In reality, the diagram for a four-cycle in an internal combustion engine is substantially different to its theoretical one. Compare for example figure 7 with figure 4 of the theoretical thermodynamic diagram of the Otto cycle.

Besides the four-stroke cycle, a two-stroke cycle was developed. The major difference between the four stroke engine and the two stroke engine is the way in which intake and exhaust process takes place. In the four stroke engine there are separate strokes for the intake and exhaust processes. In the two stroke engine however, both intake and exhaust processes take place with the same stroke, which is called scavenging. The two stroke engine can be either made into a spark ignition or compression ignition engine.

2.3 Thermochemistry

Internal combustion engines obtain their energy from the combustion of hydrocarbon fuel with air. The chemical energy stored in the fuel is converted to energy that the engine can use. The combustion process involves the chemical reaction of hydrocarbons fuel with oxygen to produce water vapor and carbon dioxide CO_2. The maximum amount of chemical energy from the hydrocarbon fuel is when it reacts with stoichiometric oxygen. The meaning of *stoichiometric oxygen* is defined as the amount of oxygen that is needed to convert all of the carbon in the fuel to CO_2 and all of the hydrogen to water. The simplest chemical reaction using the simplest hydrocarbon like methane with stoichiometric oxygen is:

$$CH_4 + 2O_2 \Rightarrow CO_2 + 2H_2O \tag{3}$$

The hydrocarbon fuel used in the engines is not a simple fuel like methane but rather consists of isooctane and various additives. The chemical reaction involving isooctane and oxygen is:

$$C_8H_{18} + 12.5O_2 \Rightarrow 8CO_2 + 9H_2O \tag{4}$$

The above two chemical reactions involve the reaction of a hydrocarbon with oxygen. Since it would be extremely expensive to use pure oxygen, the atmosphere is used as a rich source of oxygen. The hydrocarbon reacts with air which is composed of many substances. Nitrogen and oxygen are the two most found substances in air with a nitrogen composition of 78% by mole, and oxygen composition of 21%. An overall stoichiometric combustion equation is then:

$$C_aH_b + (a + \frac{b}{4})(O_2 + 3.76N_2) \Rightarrow aCO_2 + \frac{b}{2}H_2O + 3.76(a + \frac{b}{4})N_2 \tag{5}$$

Combustion can occur with either a lean or a rich mixture. If the mixture is for example 150% of stoichiometric, then there will be an excess amount of air and the products will involve an amount of extra air in unchanged form. This is called a *lean mixture* since there is a deficiency in fuel. If on the other hand the mixture is 80% of stoichiometric then there will be excess fuel and there will be insufficient oxygen to oxidize fully the fuel C and H to CO_2 and H_2O. The products will be a mixture of CO_2 and H_2O with carbon monoxide (CO) in the end. This is a *rich mixture* since the mixture has excess of fuel. Carbon monoxide is a colourless, poisonous gas which can be further burned to form CO_2. If there is a further deficiency in oxygen then more CO will go into the atmosphere as pollution.

The equivalence ratio is sometimes used to charactarize the mixture ratio, whether rich or lean. The *equivalence ratio*, Φ , is defined as the ratio of the actual fuel-air ratio to the stoichiometric fuel-ratio ratio. Thus $\Phi > 1$ represents a rich mixture and $\Phi < 1$ represents a lean mixture. In terms of air-fuel ratio, $\Phi = (A/F)_{stoich}/(A/F)$.

Homogeneous air-fuel mixtures close to stoichiometric may ignite spontaneously if the mixture temperature exceeds a temperature called the *autoignition temperature*.

2.4 The Combustion Process

In a SI engine, combustion is initiated by a spark discharge. At the beginning of combustion the flame is a smooth-surfaced, roughly spherical kernel about 1mm in diameter and grows quite spherically for the next few degrees. During this period a negligible fraction of the mass is burned and the measured initial burning speeds are close to laminar flame speeds. This stage is called *the initial burning phase*. After only a few degrees, the interaction of the flame and the turbulent flow field produces a highly wrinkled and convoluted outer surface of the flame. During this period burning speeds are equal to turbulent flame speed. This phase is called *the faster burning phase*. Experimental studies also show that a specified amount of gas is still unburned after the termination of the flame propagation process. This stage is called *the final burning phase*.

3 Mathematical Model

In general mathematical engine simulation models are divided into 2 main groups. The first is fluid-dynamics-based models and the second is thermodynamics-based models. Fluid-dynamics-based models are also called multidimensional models due to the fact that their formation is based on the conservation of mass, chemical species and energy at any location within the engine cylinder or manifolds at any time. That is, space dimensions (spatial coordinates) and time are the independent variables, hence the governing equations are partial differential equations. By using the multidimensional models, detailed information about the spatial distribution of the gas velocity, temperature and composition within the engine cylinder can be obtained. The solution of such equations is difficult and a large amount of computer time is needed.

Thermodynamic cycle models are based on the thermodynamic analysis of the cylinder contents during the engine operating cycle. In these models the First Law of Thermodynamics is applied to an open system composed of a fuel-air-residual gas charge within the engine manifold and cylinders. Thermodynamics-based models are zero dimensional because the only independent variable is time. Thus, the governing equations consist of ordinary differential equations (ODE) instead of partial differential equations (PDE) of multi dimensional equations. In such engine models, the combustion process is simulated in 2 ways for SI engines. In the first approach, the rate of burning of the charge is obtained empirically by using the Wiebe function or the cosine burn rate formula. That is, combustion chamber geometry and flame geometry are not considered. Therefore these models are called zero-dimensional models. In the second approach, the mass burning rate is determined from a mathematical model of the turbulent flame propagation. These types of thermodynamic cycle models that include detailed combustion modeling are called quasi-dimensional models.

From the above, it is well understood that the multidimensional models are useful for predicting the detailed information of the cylinder charge during the engine cycle but are not practical for parametric studies of the effects of changes in design and operating variables on engine performance, efficiency and emmisions.

3.1 Intake Process

The system is considered to consist of a single cylinder.The pressure for the intake process is given by :

$$p_a = p_0 - \Delta p_a \tag{6}$$

Pressure loss Δp is computed from the Bernoulli equation for one-dimensional in compressible flows, $area \times velocity = const.$ I this way:

$$A_v U_{in} = A_p U_m \Rightarrow U_m = \frac{A_v}{A_p} U_{in} \tag{7}$$

where U_{in} is the mixture intake velocity, A_u is the valve area (taken as the $\frac{1}{3}$ of the cylinder bore area), A_p is the cylinder bore area and U_in is the mean piston speed. Hence

$$p_0 + \frac{1}{2}\rho U_m^2 = p_a + \frac{1}{2}\rho U_{in}^2 \tag{8}$$

$$\Rightarrow p_0 - p_a = \Delta p_a = \frac{1}{2}\rho(U_m^2 - U_{in}^2) \tag{9}$$

and so we have :

$$p_a = p_0 - \frac{1}{2}\rho(U_m^2 - U_{in}^2) \tag{10}$$

Temperature can be calculated by

$$T_a = T_0 + \Delta T_R + \Delta T_W \tag{11}$$

$$\Delta T_R = T_m - T_0 \tag{12}$$

$$\Delta T_W = \frac{1}{3}[T_W - T_m] \tag{13}$$

where $T_m = \frac{P_a}{P_0} T_0$ is the fuel-air mixture temperature after being heated by the exhaust gases. Thus :

$$T_a = \frac{1}{3}[T_W + 2T_m] \tag{14}$$

3.2 Compression Process

Throughout the compression, the cylinder charge consists of a mixture of air, fuel vapor and residual gases. These gases are nonreacting ideal gases and are characterized by a single mean temperature. The total mass of the system is taken as constant during this process. By the above assumptions, using the First Law of Thermodynamics and ideal gas laws, the time rates of pressure and temperature for the compression process were arranged as follows:

$$\dot{T}_u = (\frac{B}{A})_u[-\frac{\dot{V}}{V} + \frac{-\dot{Q}_{wu}}{(Bm)_u}], \tag{15}$$

$$\dot{p}_u = 10^{-5}(\frac{\rho}{\frac{\partial\rho}{\partial p}})_u[-\frac{\dot{V}}{V} + \frac{1}{\rho_u}(\frac{\partial\rho}{\partial T})_u\dot{T}_u] \tag{16}$$

where

$$A_u = \frac{1}{\rho_u}\frac{(\frac{\partial\rho}{\partial T})_u}{(\frac{\partial\rho}{\partial p})_u} + c_{pu} \ , \ B_u = \frac{1}{(\frac{\partial\rho}{\partial p})_u} \tag{17}$$

and u denotes the unburned gas, V is the instantaneous total cylinder volume, $\dot{Q}$ the rate of the total heat transfer to the wall, ρ the density and C_p is the specific heat value at a constant pressure.

