MARMARA UNIVERSITY
FACULTY OF ENGINEERING

Honda CR250F

Dilara KOCAMAN
Neslihan BUYUKAYDIN

GRADUATION PROJECT REPORT
Department of Mechanical Engineering

Supervisor
Assoc. Dr. Mustafa YILMAZ

ISTANBUL, 2022

MARMARA UNIVERSITY

FACULTY OF ENGINEERING

Honda CR250F

by

Dilara KOCAMAN
Neslihan BUYUKAYDIN

June 25, 2022, Istanbul

SUBMITTED TO THE DEPARTMENT OF MECHANICAL ENGINEERING IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE

OF BACHELOR OF SCIENCE

AT

MARMARA UNIVERSITY

Signature of Author(s) ....
Department of Mechanical Engineering

Certified By ....
Project Supervisor, Department of Mechanical Engineering

Accepted By...
Head of the Department of Mechanical Engineering

ACKNOWLEDGEMENT

First of all, we would like to thank our supervisor Prof. Dr. Mustafa Yılmaz for the valuable guidance and advice on preparing this thesis and giving us moral and material support and Marmara University Mechanical Engineering Department.

June, 2022

Neslihan BUYUKAYDIN
Dilara KOCAMAN

CONTENTS

ABSTRACT

The Effects of Altitude, Speed and Mass Flow Rate at Air Intake on the Aerodynamics of Fighter Aircraft Using Open-Source Programs and Operating System

In this thesis, the effect of some specifications on Honda CR250F engine, like RPM, AF Ratio, compression ratio and their behavior on graphs were investigated. While evaluating the behavior of this engine, GUI Interface, MATLAB is used.

Different amount of fuel consumptions was taken from different revolutions. To reach an optimum BSFC curve, analyzes were made on ideal RPM, claimed to be between 3000-9000. To observe these curves, all parameters were created within the arrays and proper equations, internal combustion engine equations were taken mostly from Pulkrabek. After coding equations in the MATLAB, user-controlled input fields are created in the GUI, Design App. Adjusting default values from the specs of Honda CR250F the results were compared to Jeremy's experimental data.

Finally, the main purpose to generate BSFC curves and compare the changes with the determined parameters.

Keywords: Internal Combustion Engine, Honda CR250F, BSFC, MATLAB

SYMBOLS & ABBREVIATIONS

B	-	Cylinder Bore
S	-	Stroke
r	-	Connecting Rod Length
a	-	Crank Offset
s	-	Piston Position
θ	-	Crank Angle
S_c	-	Clearance Height
V_c	-	Clearance Volume
V_d	-	Displacement Volume
V_{BDC}	-	Volume at Bottom Dead Center
V_{TDC}	-	Volume at Top Dead Center
N_c	-	Number of Engine Cylinders
n	-	Number of Revolutions per Cycle
U_p	-	Average Piston Speed
S	-	Stroke Length
N	-	Revolutions per Minute (RPM)
L	-	Litres
W_b	-	Brake Work of One Revolution
$\dot{W}$	-	Engine Power
$\dot{W}_b$	-	Engine Brake Power
$\dot{W}_i$	-	Engine Indicated Power
A_p	-	Piston Face Area of All Pistons
$\dot{m}_f$	-	Rate of Fuel Flow into Engine
η_m	-	Mechanical Efficiency
η_t	-	Thermal Efficiency
η_c	-	Combustion Efficiency
τ	-	Torque
m_a	-	Mass of Air into the Engine or Cylinder
$\dot{m}_a$	-	Mass Flow of Air into the Engine
r_c	-	Compression Ratio
Q_{in}	-	Heat Transfer for One Cycle

$\dot{Q}_{in}$	-	Heat Transfer Rate
ρ_a	-	Air Density Evaluated at Atmospheric Conditions Outside the Engine
Q_{HV}	-	Heating Value of the Fuel
P	-	Gas Pressure in Cylinder
v	-	Spesific Volume of Gas
TDC	-	Top Dead Center
BDC	-	Bottom Dead Center
bsfc	-	Break Specific Fuel Consumption
IVO	-	Intake Valve Opening
IVC	-	Intake Valve Closing
EVO	-	Exhaust Valve Opening
EVC	-	Exhaust Valve Closing
imep	-	Indicated Mean Effective Pressure
bmep	-	Brake Mean Effective Pressure
mep	-	Mean Effective Pressure
sfc	-	Specific Fuel Consumption

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

In this report, it is aimed to observe break specific fuel consumption of Honda CR250F Engine and to observe mechanical behaviors of this engine and its limits via MATLAB GUI. These graphs can be examined through different coding programs and also 1-D Engine Modeling simulation programs like Lotus, GT-Power etc. While making these observations, it is also aimed to offer user-controlled, easy interface to whom operates the program.

1.1. Internal Combustion Engines

The purpose of internal combustion engines is the production of mechanical power from the chemical energy contained in the fuel. In internal combustion engines, as distinct from external combustion engines, this energy is released by burning or oxidizing the fuel inside the engine. The fuel-air mixture before combustion and the burned products after combustion are the actual working fluids. The work transfers which provide the desired power output occur directly between these working fluids and the mechanical components of the engine [1].

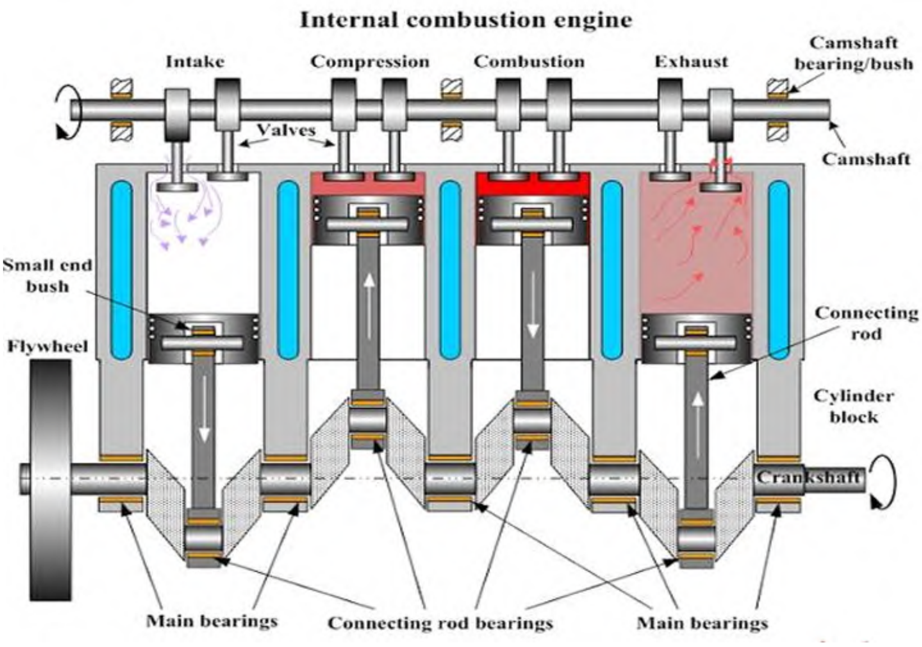


Figure 1 Internal Combustion Engine

https://tr.pinterest.com/pin/507077239273529016/

1.2. Historical Perspective

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 produced by burning the fuel and the work-producing piston- in-cylinder expander. It was not until the 1860s that the internal combustion engine became a practical reality. The early engines developed for commercial use burned coal-gas air mixtures at atmospheric pressure there was no compression before combustion. J. J. E. Lenoir (1822-1900) developed the first marketable engine of this type. 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 stroke. The cycle was completed with an exhaust stroke. Some 5000 of these engines were built between 1860 and 1865 in sizes up to six horsepower. Efficiency was at best about 5 percent. [1]

1.3. Engine Classifications

INTERNAL COMBUSTION ENGINE	**TYPES OF IGNITION**	SPARK IGNITION (SI)
		COMPRESSION IGNITION
	VALVE LOCATION	I HEAD ENGINE
		L HEAD ENGINE
		T HEAD ENGINE
		F HEAD ENGINE
	BASIC DESIGN	RECIPROCATING
		ROTARY
	POSITION AND NUMBER OF CYLINDERS OF RECIPROCATING ENGINES	SINGLE CYLINDER
		IN-LINE
		V ENGINE
		OPPOSED CYLINDER ENGINE
		W ENGINE
		OPPOSED PISTON
		RADIAL
	ENGINE CYCLE	FOUR-STROKE CYCLE
		TWO-STROKE CYCLE

1.3.1. Types of Ignition

Spark Ignition (SI)

Spark Ignition (SI) Engine is a type of engine in which the combustion takes place by the spark generated by the spark plug. It uses petrol as fuel and works on Otto cycle. In the spark ignition engine, the air-fuel mixture is inserted into the cylinder with help of carburetor. The compression of the fuel takes place but it has low compression ratio. The fuel is ignited by the spark generated by the spark plug. SI engine produces less noise and vibration and their starting is very easy. They are light in weight and have less maintenance cost. They are mostly used in light commercial vehicles such as scooters, motorcycles cars, etc.

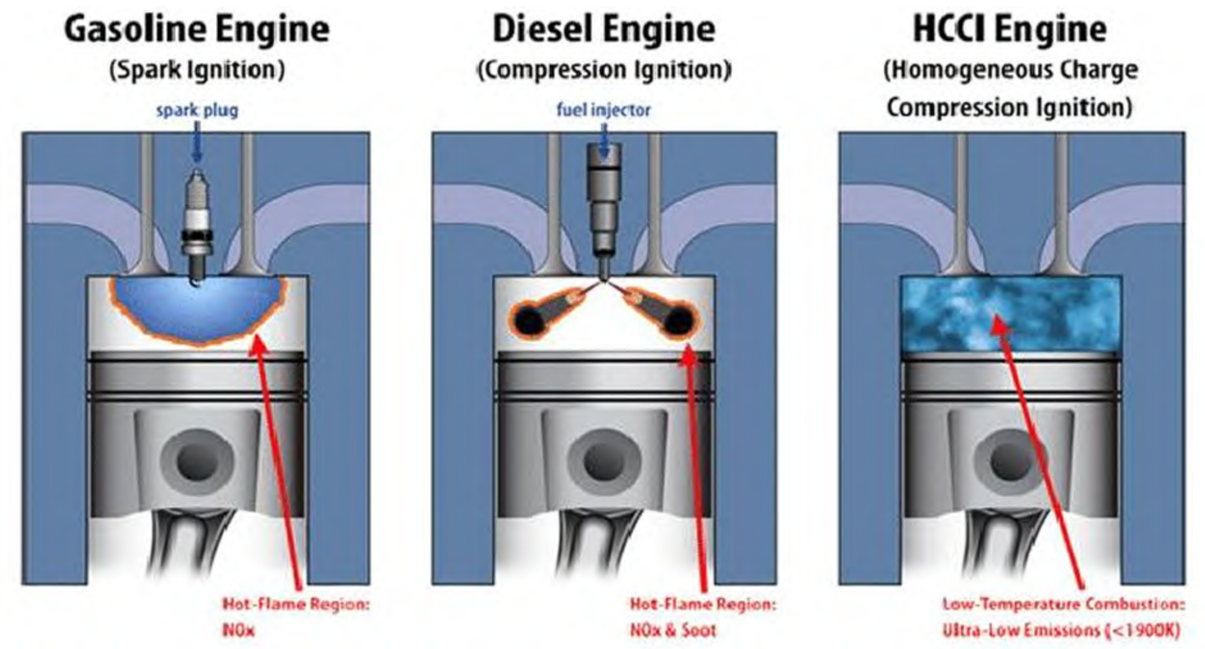


Figure 2 The differences among SI, CI and HCCI engines, reproduced from (William and Charles,2011)

Compression Ignition (CI) (Diesel)

Compression Ignition (CI) Engine is an engine in which the combustion of fuel takes place by the heat of the compressed air. It uses diesel as fuel and works on the Diesel cycle. In the compressed ignition engine, only air enters into the cylinder during suction stroke. It has high compression ratio because of the high ignition temperature of the diesel fuel. The heat of the compressed air ignites the fuel. Due to the high compression ratio it produces more power. Due to incomplete combustion of the fuel, it produces more hydrocarbons which lead to air pollution. The noise and vibration problem is there in the CI engines. The maintenance cost of the CI engine is more as compared with the SI engines. They are mostly used in heavy duty.

Table 1 Comparison of CI and SI Engine

S.no	Parameter	SI Engine	CI Engine
1.	Definition	It is an engine in which the spark is used to burn the fuel.	It is an engine in which heat of compressed air is used to burn the fuel.
2.	Fuel used	Petrol is used as fuel.	Diesel is used as fuel.
3.	Operating cycle	It operates on Otto cycle.	It operates on Diesel cycle.
4.	Compression ratio	Low compression ratio.	High compression ratio.
5.	Thermal efficiency	High thermal efficiency.	Less thermal efficiency.
6.	Method of ignition	Spark plug is used to produce spark for the ignition.	Heat of compressed air is used for the ignition.
7.	Engine Speed	High speed engines.	Low speed engines.
8.	Pressure generated	Low pressure is generated after combustion.	High pressure is generated after combustion.
9.	Constant parameter during cycle	Constant volume cycle.	Constant pressure cycle.
10.	Intake	Air + fuel.	Only air.
11	Weight of engine	Si engine has less weight.	CI engine are heavier.
12.	Noise production	It produces less noise.	It produces more noise.
13.	Production of hydrocarbon	Less Hydrocarbon is produced.	More hydrocarbon is produced.
14.	Maintenance cost	Low	High
15.	Vibration problem	Less	Very High
16.	Cost of engine	Less cost	High cost
17.	Fuel supply	Carburetor	Injector

1.3.2. Valve Location

Valves in head

Valve in head engine is an internal-combustion engine in which both inlet and exhaust valves are located in the cylinder head. (https://www.merriam-webster.com/dictionary/valve-in-head%20engine)

Valves in block

Some historic engines with valves in block had the intake valve on one side of the cylinder and the exhaust valve on the other side. These were called T Head engines. One valve in head (usually intake) and one in block, also called F Head engine; this is much less common.

1.3.3. Position and Number of Cylinders of Reciprocating Engines

Single Cylinder

Engine has one cylinder and piston connected to the crankshaft.

In-Line

Cylinders are positioned in a straight line, one behind the other along the length of the crankshaft.

V Engine

Two banks of cylinders at an angle with each other along a single crankshaft. The angle between the banks of cylinders can be anywhere from 15° to 120°, with 60°-90° being common.

Opposed Cylinder Engine

Two banks of cylinders opposite each other on a single crankshaft (a V engine with a 180°V).

W Engine

Same as a V engine except with three banks of cylinders on the same crankshaft. Usually 12 cylinders with about a *60°* angle between each bank.

Opposed Piston Engine

An opposed-piston engine is a piston engine in which each cylinder has a piston at both ends, and no cylinder head. (https://en.wikipedia.org/wiki/Opposed-piston_engine)

Radial Engine

Engine with pistons positioned in a circular plane around the central crankshaft. The connecting rods of the pistons are connected to a master rod which, in turn, is connected to the crankshaft.

1.3.4. Engine Cycle

There are two classes of internal combustion engines : two stroke and four stroke. Most motorcycles generally use two stroke cycle. 4-stroke engine goes through four stages, or two complete revolutions, to complete one power stroke. A 2-stroke engine goes through 2 stages, or one complete revolution, to complete one power stroke.

Four-Stroke Cycle

A four-stroke cycle experiences four piston movements over two engine revolutions for each cycle. This cycle has 4 main steps. These are intake compression, combustion and exhaust. Each steps have individual time.

Two-Stroke Cycle

A two-stroke cycle has two piston movements over one revolution for each cycle. A two-stroke engine combines the compression and ignition steps on the upstroke and the power and exhaust steps on the downstroke.

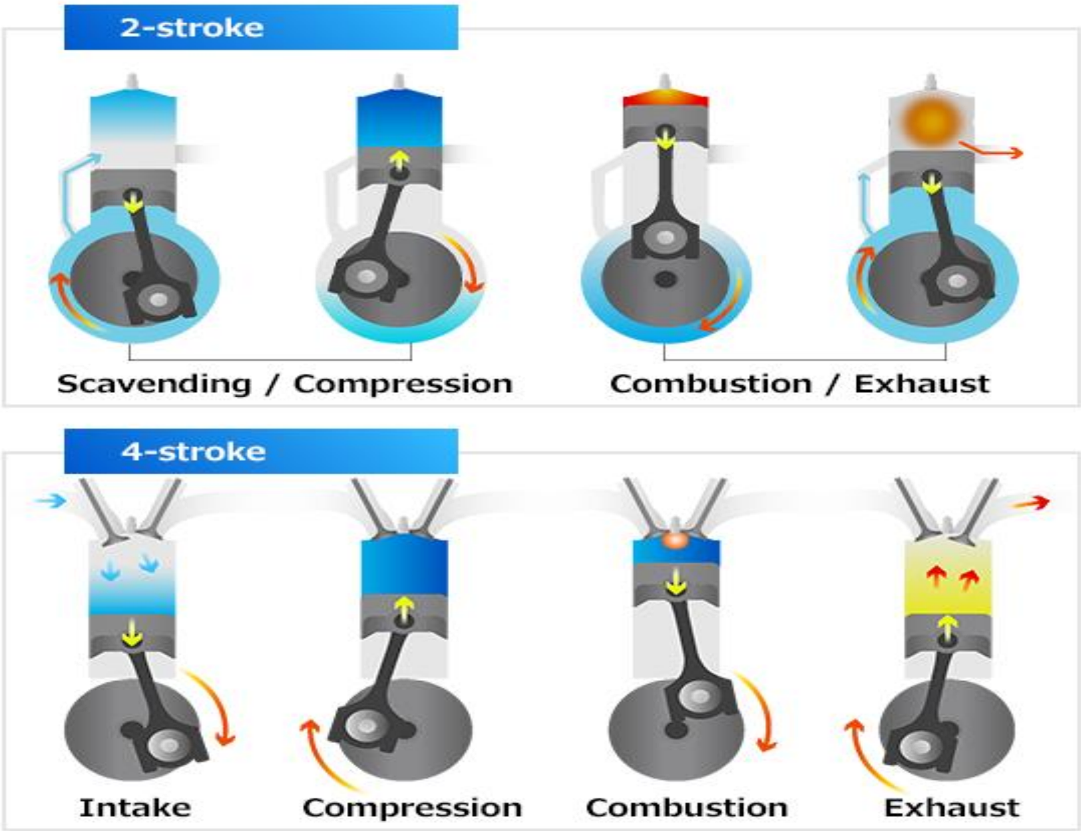


Figure 3 2-Stroke and 4-Stroke Cycles
https://global.yamaha-motor.com/business/mc/mc-tech/standard-technology/2st4st.html

1.4.Ideal Engine Cycles

1.4.1. Otto Cycle

The Otto cycle is the ideal cycle for spark-ignition (SI) engines. This cycle was presented in the late 19th century after Nikolaus Otto demonstrated the four-stroke SI engine successfully. Description of the processes is given as follows:

• Process 1-2: isentropic compression of the working fluid occurs as the piston moves from bottom dead center (BDC) to top dead center (TDC).

• Process 2-3: here, with the SI, a rapid burning occurs inside the piston; therefore, the heat addition takes place at the constant volume.

• Process 3-4: in this process, the working fluid expands isentropically and produces the useful work for the cycle.

• Process 4-1: heat removal at constant volume. In practical applications, the heat is removed by expelling the exhaust gas to the atmosphere.

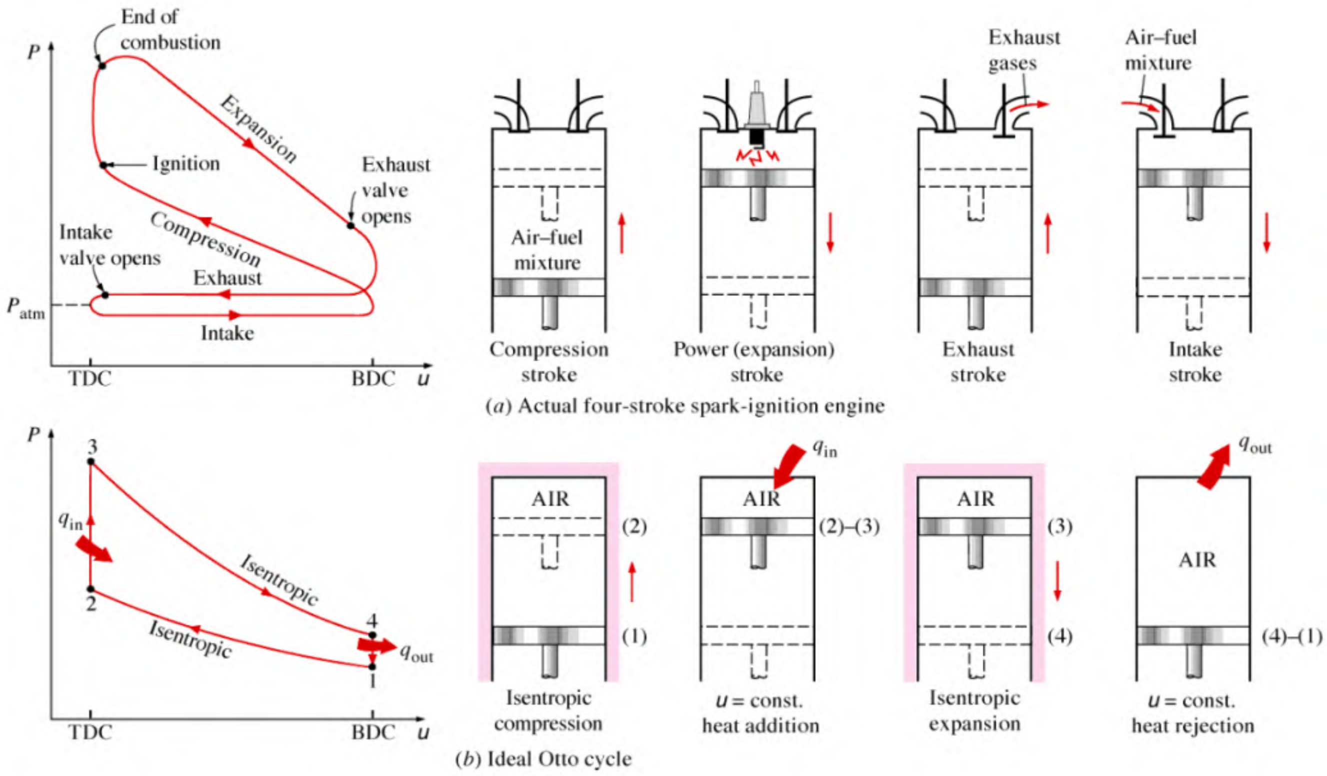


Figure 4 Ideal and Actual P-V diagram for Otto Cycle

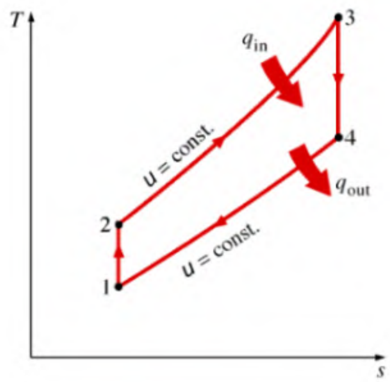


Figure 5 T-S diagram for Otto Cycle

1.4.2. Diesel Cycle

The Diesel cycle is used in numerous large-scale industrial applications for power generation. The Diesel engine is also known as a compression ignition (CI) engine, which was proposed in Germany by Rudolf Diesel in the late 19th century. In the CI engine, the ignition of the combustion process is attained by compression of the working fluid up to a larger temperature than the fuel's autoignition temperature. When the temperature of the air increases adequately, high-pressure liquid fuel is sprayed into the combustion chamber. Thus, the ignition process occurs rapidly. *P-v* and *T-s* diagrams of the ideal Diesel cycle are presented in Fig. xxxx. A description of the Diesel cycle processes is given as follows:

•Process 1-2: isentropic compression of the working fluid occurs as the piston moves from BDC to TDC. This process requires an external work input.