Due to adiabatic compression $\Delta Q = 0$, so that from the First Law of the Thermodynamics we have :

$$\begin{aligned} \Delta U = -p\Delta V \Leftrightarrow nC_v\Delta T = -p\Delta V &\Leftrightarrow nC_v\Delta T = -\Delta(pV) + V\Delta p \Leftrightarrow \\ C_v = -\frac{\Delta(pV)}{n\Delta T} + V\frac{\Delta p}{n\Delta T} &\Leftrightarrow C_v = -R + \frac{V}{n}\frac{\Delta p}{\Delta T} \Leftrightarrow \\ \frac{\Delta p}{\Delta T} = \frac{n}{V}(C_v + R) &\Leftrightarrow \frac{\Delta p}{\Delta T} = \frac{\rho}{MB}(C_v + R) \end{aligned} \tag{18}$$

Hence:

- The partial derivative of density with pressure

$$\begin{aligned} \frac{\partial\rho}{\partial p} &= \frac{\partial}{\partial p}(\frac{p}{R^*T}) = \frac{1}{R^*T} - \frac{p}{R^*T^2}\frac{\partial T}{\partial p} = \\ &= \frac{1}{R^*T}[1 - \frac{p}{T}\frac{\partial T}{\partial p}] = \frac{1}{R^*T}[1 - \frac{p}{T}\frac{MB}{\rho}\frac{1}{C_v + R}] \end{aligned} \tag{19}$$

  where $p = \rho R^*T \Leftrightarrow \frac{p}{T} = \frac{\rho}{MB}R$.
  So

$$\frac{\partial\rho}{\partial p} = \frac{1}{R^*T}[1 - R\frac{1}{C_v + R}] \Leftrightarrow \frac{\partial\rho}{\partial p} = \frac{1}{R^*T}\frac{C_v}{C_v + R} \tag{20}$$

- The partial derivative of density with temperature

$$\begin{aligned} \frac{\partial\rho}{\partial T} &= \frac{\partial}{\partial T}(\frac{p}{R^*T}) = \frac{1}{R^*T}[\frac{\partial T}{\partial p} - \frac{p}{T}] \\ &\Leftrightarrow \frac{\partial\rho}{\partial T} = \frac{\rho C_v}{RT} \end{aligned} \tag{21}$$

- The specific heat value at a constant volume

$$C_p = (\frac{\partial h}{\partial p})_p = \frac{\partial}{\partial T}(U + pV) = \tag{22}$$

$$= \frac{\partial}{\partial t}(U + nRT) = \frac{\partial U}{\partial T} + nR \tag{23}$$

  if we consider that $U = \frac{3}{2}nRT \Leftrightarrow \frac{\partial U}{\partial T} = \frac{3}{2}nR$. So, we conclude that

$$C_p = \frac{5}{2}nR \tag{24}$$

- The volume of the engine cylinder and the rate of its change can be expressed, respectively, by :

$$V = V_c + \frac{\pi B^2}{4} L \tag{25}$$

where $L = a\Big(1 - cos(\theta) - \frac{R_c}{2} sin^2(\theta)\Big)$.

Hence :

$$\frac{dV}{d\theta} = \frac{sin(\theta) + \frac{R_c}{2} sin(2\theta)}{1 - cos(\theta) - \frac{R_c}{2} sin^2(\theta)} \tag{26}$$

and $\dot{V}$ is taken as :

$$\dot{V} = \frac{dV}{d\theta}\frac{d\theta}{dt} = \frac{dV}{d\theta}\omega \tag{27}$$

3.3 Combustion Process

During the process of combustion two discrete regions need to be accounted for: a region of the unburned fuel-air mixture and a region where combustion has already taken place and there exist only fully burned gases-products of combustion (*i.e.* CO_2, H_2O *etc*). Two different sets of equations need to be developed in order for combustion to be correctly modelled. These are :

$$\begin{aligned} \dot{T}_u &= (\frac{B}{A})_u[-\frac{\dot{V}}{V} + \frac{-\dot{Q}_{wu}}{(Bm)_u}], \\ \dot{p}_u &= 10^{-5}(\frac{\rho}{\frac{\partial\rho}{\partial p}})_u[-\frac{\dot{V}}{V} + \frac{1}{\rho_u}(\frac{\partial\rho}{\partial T})_u\dot{T}_u] \end{aligned} \tag{28}$$

and

$$\begin{aligned} \dot{T}_b &= (\frac{B}{A})_b[-\frac{\dot{V}}{V} + \frac{-\dot{Q}_{wb}}{(Bm)_b}], \\ \dot{p}_b &= 10^{-5}(\frac{\rho}{\frac{\partial\rho}{\partial p}})_b[-\frac{\dot{V}}{V} + \frac{1}{\rho_b}(\frac{\partial\rho}{\partial T})_b\dot{T}_b] \end{aligned} \tag{29}$$

where A and B for the unburned and burned gases are given by the following equations :

$$A_u = \frac{1}{\rho_u}\frac{(\frac{\partial\rho}{\partial T})_u}{(\frac{\partial\rho}{\partial p})_u} + c_{pu} \quad , \quad B_u = \frac{1}{(\frac{\partial\rho}{\partial p})_u} \tag{30}$$

and

$$A_b = \frac{1}{\rho_u}\frac{(\frac{\partial\rho}{\partial T})_u}{(\frac{\partial\rho}{\partial p})_u} + c_{pu} \quad , \quad B_b = \frac{1}{(\frac{\partial\rho}{\partial p})_b} \tag{31}$$

The subscripts u and b refer to "unburned" and "burned" regions.

Obviously the combustion process starts with burned quantities all set to zero and finishes with no remaining unburned quantities. It is also essential at this point to clarify that combustion does not occur instantaneously but between a small interval of crankshaft angle rotation (a few degrees before TDC and a few degrees after TDC). In spark ignition engines, this interval is normally between 30^o and 40^o.

3.4 Expansion Process

After the end of the combustion process, the expansion process starts and the cylinder charge consists of fully burned gases. During the expansion process the piston moves from TDC to BDC. The total mass inside the cylinder is constant. The equations of time rates of pressure and temperature are as follows:

$$\dot{T}_b = (\frac{B}{A})_b[-\frac{\dot{V}}{V} + \frac{-\dot{Q}_{wb}}{(Bm)_b}], \tag{32}$$

$$\dot{p}_b = 10^{-5}(\frac{\rho}{\frac{\partial\rho}{\partial p}})_b[-\frac{\dot{V}}{V} + \frac{1}{\rho_b}(\frac{\partial\rho}{\partial T})_b\dot{T}_b] \tag{33}$$

where A_b and B_b are the same as those of the combustion process.

3.5 Exhaust Process

The exhaust pressure p_r and temperature T_r are calculated by the following equations :

$$p_r = (1.05 \div 1.25)p_0, \tag{34}$$

$$T_r = \frac{T_b}{(\frac{p_b}{p_r})^{1/3}}. \tag{35}$$

It is obvious that the exhaust pressure p_r depends on the ambient pressure p_0 and the exhaust temperature T_r is specified using the burned gas temperature T_b at the end of the expansion process.

4 Physical and Numerical Considerations

For this study, a one-cylinder engine with 0.5-litre displacement volume was considered. The simulation input data are shown below:

Symbol	Value	Explanation
B	0.08	Cylinder Bore. Measured in m.
B_{iv}	0.032	Intake Valve diameter. Measured in m.
γ	1.3	C_p/C_v. Dimensionless.
ϵ	10	Compression Ratio.Dimensionless.
L	0.1	Cylinder's Stroke. Measured in m.
L_{iv}	0.0065	Maximum intake valve lift. Measured in m.
P_0	1	Atmospheric pressure. Measured in atm.
rpm	3000	Engine Rotation. Measured in rpm.
R_c	0.3	Ratio of crank radius to connecting rod length.
T_0	15	Atmospheric temperature. Measured in C^o
T_w	200	Temperature of chamber's wall. Measured in C^o.
U_{in}	60	Intake velocity of mixture. Measured in m/s.

4.1 Physical Approximations

In this report gasoline has been used as burning fuel. The chemical reaction which simulates the burning process of gasoline is:

$$C_7H_{17} + 11.27 \cdot (O_2 + 3.76N_2) \Rightarrow 7CO_2 + \frac{17}{2}H_2O + 3.76 \cdot 11.27N_2 \tag{36}$$

From the above equation and using stoichiometric quantities it is easy to calculate the molecular weights of the reactants and the products,respectively. We observe that the air is the main component from the reactants. The molecular weights of reactants is $MB_r = 33.32$ and of the products is $MB_p = 29.88$. This results as follows :

- Percentage of C_7H_{17} in reactants is 6%.
- Percentage of air in reactants is 94%.
- Molecular weight of C_7H_{17} is 101.
- Molecular weight of air is 29.

Hence, the molecular weight of reactants is $MB_r = 0.06MB_{C_7H_{17}} + 0.94MB_{air} = 32.33$..

- Percentage of CO_2 in products is 18%.
- Percentage of H_2O in products is 10%.
- Percentage of N_2 in products is 72%.
- Molecular weight of CO_2 is 44.
- Molecular weight of H_2O is 18.
- Molecular weight of N_2 is 28.

Hence, the molecular weight of products is $MB_p = 0.18MB_{CO_2} + 0.10MB_{H_2O} + 0.72MB_{N_2} = 29.88$.
In a similar manner the specific heat capacities at constant pressure and volume result as :

- $C_{p,gasoline} = 0.85$ kJ/kg k

- $C_{p,air} = 1$ kJ/kg k

So, the specific heat at constant volume for the reactants is given by $C_{pb} = 0.06C_{p,gasoline} + 0.94C_{p,air}$.

- $C_{p,CO_2} = 0.85$ kJ/kg k
- $C_{p,H_2O} = 4.18$ kJ/kg k
- $C_{p,N_2} = 1.042$ kJ/kg k

Obviously, the specific heat at constant volume for the reactants is given by $C_{pu} = 0.18C_{p,CO_2} + 0.1C_{p,H_2O} + 0.72C_{p,N_2}$.

Specific heat values at constant volume come from the equation $C_v = C_p - R^*$.

4.2 Wiebe Function

A functional form often used to represent the mass fraction burned versus crank angle is the *Wiebe Function*:

$$x_b = 1 - exp\Big[- \alpha \Big(\frac{\theta - \theta_0}{\Delta\theta} \Big)^{m+1} \Big] \tag{37}$$

where θ is the crank angle, θ_0 is the angle where the start of combustion occurs, $\Delta\theta$ is the total combustion duration and α and m are adjustable parameters. Actual mass fraction burned curves have been fitted with $\alpha = 5$ and $m = 2$.

Obviously, the mass fraction that remains unburned is given by $(1 - x_b)$.

4.3 Burn Rate Model

During combustion, close to the flame front turbulent eddies of characteristic radius l_t develop. These eddies are entrained into the flame zone at a characteristic time t_b. In this way, the rate of change of mass of the eddies is :

$$\frac{dm_e}{dt} = \rho_u A_f (U_t + S_t) \tag{38}$$

Then, the rate of change of mass of the burned fraction is :

$$\frac{dm_b}{dt} = \rho_u A_f U_t + \frac{m_e - m_b}{\tau_b} \tag{39}$$

where $\tau_b = l_t / S_t$ is the characteristic reaction time to burn the mass of an eddy of size l_t, U_t is the characteristic speed and S_l is the laminar flame speed, which is given by the equation:

$$S_l = \frac{B}{\Delta\theta \Big/ (2\pi n/60)} \tag{40}$$

with n being the frequancy of rotation of the crankshaft (rpm); A_f is the flame surface is corresponding to enflame volume V_f where

$$V_f = V_b + \frac{m_e - m_b}{\rho_u} \tag{41}$$

The parametres U_t and l_t are given by the equations:

$$U_t = \tag{42}$$

$$l_t = \tag{43}$$

$$\tag{44}$$

The burned fraction volume V_b is taken as :

$$V_b = x_b \dot{V} \tag{45}$$

Consequently the unburned fraction volume V_u is :

$$V_u = (1 - x_b)\dot{V} \tag{46}$$

Obviously,

$$\dot{V}_b = -\dot{V}_u \tag{47}$$

Similarly,

$$\dot{m}_b = -\dot{m}_u \tag{48}$$

4.4 Heat Transfer

Heat transfer in internal combustion engines is a serious problem since it needs high temperatures for fuel combustion but the temperature must be kept in controllable levels in order to operate the engine safely. Moreover heat transfer affects engine performance efficiency and emissions.

Heat transfer to and from the cylinder wall occurs throughout the four strokes of the engine's operation and is due to the convection and radiation. During the intake process the cylinder walls are hotter than the fuel, which causes vaporization of the fuel. During the compression stroke the gases become warmer than the cylinder walls and during the expansion stroke the greatest amount of heat transfer from the gases to the cylinder wall occurs. In the exhaust stroke the gases have cooled down so there is no significant amount of heat transfer.

An empirical formula that is used in this study is :

$$\dot{Q}_{wi} = A_{wi}[a\frac{k_i}{B}R_{ei}^b(T_i - T_w) + c(T_i^4 - T_{wi}^4)] \tag{49}$$

where a, b and c are constants and are chosen as a=0.35-0.8, b=0.7 and $c = 4.310^{-9}W/m^2K^4$; B is the cylinder bore; subscript i can be u for the unburned gases and b for the burned gases; T_w is the wall temperature and finally R_e is the Reynolds number. Typical values for Re for gasoline are of the order $10^4 - 10^5$.

4.5 Mixture Properties

In this work, during combustion because of the existence of two distinct regions (burned and unburned) the properties of the mixture have been averaged.

Thus, the total density is :

$$\rho_{tot} = x_b\rho_b + (1 - x_b)\rho_u \tag{50}$$

where

$$\rho_u = \frac{m_u}{V_u} \tag{51}$$

and

$$\rho_b = \frac{m_b}{V_b} \tag{52}$$

According to Dalton's Law for pressures we have :

$$P_{tot} = P_b + P_u \tag{53}$$

Total temperature can be calculated directly from the idel gas law, i.e.

$$T_{tot} = \frac{P_{tot}}{\rho_{tot}R/MB} \tag{54}$$

4.6 Numerical Approximations

In order to solve the system of ordinary differential equation for the compression and expansion, a fourth order Runge-Kutta with variable step is used. The method maintained the coupling between the two equations for temperature and pressure.

For the equations during combustion a different approach was followed. As seen from equations (28) (29) the derivatives of density with respect to pressure and temperature come into these equations. As discussed earlier, the thermodynamic process of combust in a spark ignition engine occurs under constant volume. By trying to calculate these derivatives in a similar manner as in the case of adiabatic compression and expansion, we result to highly complicated non-linear equations with respect to $\dot{T}$ and $\dot{P}$.

There exist no formal numerical techniques in the literature for the solution of such a system of equations. Hence, a simple approach was chosen, with the derivatives of density with respect to temperature and pressure to be calculated at times (n) and (n-1) (for example: $d\rho/dp = (\rho^n - \rho^{n-1})/(p^n - p^{n-1})$) and these were used in the $\dot{T}$ and $\dot{P}$ equations in order to calculate temperature and pressure at time (n+1) through the direct application of Euler's method. The step was chosen to be very small (10% of the step used in compression) in order to minimise any numerical errors.

5 Model Inputs

The inputs to the numerical model are the following: atmospheric temperature measured in ^oC; atmospheric pressure, measured in atm; engine rotation measured in rpm; the compression ratio; the number of perform cycles; the speed heat ratio; temperature of cylinder's wall measured in ^oC; cylinder bore measured in m; cylinder's stroke measured in m; ratio of crank radius to connecting rod length; intake velocity, measured in m/s; intake valve diameter measured in m; maximum intake valve lift measured in m; the angle when combustion starts and the angle when combustion finishes.

6 Results

In this section some results of a simulation for one cycle will be presented. Compression finished at 330^o and combustion duration was for 40^o rotation of the crankshaft (combustion finished at 10^o after TDC).

Figure 8 shows pressure vs angle during the compression stage. The duration of this stage was 150^o of the crankshaft rotation (from 180^o to 330^o).

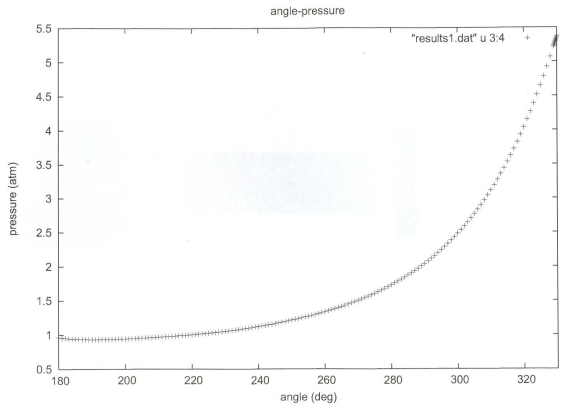


Figure 8: *Predicted pressure in cylinder during the compression process.*

The pressure as expected was seen to increase, attaining a maximum value of 5.37 Atm. This value is in reasonable agreement for an engine of similar characteristics, as seen in Figure 9.