•Process 2-3: here, with the rising temperature of the compressed gas and fuel injection, ignition occurs instantly, which results in growth in the temperature. Since the piston moves during the process, the heat addition takes place at constant pressure.

•Process 3-4: in this process, the working fluid expands isentropically and produces the useful work for the cycle.

•Process 4-1: heat removal at constant volume by expelling the exhaust gas to the atmosphere.

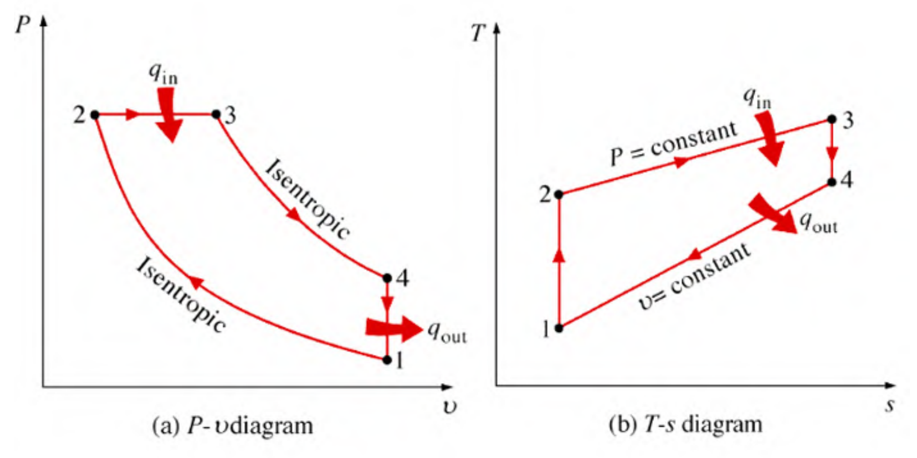


Figure 6 P-V and T-S Diagram for Ideal Diesel Cycle

1.4.3. Dual Cycle

Dual cycle will provide us better approximation to a real engine. As we can see from the name of this cycle, we can say that we will secure Dual cycle by considering the concept of Otto cycle and Diesel cycle together. Heat addition process in case of Dual cycle will be combination of heat addition process of Otto cycle and Diesel cycle and therefore this cycle is termed as mixed cycle or Dual cycle.

Heat addition in Dual cycle will be done in two parts i.e. heat energy will be supplied partially at constant volume and partially at constant pressure and hence there will be more time for fuel for combustion which will be injected in to the engine cylinder before end of the compression stroke.

•Process 1-2: Adiabatic compression of the working fluid
•Process 2-3: Heat energy addition to the working fluid at constant volume
•Process 3-4: Heat energy addition to the working fluid at constant pressure
•Process 4-5: Adiabatic expansion of the working fluid or also termed as power stroke
•Process 5-1: Rejection of heat energy at constant volume[8]

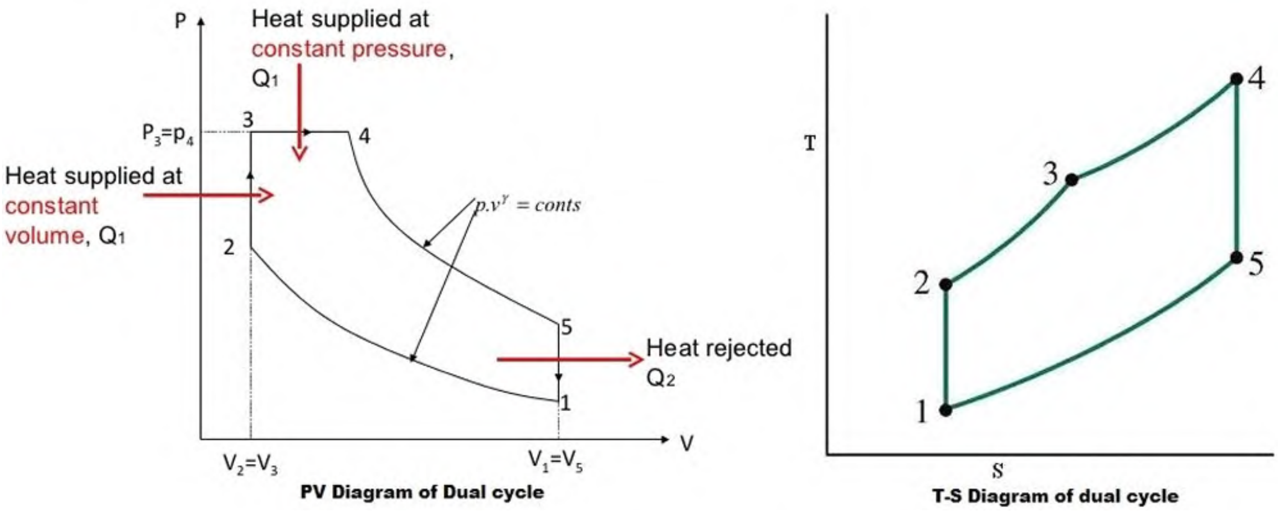


Figure 7 P-V and T-S diagram for Dual Cycle

1.5.GASOLINE ENGINE

Most modern gasoline engine use the conventional cylinder and piston arrangement operated with a slider crank mechanism common to other internal combustion engines such as diesel engine. There is very little difference between the basic structure of diesel and gasoline engines.

Conceptually, diesel engines operate by compressing air to high pressure and then injecting a small amount of fuel into the cylinder that is full with hot compressed air. This high temperature causes the small amount of highly atomized injected fuel to evaporate. This mixture in the combustion chamber release the energy that is stored in this fuel.

On the other hand, Air and fuel is mixed before the compression. The pre-mixing is formerly done in a carburetor. So, gasoline engines takes air-fuel mixture at the same time in the combustion chamber. Than compressed air-fuel mixture burns by using spark plug.

1.5.1. Major Components of a Single Cylinder Gasoline Engine

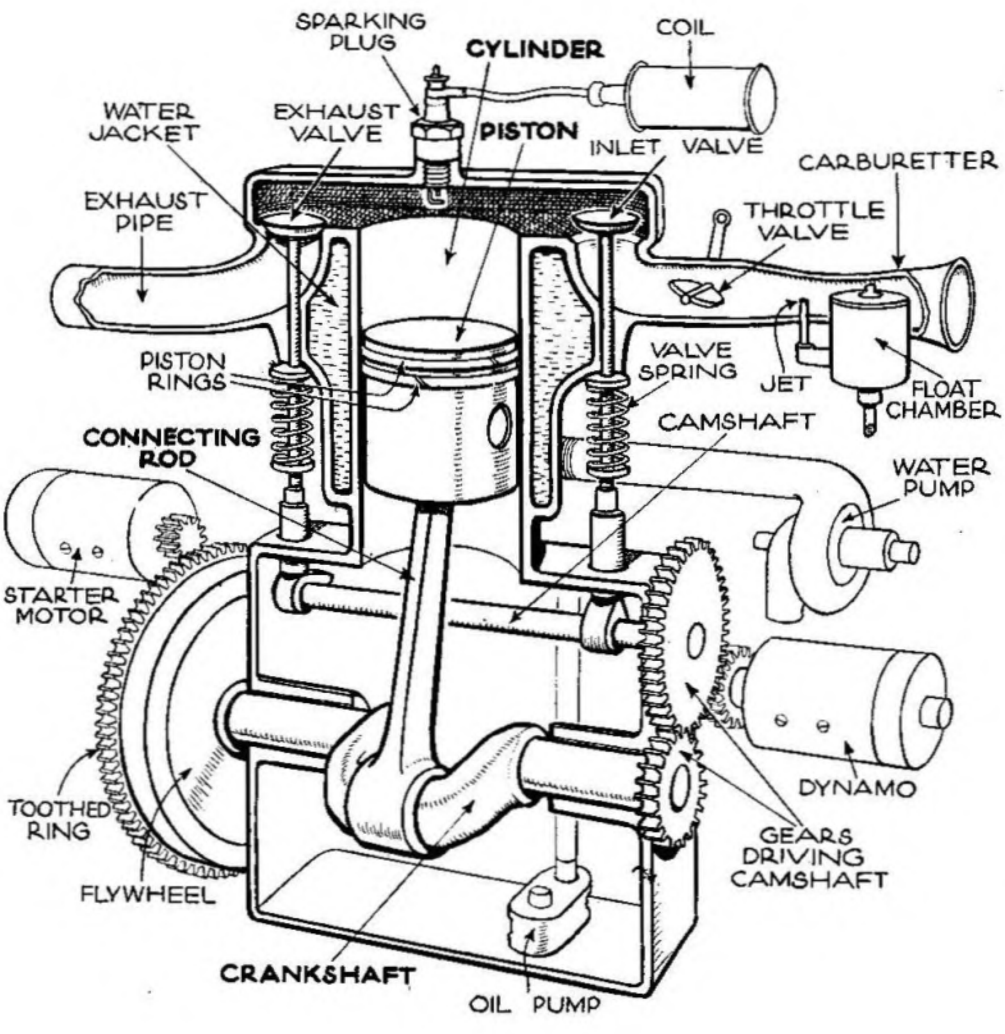


Figure 8 Parts of One Cylinder

Cylinder Block : The main structural member of all automotive engines is a cylinder block that usually extends upward from the centre line of the main support for the crankshaft to the junction with the cylinder head.

Combustion Chamber : The combustion chamber is defined by the size, location, and position of the piston within the cylinder. Bore is the inner diameter of the cylinder. The volume at bottom dead centre (VBDC) is defined as the volume occupied between the cylinder head and the piston face when the piston is farthest from the cylinder head. The volume at top dead centre (VTDC) is the volume occupied when the piston is closest to the cylinder head; the distance between the piston face and cylinder head at VTDC is called the clearance. The distance traveled by the piston between its VTDC and VBDC locations is the stroke.

Piston : The pistons are cup-shaped cylindrical castings of steel or aluminum alloy. The upper, closed end, called the crown, forms the lower surface of the combustion chamber and receives the force applied by the combustion gases.

Connecting Rod : A forged-steel connecting rod connects the piston to a throw (offset portion) of the crankshaft and converts the reciprocating motion of the piston to the rotating motion of the cranks. **https://www.britannica.com/technology/gasoline-engine/Cylinder-block**

Connecting Rod Bearing : Bearing where connecting rod fastens to crankshaft.

Crankshaft : A crankshaft is a shaft driven by a crank mechanism consisting of a series of cranks and crankpins to which the connecting rods of an engine is attached. It is a mechanical part able to perform a conversion between reciprocating motion and rotational motion. https://en.wikipedia.org/wiki/Crankshaft#:~:text=A%20crankshaft%20is%20a%20shaft,reciprocating%20motion%20and%20rotational%20motion.

Valves : Valves for controlling intake and exhaust may be located overhead, on one side, on one side and overhead, or on opposite sides of the cylinder.

Cam Shaft : Varying from vehicle to vehicle, the camshaft may either be located within the engine block or in the cylinder heads. Many modern vehicles have them in the cylinder heads, also known as Dual Overhead Camshaft (DOHC) or Single Overhead Camshaft (SOHC), and supported by a sequence of bearings that are lubricated in oil for longevity. The role of the camshaft is to regulate the timing of the opening and closing of valves and take the rotary motion from the crankshaft and transfer it to an up and down motion to control the movement of the lifters, moving the pushrods, rockers, and valves.

Crankcase : This component houses the crankshaft and is found under the modern engine block.

Cylinder: Cylinders are the parts in which the movement of the piston takes place. They are generally made of large size and have holes to form a seal with the piston. The number of cylinders holds the power and size of the engine. The cylinder is the space through which the piston travels, propelled to the energy generated from the combustion of the air/fuel mixture in the combustion chamber

Oil Filter : It is typically located either on the flank or under the engine block. There is an oil filter that keeps as many contaminants out of the lubricant that circulates the engine.

Flywheel : The cycle of the internal-combustion engine is such that torque (turning force) is applied only intermittently as each cylinder fires. Between these power impulses, the pistons rising on compression and the opposition to rotation caused by the load carried by the engine apply negative torque. The alternating acceleration caused by the power impulse and deceleration caused by compression result in nonuniform rotation. To counter this tendency to slow down and speed up is the function of the flywheel, attached to one end of the crankshaft. The flywheel consists of a heavy circular cast-iron disk with a hub for attachment to the engine.

Fuel Injector : The fuel injector injects/sprays fuel into the intake manifold at a very precise angle. Within the intake manifold, air and fuel mix. The air-fuel mixture is compressed in the combustion chamber, igniting the chemical reaction required to power your engine.

Exhaust Manifold : Exhaust manifold collects the exhaust gases from multiple cylinders into one pipe, usually made of cast iron.

Piston Ring : A piston ring is a metallic split ring that is attached to the outer diameter of a piston in an internal combustion engine or steam engine. The main functions of piston rings in engines are: Sealing the combustion chamber so that there is minimal loss of gases to the crank case, Improving heat transfer from the piston to the cylinder Wall, maintaining the proper quantity of the oil between the piston and the cylinder Wall and regulating engine oil consumption by scraping oil from the cylinder walls back to the sump.

Throttle Valve : In a conventional gasoline engine, a throttle valve is used in the intake manifold to reduce the power output for part-load operation. Varying the throttle position (TP) controls the amount of mixture (air + fuel) inside the engine, while the AFR is kept constant.

1.5.2. Principle Operation of Four-Stroke Engine

An internal-combustion engine goes through four strokes: intake, compression, combustion (power), and exhaust. As the piston moves during each stroke, it turns the crankshaft.[4]

Intake stroke

- Piston moves down the cylinder bore from top dead center (TDC) to bottom dead center (BDC).
- Intake valve is open, the exhaust valve is closed.
- Downward piston motion creates a vacuum (negative air pressure) that draws that air/fuel mixture into the engine via the open intake valve.

Compression stroke

- Piston moves up the cylinder bore from bottom dead center to top dead center.
- Both the intake and exhaust valves are closed.
- Upward piston motion compresses air/fuel mixture in the combustion chamber.

Power stroke

- At the end of the compression (previous) stroke, the spark plug fires and ignites the compressed air/fuel mixture. This ignition/explosion forces the piston back down the cylinder bore and rotates the crankshaft, propelling the vehicle forward.
- Piston moves down the cylinder bore from top dead center to bottom dead center.
- Both the intake and exhaust valve are closed.

Exhaust stroke

- Piston moves up the cylinder bore from bottom dead center to top dead center. The momentum caused by the power stroke is what continues the crankshaft movement and the other 3 strokes consecutively.
- Intake valve is closed, the exhaust valve is open
- This final stroke forces the spent gasses/exhaust out of the cylinder. The cycle in now complete and the piston is ready to begin the intake stroke.

1.6.MODELING

Modeling is indispensable in engineering. Modeling studies developed in parallel with computer technology have shown how profitable a real size prototype can be when

production would be impossible or very expensive. We can list the advantages of using models in experimental designs as follows:
If the analytical solution of the problem is too complex, experiential solution methods can be developed. By comparing the real behavior of the model with analytical techniques, its possible behavior can be proven. If the prototype is too large, it is under environmental factors that cannot be imitated, it has a very small structure, it is under dangerous working conditions, etc., studies can be carried out on it.
Modeling that created by using finite element method. is important factor for the design process. Modeling that is created correctly could decrease the cost and provide safety design. Detailed low-dimensional models of system behavior can provide valuable insights into system performance and function thus guiding the design process.
1D modeling : The aim to increase system efficiency, by helping engineers understand the interaction of the different components within the system. The process of building a model for a system consists of a few steps:

1. Input the correct components and link them together;
2. Assign the correct physical models to the different components (e.g. simplified turbomachines/heat-exchangers, pipelines, chambers, resistance etc.);
3. Enter the correct parameters for the physical models; and
4. Run the simulation.

Each component is simulated separately with its own input and parameters. This means that the model gives an accurate display of the performance of the system, because it takes the interactions of the different components with each other into account as well.

3D Modeling : Themain difference here is that 3D simulations show the interaction of individual components with their surroundings, whereas 1D simulations show the entire design of a system and the interactions of the different components of this larger system. This makes a 1D simulation excellent for optimizing the design of an entire system.
System-level simulation is a tool that can be used across the entire development cycle, from initial design to testing or optimization. This means that information can be transferred across one platform and the risk of miscommunication is reduced. Simulation allows you to thoroughly prototype your product digitally, reducing the extent and costs of physical testing without compromising quality. These are just some of the benefits that make 1D simulation.

1D engine modeling can be done alone or can be an input provider to other software similar as CFD to perform more complex calculations. So, 1D system simulation or 1D simulation in short is very effective method to get basic inputs required for the system design at the concept stage of product development. It also helps to simulate different operating parameters quickly and deliver inputs required for 3D simulations.

There are lots of 1D and 3D simulation program. Also, Lotus is the one of this type of program. Lotus Engine Simulation Program help to create 1D modeling for internal combustion engine. In this thesis Lotus Engine Simulation and Matlab is used.

2. METHODOLOGY AND MATERIALS

2.1. Introduction to Lotus Engine Simulation Software

Lotus Engine Simulation software is a very helpful American Software company found in Massachusetts, USA which offers simulation tools that enable the user to create models. This program is capable of predicting the complete performance of an engine system. This program can calculate the full and part load performance of the engine under steady-state and transient operating conditions, heat transfer data, turbocharger and supercharger matching conditions and instantaneous gas property variations within the engine manifolds.

Figure 9 Lotus Engine Simulation Programme Logo

2.2. Selected Engine Model & Specifications

Figure 10 Honda CRF250F Motorcycle

Table 2 Specifications of CRF250F

Manufacturer	**Honda**
Displacement (cc)	249 cc, air cooled
Number of cylinders	4
Valvetrain Layout	SOHC
Number of Valves	4
Bore(mm)	71 mm
Stroke(mm)	63 mm

Compression Ratio	9.6:1
Fuel Type	Gasoline
Stroke	Four Stroke

Engine specifications from the original brochure can be seen in appendix.

2.3. Engine Technology behind Honda CRF Series

The Honda CRF Series is a inline of four stroke motocross, trail and dual sport motorcycles manufactured and marketed. by Honda.

The engines in these bikes use an over-square design, which means that the diameter of the cylinder is larger than the stroke of the piston. This allows for higher engine speeds and a reduction of reciprocating mass for a given displacement. Another technology that is used is short piston skirts. The "skirt" area of the piston is the portion on the side of the piston which encounters the cylinder wall and aids piston stability. While the introduction of the shorter skirt on the piston helps to reduce reciprocating mass, it also leads to more "rocking" of the piston, or minute unwanted rotation of the piston around the axis of its wrist pin. This leads to more frequent maintenance intervals for the pistons, piston rings, and cylinder walls.[4]

2.4. Engine Modeling on Lotus

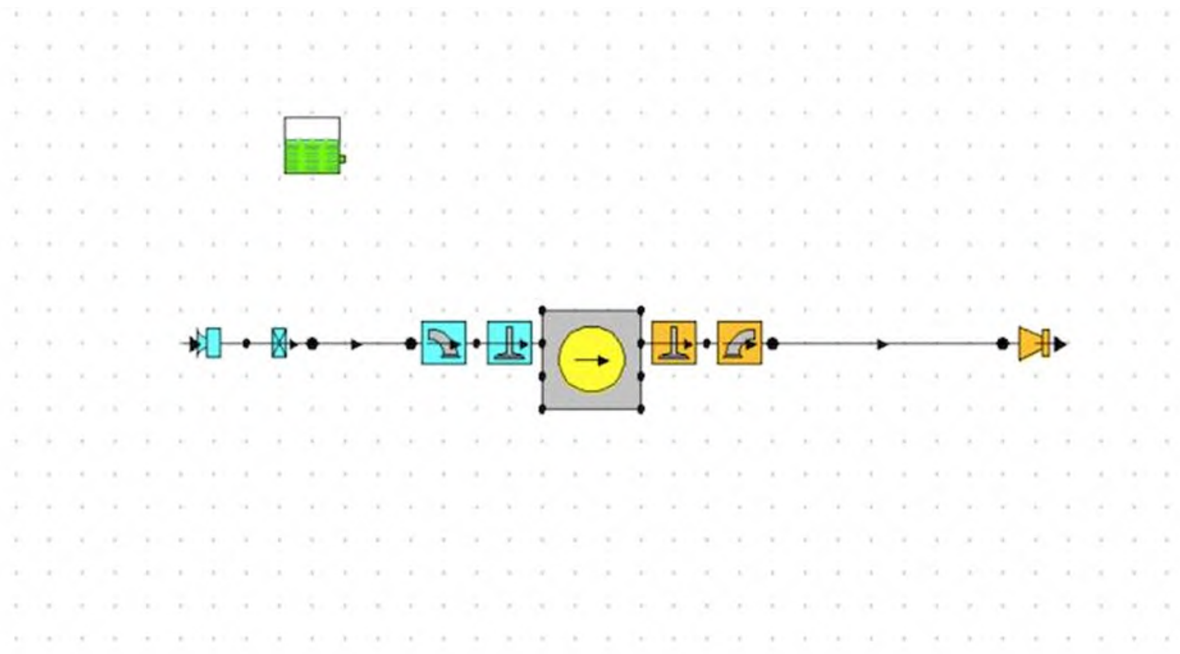


Figure 11 Lotus Schematic for Honda CRF250F

Label	HONDA CR250F
Bore (mm)	71,0000
Stroke (mm)	63,0000
Cyl Swept Volume (l)	0,24943
Total Swept Volume (l)	0,24943
Con-rod Length (mm)	90,00
Pin Off-Set (mm)	0,00
Compression Ratio	9,60
Clearance Volume (l)	0,029003
Phase (ATDC)	0,00

Figure 12 Values CRF250F from Lotus

Figure (xxx) above shows the 1-D Model of Honda CR250F. Even though in Lotus, all single cylinder simulations have same appearance. Specifications of the engine has entered in cylinder part in Lotus. Figure(xxx2) shows the Honda engine's specs.