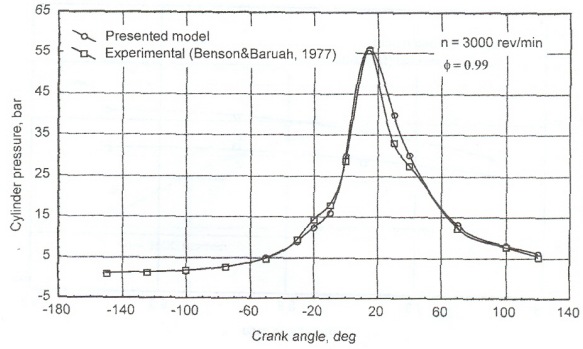


Figure 9: *Pressure evolution (theoretical and experimental) during compression and combustion in am internal combustion engine from Bayractar and Durgun, 2003.*

Figure 10 shows the predicted burned mass fraction during combustion.

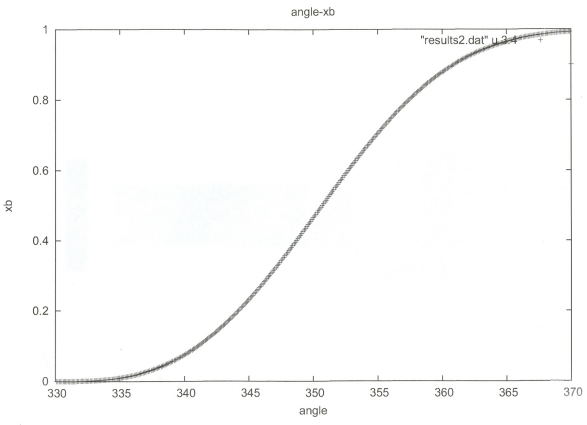


Figure 10: *Predicted burned mass fraction during combustion.*

It was seen to increase very slowly at first, then suddenly most of the fuel was burned (around TDC) and then slowly the fraction approached unity at the end of the combustion. This result was in good agreement with theoretical and experimental observations as those given in Figure 11 for an engine of similar characteristics.

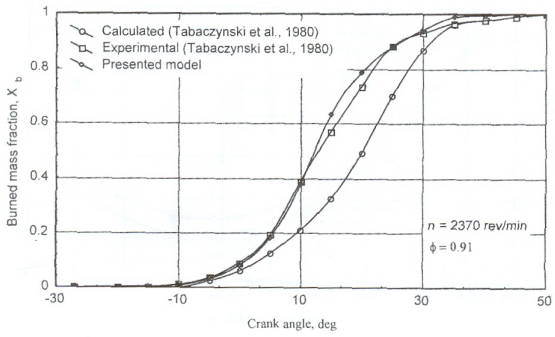


Figure 11: *Calculated and experimental burned mass fractions during combustion in an internal combustion engine from Bayractar and Durgun, 2003.*

7 Conclusions

In this report, a mathematical and numerical model was presented for a four stroke spark ignition engine, based on a quasi-dimensional thermodynamic cycle model.

Simulations were performed for one cycle of the engine and the obtained results were in reasonable agreement with theoretical observations and experimental data.

Further work is required for the construction of robust numerical techniques for the solution of the non-linear coupled ordinaries differential equations for combustion.

Despite their limitations, these models have the potential for fast and accurate predictions of the evolution of temperature and pressure in engine cylinders and are in a position to simulate very well the combustion process in spark ignition engines.

8 Appendix - Computer Code in C

```
#include <stdlib.h> #include <stdio.h> #include <math.h> #include
"struct.h" #define pi M_PI #define Rgas 8314. #define Re 80000.
#define tol 1.e-1 #define MBr 33.32 #define MBp 28.88

void InitDat(char *file,struct data *pdat); void check(double
*ch); double error(double er1, double er2); double enthalpy(double
t, double MB); double volume(double angle, struct data *pdat);
double density(double temp, double pres, double MB); double
dQwall(int cycl, double angle, double t, struct data *pdat); void
Wiebe(double angle, double angleS, double angleF, double *xb,
double *dxb); double dhdp(char *gas, double p1, double p2, double
h1, double h2, struct data *pdat); double drdp(char *process,
double t, double p1, double p2, double r1, double r2, struct data
*pdat); double drdt(char *process, double t1, double t2, double
r1, double r2, struct data *pdat); double Bhta(char *process, char
*gas, double p1, double p2, double t1, double t2, double r1,
double r2, double h1, double h2, struct data *pdat); double
Alpha(char *process, char *gas, double p1, double p2, double t1,
double t2, double r1, double r2, double h1, double h2, struct data
*pdat); double funcf1(char *process, char *gas, int cycl, double
angle, double t1, double t2, double t3, double p1,\
              double p2, double v1, double v2, double v1p, double v2p, double dv1, double dv2\
        , double m1, double m2, double m1p, double m2p, double dm1, double dm2, struct data *pdat);
double funcf2(char *process, char *gas, int cycl, double angle,
double t1, double t2, double t3, double p1,\
              double p2, double v1, double v2, double v1p, double v2p, double dv1, double dv2,\
         double m1, double m2, double m1p, double m2p, double dm1, double dm2, struct data *pdat);
void rk4(char *process, char *gas, int cycl, double sangle, double
angle, double p1, double t1, double *p2, double *t2, double v,
double m, struct data *pdat); void engine(struct data *pdat);

// ************************************************
// *               Main Programm                  *
// ************************************************
int main(int argc,char **argv){

    int i,j;
    struct data dat, *pdat;
    int cycl;
    char file[10];

    printf("\n\nMathematical and Numerical Modeling of Flow\n");
    printf("and Combustion Processes in a SI Engine.\n\n");

    if (argc!=2){
        printf("Error in opening the file.\n");
        printf("Usage:\n %s datafile\n",argv[0]);
        exit(1);
    }

    pdat=&dat;

    InitDat(argv[1],&dat);

    engine(pdat);

    return 0;
}
```

```
// ======================================================================
// ======================================================================
// Function which read the data from a dat file
void InitDat(char *file,struct data *pdat){

    FILE *fp;
    double at, ap, tw, a1, a2;
    at=ap=tw=a1=a2=0.;

    if ((fp=fopen(file, "r")) == NULL){

        printf("Error in opening the file.\n");
        exit(1);
        }

    fscanf(fp,"%lf %lf %lf %lf %d %lf %lf %lf %lf %lf %lf %lf %lf %lf %lf",
    &at,&ap,
    &(*pdat).rpm,&(*pdat).compratio,&(*pdat).ncycl,
    &(*pdat).gamma,
    &tw,
    &(*pdat).B,&(*pdat).L,&(*pdat).R,
    &(*pdat).Ui,&(*pdat).Bv,&(*pdat).Liv,
    &a1,&a2);

    (*pdat).Ap=pi*((*pdat).B/2.)*((*pdat).B/2.);
    (*pdat).Av=pi*((*pdat).Bv/2.)*((*pdat).Bv/2.);

    (*pdat).atmopres=ap;

    (*pdat).atmotemp=at + 273.;
    (*pdat).Tw=tw + 273.;

    (*pdat).sacomb=a1*pi/180.;
    (*pdat).facomb=a2*pi/180.;

    (*pdat).Vc=(1./((*pdat).compratio-1.))*0.25*pi*(*pdat).B*(*pdat).B*(*pdat).L;

    (*pdat).omega=2.*pi*(*pdat).rpm/60.;
}
// ======================================================================
// Function which simulates the engine.
void engine(struct data *pdat){

    FILE *fp1, *fp2;
    char b, u;
    char COMBUSTION, COMPRESSION, EXPANSION;
    int cycl, logical;
    double xb, dxb;
    double angle, sangle;
    double Af, Ut, lt, Sl, hc;
    double Vf, Vu, dVu, Vb, dVb;
    double densi, denstot, densu ,densb;
    double me, dme, mu, dmu, mb, dmb, MBtot;
    double Temp, Pres, Temp1, Pres1, Temp2, Pres2, Temp3, Pres3;
    double Tempu, Presu, Tempb, Presb,\
         Tempu1, Presu1, Tempb1, Presb1, Tempu2, Tempb2, Presb2, Presu2;
    double dTempu, dPresu, dTempb, dPresb, dTempu1, dPresu1, dTempb1, dPresb1;
        double mu_p, mb_p, Vu_p, Vb_p;

    cycl=0;
```

```
    xb=dxb=0.;
    sangle=pi/180.;
    Af=Ut=lt=Sl=hc=0.;
    Vf=Vu=dVu=Vb=dVb=0.;
    densi=denstot=densu=densb=0.;
    MBtot=me=dme=mu=dmu=mb=dmb=0.;
    Temp=Pres=Temp1=Pres1=Temp2=Pres2=Temp3=Pres3=0.;
    Tempu=Presu=Tempb=Presb=Tempu1=
        =Presu1=Tempb1=Presb1=Tempu2=Presu2=Tempb2=Presb2=0.;
    dTempu=dPresu=dTempb=dPresb=dTempu1=dPresu1=dTempb1=dPresb1=0.;
        mu_p = mb_p = Vu_p = Vb_p = 0.0;

    if (((fp1=fopen("results1.dat","w")) == NULL)){
                printf("Programm couldn't open the file.\n");
                exit(1);
    }else if (((fp2=fopen("results2.dat","w")) == NULL)){
        printf("Programm couldn't open the file.\n");
        exit(1);
    }else{
        for(cycl=1;cycl<=(*pdat).ncycl;cycl++){

//      Beginning of Intake Process
            densi=density((*pdat).atmotemp,(*pdat).atmopres,MBr);

            if(cycl == 1){
                Pres=(*pdat).atmopres-1.e-5*.5*densi*pow((*pdat).Ui,2.)\
                    *(pow((*pdat).Ap,2.)-pow((*pdat).Av,2.))/pow((*pdat).Ap,2.);
                Temp=(*pdat).atmotemp;
                printf(".");
                fprintf(fp1,"%d \t   %lf %lf %lf \n",cycl,270.,Pres,Temp);
            }else{
                Pres=(*pdat).atmopres-1.e-5*.5*densi*pow((*pdat).Ui,2.)\
                    *(pow((*pdat).Ap,2.)-pow((*pdat).Av,2.))/pow((*pdat).Ap,2.);
                Temp=(1./3.)*((*pdat).Tw + 2*Temp1);
                printf(".");
                fprintf(fp1,"%d \t   %lf %lf %lf \n",cycl,270.,Pres,Temp);
            }

            Vu=(*pdat).Vc+.25*pi*(*pdat).B*(*pdat).B*(*pdat).L;
            mu=density(Temp,Pres,MBr)*Vu;

//      Beginning of compression Process
            angle=pi;
            sangle=pi/180.;
            Temp1=Pres1=Temp2=Pres2=0.;
            while(angle<2*pi-(*pdat).sacomb-2.*pi/180.){
                logical=1;
                while(logical == 1){
                    Pres1=Pres;Temp1=Temp;
                    Pres2=Pres;Temp2=Temp;
                    Vu=volume(angle,pdat);              check(&Vu);
                    rk4("COMPRESSION","u",cycl,sangle,angle,Pres,Temp,&Pres1,&Temp1,Vu,mu,pdat);
//                  BYPASS PRESSURE CALCULATION-CALCULATE USING IDEAL GAS LAW
                    Pres1 = (mu/Vu) * ( Rgas*Temp1/MBr ) *1.e-5;
                    rk4("COMPRESSION","u",cycl,.5*sangle,angle,Pres,Temp,&Pres2,&Temp2,Vu,mu,pdat);
                    Pres2 = (mu/Vu) * ( Rgas*Temp2/MBr ) *1.e-5;
                    Vu=volume(angle+.5*sangle,pdat) ;       check(&Vu);
                    rk4("COMPRESSION","u",cycl,.5*sangle,angle+.5*sangle,Pres,Temp,&Pres3,&Temp3,\
                        Vu,mu,pdat);
                    Pres3 = (mu/Vu) * ( Rgas*Temp3/MBr ) *1.e-5;
```

```
                if((error(Pres1,Pres3)<=tol)&&(error(Temp1,Temp3)<=tol)){
                    Pres=Pres1;Temp=Temp1;
                    printf(".");
                    fprintf(fp1,"%d %lf %lf %lf %lf \n",cycl,180.*sangle/pi,180.*angle/pi,Pres,Temp);
                    Temp1=Pres1=Temp2=Pres2=Temp3=Pres3=0.;
                    angle += sangle;
                    logical=0;
                }else{
                    sangle=.5*sangle;
                    if(sangle<1.e-6){