When modeling 1-D model, fuel type gasoline is selected. For the load ,throttle valve is added firstly then intake manifold and intake valve is placed just before the cylinder. After cylinder data's is given model continued by exhaust valve and exhaust manifold.

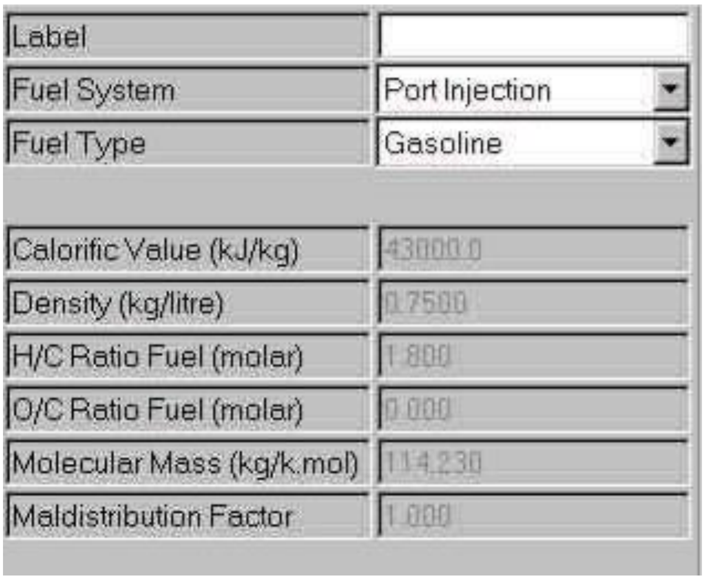


Figure 13 Values CRF250F from Lotus -2

The fuel type is Port Injection in this project. Fuel properties are defined by clicking on the fuel element shown in figure(xx) above.

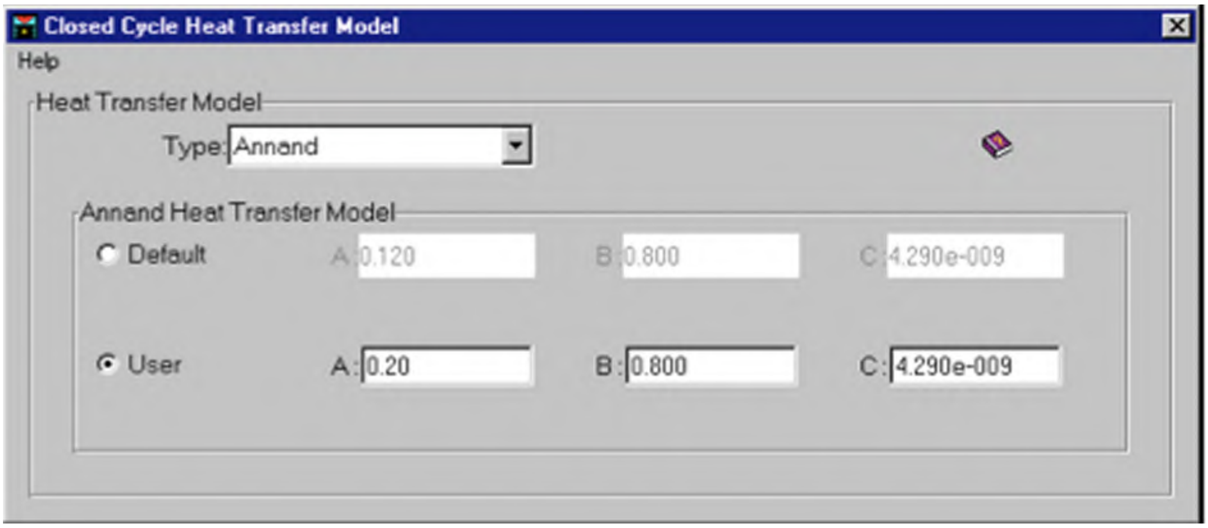


Figure 14 Model Type Set from Lotus

2.5. What is MatLab GUI ?

Graphical Users (GUIs), also known as apps, provide point-and-click control of software applications, eliminating the need for others to learn language or type commands in order to run the application. In general, it converts scripts into a simple app.
In this graduation project, both of Matlab Script and GUI is used. This simple app can be seen in figure (xxxx) aimed to take inputs from user and evaluate some mathematical models and calculations also as a results computes several graphs about characteristics of engine. Also this app allows user to see outputs directly on the interface.

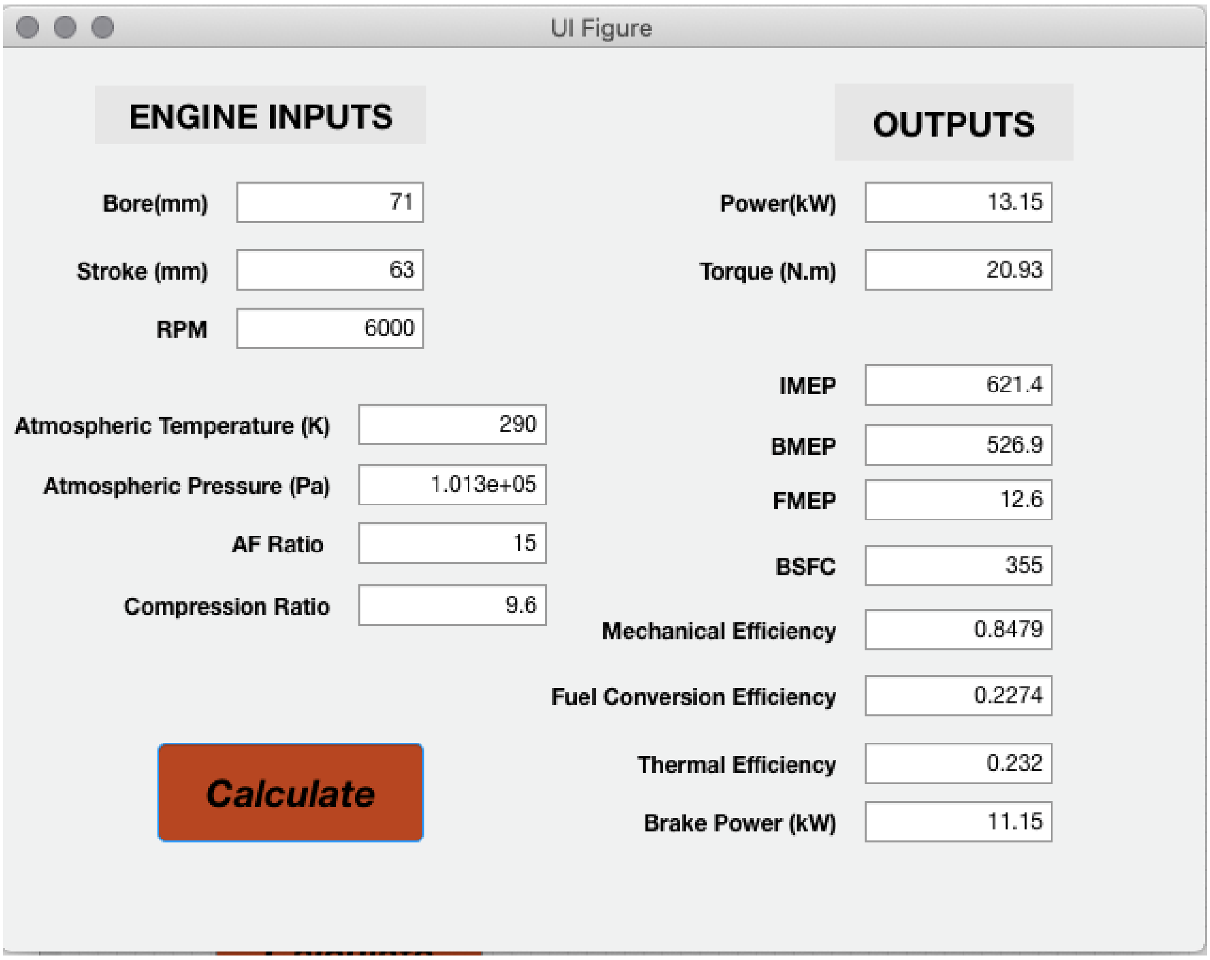


Figure 15 GUI Interface from MatLab

As seen in figure(xxx) above user interface is quite simple. Left side is engine inputs. Which we are expected to user to enter bore, stroke, RPM etc. Right side shows the results after some mathematical calculations. Formulas are taken usually from Engineering Fundamentals of the Internal Combustion Engine, Willard W. Pulkrabek. Formulas and will be mentioned in next sections.

```
value = app.CalculateButton.Value;
            %Engine Inputs
Load =1  %Engine Load (Affects Inlet Pressure)
RPM = app.RPMEditField.Value; %Revolutions Per Minute [1/min]
L = (app.StrokemmEditField.Value)/1000; %Stroke of Engine [m]
B = (app.BoremmEditField.Value)/1000; %Bore of Engine [m]
l = .0935; %Length of Engine Connecting Rod [m]
N_cyl = 1; %Number of Cylinders [unitless]
C_r = app.CompressionRatioEditField.Value; %Compression Ratio [unitless]
N_r = 2; %Number of Revolutions Per Power Stroke
theta_b = 85; %Combustion Burn Duration [degrees]
theta_0 = 145; %Crank Angle At Start of Combustion [degrees]
theta_f = theta_0+theta_b; %Final Comb. Angle [degrees]
IVC = 0; %Time [degrees] when Intake Valve Closes
EVO = 314; %Time [degrees] when Exhaust Valve Opens
U_p = 2*L*RPM/60;
```

Figure 16 Default Input from MatLab

Method of this app is dependent on the user input. To be sure of that such as bore stroke dimensions and so on are taken from the app directly. This figure(xx) indicates that. Red arrows shows the parameters that directly taken from the interface.

. Heat transfer model is chosen as Annand as in Lotus Engineering Software. Friction Model is taken from Jeremy's thesis.[6]

```
%_________________________
%Incorporating The Annand Method To Predict Heat Transfer
%Calculating Reynolds Number
Re(i)=rho(i)*S_bar_p*B/mu(i);
%Calculating Nusselt Number (constant=.26 two stroke, .49 4 stroke)
Nus(i)=.49*Re(i)^(.7);
%Calculating Heat Transfer Coefficient Using Annand Method
h_g(i)=C_k(i)*Nus(i)/B;
```

Figure 17 Annand Method Calculations from MatLab

This figure(xxxx) shows the heat transfer calculation with the Annand Method. As a Nusselt number 0.49 is used due to Honda is a 4 stroke engine.

```
%FOR FRICTION MODEL
%fmep = (0.018)*RPM+59.993;
fmep= 250*L*RPM*10^-3;
```

Figure 18 FMEP Calculation from MatLab

As a friction model, as mentioned above Jeremy's thesis is used but above equation is derived manually. For our FMEP equation, we created a excel plot which x-axis is RPM and y-axis is FMEP Values. This FMEP values are taken from Lotus analysis for each RPM value. With the use of this equation, we obtained almost the same result with Jeremy's equation. To compatibility to other engines than ours we chose to use Jeremy's equation.

2.6. Equations

Equations that are used in between MATLAB codes are taken and obtained from below formulas.

```
A_p = (pi/4)*B^2; %Cross Sectional Piston Area [m^2]
A_ch = 2*A_p; %Cylinder Head Surface Area (in chamber)
V_d = N_cyl*A_p*L; %Displaced Volume Of Engine [m^3]
N = RPM/60; %Converts RPM to RPS [1/s]
S_bar_p = 2*L*N; %Calculates Mean Piston Speed [m/s]
a = L/2; %Calculates Crank Radius (1/2 stroke)[m]
V_TDC = (V_d/(C_r-1))/N_cyl; %Calculates Clearance Volume [m^3]
V_BDC = (V_d/N_cyl)+V_TDC; %Cyl. Volume At BDC [m^3]
```

Figure 19 Input Codes from MatLab

This part calculates the engine inputs for the further formulas. This characteristics will be used to obtain graphs later.

```
%Specified Outputs (On Matlab Screen)
W_dot_indicated=W_dot(360);
bmep=T_indicated(i)*(4*pi)/V_d
bmep = imep-fmep;
imep=bmep+fmep
W_dot_ac =(bmep*V_d*1000*N/(N_r*1000))
T_ac = W_dot(i)/(2*pi*N*10^(-3))
```

Figure 20 Calculations from MatLab

This part of code simply calculates work, power, imep and bmep. To calculate this arrays were created to the proper crank angle. Whole code can be seen in appendix.

```
% eta_m = bmep/imep %Calculates Mechanical Efficiency
eta_m=W_dot_ac/W_dot(i)
Qin = M_F*LHV*1;
thermalefficiency= W(i)/(m_f*LHV*0.98);
```

Figure 21 Efficiency Calculation from MatLab

Average piston speed:

$$U_p = 2 \times S \times N$$

Displacement Volume:

$$V_d = V_{BDC} - V_{TDC}$$

$$V_d = \frac{\pi \times B^2 \times S}{4}$$

Clearance Volume:

$$V_C = V_{TDC}$$

Compression ratio:

$$r_c = \frac{V_{BDC}}{V_{TDC}}$$

Cross-sectional area of cylinder:

$$A_P = \frac{\pi \times B^2}{4}$$

Mechanical efficiency:

$$\eta_m = \frac{W_B}{W_i}$$

Brake Work:

$$W_B = 2 \times \pi \times \tau$$

$$W_B = \frac{bmep \times V_d}{n}$$

BMEP:

$$bmep = \eta_m \times imep$$

$$bmep = \frac{2 \times \pi \times \tau}{V_d}$$

$$bmep = imep - fmep$$

Power:

$$\dot{W} = \frac{W \times N}{n}$$

$$\dot{W} = 2 \times \pi \times N \times \tau$$

Brake Power:

$$\dot{W}_b = \eta_m \times \dot{W}_i$$

$$\dot{W}_b = \dot{W}_i - \dot{W}_f$$

Brake Specific Fuel Consumption (BSFC):

$$bsfc = \frac{\dot{m}_f}{\dot{W}_b}$$

Heat Transfer and Heat Transfer Rate:

$$Q_{in} = m_f \times Q_{HV} \times \eta_c$$

$$\dot{Q}_{in} = \dot{m}_f \times Q_{HV} \times \eta_c$$

Thermal Efficiency:

$$\eta_t = \frac{W}{Q_{in}} = \frac{\dot{W}}{\dot{Q}_{in}} = \frac{\eta_f}{\eta_c}$$

Fuel Conversion Efficiency:

$$\eta_f = \frac{W}{m_f \times Q_{HV}}$$

3. RESULTS AND DISCUSSION

In this section collected datas and obtained graphs will be visualized. There are two different outputs.

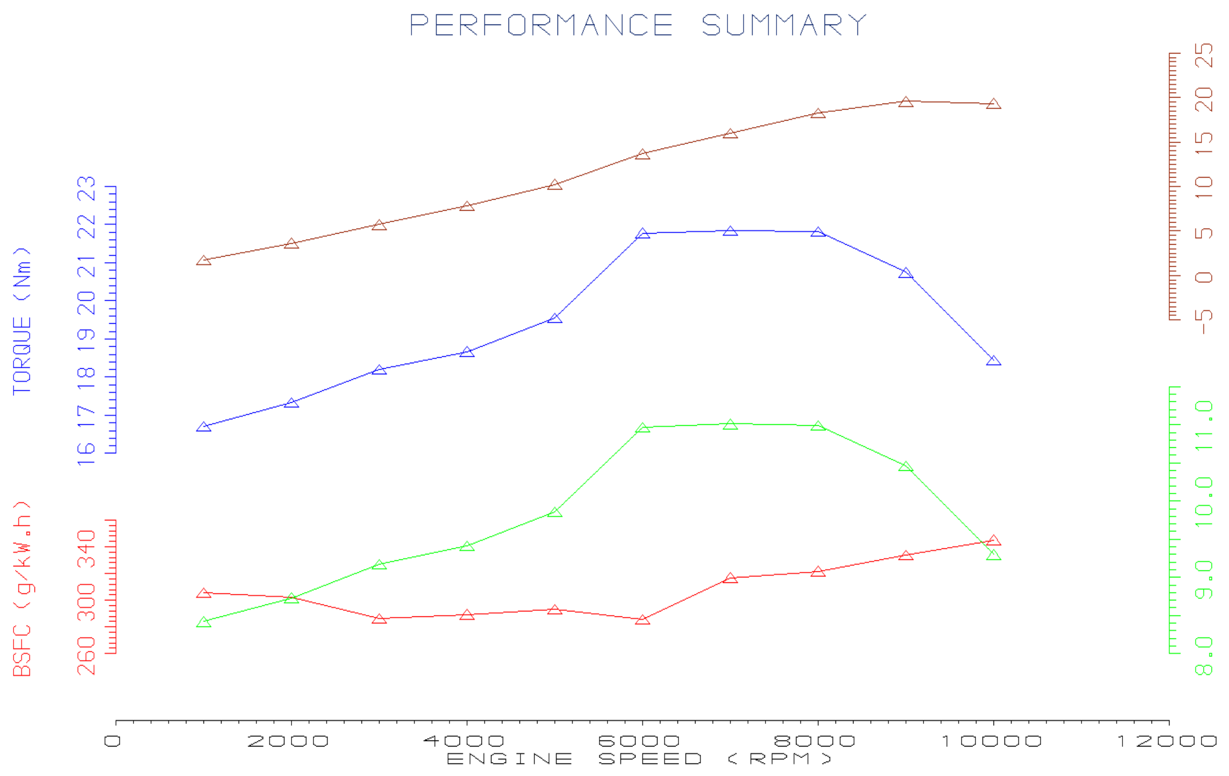
3.1. Lotus Outputs

```
================================================================================
                                 INPUT DATA
                                 ~~~~~~~~~~
        Simple single cylinder engine
        model with no pipes

   1 Cylinder /Port Injection /4 STROKE / Gasoline Engine
   Cylinder . . . . . . .        1       Phase Angle. . . . . .     0.00deg
   Bore . . . . . . . . .    71.00 mm    Stroke . . . . . . . .    63.00 mm
   Con-Rod Length . . . .    93.50 mm    Piston Pin Offset  . .     0.00 mm
   Compression Ratio  . .     9.60

OPERATING CONDITION
~~~~~~~~~~~~~~~~~~~
  "Load Case 1 - ( Created by Test Wizard )"
  Engine Speed . . . . .  1000.0  rpm
  50% Combustion Angle .    10.0 ATDC                     Default Timing
  Burn Angle (10%-90%) .    17.2 deg.                     Default Period
  Overall Equivelent Ratio  .    1.10
  Ambient Pressure . . .   1.000 bar    Ambient Temperature  .    20.0  C
  Relative Humidity  . .   0.875        Specific Humidity  . . 0.0130 kg/kg
  Inlet    1 Pressure . .   1.000 bar    Temperature. . . . . .    20.0  C
  Exit     1 Pressure . .   1.000 bar
  Mechanical Friction  .                                 Default Gasoline
```

Performance Summary

Eng.Speed [rev/min]	B.Power [kW]	B.Torque [Nm]	BMEP [bar]	BSFC [g/kW/hr]	V.Eff. [%]	Converged/Cycles
1000.0	1.95	18.60	9.37	272.84	79.5	Yes (5)
2000.0	4.07	19.44	9.79	268.63	81.8	Yes (6)
3000.0	6.22	19.79	9.97	269.25	83.5	Yes (6)
4000.0	8.48	20.25	10.20	271.45	86.2	Yes (6)
5000.0	10.91	20.84	10.50	274.48	89.7	Yes (6)
6000.0	14.95	23.79	11.98	274.90	102.6	Yes (6)
7000.0	18.35	25.03	12.61	278.58	109.3	Yes (6)
8000.0	20.80	24.83	12.51	283.33	110.3	Yes (6)
9000.0	22.69	24.07	12.13	292.21	110.3	Yes (6)
10000.0	22.18	21.18	10.67	302.93	100.6	Yes (6)

Figure 22 Lotus Summary



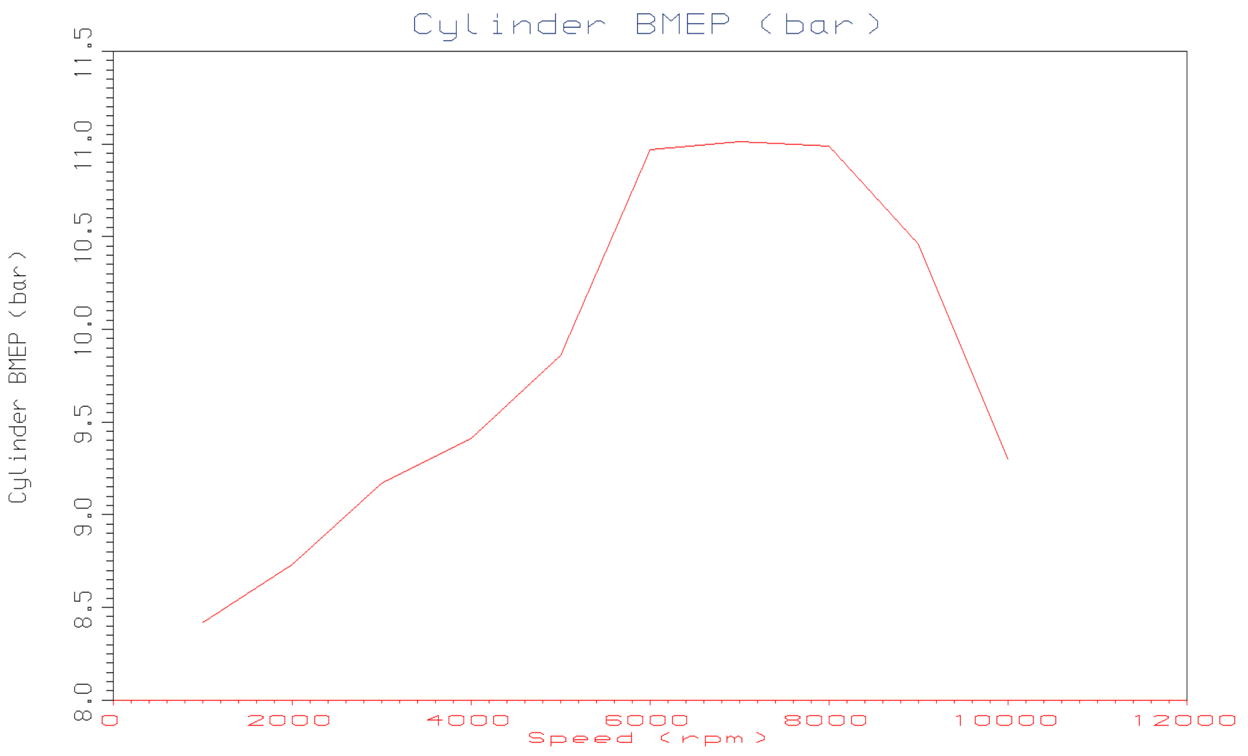
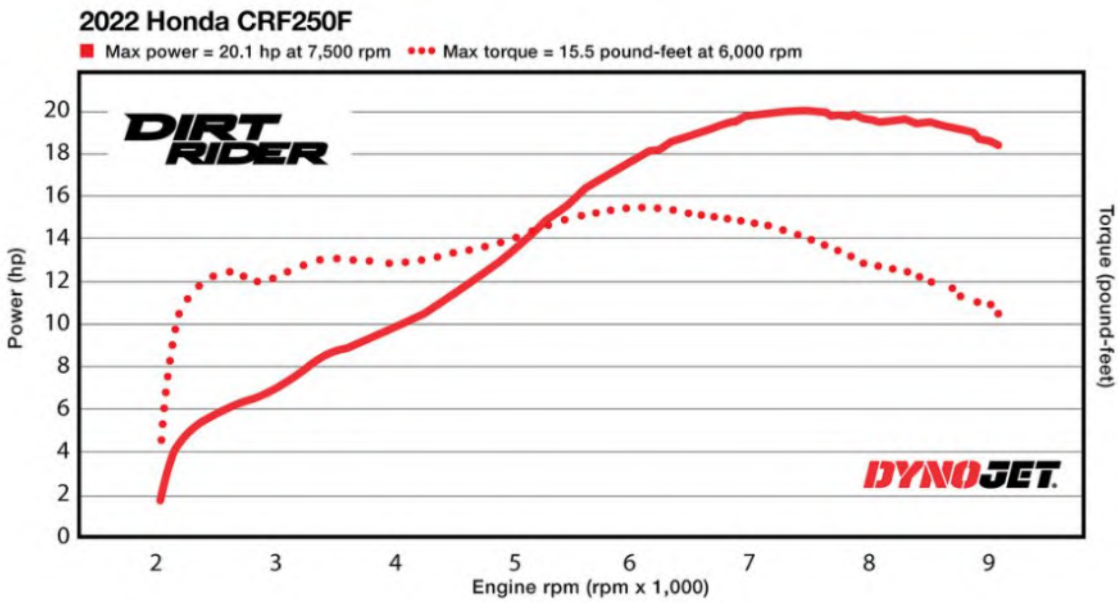


Figure 23 Performance Summary and BMEP graph from Lotus

3.1.1. Real Data



3.1.2. Comparison of Power-Torque-RPM graph between Lotus MATLAB and DynoJet

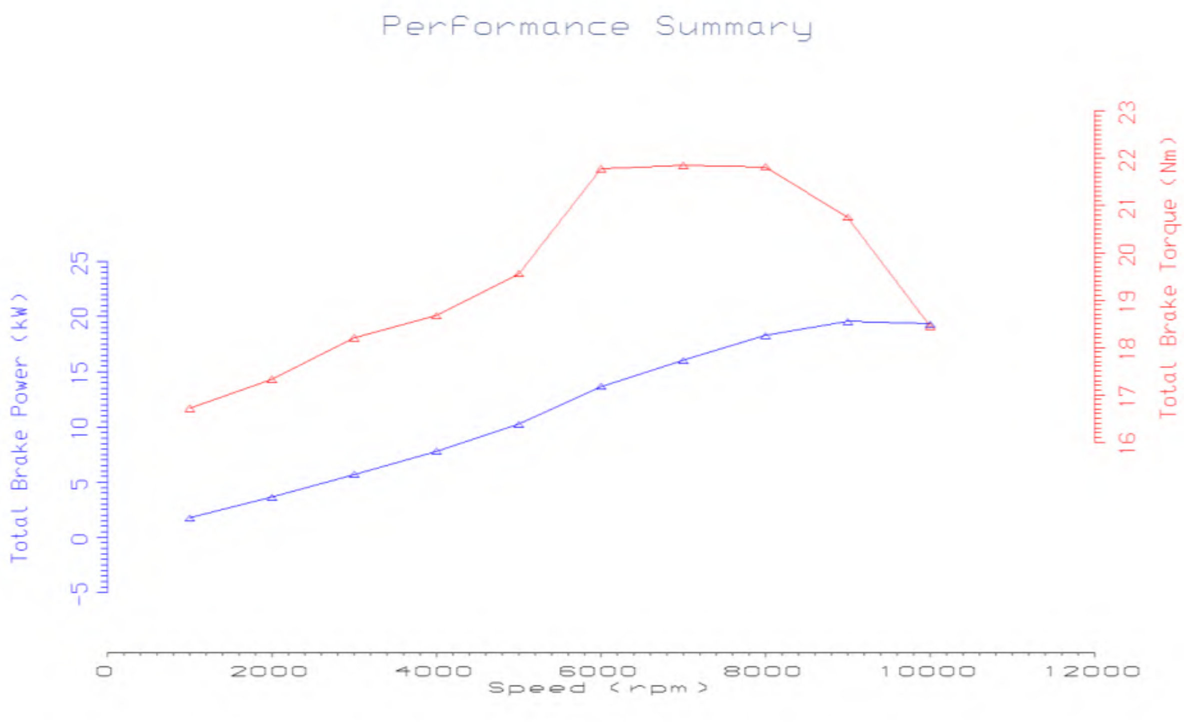


Figure 24 Brake Power - Brake Torque -RPM graph from Lotus

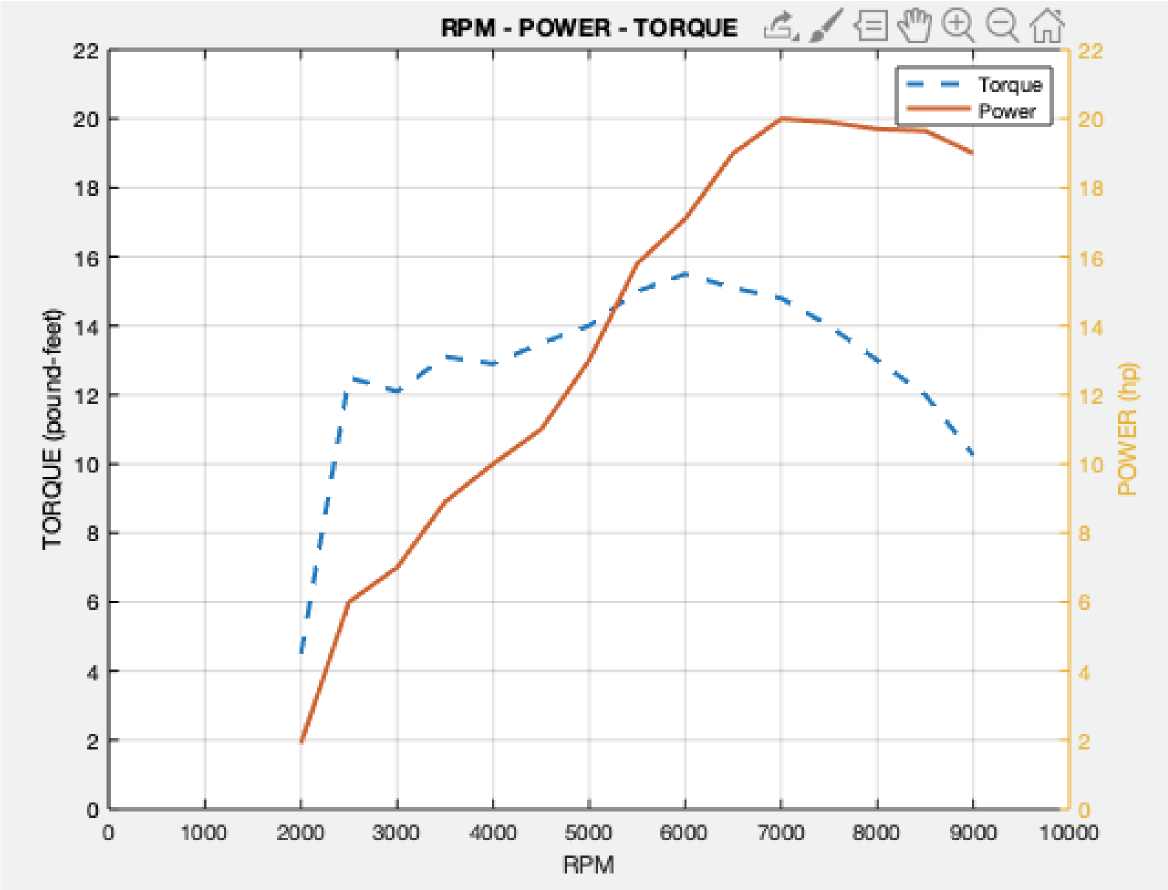
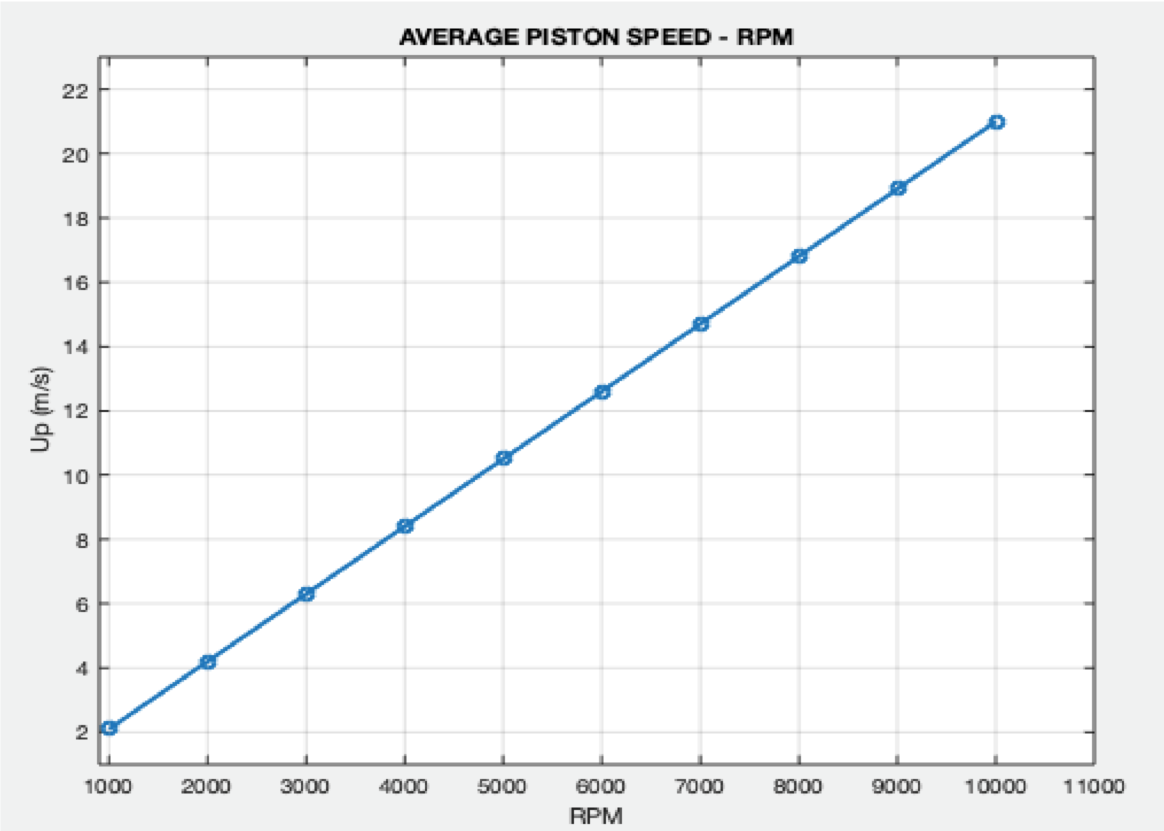
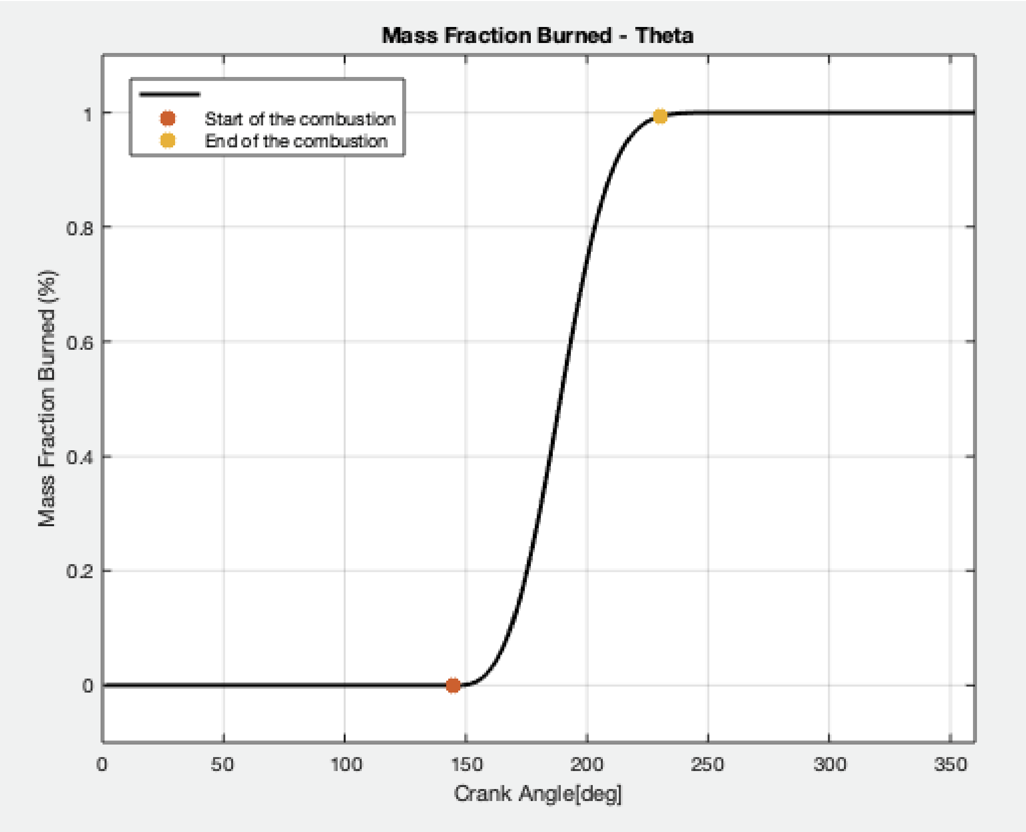


Figure 25 Brake Power - Brake Torque -RPM graph from MatLab

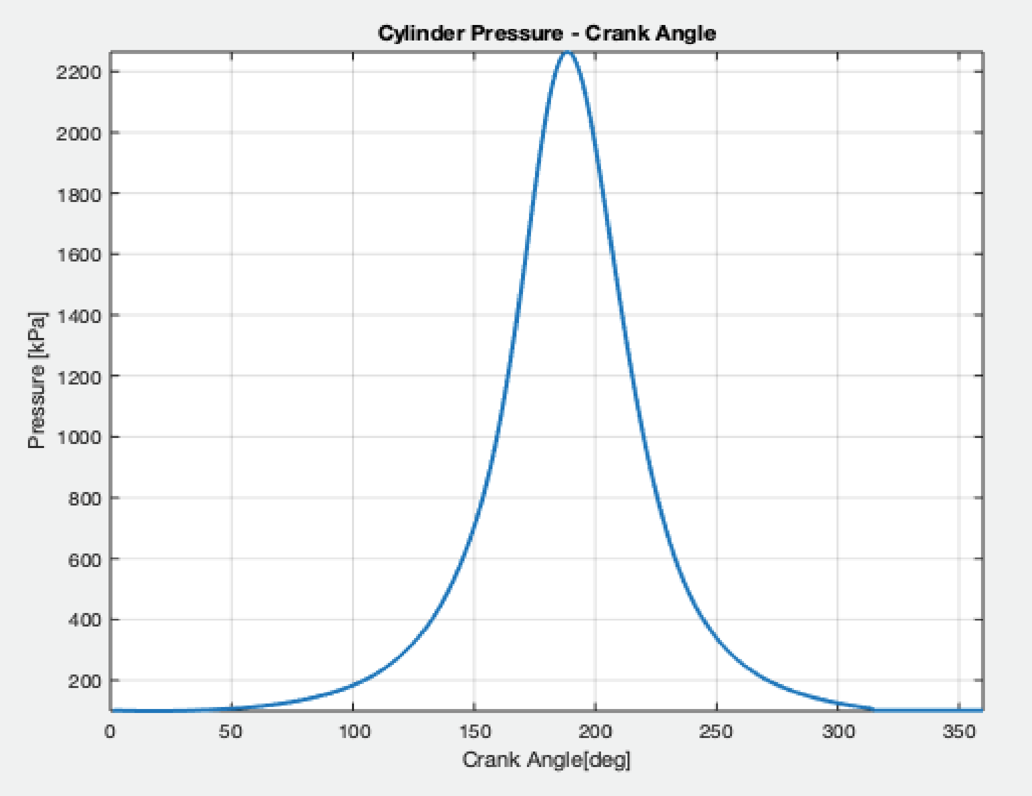
3.1.3. Average Piston Speed – RPM



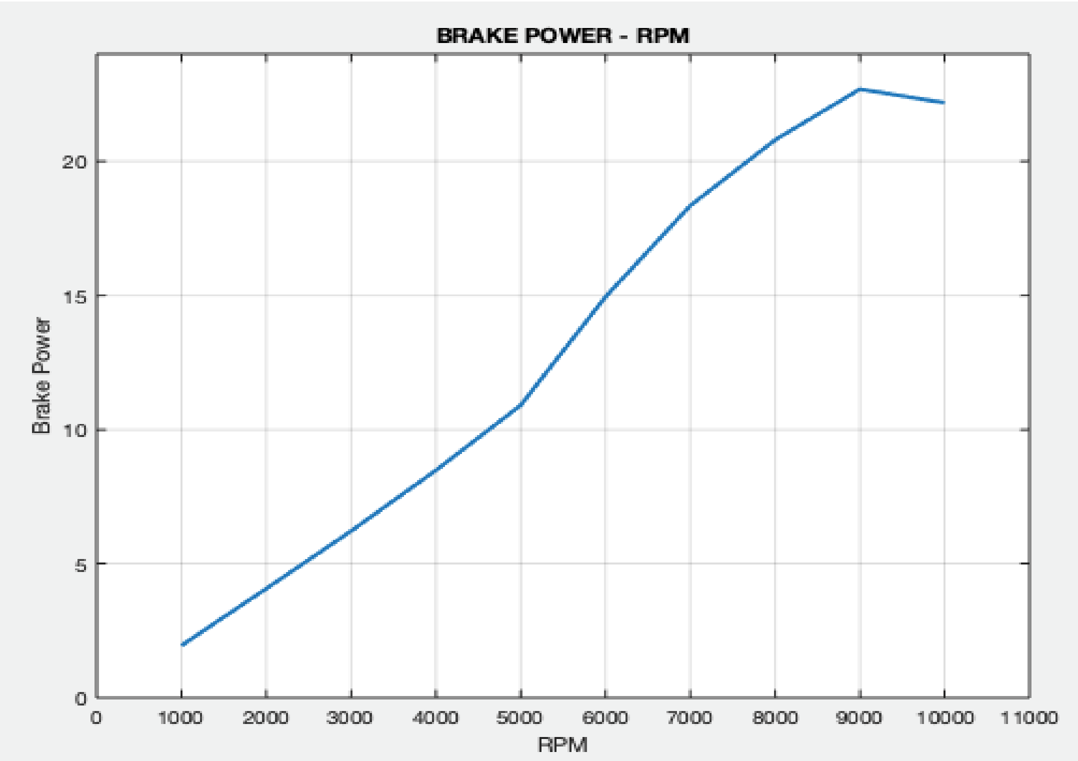
3.1.4. Mass Fraction Burned – Crank Angle



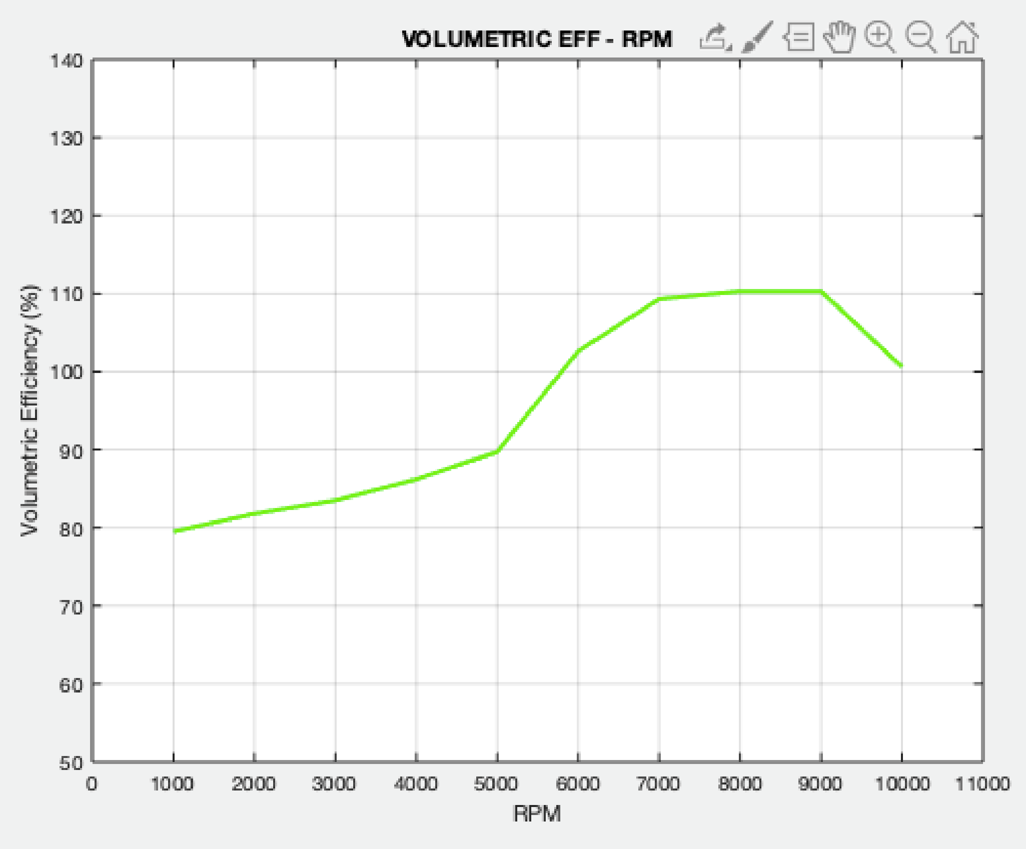
3.1.5. Cylinder Pressure – Crank Angle



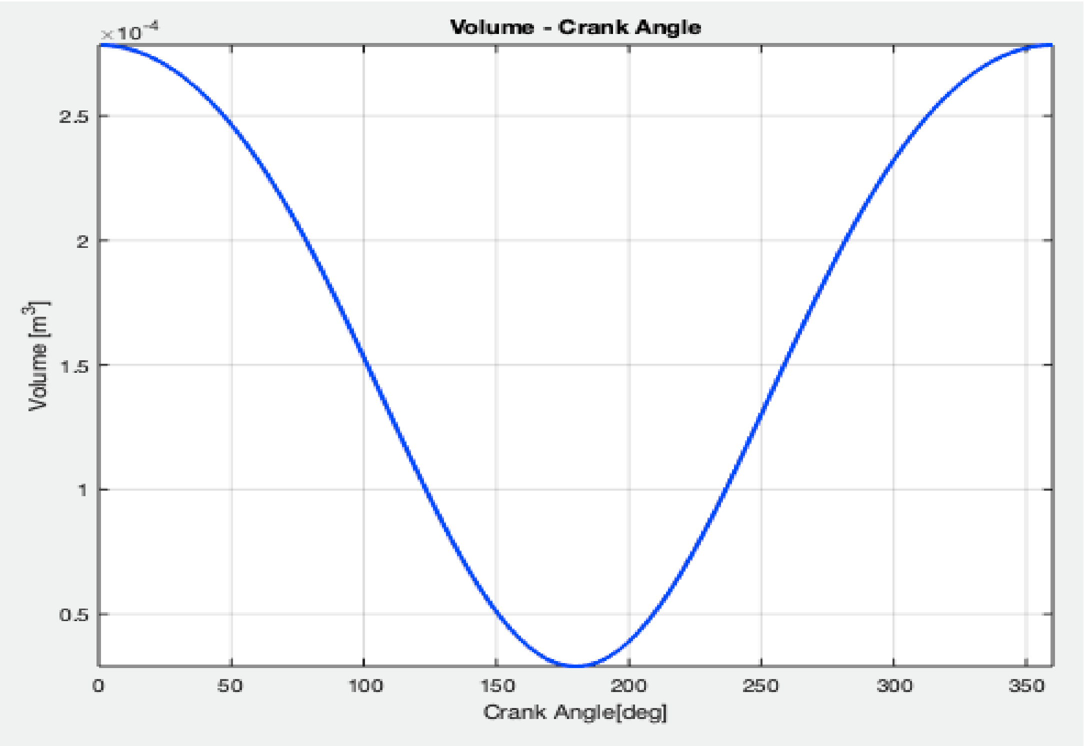
3.1.6. Brake Power – RPM



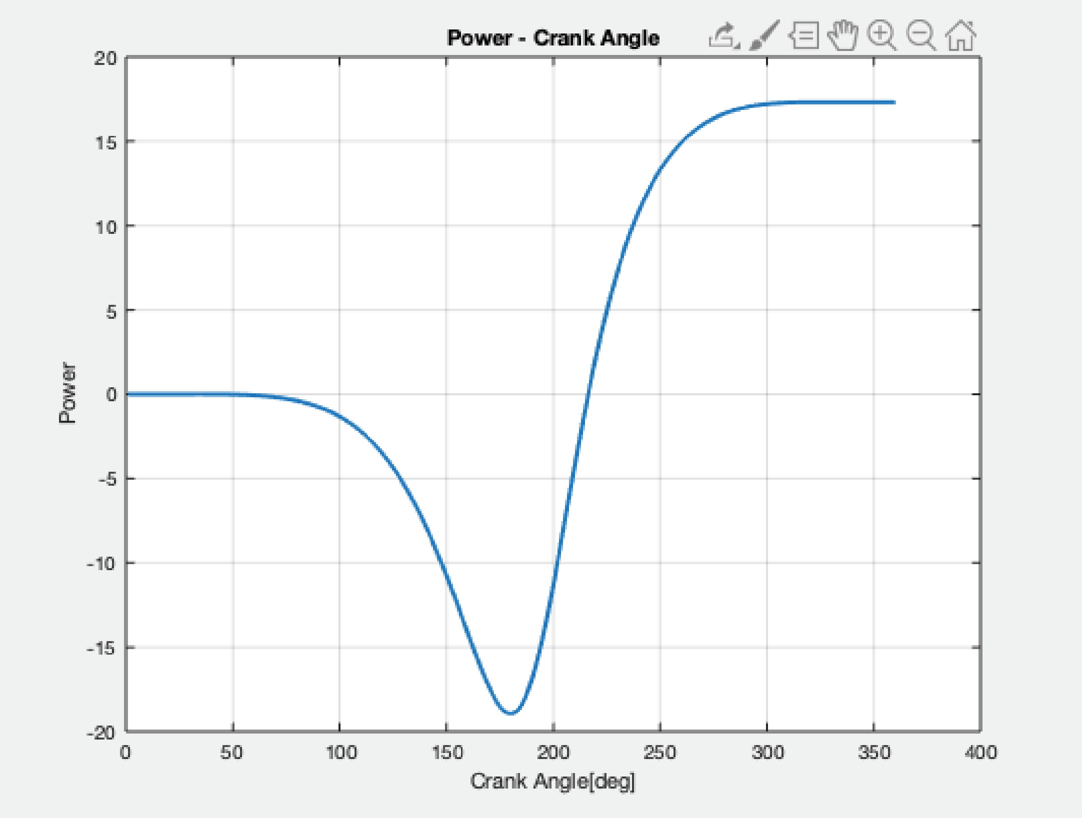
3.1.7. Volumetric Efficiency – Crank Angle



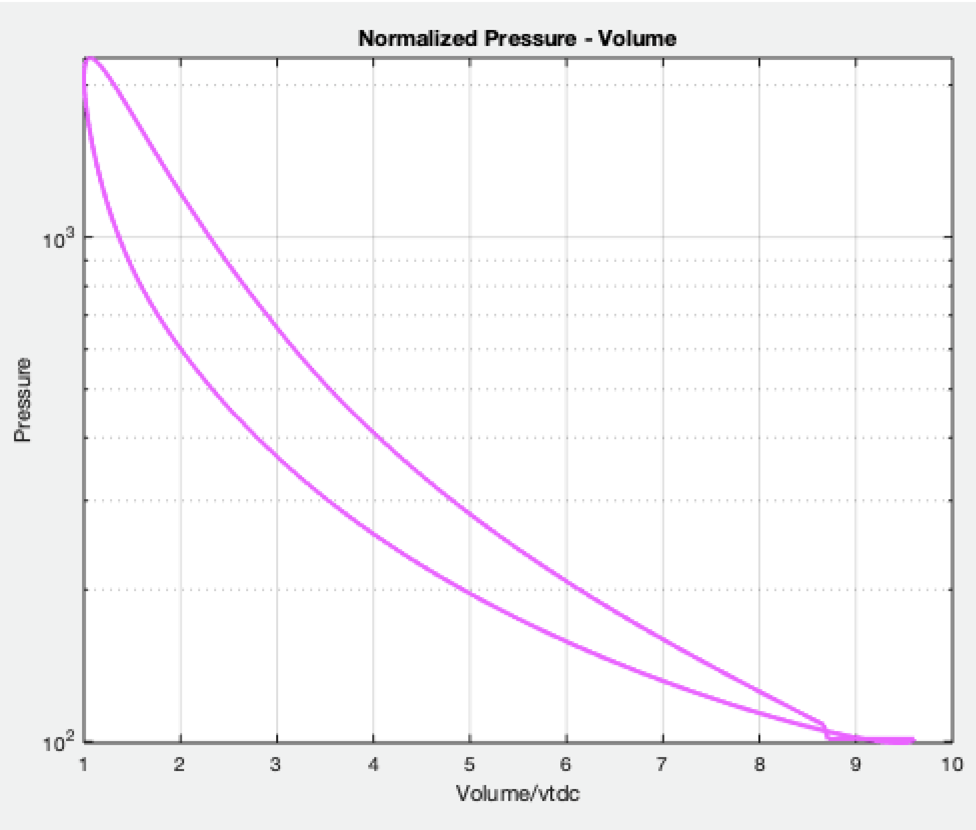
3.1.8. Volume – Crank Angle



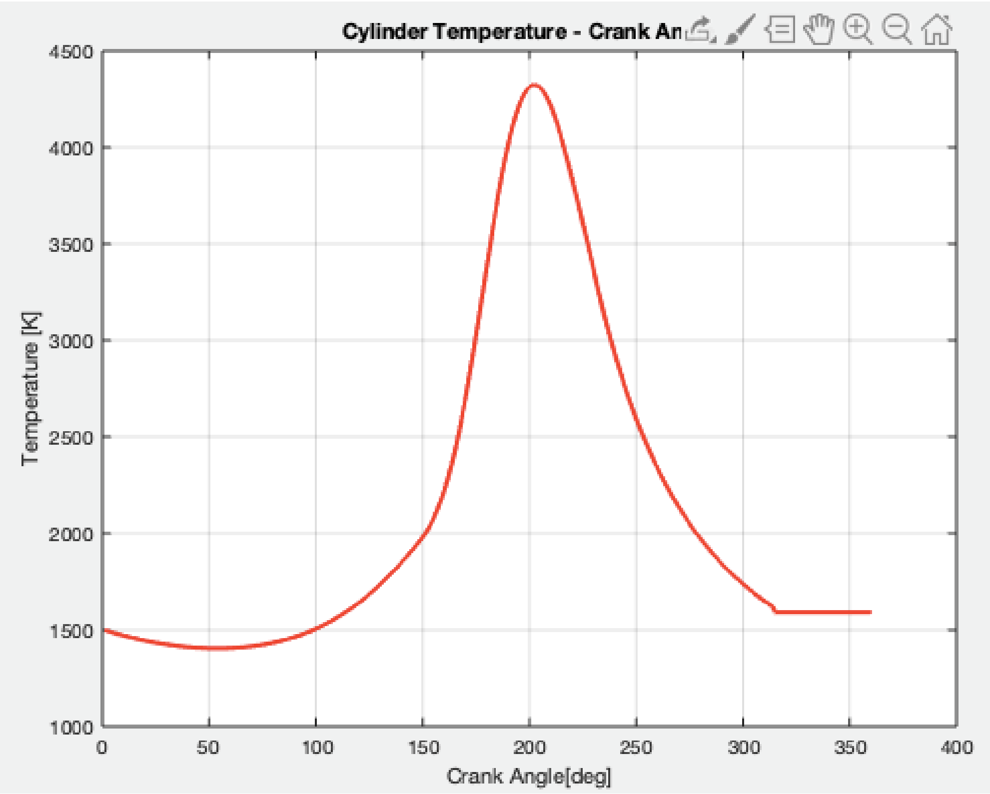
3.1.9. Power – Crank Angle



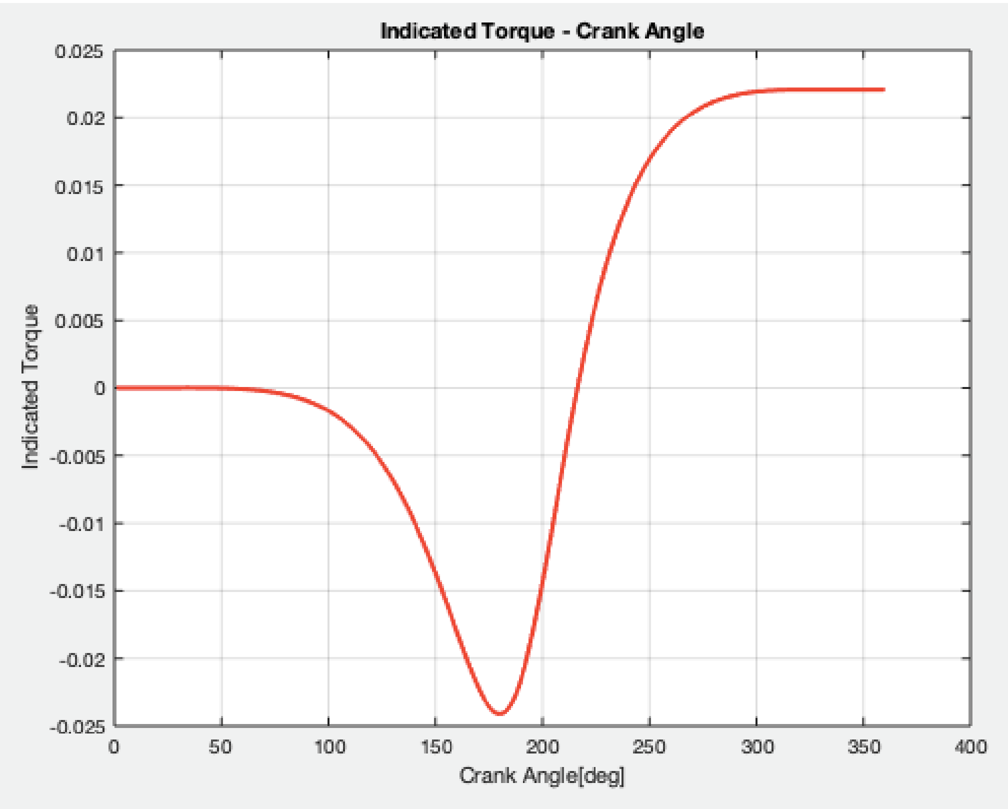
3.1.10. Normalized Pressure – Volume



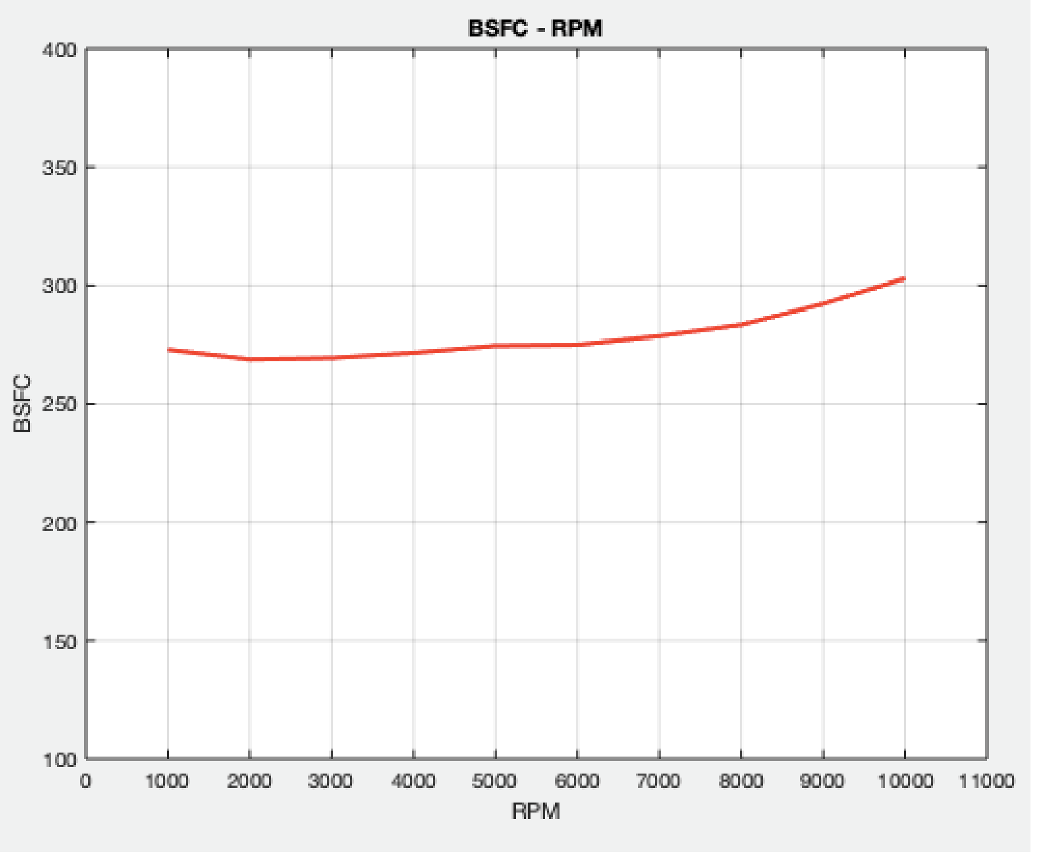
3.1.11. Cylinder Temperature – Crank Angle



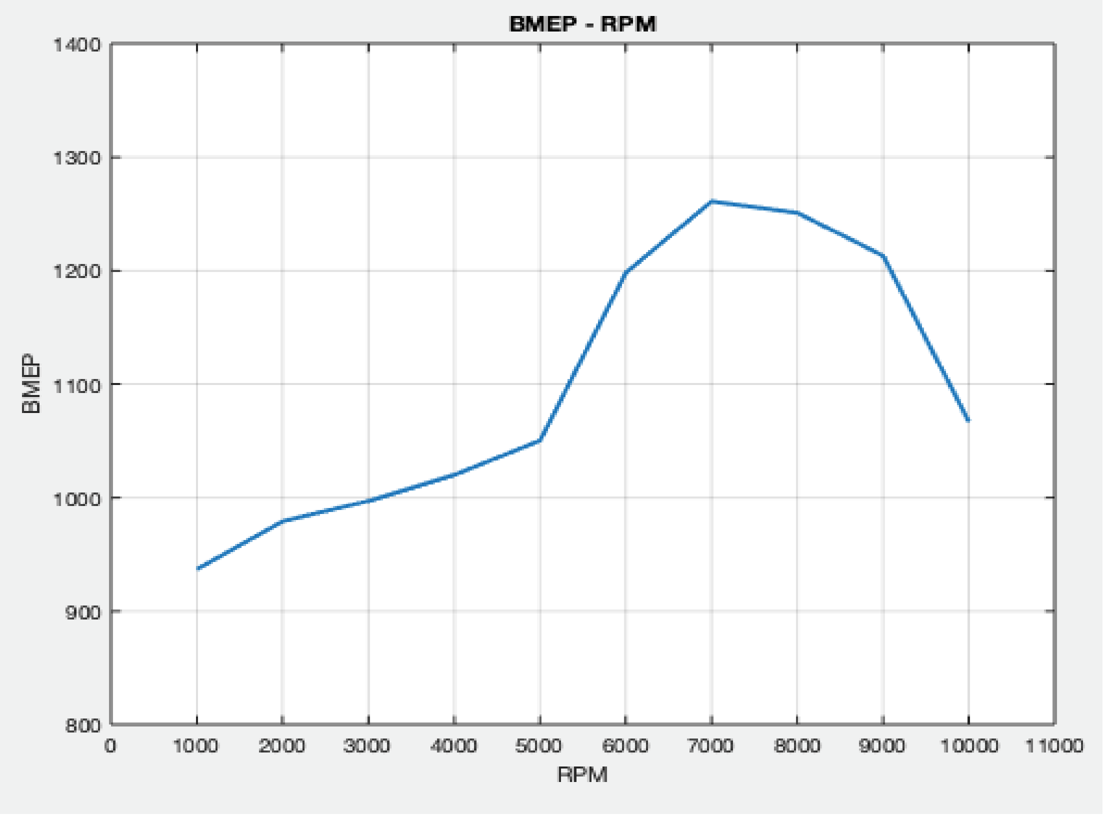
3.1.12. Indicated Torque – Crank Angle



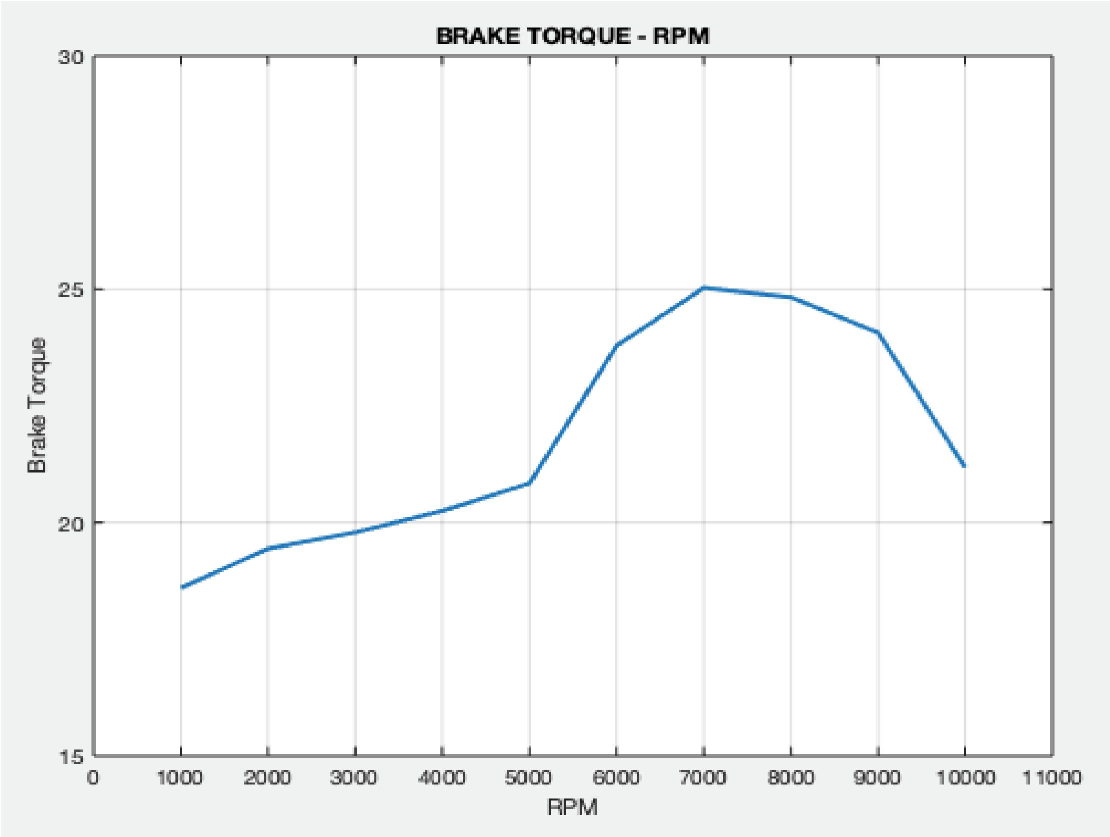
3.1.13. BSFC – RPM



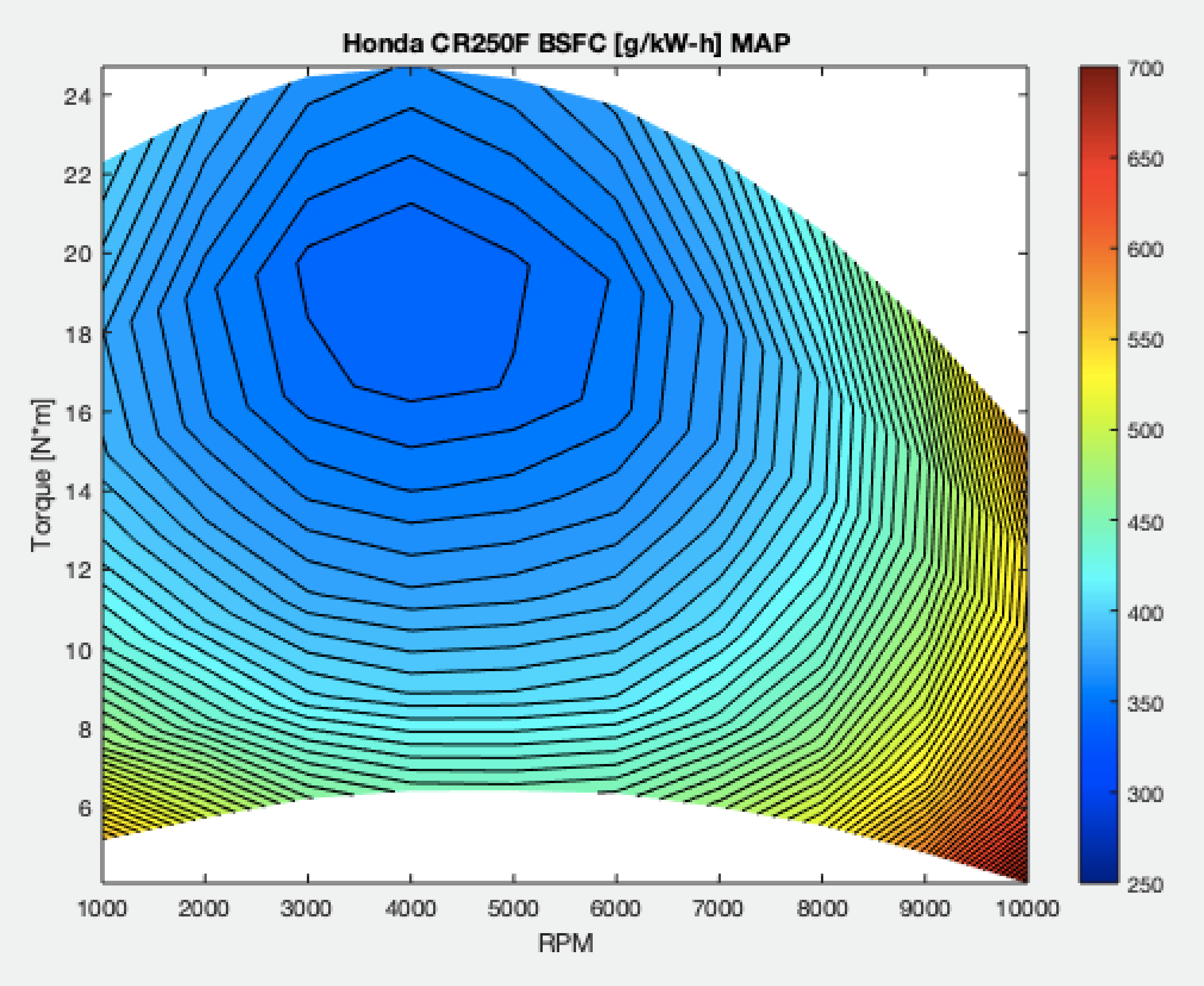
3.1.14. BMEP – RPM



3.1.15. Brake Torque – RPM



3.1.16. Torque – BSFC – RPM Performance Curve


Honda CR250F BSFC [g/kW-h] MAP
Torque [N*m]
24
22
20
18
16
14
12
10
8
6
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
RPM
700
650
600
550
500
450
400
350
300
250

4. FEASIBILITY AND COST ANALYSIS

Table 3 Fuel Consumption according to RPM

RPM	2000	5000	10000
Power (kW)	4.07	10.91	22.18
BSFC (g/kWh)	268.63	274.48	302.93
Fuel Consumption (l/h)	1.540	4.218	9.463
The cost of Gasoline (TL/l)	27.26		
The cost of consumption (TL/h)	41.98	114.98	257.96

https://buildingclub.info/calculator/g-kwh-to-l-h-online-from-gram-kwh-to-liters-per-hour/

$$Q = \frac{N \times q}{R}$$

Q = (in l / h) maximum theoretical fuel consumption in grams per 1 hour of engine operation at maximum power;

q = (in g / kWh) specific fuel consumption each RPM

N = (in kW) engine power;

R = (kg / m3) fuel density.

Gasoline (petrol) fuel density: 710 – 760 kg / m3

For 2000 RPM ;

$$Q = \frac{N \times q}{R} = \frac{4.07 \times 268.63}{710} = 1.539 \ \cong 1.540 \ l/h$$

For 5000 RPM;

$$\frac{10.91 \times 274.48}{710} = 4.2177 \ \cong 4.218 \ l/h$$

For 10000 RPM;

$$Q = \frac{N \times q}{R} = \frac{22.18 \times 302.93}{710} = 9.4633 \ \cong 9.463 \ l/h$$

As shown in calculations above, at 2000 RPM the fuel consumption is 1.54 liter per hour and cost of fuel of gasoline is 27.26 TL per liter. So, the cost of consumption for an hour is 41.98 TL.This shows us fuel consumption is really high for 10000 RPM but considering the engine is not working this high RPM constantly, this consumption can be acceptable.

5. CONCLUSION

In this investigation, the graphical results of a certain engine and simulation results were studied. Simulated results on Torque and Power appear to be close to the experimental results in the Dyno Jet testing. The results appear to be closest to each other near optimum operating conditions. This engine is a single cylinder engine so it’s lightweight, easy to handle, fuel-efficient. Also, it’s met with the California’s strict CARB emissions standards for off-road motorcycles. Compared to a bike with carburetor, this motorcycle helps reduce emissions and improve the performance and efficiency of a fuel.

Graphs are taken from Lotus Engine Software and MATLAB GUI. Lotus’s performance summary includes bmep, torque and power. When it comes to comparing these to programs, they are nearly the same, numerically. There is almost 2 kN difference between the experimental result and MATLAB result but reason of this error cannot be found yet. Possible reason of this difference is; inside the codes may have cumulative rounding errors that leads to the difference.

REFERENCES

[1]Internal Combustion Engine Fundamentals, John B. Heywood https://gctbooks.files.wordpress.com/2016/02/internal-combustion-engine-fundamentals-by-j-b-heywood.pdf

https://dirtbikemagazine.com/2022-250-mx-bikes-on-the-dyno-the-wrap/

[3] https://core.ac.uk/download/pdf/159181546.pdf

APPENDICES

MATLAB GUI CODES