                        Pres=Pres1;Temp=Temp1;
                        Temp1=Pres1=Temp2=Pres2=Temp3=Pres3=0.;
                        angle += sangle;
                        logical=0;
                    }
                }
            }
        }

        /*Still in Compression, 1 deg before the start of Combustion
          Reduce step to 1/10 of what it is
          Save values at time (n-1) in order to use in combustion      */
        sangle=.1*sangle;
        while(angle<2.*pi-(*pdat).sacomb){
            Pres1=Pres;Temp1=Temp;
            Vu=volume(angle,pdat);                  check(&Vu);
            rk4("COMPRESSION","u",cycl,sangle,angle,Pres,Temp,&Pres1,&Temp1,Vu,mu,pdat);
//          BYPASS PRESSURE CALCULATION-CALCULATE USING IDEAL GAS LAW
            Pres1 = (mu/Vu) * ( Rgas*Temp1/MBr ) *1.e-5;
            Temp2=Pres2=0.;
            Pres2=Pres;Temp2=Temp;
            Pres=Pres1;Temp=Temp1;
            Pres1=Pres2;Temp1=Temp2;
            printf(".");
            fprintf(fp1,"%d %lf %lf %lf %lf \n",cycl,180.*sangle/pi,180.*angle/pi,Pres,Temp);
                        mu_p = mu;
                        Vu_p = Vu;
            angle += sangle;
        }


//      Beginning of Combustion Process
                    Vu = volume(angle,pdat);
            densu=mu/Vu;                            check(&densu);
            Presu=Pres;Presu1=Pres1;
            Tempu=Temp;Tempu1=Temp1;
            while(angle<2.*pi+(*pdat).facomb){
                Wiebe(angle,2.*pi-(*pdat).sacomb,2.*pi+(*pdat).facomb,&xb,&dxb);

                //CAREFUL!!  INCONSISTENCY BETWEEN VOLUMES
                //SWITCH SUDDENLY BETWEEN Vc+sthing small TO FRACTION OF Vc
                Vb=xb*volume(angle,pdat);                   check(&Vb);
                dVb=dxb*volume(angle,pdat)+\
                    +xb*(*pdat).Vc+.25*pi*(*pdat).B*(*pdat).B*\
                    *.5*(*pdat).L*(sin(angle)+(*pdat).R*sin(angle)*cos(angle));
                                            check(&dVb);
                Vu=(1-xb)*volume(angle,pdat);               check(&Vb);
                dVu=-dVb;                   check(&dVu);
                if((mb<=1.e-9)&&(Vb<=1.e-10)){  densb=1.e-10;}
                else{ if(mb<=1.e-9){    densb=1.e-10;}
```

```
                    else{   densb=mb/Vb;                check(&densb);}
                }
                Vf=Vb+(me-mb)/densu;
                if (Vf<=0.0){Vf = 1.e-10;}              check(&Vf);
                hc=Vf/(pi*pow((*pdat).B,2.)/4.);            check(&hc);
                Af=.25*pi*pow(.5*hc,2.);                check(&Af);
                Ut=.08*(*pdat).Ui*sqrt(densu/densi);            check(&Ut);
                lt=.8*(*pdat).Liv*pow(densi/densu,3./4.);       check(&lt);
                Sl=(*pdat).B*(*pdat).omega/( (*pdat).facomb+(*pdat).sacomb );
                                        check(&Sl);
                dme=densu*Af*(Ut+Sl)*(1./(*pdat).omega);        check(&dme);
                me += sangle*dme;                   check(&me);
                dmb=(densu*Af*Sl+Sl*(me-mb)/lt)*(1./(*pdat).omega); check(&dmb);
                mb += sangle*dmb;                   check(&mb);
                dmu=-dmb;                       check(&dmu);
                mu += sangle*dmu;                   check(&mu);

                dTempb=funcf1("COMBUSTION","b",cycl,angle,\
                Tempb,Tempb1,Tempu,Presb,Presb1, \
                Vu,Vb,Vu_p,Vb_p,dVu,dVb,mu,mb,mu_p,mb_p,dmu,dmb,pdat);
                Tempb2 += sangle*dTempb;

                dPresb=funcf2("COMBUSTION","b",cycl,angle,\
                Tempb,Tempb1,Tempu,Presb,Presb1,\
                Vu,Vb,Vu_p,Vb_p,dVu,dVb,mu,mb,mu_p,mb_p,dmu,dmb,pdat);
                Presb2 += sangle*dPresb;

//              Bypass Pressure calculation-Use ideal gas law
                        if (mb<=1.e-5) {Presb2 = 1.e-10;}
                        else
                        {Presb2 = (mb/Vb) * (Rgas/MBp) * Tempb2 *1.e-5;}

                dTempu=funcf1("COMBUSTION","u",cycl,angle,\
                Tempu,Tempu1,0.,Presu,Presu1,\
                Vu,Vb,Vu_p,Vb_p,dVu,dVb,mu,mb,mu_p,mb_p,dmu,dmb,pdat);
                Tempu2 += sangle*dTempu;

                dPresu=funcf2("COMBUSTION","u",cycl,angle,\
                Tempu,Tempu1,0.,Presu,Presu1,\
                Vu,Vb,Vu_p,Vb_p,dVu,dVb,mu,mb,mu_p,mb_p,dmu,dmb,pdat);
                Presu2 += sangle*dPresu;

//              Bypass Pressure calculation-Use ideal gas law
                        if (mu<=1.e-5) {Presu2 = 1.e-10;}
                        else
                        Presu2 = (mu/Vu) * (Rgas/MBr) * Tempu2 *1.e-5;

                Pres=Presu2+Presb2;
                MBtot=xb*MBp+(1.-xb)*MBr;
                denstot=xb*densb+(1.-xb)*densu;
                Temp=(Pres*MBtot)/(denstot*Rgas);
                printf(".");
                fprintf(fp1,"%d %lf %lf %lf %lf \t %lf %lf %lf %lf \n"\
                ,cycl,180.*sangle/pi,180.*angle/pi,Pres,Temp,Presu2,Tempu2,Presb2,Tempb2);
                fprintf(fp2,"%d %lf %lf %lf %lf \n"\
                ,cycl,180.*sangle/pi,180.*angle/pi,xb,1.-xb);
                Presu1=Presu;Presb1=Presb;
                Tempu1=Tempu;Presb1=Tempb;
                Presu=Presu2;Presb=Presb2;
                Tempu=Tempu2;Presb=Tempb2;
```

```
                if((mu<=1.e-10)&&(Vu<=1.e-10)){ densu=1.e-10;}   //Added conditions
                else{ if(mu<=1.e-10){   densu=1.e-10;}
                    else{   densu=mu/Vu;                check(&densu);}
                }

                            //Allocate mass-volume previous values
                mu_p = mu;
                Vu_p = Vu;
                mb_p = mb;
                Vb_p = Vb;

                angle += sangle;
            }

//      Beginning of Expansion Process
            sangle=pi/180.;
            Temp1=Pres1=Temp2=Pres2=0.;
            while(angle<=3*pi){
                logical=1;
                while(logical == 1){
                    Pres1=Pres;Temp1=Temp;
                    Pres2=Pres;Temp2=Temp;
                    rk4("EXPANSION","b",cycl,sangle,angle\
                    ,Pres,Temp,&Pres1,&Temp1,Vb,mb,pdat);
                    rk4("EXPANSION","b",cycl,.5*sangle,angle\
                    ,Pres,Temp,&Pres2,&Temp2,Vb,mb,pdat);
                    Vb=volume(angle+.5*sangle,pdat);            check(&Vu);
                    rk4("EXPANSION","b",cycl,.5*sangle,angle+.5*sangle\
                    ,Pres2,Temp2,&Pres3,&Temp3,Vb,mb,pdat);
                    if((error(Pres1,Pres3)<=tol)&&(error(Temp1,Temp3)<=tol)){
                        Pres=Pres1;Temp=Temp1;
                        printf(".");
                        fprintf(fp1,"%d %lf %lf %lf \n",cycl,180.*angle/pi,Pres,Temp);
                        Pres1=Temp1=Pres2=Temp2=Pres3=Temp3=0.;
                        angle += sangle;
                        logical=0;
                    }else{
                        sangle=.5*sangle;
                    }
                }
            }

//      Beginning of Exhaust Process
            Pres1=1.15*(*pdat).atmopres;
            Temp1=Temp / pow((Pres/Pres1),(1./3.));
            Pres=Pres1;Temp=Temp1;
            printf(".");
            fprintf(fp1,"%d \t   %lf %lf %lf",cycl,180.*angle/pi,Pres,Temp);

            fprintf(fp1,"\n");
            fprintf(fp2,"\n");
        }
    }
    fclose(fp1);
    fclose(fp2);
}
// ===================================================================
// Function for the Fourth order Runge-Kutta
void rk4(char *process, char *gas, int cycl, double sangle, double
```

```
    angle, double p1, double t1, double *p2, double *t2, double v,