```
classdef tezson < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        UIFigure                          matlab.ui.Figure
        BoremmEditFieldLabel              matlab.ui.control.Label
        BoremmEditField                   matlab.ui.control.NumericEditField
        StrokemmEditFieldLabel            matlab.ui.control.Label
        StrokemmEditField                 matlab.ui.control.NumericEditField
        RPMEditFieldLabel                 matlab.ui.control.Label
        RPMEditField                      matlab.ui.control.NumericEditField
        AtmosphericTemperatureKEditFieldLabel  matlab.ui.control.Label
        AtmosphericTemperatureKEditField  matlab.ui.control.NumericEditField
        AtmosphericPressurePaEditFieldLabel  matlab.ui.control.Label
        AtmosphericPressurePaEditField  matlab.ui.control.NumericEditField
        AFRatioEditFieldLabel             matlab.ui.control.Label
        AFRatioEditField                  matlab.ui.control.NumericEditField
        CompressionRatioEditFieldLabel  matlab.ui.control.Label
        CompressionRatioEditField         matlab.ui.control.NumericEditField
        PowerkWEditFieldLabel             matlab.ui.control.Label
        PowerkWEditField                  matlab.ui.control.NumericEditField
        TorqueNmEditFieldLabel            matlab.ui.control.Label
        TorqueNmEditField                 matlab.ui.control.NumericEditField
        IMEPEditFieldLabel                matlab.ui.control.Label
        IMEPEditField                     matlab.ui.control.NumericEditField
        BMEPEditFieldLabel                matlab.ui.control.Label
        BMEPEditField                     matlab.ui.control.NumericEditField
        BSFCEditFieldLabel                matlab.ui.control.Label
        BSFCEditField                     matlab.ui.control.NumericEditField
```

```
        MechanicalEfficiencyEditFieldLabel  matlab.ui.control.Label
        MechanicalEfficiencyEditField   matlab.ui.control.NumericEditField
        FuelConversionEfficiencyEditFieldLabel  matlab.ui.control.Label
        FuelConversionEfficiencyEditField  matlab.ui.control.NumericEditField
        ThermalEfficiencyEditFieldLabel  matlab.ui.control.Label
        ThermalEfficiencyEditField      matlab.ui.control.NumericEditField
        CalculateButton               matlab.ui.control.StateButton
        ENGINEINPUTSLabel             matlab.ui.control.Label
        OUTPUTSLabel                  matlab.ui.control.Label
        BrakePowerkWEditFieldLabel     matlab.ui.control.Label
        BrakePowerkWEditField          matlab.ui.control.NumericEditField
        FMEPEditFieldLabel            matlab.ui.control.Label
        FMEPEditField                 matlab.ui.control.NumericEditField
    end

    % Callbacks that handle component events
    methods (Access = private)

        % Value changed function: CalculateButton
        function CalculateButtonValueChanged(app, event)
            value = app.CalculateButton.Value;
            % Dilara Kocaman
            % Neslihan Büyükaydın
            % HONDA CRF250F




value = app.CalculateButton.Value;
            %Engine Inputs
Load =1  %Engine Load (Affects Inlet Pressure)
RPM = app.RPMEditField.Value; %Revolutions Per Minute [1/min]
L = (app.StrokemmEditField.Value)/1000; %Stroke of Engine [m]
B = (app.BoremmEditField.Value)/1000; %Bore of Engine [m]
l = .0935; %Length of Engine Connecting Rod [m]
N_cyl = 1; %Number of Cylinders [unitless]
C_r = app.CompressionRatioEditField.Value; %Compression Ratio [unitless]
N_r = 2; %Number of Revolutions Per Power Stroke
theta_b = 85; %Combustion Burn Duration [degrees]
theta_0 = 145; %Crank Angle At Start of Combustion [degrees]
theta_f = theta_0+theta_b; %Final Comb. Angle [degrees]
IVC = 0; %Time [degrees] when Intake Valve Closes
EVO = 314; %Time [degrees] when Exhaust Valve Opens
U_p = 2*L*RPM/60;
%________________________
```

```
%Engine Calculations Based On Previous Inputs
%Assumes Average Surface Area In Which Heat Transfer Occurs
A_p = (pi/4)*B^2; %Cross Sectional Piston Area [m^2]
A_ch = 2*A_p; %Cylinder Head Surface Area (in chamber)
V_d = N_cyl*A_p*L; %Displaced Volume Of Engine [m^3]
N = RPM/60; %Converts RPM to RPS [1/s]
S_bar_p = 2*L*N; %Calculates Mean Piston Speed [m/s]
a = L/2; %Calculates Crank Radius (1/2 stroke)[m]
V_TDC = (V_d/(C_r-1))/N_cyl; %Calculates Clearance Volume [m^3]
V_BDC = (V_d/N_cyl)+V_TDC; %Cyl. Volume At BDC [m^3]
%________________________
%Calculating Losses Due To Friction
%Engine friction model is formulated below for motorcycle engines
%This formula taken from engine friction model used in Jeremy's Thesis


%fmep = (0.018)*RPM+59.993;
fmep= 250*L*RPM*10^-3;


  % Constants
        Cp = 1.108;
        Cv = 0.821;
        Rc = Cp-Cv;
        k = Cp/Cv;



%Initial Preallocation Of Matrices (Second Preallocation In Loops Needs To
%Be Included (Do Not Delete)
V(1:360)=zeros;DV(1:360)=zeros;rho(1:360)=zeros;mu(1:360)=zeros;
C_k(1:360)=zeros;C_R(1:360)=zeros;X(1:360)=zeros;M_F(1:360)=zeros;
DX(1:360)=zeros;Re(1:360)=zeros;Nus(1:360)=zeros;h_g(1:360)=zeros;
DQ_w(1:360)=zeros;DQ(1:360)=zeros;Q(1:360)=zeros;DT(1:360)=zeros;
DP(1:360)=zeros;P(1:360)=zeros;T(1:360)=zeros;W_dot(1:360)=zeros;
W(1:360)=zeros;T_indicated(1:2)=zeros;Q_dot(1:360)=zeros;u(1:360)=zeros;
du(1:360)=zeros;cv(1:360)=zeros;m_b(1:360)=zeros;m_u(1:360)=zeros;
V_u(1:360)=zeros;V_b(1:360)=zeros;T_u(1:360)=zeros;T_b(1:360)=zeros;
A_u(1:360)=zeros;A_b(1:360)=zeros;DT_u(1:360)=zeros;gamma_u(1:360)=zeros;
u_u(1:360)=zeros;du_u(1:360)=zeros;cv_u(1:360)=zeros;DQ2(1:360)=zeros;
DQ_w2(1:360)=zeros;Q2(1:360)=zeros;
%________________________
%Fuel Inputs/Efficiencies
%Below values are taken from Jeremy's Thesis
AF_ratio_stoich = 15.09; %Gravimetric Air Fuel Ratio (Stoich)
AF_ratio_mol_sotich=14.7; %Molar Air_Fuel Ratio (Stoich)
lambda = .90; %Excess Air Coefficient
AF_ratio_ac = lambda*AF_ratio_stoich; %Actual Air Fuel Ratio
AF_ratio_mol=lambda*AF_ratio_mol_sotich;
```

```
LHV = 44.6*10^6; %Lower Heating Value Of Fuel Mixture [J/kg]
eta_combmax = .98; %Assumed MAX COmb. Efficiency


%Predicts Combustion Efficiency (Reference To Blair)
eta_comb=eta_combmax*(-1.6082+4.6509*lambda-2.0764*lambda^2);


%Atmospheric Inputs
P_atm = app.AtmosphericPressurePaEditField.Value;
T_atm = app.AtmosphericTemperatureKEditField.Value;
P_BDC = Load*P_atm; %Inlet Pressure[Pa] Turkiye,Istanbul
R_air = 287; %Gas Constant For Air [J/kg-K]
gamma(1:360) = 1.4; %Preallocate Gamma Array (sets initial value)
T_w =350; %Assumed Wall Temperature (Reference Stone)
%_________________________
%Polynomials Used To Calculate Gamma As A Function Of RPM
a_1 = .692; a_2 = 39.17e-06; a_3 = 52.9e-09; a_4 = -228.62e-13;
a_5 = 277.58e-17;b_0 = 3049.33; b_1 = -5.7e-02; b_2 = -9.5e-05;
b_3 = 21.53e-09;b_4 = -200.26e-14;c_u = 2.32584; c_r = 4.186e-03;
d_0 = 10.41066; d_1 = 7.85125; d_3 = -3.71257;e_0 = -15.001e03;
e_1 = -15.838e03; e_3 = 9.613e03;f_0 = -.10329; f_1 = -.38656;
f_3 = .154226; f_4 = -14.763; f_5 = 118.27; f_6 = 14.503;
r_0 = -.2977; r_1 = 11.98; r_2 = -25442; r_3 = -.4354;
%_________________________


R=R_air/1000;
T_corr=1500;


for k = 1:2
%Corrects Temperature Based On Exhaust Gas Residuals
if k==1
T_BDC = T_atm; %Assumed Inlet Temperature [K]
else
T_BDC=T_corr;
end


%Calculate Mass of Air In Cylinder/ Mass Of Fuel Based On AFR
rho_a = P_BDC/(R_air*T_BDC); %Air Density kg/m^3
m_a = rho_a*V_d; %Mass of Air In Cylinder [kg]
m_f = m_a/AF_ratio_ac; %Mass Of Fuel In Cylinder [kg]
m_c = m_a+m_f; %Mass In Cylinder
%________________________
%Specifying Initial Conditions For Loops
%DV,DX,etc. Are Relative To Change In Theta (i.e. DV/Dtheta)
theta(1:360)=zeros; %Starting Crank Angle [deg]
V(1:360)=zeros; %Preallocate Volume Array
```

```
V(1)=V_BDC; %Starting Combustion Chamber Volume [m^3]
DV(1:360) = zeros; %Preallocate Change In Volume Array
DV(1) = 0; %Specifying Initial Change In Volume [m^3}
P(1:360)=P_BDC; %Preallocate Pressure Array
DP(1:360) = zeros; %Specifying Initial Change In Pressure
T(1:360)=zeros; %Preallocate Temperature Array
T(1) = T_BDC; %Inlet Temperature [K]
T_u(1)=T_BDC; %Initial Unburned Temperature[K]
DT(1:360) = zeros; %Specifying Initial Change In Temperature
DT_u(1:360)=zeros; %Preallocate Change In Unburned Temperature
gamma(1)=1.4; %Initial Gamma Input
gamma_u(1)=1.4; %Initial Gamma Input
X(1:360) = 0; %Preallocate Mass Burn Array
DX(1:360) = zeros; %Preallocate Change In Mass Burn Fraction [unitless]
DQ(1:360) = zeros; %Preallocate Heat Release Array
DQ2(1:360)=zeros; %Preallocate Two Zone Heat Release Array
Q(1:360)=zeros; %Preallocate Heat Array
Q2(1:360)=zeros; %Preallocate 2 zone Heat Array
M_F(1:360) = 0; %Preallocate Mass In Comubstion Chamber Array
rho(1:360) = zeros; %Preallocates Ideal Gas Law array
rho(1) = P(1)/(R_air*T(1)); %Initial Value Ideal Gas Array
mu(1:360)=zeros; %Preallocate Viscosity Array
mu(1)=7.457*10^(-6)+4.1547*10^(-8)*T_BDC-7.4793*10^(-12)*T_BDC^(2);
C_k(1:360)=zeros; %Preallocate Thermal COnductivity Array
C_k(1) = 6.1944*10^(-3)+7.3814*10^(-5)*T_BDC-1.2491*10^(-8)*T_BDC^(2);
C_R(1:360) = zeros; %Preallocate Radiation Coefficient Array
C_R(1) = 4.25*10^(-09)*((T(1)^4-T_w^4)/(T(1)-T_w)); %Initial Rad. Coeff
Re(1:360)=zeros; %Preallocate Reynolds Value Array
Re(1)=rho(1)*S_bar_p*B/mu(1); %Initial Reynolds Value
Nus(1:360)=zeros; %Preallocating Nusselt Number Array
Nus(1)=.49*Re(1)^(.7); %Initial Nusselt Number
h_g(1:360)=zeros; %Preallocate Heat Transfer Coefficient Array
h_g(1)=C_k(1)*Nus(1)/B; %Initial Heat Transfer Coefficient
s(1:360)=zeros; %Preallocates Distance Crank/Piston Axes Array
s(1) = -a*cosd(theta(1))+sqrt(l^2 - a^2*sind(theta(1))^2);%Initial Val.
W(1:360) = zeros; %Preallocate Work Array
W_dot(1:360) = zeros; %Preallocate Power Array
T_indicated(1:360) = zeros; %Preallocate Torque Array
Q_dot(1:360) = zeros; %Preallocate Heat Transfer Array
u(1:360) = zeros; %Preallocate Internal Energy Array
du(1:360) = zeros; %Preallocates Change In Internal Energy Array
cv(1:360) = zeros; %Preallocates Heat Capacity Array
DQ_w(1:360)=zeros; %Preallocate Convective Heat Loss Array
DQ_w2(1:360)=zeros; %Preallocate Convective Heat Loss Array 2 zone
m_b(1:360)= zeros; %Preallocate mass burned array
m_u(1:360)=m_c; %Preallocate unburned mass array
V_u(1:360)=zeros; %Preallocate unburned Volume Array
V_u(1) = V(1); %Initial Unburned Volume
%________________________
```

```
theta=1:360;


for i = 2:360
%Specifies Distance Between Crank/Piston Axes As A Function Of theta
s = -a*cosd(theta(i))+sqrt(l^2 - a^2*sind(theta(i))^2);
%Specifies Volume As A Function Of Crank Angle
V(i) = V_TDC +((pi/4)*B^2)*(l + a - s);
%Specifies Change In Volume As A Function Of Crank Angle
DV(i) = V(i)-V(i-1);
%Calculates Density As A Function Of Crank Angle
rho(i) = P(i-1)/(R_air*T(i-1)); %% degisti i-1
%Calculates Viscosity As A Function Of Temperature
mu(i)=7.457*10^(-6)+4.1547*10^(-8)*T(i-1)-7.4793*10^(-12)*T(i-1)^(2);
%Calculating Instantaneous Thermal Conductivity of Cylinder Gas
C_k(i) = 6.1944*10^(-3)+7.3814*10^(-5)*T(i-1)-1.2491*10^(-8)*T(i-1)^(2);
%Calculating The Radiation Heat Transfer Coefficient
C_R(i) = 4.25*10^(-09)*((T(i-1)^4-T_w^4)/(T(i-1)-T_w));
%Instantaneous Suface Area (For Heat Transfer)
A = A_ch + A_p + pi*B*(l+a-s);


if i<=2
A_u=A;
end


%_________________________
%Specifies Mass Fraction Burn As A Function Of Crank Angle (Weibe Fcn.)
%Also Specifies Mass Of Fuel In Combustion Chamber As A Function Of
%Theta
if theta(i)<theta_0
X(i)=0;
else
X(i) = 1-exp(-5*((theta(i)-theta_0)/theta_b)^3);
if theta(i) < theta_f
M_F(i) = V(theta_0-1)*rho(theta_0-1)/(lambda*AF_ratio_mol);
end
end
%__________________________
%Specifies Change In Mass Fraction Burn As A Function Of Crank Angle
DX(i) = X(i) - X(i-1);


%__________________________
%Incorporating The Annand Method To Predict Heat Transfer
%Calculating Reynolds Number
Re(i)=rho(i)*S_bar_p*B/mu(i);
%Calculating Nusselt Number (constant=.26 two stroke, .49 4 stroke)
Nus(i)=.49*Re(i)^(.7);
```

```
%Calculating Heat Transfer Coefficient Using Annand Method
h_g(i)=C_k(i)*Nus(i)/B;
%Calculates Convective Losses Into Wall As A Function Of Crank Angle
DQ_w(i) = (h_g(i)+C_R(i))*A*(T(i-1)-T_w)*(60/(360*RPM));
%Calculates Change In Heat Transfer (total) As A Function Of Crank
%Angle
DQ(i) = eta_comb*LHV*M_F(i)*DX(i)-DQ_w(i);
%Calculates Total Heat Transfer (Per Cycle)
Q(i) = Q(i-1)+DQ(i);
%__
%
thermalefficiency= W(i)/(m_f*LHV*0.95);



%Specifies Pressure and Temperature Increases Between Intake Valve
%Closing and Exhaust Valve Opening
if IVC< theta(i)
DT(i)=T(i-1)*(gamma(i-1)-1)*((1/(P(i-1)*V(i-1)))*DQ(i)...
-(1/V(i-1))*DV(i));
DP(i)=(-P(i-1)/V(i-1))*DV(i)+(P(i-1)/T(i-1))*DT(i);
P(i) = P(i-1)+DP(i);
end
if EVO < theta(i)
P(i) = P_atm;
end
if 200 < theta(i)
if P(i)<=P_atm
P(i)=P_atm;
end
end




%________________________
%Returns Temperature Values To Beginning Of Loop
%Assumes Temperature Drops Back To ATM Temp After Exhaust Is Extracted
T(i) = T(i-1)+DT(i);