    double m, struct data *pdat){

    double k1,k2,k3,k4,k;
    double l1,l2,l3,l4,l;

    k1=k2=k3=k4=k=0.;
    l1=l2=l3=l4=l=0.;

    k1=sangle*funcf1(process,gas,cycl,angle,\
        t1,0.,0.,p1,0.,v,0.,0.,0.,0.,0.,m,0.,0.,0.,0.,0.,pdat);
    l1=sangle*funcf2(process,gas,cycl,angle,\
        t1,0.,0.,p1,0.,v,0.,0.,0.,0.,0.,m,0.,0.,0.,0.,0.,pdat);

    k2=sangle*funcf1(process,gas,cycl,angle+.5*sangle,\
        t1+.5*k1,0.,0.,p1+.5*l1,0.,v,0.,0.,0.,0.,0.,m,0.,0.,0.,0.,0.,pdat);
    l2=sangle*funcf2(process,gas,cycl,angle+.5*sangle,\
        t1+.5*k1,0.,0.,p1+.5*l1,0.,v,0.,0.,0.,0.,0.,m,0.,0.,0.,0.,0.,pdat);

    k3=sangle*funcf1(process,gas,cycl,angle+.5*sangle,\
        t1+.5*k2,0.,0.,p1+.5*l2,0.,v,0.,0.,0.,0.,0.,m,0.,0.,0.,0.,0.,pdat);
    l3=sangle*funcf2(process,gas,cycl,angle+.5*sangle,\
        t1+.5*k2,0.,0.,p1+.5*l2,0.,v,0.,0.,0.,0.,0.,m,0.,0.,0.,0.,0.,pdat);

    k4=sangle*funcf1(process,gas,cycl,angle+sangle,\
        t1+k3,0.,0.,p1+l3,0.,v,0.,0.,0.,0.,0.,m,0.,0.,0.,0.,0.,pdat);
    l4=sangle*funcf2(process,gas,cycl,angle+sangle,\
        t1+k3,0.,0.,p1+l3,0.,v,0.,0.,0.,0.,0.,m,0.,0.,0.,0.,0.,pdat);

    k=(1./6.)*(k1+2.*k2+2.*k3+k4);
    *p2 =p1 + k;

    l=(1./6.)*(l1+2.*l2+2.*l3+l4);
    *t2 =t1 +  l;
}
// ====================================================================
// Function which calculates the derivative of temperature
double funcf1(char *process, char *gas, int cycl, double angle,
        double t1, double t2, double t3, double p1,
        double p2, double v1, double v2, double v1p, double v2p, double
        dv1, double dv2, double m1, double m2, double m1p, double m2p,
        double dm1, double dm2, struct data *pdat){

    double w,rho1,rho2,h1,h2;
    double AlphaU, BhtaU, AlphaB, BhtaB;
    w=rho1=rho2=h1=h2=0.;

    if(strcmp(process,"COMPRESSION")==0){
        if(v1<=(*pdat).Vc){
            rho1=m1/(*pdat).Vc;     check(&rho1);
        }else{
            rho1=m1/v1;             check(&rho1);
        }
        w=(Bhta(process,gas,p1,p2,t1,t2,rho1,0.,0.,0.,pdat)\
            /Alpha(process,gas,p1,p2,t1,t2,rho1,0.,0.,0.,pdat));
        w*=(-((sin(angle)+.5*sin(2.*angle)*(*pdat).R)/(1.-cos(angle)\
           +.5*(*pdat).R*pow(sin(angle),2.)))-(1./(*pdat).omega)\
            *dQwall(cycl,angle,t1,pdat)/(Bhta(process,gas,p1,p2,t1,t2,rho1,0.,0.,0.,pdat)*m1));
    }
    else if(strcmp(process,"COMBUSTION")==0){
```

```
        if(strcmp(gas,"u")==0){
            h1=enthalpy(t1,MBr);
            if((m1<=1.e-10)&&(v1<=1.e-10)){ rho1=1.e-10;}
            else{ if(m1<=1.e-10){   rho1=1.e-10;}
                else{   rho1=m1/v1;             check(&rho1);}
            }
            h2=enthalpy(t2,MBr);
            if((m1p<=1.e-10)&&(v1p<=1.e-10)){   rho2=1.e-10;}
            else{ if(m1p<=1.e-10){  rho2=1.e-10;}
                else{   rho2=m1p/v1p;               check(&rho2);}
            }
                        AlphaU = Alpha(process,gas,p1,p2,t1,t2,rho1,rho2,h1,h2,pdat); //check(&AlphaU);
                        if (AlphaU >=1.+20) AlphaU = 1.e+20;
                        if (AlphaU <=-1.+20) AlphaU = -1.e+20;
                        BhtaU = Bhta(process,gas,p1,p2,t1,t2,rho1,rho2,h1,h2,pdat); //check(&BhtaU);
                        if (BhtaU >=1.+20) BhtaU = 1.e+20;
                        if (BhtaU <=-1.+20) BhtaU = -1.e+20;
            w = BhtaU / AlphaU;
        printf("%lf \t bhtaU=%lg, alphaU=%lg\n",180.*angle/pi,BhtaU, AlphaU);
            w*=((1./(*pdat).omega)*(dm1/m1)*(1.-h1/BhtaU)-(dv1/v1)+(1./(*pdat).omega)*\
                (-dQwall(cycl,angle,t1,pdat)+dm1*(5./2.)*(Rgas/MBr)*t1)/(BhtaU*m1));
        }
        else if(strcmp(gas,"b")==0){
            h1=enthalpy(t1,MBp);
            if((m2<=1.e-10)&&(v2<=1.e-10)){ rho1=1.e-10;}
            else{ if(m2<=1.e-10){   rho1=1.e-10;}
                else{   rho1=m2/v2;             check(&rho1);}
            }
            h2=enthalpy(t2,MBp);
            if((m2p<=1.e-10)&&(v2p<=1.e-10)){   rho2=1.e-10;}
            else{ if(m2p<=1.e-10){  rho2=1.e-10;}
                else{   rho2=m2p/v2p;               check(&rho2);}
            }
            AlphaB = Alpha(process,gas,p1,p2,t1,t2,rho1,rho2,h1,h2,pdat); //check(&AlphaB);
                        if (AlphaB >=1.+20) AlphaB = 1.e+20;
                        if (AlphaB <=-1.+20) AlphaB = -1.e+20;
            BhtaB = Bhta(process,gas,p1,p2,t1,t2,rho1,rho2,h1,h2,pdat); //check(&BhtaB);
                        if (BhtaB >=1.+20) BhtaB = 1.e+20;
                        if (BhtaB <=-1.+20) BhtaB = -1.e+20;
            w = BhtaB / AlphaB;
        printf("%lf \t bhtaB=%lg, alphaB=%lg\n",180.*angle/pi,BhtaB, AlphaB);
            w*=((1./(*pdat).omega)*(dm2/m2)*(1.-h1/BhtaB)-(dv2/v2)+(1./(*pdat).omega)*\
                (-dQwall(cycl,angle,t1,pdat)+dm2*(5./2.)*(Rgas/MBr)*t3)/(BhtaB*m2));
        }
    }
    else if(strcmp(process,"EXPANSION")==0){
        if(v1<=(*pdat).Vc){
            rho1=m1/(*pdat).Vc;     check(&rho1);
        }else{
            rho1=m1/v1;         check(&rho1);
        }
        w=(Bhta(process,gas,p1,p2,t1,t2,rho1,0.,0.,0.,pdat)\
            /Alpha(process,gas,p1,p2,t1,t2,rho1,0.,0.,0.,pdat));
        w*=(-((sin(angle)+.5*sin(2*angle)*(*pdat).R)/(1.-cos(angle)\
            +.5*(*pdat).R*pow(sin(angle),2.)))-(1./(*pdat).omega)\
            *(dQwall(cycl,angle,t1,pdat))/(Bhta(process,gas,p1,p2,t1,t2,rho1,0.,0.,0.,pdat)*m2));
    }
    return(w);
}
// Function which calculates the derivative of pressure
```

```
double funcf2(char *process, char *gas, int cycl, double
angle,double t1, double t2, double t3, double p1, double p2,
double v1, double v2, double v1p, double v2p, double dv1, double
dv2, double m1, double m2, double m1p, double m2p, double dm1,
double dm2, struct data *pdat){

    double w,rho1,rho2,h1,h2;
    w=rho1=rho2=h1=h2=0.;

    if(strcmp(process,"COMPRESSION")==0){
        if(v1<=(*pdat).Vc){
            rho1=m1/(*pdat).Vc;     check(&rho1);
        }else{
            rho1=m1/v1;         check(&rho1);
        }
        w=1.e-5*rho1*(1./drdp(process,t1,p1,p2,rho1,rho2,pdat));
        w*=(-((sin(angle)+.5*sin(2.*angle)\
            *(*pdat).R)/(1.-cos(angle)+.5*(*pdat).R*pow(sin(angle),2.)))-\
            (1./rho1)*drdt(process,t1,t2,rho1,0.,pdat)*funcf1(process,gas,cycl,angle,\
               t1,t2,t3,p1,p2,v1,v2,v1p,v2p,dv1,dv2,m1,m2,m1p,m2p,dm1,dm2,pdat));
    }
    else if(strcmp(process,"COMBUSTION")==0){
        if(strcmp(gas,"u")==0){
            h1=enthalpy(t1,MBr);
            if((m1<=1.e-10)&&(v1<=1.e-10)){ rho1=1.e-10;}
            else{ if(m1<=1.e-10){   rho1=1.e-10;}
                else{   rho1=m1/v1;             check(&rho1);}
            }
            h2=enthalpy(t2,MBr);
            if((m1p<=1.e-10)&&(v1p<=1.e-10)){   rho1=1.e-10;}
            else{ if(m1p<=1.e-10){  rho2=1.e-10;}
                else{   rho2=m1p/v1p;               check(&rho2);}
            }
            w=rho1*(1./drdp(process,t1,p1,p2,rho1,rho2,pdat))*(-(dv1/v1)-(1./rho1)*\
                        drdt(process,t1,t2,rho1,rho2,pdat)*\
                        funcf1(process,gas,cycl,angle,\
                        t1,t2,t3,p1,p2,v1,v2,v1p,v2p,dv1,dv2,\
                        m1,m2,m1p,m2p,dm1,dm2,pdat)+(dm1/m1));
        }
        else if(strcmp(gas,"b")==0){
            h1=enthalpy(t1,MBp);
            if((m2<=1.e-10)&&(v2<=1.e-10)){ rho1=1.e-10;}
            else{ if(m2<=1.e-10){   rho1=1.e-10;}
                else{   rho1=m2/v2;             check(&rho1);}
            }
            h2=enthalpy(t2,MBp);
            if((m2p<=1.e-10)&&(v2p<=1.e-10)){   rho1=1.e-10;}
            else{ if(m2p<=1.e-10){  rho2=1.e-10;}