%Calculate The Residual Gas Fraction
%Assume A Polytropic Constant Of 1.3
R_frac = (1/C_r)*(P_BDC/P_atm)^(1/1.3)*(1/lambda);


%Calculates Cylinder Work [J] As A Function Of Crank Angle
```

```
%Treats Atmospheric Pressure As Reference State
 W(i) = W(i-1)+(P(i)-P_atm)*DV(i);


% %Calculates Power [kW] As A Function Of Crank Angle
 %W_dot(i)=(N_cyl*W(i)*N/N_r)/1000;


W_dot(i)=(W(i)*RPM/N_r)*10^-4
mep=W_dot*N_r/(V_d*N);

%Indicated Mean Effective Pressure
CF=0.99;
imep = CF*W_dot(i)*N_r*1000/(V_d*1000*N);


%Calculates Torque[N*m] As A Function Of Crank Angle
T_indicated(i) = (W_dot(i))/(2*pi*N);


%Calculates Heat Loss [kW] As A Function Of Crank Angle
Q_dot(i) = (N_cyl*Q(i)*N/N_r)/1000;


%Calculate Temperature Of Exhaust Based On Polytropic Relations
   if EVO < theta(i)
   T(i)=T(EVO)*(P_BDC/P(EVO))^((gamma(i)-1)/gamma(i));

   end
end
end


%Specified Outputs (On Matlab Screen)
W_dot_indicated=W_dot(360);
bmep=T_indicated(i)*(4*pi)/V_d
bmep = imep-fmep;
imep=bmep+fmep
W_dot_ac =(bmep*V_d*1000*N/(N_r*1000))
T_ac = W_dot(i)/(2*pi*N*10^(-3))


%Calculated Mechanical Efficiency (Based On Previous Inputs)
% eta_m = bmep/imep %Calculates Mechanical Efficiency
eta_m=W_dot_ac/W_dot(i)
Qin = M_F*LHV*1;
thermalefficiency= W(i)/(m_f*LHV*0.98);
%___________________________
%Calculates Brake Specific Fuel Consumption
 m_ta = P_BDC*V_d/(R_air*T_BDC); %Calculate Trapped Air In Cylinder
```

```
 % eta_v = CF*((m_ta)/(rho_a*V_d)); %Corrected Volumetric Efficiency
m_dot_f = W(i)/(thermalefficiency*LHV*0.98)


%N_cyl*M_F(theta_0)*(N/N_r); %Mass Flow Rate Of Fuel
m_dot_a = AF_ratio_ac*m_dot_f; %Mass Flow Rate Of Air
nf= W(i)/(m_f*LHV)
BSFC=3600/(nf*LHV*10^-6) %BSFC [g/kW*h]
eta_f = 3600/(BSFC*(LHV*10^(-6))) %Fuel Conversion Efficiency




if RPM>8500
  BSFC=BSFC*1.1
end


%Specifies Conditions For Minimum and Maximum Plot Values
v_min = min(V); v_max = max(V);
p_min = min(P); p_max = max(P);
w_min = min(W_dot); w_max = max(W_dot);
T_min = min(T); T_max = max(T);
Q_min = min(Q_dot); Q_max = max(Q_dot);
Tmin = min(T_indicated); Tmax = max(T_indicated);


bmep2=6.28*N_cyl*T_indicated(i)/V_d;
imep2=fmep+bmep2;
mek= bmep2/imep2;
wbrake=mek*W_dot(i)

%
%   if RPM > 12000
% mek = 0
% nf = 0
% thermalefficiency = 0
% bmep2= 0
% imep2 = 0
%


% end




app.BrakePowerkWEditField.Value=wbrake
```

```
app.PowerkWEditField.Value= W_dot(i);
app.TorqueNmEditField.Value=T_ac;
app.BSFCEditField.Value=BSFC;
app.IMEPEditField.Value=imep2;
app.MechanicalEfficiencyEditField.Value=mek;
app.FuelConversionEfficiencyEditField.Value=nf;
app.ThermalEfficiencyEditField.Value=thermalefficiency;
app.BMEPEditField.Value=bmep2;
app.ThermalEfficiencyEditField.Value= thermalefficiency;
app.FMEPEditField.Value= U_p;



TORK=bmep*V_d/(4*pi)*1300
%app.ThermalEfficiencyEditField.Value= U_p;


% % %Plot Statements
 figure(1)
 plot(theta,V,'b',"LineWidth",2)
 title('Volume - Crank Angle')
 xlabel('Crank Angle[deg]')
 ylabel('Volume [m^3]')
 %axis([0 360 v_min v_max])
 ylim ([v_min v_max])
 xlim ([0 360])
 grid on
 %
  figure(2)
 plot(theta,T,'r',"LineWidth",2)
 title('Cylinder Temperature - Crank Angle')
 xlabel('Crank Angle[deg]')
 ylabel('Temperature [K]')
 %axis([0 360 T_min T_max])
%  ylim ([T_min T_max])
%  xlim ([0 360])
grid on


 figure(3)
 plot(theta,P/1000,"LineWidth",2)
 title('Cylinder Pressure - Crank Angle')
 xlabel('Crank Angle[deg]')
 ylabel('Pressure [kPa]')
%axis([0 360 p_min/1000 p_max/1000])
 ylim ([p_min/1000 p_max/1000])
 xlim ([0 360])
grid on
```

```
figure(4)
plot(theta,X,'k',"LineWidth",2)
hold on
plot(145,0,'*',"LineWidth",5)
plot(230,0.993262,'*',"LineWidth",5)
title('Mass Fraction Burned - Theta')
xlabel('Crank Angle[deg]')
ylabel('Mass Fraction Burned (%)')
%axis([0 360 -.1 1.1])
ylim ([-.1 1.1])
xlim ([0 360])
legend('','Start of the combustion','End of the combustion' )
grid on


%
figure(5)
semilogy(V/V_TDC,P/1000,'m',"LineWidth",2);
grid on
title('Normalized Pressure - Volume')
xlabel('Volume/vtdc')
ylabel('Pressure')


%
figure(6)
plot(theta,W_dot,"LineWidth",2)
title('Power - Crank Angle')
xlabel('Crank Angle[deg]')
ylabel('Power')
grid on
%
figure(7)
plot(theta,T_indicated,'r',"LineWidth",2)
title('Indicated Torque - Crank Angle')
xlabel('Crank Angle[deg]')
ylabel('Indicated Torque')
grid on




%expected torque due to RPM graph
figure
RPM1= [2000:500:9000]
TORK1= [4.5 12.5 12.1 13.1 12.9 13.5 14 15 15.5 15.1 14.8 14 13 12 10.25]
POWER1=[1.9 6 7 8.9 10 11 13 15.8 17.1 19 20 19.9 19.7 19.65 19]
plot(RPM1,TORK1,'--',"LineWidth",2)
ylim ([0 22])
```

```
xlim ([0 10000])
hold on
plot (RPM1,POWER1,"LineWidth",2)
title ('RPM - POWER - TORQUE')
yyaxis right
ylim([0 22])
ylabel('POWER (hp)')
yyaxis left
ylabel('TORQUE (pound-feet)')
xlabel('RPM')
legend('Torque','Power')
grid on


figure


RPM3=[1000:1000:10000]
BPOWER=[1.95 4.07 6.22 8.48 10.91 14.95 18.35 20.8 22.69 22.18]
plot(RPM3,BPOWER,"LineWidth",2)
title('BRAKE POWER - RPM')
xlim([0 11000])
ylim([0 24])
xlabel('RPM')
ylabel('Brake Power')
grid on



figure
BTORK=[18.6 19.44 19.79 20.25 20.84 23.79 25.03 24.83 24.07 21.18]
plot(RPM3,BTORK,"LineWidth",2)
title('BRAKE TORQUE - RPM')
xlim([0 11000])
ylim([15 30])
xlabel('RPM')
ylabel('Brake Torque')
grid on


figure
BMEPLOTUS=[937 979 997 1020 1050 1198 1261 1251 1213 1067]
plot(RPM3,BMEPLOTUS,"LineWidth",2)
title('BMEP - RPM')
xlim([0 11000])
ylim([800 1400])
xlabel('RPM')
ylabel('BMEP')
grid on
```

```
figure
BSFCLOTUS=[272.84 268.63 269.25 271.45 274.48 274.9 278.58 283.33 292.21 302.93]
plot(RPM3,BSFCLOTUS,'-r',"LineWidth",2)
title('BSFC - RPM')
xlim([0 11000])
ylim([270 310])
xlabel('RPM')
ylabel('BSFC')
grid on


figure
VOLUMETRIC=[79.5 81.8 83.5 86.2 89.7 102.6 109.3 110.3 110.3 100.6]
plot(RPM3,VOLUMETRIC,'-g',"LineWidth",2)
title('VOLUMETRIC EFF - RPM')
xlim([0 11000])
ylim([50 140])
xlabel('RPM')
ylabel('Volumetric Efficiency (%)')
grid on


figure
RPM2=[1000:1000:10000]
UP=[2.1 4.2 6.3 8.4 10.5 12.6 14.7 16.8 18.9 21]
plot(RPM2,UP,'-o','LineWidth',2)
xlim([900 11000])
ylim([1 23])
xlabel('RPM')
ylabel('Up (m/s)')
title('AVERAGE PISTON SPEED - RPM')
grid on

        end
    end

    % Component initialization
    methods (Access = private)
```

```
% Create UIFigure and components
function createComponents(app)

    % Create UIFigure and hide until all components are created
    app.UIFigure = uifigure('Visible', 'off');
    app.UIFigure.Position = [100 100 640 480];
    app.UIFigure.Name = 'UI Figure';

    % Create BoremmEditFieldLabel
    app.BoremmEditFieldLabel = uilabel(app.UIFigure);
    app.BoremmEditFieldLabel.HorizontalAlignment = 'right';
    app.BoremmEditFieldLabel.FontWeight = 'bold';
    app.BoremmEditFieldLabel.Position = [50 388 61 22];
    app.BoremmEditFieldLabel.Text = 'Bore(mm)';

    % Create BoremmEditField
    app.BoremmEditField = uieditfield(app.UIFigure, 'numeric');
    app.BoremmEditField.Position = [126 388 100 22];
    app.BoremmEditField.Value = 71;

    % Create StrokemmEditFieldLabel
    app.StrokemmEditFieldLabel = uilabel(app.UIFigure);
    app.StrokemmEditFieldLabel.HorizontalAlignment = 'right';
    app.StrokemmEditFieldLabel.FontWeight = 'bold';
    app.StrokemmEditFieldLabel.Position = [36 352 75 22];
    app.StrokemmEditFieldLabel.Text = 'Stroke (mm)';

    % Create StrokemmEditField
    app.StrokemmEditField = uieditfield(app.UIFigure, 'numeric');
    app.StrokemmEditField.Position = [126 352 100 22];
    app.StrokemmEditField.Value = 63;

    % Create RPMEditFieldLabel
    app.RPMEditFieldLabel = uilabel(app.UIFigure);
    app.RPMEditFieldLabel.HorizontalAlignment = 'right';
    app.RPMEditFieldLabel.FontWeight = 'bold';
    app.RPMEditFieldLabel.Position = [78 321 33 22];
    app.RPMEditFieldLabel.Text = 'RPM';

    % Create RPMEditField
    app.RPMEditField = uieditfield(app.UIFigure, 'numeric');
    app.RPMEditField.Position = [126 321 100 22];
    app.RPMEditField.Value = 7500;
```

```
% Create AtmosphericTemperatureKEditFieldLabel
app.AtmosphericTemperatureKEditFieldLabel = uilabel(app.UIFigure);
app.AtmosphericTemperatureKEditFieldLabel.HorizontalAlignment = 'right';
app.AtmosphericTemperatureKEditFieldLabel.FontWeight = 'bold';
app.AtmosphericTemperatureKEditFieldLabel.Position = [3 270 173 22];
app.AtmosphericTemperatureKEditFieldLabel.Text = 'Atmospheric Temperature (K)';

% Create AtmosphericTemperatureKEditField
app.AtmosphericTemperatureKEditField = uieditfield(app.UIFigure, 'numeric');
app.AtmosphericTemperatureKEditField.Position = [191 270 100 22];
app.AtmosphericTemperatureKEditField.Value = 290;

% Create AtmosphericPressurePaEditFieldLabel
app.AtmosphericPressurePaEditFieldLabel = uilabel(app.UIFigure);
app.AtmosphericPressurePaEditFieldLabel.HorizontalAlignment = 'right';
app.AtmosphericPressurePaEditFieldLabel.FontWeight = 'bold';
app.AtmosphericPressurePaEditFieldLabel.Position = [18 238 158 22];
app.AtmosphericPressurePaEditFieldLabel.Text = 'Atmospheric Pressure (Pa)';

% Create AtmosphericPressurePaEditField
app.AtmosphericPressurePaEditField = uieditfield(app.UIFigure, 'numeric');
app.AtmosphericPressurePaEditField.Position = [191 238 100 22];
app.AtmosphericPressurePaEditField.Value = 101300;

% Create AFRatioEditFieldLabel
app.AFRatioEditFieldLabel = uilabel(app.UIFigure);
app.AFRatioEditFieldLabel.HorizontalAlignment = 'right';
app.AFRatioEditFieldLabel.FontWeight = 'bold';
app.AFRatioEditFieldLabel.Position = [118 207 58 22];
app.AFRatioEditFieldLabel.Text = 'AF Ratio ';

% Create AFRatioEditField
app.AFRatioEditField = uieditfield(app.UIFigure, 'numeric');
app.AFRatioEditField.Position = [191 207 100 22];
app.AFRatioEditField.Value = 15;

% Create CompressionRatioEditFieldLabel
app.CompressionRatioEditFieldLabel = uilabel(app.UIFigure);
app.CompressionRatioEditFieldLabel.HorizontalAlignment = 'right';
app.CompressionRatioEditFieldLabel.FontWeight = 'bold';
app.CompressionRatioEditFieldLabel.Position = [61 174 115 22];
app.CompressionRatioEditFieldLabel.Text = 'Compression Ratio';
```

```
% Create CompressionRatioEditField
app.CompressionRatioEditField = uieditfield(app.UIFigure, 'numeric');
app.CompressionRatioEditField.Position = [191 174 100 22];
app.CompressionRatioEditField.Value = 9.6;

% Create PowerkWEditFieldLabel
app.PowerkWEditFieldLabel = uilabel(app.UIFigure);
app.PowerkWEditFieldLabel.HorizontalAlignment = 'right';
app.PowerkWEditFieldLabel.FontWeight = 'bold';
app.PowerkWEditFieldLabel.Position = [378 388 67 22];
app.PowerkWEditFieldLabel.Text = 'Power(kW)';

% Create PowerkWEditField
app.PowerkWEditField = uieditfield(app.UIFigure, 'numeric');
app.PowerkWEditField.Position = [460 388 100 22];

% Create TorqueNmEditFieldLabel
app.TorqueNmEditFieldLabel = uilabel(app.UIFigure);
app.TorqueNmEditFieldLabel.HorizontalAlignment = 'right';
app.TorqueNmEditFieldLabel.FontWeight = 'bold';
app.TorqueNmEditFieldLabel.Position = [367 352 78 22];
app.TorqueNmEditFieldLabel.Text = 'Torque (N.m)';

% Create TorqueNmEditField
app.TorqueNmEditField = uieditfield(app.UIFigure, 'numeric');
app.TorqueNmEditField.Position = [460 352 100 22];

% Create IMEPEditFieldLabel
app.IMEPEditFieldLabel = uilabel(app.UIFigure);
app.IMEPEditFieldLabel.HorizontalAlignment = 'right';
app.IMEPEditFieldLabel.FontWeight = 'bold';
app.IMEPEditFieldLabel.Position = [409 291 36 22];
app.IMEPEditFieldLabel.Text = 'IMEP';

% Create IMEPEditField
app.IMEPEditField = uieditfield(app.UIFigure, 'numeric');
app.IMEPEditField.Position = [460 291 100 22];

% Create BMEPEditFieldLabel
app.BMEPEditFieldLabel = uilabel(app.UIFigure);
app.BMEPEditFieldLabel.HorizontalAlignment = 'right';
```

```
app.BMEPEditFieldLabel.FontWeight = 'bold';
app.BMEPEditFieldLabel.Position = [404 259 41 22];
app.BMEPEditFieldLabel.Text = 'BMEP';

% Create BMEPEditField
app.BMEPEditField = uieditfield(app.UIFigure, 'numeric');
app.BMEPEditField.Position = [460 259 100 22];

% Create BSFCEditFieldLabel
app.BSFCEditFieldLabel = uilabel(app.UIFigure);
app.BSFCEditFieldLabel.HorizontalAlignment = 'right';
app.BSFCEditFieldLabel.FontWeight = 'bold';
app.BSFCEditFieldLabel.Position = [407 195 38 22];
app.BSFCEditFieldLabel.Text = 'BSFC';

% Create BSFCEditField
app.BSFCEditField = uieditfield(app.UIFigure, 'numeric');
app.BSFCEditField.Position = [460 195 100 22];

% Create MechanicalEfficiencyEditFieldLabel
app.MechanicalEfficiencyEditFieldLabel = uilabel(app.UIFigure);
app.MechanicalEfficiencyEditFieldLabel.HorizontalAlignment = 'right';
app.MechanicalEfficiencyEditFieldLabel.FontWeight = 'bold';
app.MechanicalEfficiencyEditFieldLabel.Position = [315 161 130 22];
app.MechanicalEfficiencyEditFieldLabel.Text = 'Mechanical Efficiency';

% Create MechanicalEfficiencyEditField
app.MechanicalEfficiencyEditField = uieditfield(app.UIFigure, 'numeric');
app.MechanicalEfficiencyEditField.Position = [460 161 100 22];

% Create FuelConversionEfficiencyEditFieldLabel
app.FuelConversionEfficiencyEditFieldLabel = uilabel(app.UIFigure);
app.FuelConversionEfficiencyEditFieldLabel.HorizontalAlignment = 'right';
app.FuelConversionEfficiencyEditFieldLabel.FontWeight = 'bold';
app.FuelConversionEfficiencyEditFieldLabel.Position = [288 126 157 22];
app.FuelConversionEfficiencyEditFieldLabel.Text = 'Fuel Conversion Efficiency';

% Create FuelConversionEfficiencyEditField
app.FuelConversionEfficiencyEditField = uieditfield(app.UIFigure, 'numeric');
app.FuelConversionEfficiencyEditField.Position = [460 126 100 22];

% Create ThermalEfficiencyEditFieldLabel
```

```
app.ThermalEfficiencyEditFieldLabel = uilabel(app.UIFigure);
app.ThermalEfficiencyEditFieldLabel.HorizontalAlignment = 'right';
app.ThermalEfficiencyEditFieldLabel.FontWeight = 'bold';
app.ThermalEfficiencyEditFieldLabel.Position = [334 90 111 22];
app.ThermalEfficiencyEditFieldLabel.Text = 'Thermal Efficiency';

% Create ThermalEfficiencyEditField
app.ThermalEfficiencyEditField = uieditfield(app.UIFigure, 'numeric');
app.ThermalEfficiencyEditField.Position = [460 90 100 22];

% Create CalculateButton
app.CalculateButton = uibutton(app.UIFigure, 'state');
app.CalculateButton.ValueChangedFcn = createCallbackFcn(app,
@CalculateButtonValueChanged, true);
app.CalculateButton.Text = 'Calculate';
app.CalculateButton.BackgroundColor = [0.851 0.3255 0.098];
app.CalculateButton.FontSize = 20;
app.CalculateButton.FontWeight = 'bold';
app.CalculateButton.FontAngle = 'italic';
app.CalculateButton.Position = [84 59 142 53];

% Create ENGINEINPUTSLabel
app.ENGINEINPUTSLabel = uilabel(app.UIFigure);
app.ENGINEINPUTSLabel.BackgroundColor = [0.902 0.902 0.902];
app.ENGINEINPUTSLabel.HorizontalAlignment = 'center';
app.ENGINEINPUTSLabel.FontSize = 18;
app.ENGINEINPUTSLabel.FontWeight = 'bold';
app.ENGINEINPUTSLabel.Position = [51 430 176 31];
app.ENGINEINPUTSLabel.Text = 'ENGINE INPUTS';

% Create OUTPUTSLabel
app.OUTPUTSLabel = uilabel(app.UIFigure);
app.OUTPUTSLabel.BackgroundColor = [0.902 0.902 0.902];
app.OUTPUTSLabel.HorizontalAlignment = 'center';
app.OUTPUTSLabel.FontSize = 18;
app.OUTPUTSLabel.FontWeight = 'bold';
app.OUTPUTSLabel.Position = [444 421 127 41];
app.OUTPUTSLabel.Text = 'OUTPUTS';

% Create BrakePowerkWEditFieldLabel
app.BrakePowerkWEditFieldLabel = uilabel(app.UIFigure);
app.BrakePowerkWEditFieldLabel.HorizontalAlignment = 'right';
app.BrakePowerkWEditFieldLabel.FontWeight = 'bold';
app.BrakePowerkWEditFieldLabel.Position = [337 59 108 22];
app.BrakePowerkWEditFieldLabel.Text = 'Brake Power (kW)';
```

```
        % Create BrakePowerkWEditField
        app.BrakePowerkWEditField = uieditfield(app.UIFigure, 'numeric');
        app.BrakePowerkWEditField.Position = [460 59 100 22];

        % Create FMEPEditFieldLabel
        app.FMEPEditFieldLabel = uilabel(app.UIFigure);
        app.FMEPEditFieldLabel.HorizontalAlignment = 'right';
        app.FMEPEditFieldLabel.FontWeight = 'bold';
        app.FMEPEditFieldLabel.Position = [406 230 39 22];
        app.FMEPEditFieldLabel.Text = 'FMEP';

        % Create FMEPEditField
        app.FMEPEditField = uieditfield(app.UIFigure, 'numeric');
        app.FMEPEditField.Position = [460 230 100 22];

        % Show the figure after all components are created
        app.UIFigure.Visible = 'on';
    end
end

% App creation and deletion
methods (Access = public)

    % Construct app
    function app = tezson

        % Create UIFigure and components
        createComponents(app)

        % Register the app with App Designer
        registerApp(app, app.UIFigure)

        if nargout == 0
            clear app
        end
    end

    % Code that executes before app deletion
    function delete(app)
```

```
            % Delete UIFigure when app is deleted
            delete(app.UIFigure)
        end
    end
end
```

MATLAB CODES FOR BSFC GRAPH

Bsfc Annand Function