                else{   rho2=m2p/v2p;               check(&rho2);}
            }
            w=rho1*(1./drdp(process,t1,p1,p2,rho1,rho2,pdat))*(-(dv2/v2)-(1./rho2)*\
                        drdt(process,t1,t2,rho1,rho2,pdat)*\
                        funcf1(process,gas,cycl,angle,\
                        t1,t2,t3,p1,p2,v1,v2,v1p,v2p,dv1,dv2,\
                        m1,m2,m1p,m2p,dm1,dm2,pdat)+(dm2/m2));
        }
    }
    else if(strcmp(process,"EXPANSION")==0){
        if(v1<=(*pdat).Vc){
            rho1=m1/(*pdat).Vc;     check(&rho1);
```

```
        }else{
            rho1=m1/v1;         check(&rho1);
        }
        w=1.e-5*rho1*(1./drdp(process,t1,p1,p2,rho1,0.,pdat));
        w*=(-((sin(angle)+.5*sin(2.*angle)*(*pdat).R)/(1.-cos(angle)+.5*(*pdat).R*\
                    pow(cos(angle),2.)))-(1./rho1)*drdt(process,t1,t2,rho1,0.,pdat)*\
                    funcf1(process,gas,cycl,angle,\
                    t1,t2,t3,p1,p2,v1,v2,v1p,v2p,dv1,dv2,\
                    m1,m2,m1p,m2p,dm1,dm2,pdat));
    }
    return(w);
}
// Function which calculates A
double Alpha(char *process, char *gas, double p1, double p2,
double t1, double t2, double rho1, double rho2, \
       double h1, double h2, struct data *pdat){

    double w, Cpu, Cpb;
    w=Cpu=Cpb=0.;

    if(strcmp(process,"COMPRESSION")==0){
        Cpu=.08*2220. + .92*1000.;
        w=(drdt(process,t1,t2,rho1,rho2,pdat)\
            /drdp(process,t1,p1,p2,rho1,rho2,pdat))*(1./rho1)+Cpu;
    }
    else if(strcmp(process,"COMBUSTION")==0){
         if(strcmp(gas,"u")==0){
            Cpu=.08*2220. + .92*1000.;
            w=(drdt(process,t1,t2,rho1,rho2,pdat)\
                /drdp(process,t1,p1,p2,rho1,rho2,pdat))*(1./rho1)+Cpu;
        }
        else if(strcmp(gas,"b")==0){
            Cpb=.18*850.+.1*4180.+.72*1042.;
            w=(drdt(process,t1,t2,rho1,rho2,pdat)\
                /drdp(process,t1,p1,p2,rho1,rho2,pdat));
            w*=((1./rho1)-(1.e-5)*dhdp(gas,p1,p2,h1,h2,pdat))+Cpb;
        }
    }
    else if(strcmp(process,"EXPANSION")==0){
        Cpb=.18*850.+.1*4180.+.72*1042.;
        w=(drdt(process,t1,t2,rho1,rho2,pdat)\
            /drdp(process,t1,p1,p2,rho1,rho2,pdat));
        w*=((1./rho1)-(1.e-5)*dhdp(gas,p1,p2,h1,h2,pdat))+Cpb;
    }
    return(w);
}
// Function which calculates Bhta
double Bhta(char *process, char *gas, double p1, double p2, double
t1, double t2, double rho1, \
            double rho2, double h1, double h2, struct data *pdat){

    double w=0.;

    if(strcmp(process,"COMPRESSION")==0){
        w=(1./drdp(process,t1,p1,p2,rho1,rho2,pdat));
    }
    else if(strcmp(process,"COMBUSTION")==0){
         if(strcmp(gas,"u")==0){
            w=(1./drdp(process,t1,p1,p2,rho1,rho2,pdat));
        }
```

```
        else if(strcmp(gas,"b")==0){
            w=(1./drdp(process,t1,p1,p2,rho1,rho2,pdat))\
                *(1.-(1.e-5)*rho1*dhdp(gas,p1,p2,h1,h2,pdat));
        }
    }
    else if(strcmp(process,"EXPANSION")==0){
        w=(1./drdp(process,t1,p1,p2,rho1,rho2,pdat))\
            *(1.-(1.e-5)*rho1*dhdp(gas,p1,p2,h1,h2,pdat));
    }
    return(w);
}
// Function which calculates the pressure derivative of density
double drdp(char *process, double t, double p1, double p2, double
rho1, double rho2, struct data *pdat){

    double w=0.;

    if(strcmp(process,"COMPRESSION")==0){
        w=MBr/(Rgas*t*(*pdat).gamma);
    }
    else if(strcmp(process,"COMBUSTION")==0){
        if(fabs(rho1-rho2)<=1.e-20){    w=1.e-10;}
        else{   w=(rho1-rho2)/( 1.e5*(p1-p2) );}   // Pressure converted in Pa for consistency
    }
    else if(strcmp(process,"EXPANSION")==0){
        w=MBp/(Rgas*t*(*pdat).gamma);
    }
    return(w);
}
// Function which calculates the temperature derivative of density
double drdt(char *process, double t1, double t2, double rho1,
double rho2, struct data *pdat){

    double w, Cvu, Cvb;
    w=Cvu=Cvb=0.;

    if(strcmp(process,"COMPRESSION")==0){
        Cvu=.08*2220. + .92*1000. - (Rgas/MBr);
        w=rho1*Cvu/(Rgas*t1);
    }
    else if(strcmp(process,"COMBUSTION")==0){
        if(fabs(rho1-rho2)<=1.e-20){    w=1.e-10;}
        else{   w=(rho1-rho2)/(t1-t2);}
    }
    else if(strcmp(process,"EXPANSION")==0){
        Cvb=.18*850.+.1*4180.+.72*1042. - (Rgas/MBp);
        w=rho1*Cvb/(Rgas*t1);
    }
    return(w);
}
// Function which calculates the pressure derivative of enthalpy
double dhdp(char *gas, double p1, double p2, double h1, double h2,
struct data *pdat){
    double w=0.;

    if(fabs(h1-h2)<=1.e-20){    w=1.e-10;}
    else{   w=(h1-h2)/(p1-p2);}
    return(w);
}
// Function which calculates the derivative of heat at wall
```

```
double dQwall(int cycl, double angle, double t, struct data
*pdat){

    double q, aw, awm, Twd;
    q=aw=Twd=awm=0.;

    if((cycl == 1)&&(angle<=2.*pi+(*pdat).facomb)){
        Twd=273.;
    }else{
        Twd=(*pdat).Tw;
    }
    awm=2.*pi*((*pdat).B)*(*pdat).L/((*pdat).compratio-1.);
    aw=2.*pi*((*pdat).B)*(.5*(*pdat).L)*(1.-cos(angle)+.5*(*pdat).R*sin(angle)*sin(angle)) + awm;
    if(aw<=awm){aw=awm;}
    q=aw*(.575*(155.)*(1./(*pdat).B)*pow(Re,0.7)*(t-Twd)+4.3e-9*(pow(t,4)-pow(Twd,4)));
    q=1.e+0*q;
    return(q);
}
// Function which calculates enthalpy
double enthalpy(double t, double MB){
    double w=0.;

    w=(5./2.)*Rgas*(1./MB)*t;
    return(w);
}
// Function which calculates the burned mass fraction versus
// crank angle curve and its derivative.Wiebe Function
void Wiebe(double angle, double angleS, double angleF, double *xb,
double *dxb){
    double w=0.;

    w=(angle-angleS)/(angleF-angleS);
    *xb=1.-(1./exp(5.*pow(w,3.)));
    if(fabs(*xb)<=1.e-10){  *xb=1.e-10;}
    *dxb =5.*(1./exp(5.*pow(w,3.)))*(1./pow(angleF-angleS,3.));/
    if(fabs(*dxb)<=1.e-10){ *dxb=1.e-10;}
        if (*xb>1.0){ *xb = 1.0; *dxb = 0.0;      }     //Check that xb can not exceed the value of 1

}
// Function which calculates density
double density(double t, double p, double MB){
    double w=0.;

        if (fabs(t)<=1.e-10){w = 0.0;}
        else{
         w=1.e5*p/(t*(Rgas/MB));
            }
    return(w);
}
// Function which calculates volume by angle
double volume(double angle, struct data *pdat){
    double w=0.;

    w=(*pdat).Vc+.25*pi*(*pdat).B*(*pdat).B*\
    *.5*(*pdat).L*(1.-cos(angle)+.5*(*pdat).R*sin(angle)*sin(angle));
    return(w);
}
// Function which ensures that values do not become zero
void check(double *ch){
    double w,tol1;
```

```
    w=tol1=0.;

    w=*ch;
    tol1=1.e-10;
    if(fabs(*ch)<=tol1){ *ch=tol1;
    }else{  *ch=w;
    }
}
// Function which calculates the percent difference between two values
double error(double er1, double er2){
    double w1, w2;
    w1=w2=0.;

    w1=(fabs(er1) > fabs(er2)) ? fabs(er1) : fabs(er2);

    w2=fabs(er1 - er2)/fabs(w1);
    return(w2);
}
```

9 References

1. G.H. Abd Alla , Computer simulation of a four stroke spark ignition engine, Energy Conversion and Management , 43, pages 1043-1061, 2002.

2. Bayraktar H. , Durgun O. , Mathematical modeling of spark ignition engine cycles, Energy Conversion and Management , pages 651–666, 2003.

3. Bayraktar H. , Durgun O. , Investigating the effects of lpg on spark ignition engine combustion and performance, Energy Conversion and Management , pages 2317–2333, 2004.

4. Cengel Y.A. , Boles M.A. , Thermodynamics:An engineering approach, Fourth Edition, McGraw-Hill Book Co. , 2002.

5. Heywood J.B. , Internal Combustion Engines Fundamentals, McGraw-Hill Book Co. , 1988.

6. *Fröberg* Carl-Erik , Numerical Mathematics Theory and Computer Applications, Benjamin/Cummings Publishing Co. , 1985.

7. The Engineering Toolbox, http:\\ www.engineeringtoolbox.com.