```
function[BSFC,T_ac,eta_f]=BSFCAnnand(RPM,Load)
%University Of Idaho Engine Simulation
%Uses "Two Zone" Combustion Analysis With Variable Specific Heats Ratios
%Only Models The Compression And Expansion Strokes
%__________________________________________________________________________
% clear all;
% close all;
% clc;
%__________________________________________________________________________
%Engine Inputs
%Load = 1;                       %Engine Load (Affects Inlet Pressure)
%RPM = 7600;                     %Revolutions Per Minute [1/min]
L = (63/1000);                   %Stroke of Engine [m]
B = (71/1000);                   %Bore of Engine [m]
l = .09;                        %Length of Engine Connecting Rod [m]
N_cyl = 1;                       %Number of Cylinders [unitless]
C_r = 9.6;                         %Compression Ratio [unitless]
N_r = 2;                         %Number of Revolutions Per Power Stroke
theta_b = 85;                    %Combustion Burn Duration [degrees]
theta_0 = 145;                   %Crank Angle At Start of Combustion
[degrees]
k=1;                             %EGR off
if Load==1
    theta_0=ceil(-.0013*RPM+154.82);
    theta_b=ceil(.0038*RPM+40);
end
if Load<1
    theta_0=ceil(-.0013*RPM+154.82)-(10-Load*10);
    theta_b=ceil(.0038*RPM+40)-(10-Load*10);
end

theta_f = theta_0+theta_b;       %Final Comb. Angle [degrees]
IVC = 0;                         %Time [degrees] when Intake Valve Closes
EVO = 314;                       %Time [degrees] when Exhaust Valve Opens

%__________________________________________________________________________

%Engine Calculations Based On Previous Inputs
%Assumes Average Surface Area In Which Heat Transfer Occurs
A_p = (pi/4)*B^2;                %Cross Sectional Piston Area [m^2]
A_ch = 2*A_p;                    %Cylinder Head Surface Area (in chamber)
V_d = N_cyl*A_p*L;               %Displaced Volume Of Engine [m^3]
N = RPM/60;                      %Converts RPM to RPS [1/s]
S_bar_p = 2*L*N;                 %Calculates Mean Piston Speed [m/s]
a = L/2;                         %Calculates Crank Radius (1/2 stroke)[m]
V_TDC = (V_d/(C_r-1))/N_cyl;     %Calculates Clearance Volume [m^3]
V_BDC = (V_d/N_cyl)+V_TDC;       %Cyl. Volume At BDC [m^3]
%__________________________________________________________________________

%Calculating Losses Due To Friction
%fmep (obtained from Blair) Based On Displacement, RPM

% if V_d>500*10^(-6)
```

```
%     fmep=(100000+350*L*RPM)*10^(-3);
% end
% if V_d<500*10^-6
%     fmep=(100000+100*(500-V_d*10^(-6))+350*L*RPM)*10^(-3);
% end
%For Motorcycles, Use "Rolling" Bearings (For Automobiles, Use Previous)
fmep = (250*L*RPM)*10^-3;

%Volumetric Efficiency Correction Factor
CF = correction( Load,RPM );

%Initial Preallocation Of Matrices (Second Preallocation In Loops Needs To
%Be Included (Do Not Delete)
V(1:360)=zeros;DV(1:360)=zeros;rho(1:360)=zeros;mu(1:360)=zeros;
C_k(1:360)=zeros;C_R(1:360)=zeros;X(1:360)=zeros;M_F(1:360)=zeros;
DX(1:360)=zeros;Re(1:360)=zeros;Nus(1:360)=zeros;h_g(1:360)=zeros;
DQ_w(1:360)=zeros;DQ(1:360)=zeros;Q(1:360)=zeros;DT(1:360)=zeros;
DP(1:360)=zeros;P(1:360)=zeros;T(1:360)=zeros;W_dot(1:360)=zeros;
W(1:360)=zeros;T_indicated(1:2)=zeros;Q_dot(1:360)=zeros;u(1:360)=zeros;
du(1:360)=zeros;cv(1:360)=zeros;m_b(1:360)=zeros;m_u(1:360)=zeros;
V_u(1:360)=zeros;V_b(1:360)=zeros;T_u(1:360)=zeros;T_b(1:360)=zeros;
A_u(1:360)=zeros;A_b(1:360)=zeros;DT_u(1:360)=zeros;gamma_u(1:360)=zeros;
u_u(1:360)=zeros;du_u(1:360)=zeros;cv_u(1:360)=zeros;DQ2(1:360)=zeros;
DQ_w2(1:360)=zeros;Q2(1:360)=zeros;

%_______________________________________________________________________
%Fuel Inputs/Efficiencies
AF_ratio_stoich = 15.09;    %Theoretical Air Fuel Ratio (gravimetric)
% lambda = .85;             %Excess Air Coefficient
if Load ==1
    lambda = .85;
end
if Load==.9
    lambda=.925;
end
if Load==.8
    lambda=.95;
end
if Load<.8
    lambda=.95;
end
AF_ratio_ac = lambda*AF_ratio_stoich; %Actual Air Fuel Ratio
AF_ratio_mol_sotich=14.7; %Molar Air_Fuel Ratio (Stoich)
AF_ratio_mol=lambda*AF_ratio_mol_sotich;
LHV = 44.6e6; %Lower Heating Value Of Fuel Mixture [J/kg]
eta_combmax = .95; %Assumed MAX COmb. Efficiency
%Predicts Combustion Efficiency (Reference To Blair)
eta_comb=eta_combmax*(-1.6082+4.6509*lambda-2.0764*lambda^2);
%Atmospheric Inputs
P_atm = (101325);
T_atm = 278;
P_BDC = Load*P_atm; %Inlet Pressure[Pa] Moscow,ID
R_air = 287; %Gas Constant For Air [J/kg-K]
gamma(1:360) = 1.4; %Preallocate Gamma Array (sets initial value)
T_w =350; %Assumed Wall Temperature (Reference Stone)

%______________________________________________________________________

%Polynomials Used To Calculate Gamma As A Function Of RPM
a_1 = .692; a_2 = 39.17e-06; a_3 = 52.9e-09; a_4 = -228.62e-13;
```

```
a_5 = 277.58e-17;b_0 = 3049.33; b_1 = -5.7e-02; b_2 = -9.5e-05;
b_3 = 21.53e-09;b_4 = -200.26e-14;c_u = 2.32584; c_r = 4.186e-03;
d_0 = 10.41066; d_1 = 7.85125; d_3 = -3.71257;e_0 = -15.001e03;
e_1 = -15.838e03; e_3 = 9.613e03;f_0 = -.10329; f_1 = -.38656;
f_3 = .154226; f_4 = -14.763; f_5 = 118.27; f_6 = 14.503;
r_0 = -.2977; r_1 = 11.98; r_2 = -25442; r_3 = -.4354;

%______________________________________________________________________
R=R_air/1000;
if k==1
    T_BDC = T_atm; %Assumed Inlet Temperature [K]
else
    T_BDC=T_corr;
end
%Calculate Mass of Air In Cylinder/ Mass Of Fuel Based On AFR
rho_a = P_BDC/(R_air*T_BDC); %Air Density kg/m^3
m_a = rho_a*V_d; %Mass of Air In Cylinder [kg]
m_f = m_a/AF_ratio_ac; %Mass Of Fuel In Cylinder [kg]
m_c = m_a+m_f; %Mass In Cylinder
%Specifying Initial Conditions For Loops
%DV,DX,etc. Are Relative To Change In Theta (i.e. DV/Dtheta)
theta(1:360)=zeros;                          %Starting Crank Angle [deg]
V(1:360)=zeros;                              %Preallocate Volume Array
V(1)=V_BDC;                                  %Starting Combustion Chamber
Volume [m^3]
DV(1:360) = zeros;                           %Preallocate Change In Volume
Array
DV(1) = 0;                                   %Specifying Initial Change In
Volume [m^3}
P(1:360)=P_BDC;                              %Preallocate Pressure Array
DP(1:360) = zeros;                           %Specifying Initial Change In
Pressure
T(1:360)=zeros;                              %Preallocate Temperature Array
T(1) = T_BDC;                                %Inlet Temperature [K]
T_u(1)=T_BDC;                                %Initial Unburned
Temperature[K]
DT(1:360) = zeros;                           %Specifying Initial Change In
Temperature
DT_u(1:360)=zeros;                           %Preallocate Change In Unburned
Temperature
gamma(1)=1.4;                                %Initial Gamma Input
gamma_u(1)=1.4;                              %Initial Gamma Input
X(1:360) = 0;                                %Preallocate Mass Burn Array
DX(1:360) = zeros;                           %Preallocate Change In Mass
Burn Fraction [unitless]
DQ(1:360) = zeros;                           %Preallocate Heat Release Array
DQ2(1:360)=zeros;                            %Preallocate Two Zone Heat
Release Array
Q(1:360)=zeros;                              %Preallocate Heat Array
Q2(1:360)=zeros;                             %Preallocate 2 zone Heat Array
M_F(1:360) = 0; %Preallocate Mass In Comubstion Chamber Array
rho(1:360) = zeros; %Preallocates Ideal Gas Law array
rho(1) = P(1)/(R_air*T(1)); %Initial Value Ideal Gas Array
mu(1:360)=zeros; %Preallocate Viscosity Array
mu(1)=7.457*10^(-6)+4.1547*10^(-8)*T_BDC-7.4793*10^(-12)*T_BDC^(2);
C_k(1:360)=zeros; %Preallocate Thermal COnductivity Array
C_k(1) = 6.1944*10^(-3)+7.3814*10^(-5)*T_BDC-1.2491*10^(-8)*T_BDC^(2);
C_R(1:360) = zeros; %Preallocate Radiation Coefficient Array
C_R(1) = 4.25*10^(-09)*((T(1)^4-T_w^4)/(T(1)-T_w)); %Initial Rad. Coeff
Re(1:360)=zeros; %Preallocate Reynolds Value Array
Re(1)=rho(1)*S_bar_p*B/mu(1); %Initial Reynolds Value
```

```
Nus(1:360)=zeros; %Preallocating Nusselt Number Array
Nus(1)=.49*Re(1)^(.7); %Initial Nusselt Number
h_g(1:360)=zeros; %Preallocate Heat Transfer Coefficient Array
h_g(1)=C_k(1)*Nus(1)/B; %Initial Heat Transfer Coefficient
s(1:360)=zeros; %Preallocates Distance Crank/Piston Axes Array
s(1) = -a*cosd(theta(1))+sqrt(l^2 - a^2*sind(theta(1))^2);%Initial Val.
W(1:360) = zeros; %Preallocate Work Array
W_dot(1:360) = zeros; %Preallocate Power Array
T_indicated(1:360) = zeros; %Preallocate Torque Array
Q_dot(1:360) = zeros; %Preallocate Heat Transfer Array
u(1:360) = zeros; %Preallocate Internal Energy Array
du(1:360) = zeros; %Preallocates Change In Internal Energy Array
cv(1:360) = zeros; %Preallocates Heat Capacity Array
DQ_w(1:360)=zeros; %Preallocate Convective Heat Loss Array
DQ_w2(1:360)=zeros; %Preallocate Convective Heat Loss Array 2 zone
m_b(1:360)= zeros; %Preallocate mass burned array
m_u(1:360)=m_c; %Preallocate unburned mass array
V_u(1:360)=zeros; %Preallocate unburned Volume Array
V_u(1) = V(1); %Initial Unburned Volume

%_______________________________________________________________________
theta=1:360;
for i = 2:360
    %Specifies Distance Between Crank/Piston Axes As A Function Of theta
    s = -a*cosd(theta(i))+sqrt(l^2 - a^2*sind(theta(i))^2);
    %Specifies Volume As A Function Of Crank Angle
    V(i) = V_TDC +((pi/4)*B^2)*(l + a - s);
    %Specifies Change In Volume As A Function Of Crank Angle
    DV(i) = V(i)-V(i-1);
    %Calculates Density As A Function Of Crank Angle
    rho(i) = P(i-1)/(R_air*T(i-1));
    %Calculates Viscosity As A Function Of Temperature
    mu(i)=7.457*10^(-6)+4.1547*10^(-8)*T(i-1)-7.4793*10^(-12)*T(i-1)^(2);
    %Calculating Instantaneous Thermal Conductivity of Cylinder Gas
    C_k(i) = 6.1944*10^(-3)+7.3814*10^(-5)*T(i-1)-1.2491*10^(-8)*T(i-
1)^(2);
    %Calculating The Radiation Heat Transfer Coefficient
    C_R(i) = 4.25*10^(-09)*((T(i-1)^4-T_w^4)/(T(i-1)-T_w));
    %Instantaneous Suface Area (For Heat Transfer)
    A = A_ch + A_p + pi*B*(l+a-s);
    if i<=2
        A_u=A;
    end
    %_____________________________________________________________________

    %Specifies Mass Fraction Burn As A Function Of Crank Angle (Weibe Fcn.)
    %Also Specifies Mass Of Fuel In Combustion Chamber As A Function Of
Theta
    if theta(i)<theta_0
        X(i)=0;
    else
        X(i) = 1-exp(-5*((theta(i)-theta_0)/theta_b)^3);
        if theta(i) < theta_f
            M_F(i) = V(theta_0-1)*rho(theta_0-1)/(lambda*AF_ratio_mol);
        end
    end
    %Specifies Change In Mass Fraction Burn As A Function Of Crank Angle
    DX(i) = X(i) - X(i-1);
    %____________________________________________________________________

    %Incorporating The Annand Method To Predict Heat Transfer
```

```
    %Calculating Reynolds Number
    Re(i)=rho(i)*S_bar_p*B/mu(i);
    %Calculating Nusselt Number (constant=.26 two stroke, .49 4 stroke)
    Nus(i)=.49*Re(i)^(.7);
    %Calculating Heat Transfer Coefficient Using Annand Method
    h_g(i)=C_k(i)*Nus(i)/B;
    %Calculates Convective Losses Into Wall As A Function Of Crank Angle
    DQ_w(i) = (h_g(i)+C_R(i))*A*(T(i-1)-T_w)*(60/(360*RPM));
    %Calculates Change In Heat Transfer (total) As A Function Of Crank
Angle
    DQ(i) = eta_comb*LHV*M_F(i)*DX(i)-DQ_w(i);
    %Calculates Total Heat Transfer (Per Cycle)
    Q(i) = Q(i-1)+DQ(i);
    %___________________________________________________________________
    %Specifies Pressure and Temperature Increases Between Intake Valve
    %Closing and Exhaust Valve Opening
    if IVC< theta(i)
        DT(i)=T(i-1)*(gamma(i-1)-1)*((1/(P(i-1)*V(i-1)))*DQ(i)-(1/V(i-
1))*DV(i));
        DP(i)=(-P(i-1)/V(i-1))*DV(i)+(P(i-1)/T(i-1))*DT(i);
        P(i) = P(i-1)+DP(i);
    end
    if EVO < theta(i)
        P(i) = P_atm;
    end
    if 200 < theta(i)
        if P(i)<=P_atm
            P(i)=P_atm;
        end
    end
    %___________________________________________________________________
    %Calculate Burned, Unburned Mass Fractions
    m_b(i) = m_b(i-1)+DX(i)*m_c; %Burned Mass
    m_u(i) = m_u(i-1)-DX(i)*m_c; %Unburned Mass
    %Calculating Burned, Unburned Volumes
    if theta(i)<=theta_0
        V_u(i)=N_cyl*V(i);
    end
    if theta(i)>theta_0
        V_u(i)=((m_u(i)*V_u(i-1))/m_u(i-1))*(P(i)/P(i-1))^(-1/gamma_u(i-
1));
    end
    V_b(i)=N_cyl*V(i)-V_u(i);
    if V_b(i)<0
        V_b(i)=0;
    end
    %Calculating Burned, Unburned Temperatures
    T_u(i)=P(i)*V_u(i)/(m_u(i)*R*1000);
    if theta(i) <= theta_0+4
        T_b(i)=0;
    end
    if theta(i)>theta_0+4
        T_b(i)=P(i)*V_b(i)/(m_b(i)*R*1000);
    end
    %Calculate Unburned, Burned Areas Based On Volume Ratio
    A_u(i)=A*(1-sqrt(X(i)));
    A_b(i)=A*(X(i)/sqrt(X(i)));
    DT_u(i)=T_u(i)-T_u(i-1);

    %___________________________________________________________________
    %Returns Temperature Values To Beginning Of Loop
```

```
    %Assumes Temperature Drops Back To ATM Temp After Exhaust Is Extracted
    T(i) = T(i-1)+DT(i);
    %Calculate The Residual Gas Fraction Assume A Polytropic Constant Of
1.3
    R_frac = (1/C_r)*(P_BDC/P_atm)^(1/1.3)*(1/lambda);
    %Calculates Cylinder Work [J] As A Function Of Crank Angle
    %Treats Atmospheric Pressure As Reference State
    W(i) = W(i-1)+(P(i)-P_atm)*DV(i);
    %Calculates Power [kW] As A Function Of Crank Angle
    W_dot(i)=(N_cyl*W(i)*N/N_r)/1000;
    %Indicated Mean Effective Pressure
    imep = CF*W_dot(360)*N_r*1000/(V_d*1000*N);
    %imep = ((-6E-05)*RPM^2) + (0.4603*RPM) + 1163.2;
    %Calculates Torque[N*m] As A Function Of Crank Angle
    T_indicated(i) = (W_dot(i)*1000)/(2*pi*N);
    %Calculates Heat Loss [kW] As A Function Of Crank Angle
    Q_dot(i) = (N_cyl*Q(i)*N/N_r)/1000;
    %______________________________________________________________________
    % The Following Section Of Code Calculates An Updated Value Of Gamma
    % Using The "Polynomial Method" Developed By Krieger-Borman
    % User Of This Code Must Be Careful Because Accuracy Of This Method
    % Drops As The Fuel Mixture Becomes Increasingly Rich

    %Calculates A,B Factors For Following Block Of Code
    A_t = a_1*T(i)+a_2*T(i)^2+a_3*T(i)^3+a_4*T(i)^4+a_5*T(i)^5;
    A_tu = a_1*T_u(i)+a_2*T_u(i)^2+a_3*T_u(i)^3+a_4*T_u(i)^4+a_5*T_u(i)^5;
    B_t = b_0+b_1*T(i)+b_2*T(i)^2+b_3*T(i)^3+b_4*T(i)^4;
    B_tu = b_0+b_1*T_u(i)+b_2*T_u(i)^2+b_3*T_u(i)^3+b_4*T_u(i)^4;
    %Calculates Factor "D" As A Function Of lambda
    D_lambda = d_0 + d_1*lambda^(-1)+ d_3*lambda^(-3);
    %Calculates Factor "F" As A Function Of Temperature,lambda
    E_TLambda = (e_0 + e_1*lambda^(-1)+ e_3*lambda^(-3))/T(i);
    E_TLambdau = (e_0 + e_1*lambda^(-1)+ e_3*lambda^(-3))/T_u(i);
    F_TPLambda = (f_0 + f_1*lambda^(-1) + f_3*lambda^(-3) + ...
        ((f_4 + f_5*lambda^(-1))/T(i)))*log(f_6*P(i));
    F_TPLambdau = (f_0 + f_1*lambda^(-1) + f_3*lambda^(-3) + ...
        ((f_4 + f_5*lambda^(-1))/T_u(i)))*log(f_6*P(i));
    %Calculates Correction Factor For Internal Energy
    u_corr = c_u*exp(D_lambda +E_TLambda + F_TPLambda);
    u_corr_u=c_u*exp(D_lambda +E_TLambdau + F_TPLambdau);
    %Calculates Internal Energy As A Function Of Crank Angle
    u(i) = A_t - B_t/lambda + u_corr;
    u_u(i) = A_tu - B_tu/lambda + u_corr_u;
    %Calculates Change In Internal Energy
    du(i) = u(i) - u(i-1);
    du_u(i) = u_u(i) - u_u(i-1);
    %Calculates Heat Capacity "C_v" As A Function Of Crank Angle
    cv(i) = du(i)/DT(i);
    cv_u(i)=du_u(i)/DT_u(i);
    %Calculates Correction Factor For "R" Value As A Function Of Crank
Angle
    R_corr = c_r*exp(r_0*log(lambda) + (r_1+r_2/T(i) + ...
        r_3*log(f_6*P(i)))/lambda);
    R_corr_u = c_r*exp(r_0*log(lambda) + (r_1+r_2/T_u(i-1) + ...
        r_3*log(f_6*P(i)))/lambda);
    %Calculates Actual "R" Value
    R = .287 + .020/lambda + R_corr;
    R_u = .287 + .020/lambda + R_corr_u;
    %Calculates Actual Gamma Value And Returns To Beginning Of Code
    gamma_u(i)=1+R_u/cv_u(i);
    gamma(i) = 1 + R/cv(i);
```

```
    if gamma(i)<1.2
        gamma(i)=1.4;
        gamma_u(i)=1.4;
    end
    if theta(i)>=EVO
        gamma(i)=1.4;
        gamma_u(i)=1.4;
    end


%_________________________________________________________________________

    %Calculate Temperature Of Exhaust Based On Polytropic Relations
    if EVO < theta(i)
        T(i)=T(EVO)*(P_BDC/P(EVO))^((gamma(i)-1)/gamma(i));
        T_b(i)=T_b(EVO)*(P_BDC/P(EVO))^((gamma(i)-1)/gamma(i));
    end
    %Calculates A Corrected Inlet Temperature Based On EGR
    %T_corr = R_frac*T(360)+(1-R_frac)*T_BDC;
    T_corr = T_BDC;
end

%_________________________________________________________________________

%Specified Outputs (On Matlab Screen)
W_dot_indicated=W_dot(360);
bmep = imep-fmep;
W_dot_ac = (bmep*V_d*1000*N/(N_r*1000));
T_ac = W_dot_ac/(2*pi*N*10^(-3));
%Calculated Mechanical Efficiency (Based On Previous Inputs)
eta_m = bmep/imep; %Calculates Mechanical Efficiency

%_________________________________________________________________________

%Calculates Brake Specific Fuel Consumption
m_ta = P_BDC*V_d/(R_air*T_BDC); %Calculate Trapped Air In Cylinder
eta_v = CF*((m_ta)/(rho_a*V_d)); %Corrected Volumetric Efficiency
m_dot_f = N_cyl*M_F(theta_0)*(N/N_r); %Mass Flow Rate Of Fuel
m_dot_a = AF_ratio_ac*m_dot_f; %Mass Flow Rate Of Air
BSFC = (m_dot_f*1000*3600)/(W_dot_ac); %BSFC [g/kW*h]
eta_f = 3600/(BSFC*(LHV*10^(-6))); %Fuel Conversion Efficiency
% %Calculate Emissions
% T_NO=.875*max(T_b); %Calculate Avg. Burn Temp
% P_NO=max(P); %Assuming Pressure is peak
% P_EXH=(P(EVO)+P_atm)/2; %Calculating Exhaust Press.
% [ PPM_NO ] = NOX( T_NO,P_atm,lambda,P_NO,T_BDC,P_BDC,P_EXH);
% P_peak = max(P); %Peak Pressure
% %disp('Percentage of Fuel Mass Reaching Exhaust')
% [ HC ] =
hydrocarbons(R_frac,AF_ratio_ac,B,P_peak,imep,C_r,V_d,N_cyl,T_w,N );
% HC = 100*HC;
end
```

Correction and Load Function

```
function [ CF ] = correction( Load,RPM )
if Load<=1
    %CF = (-3*10^(-9))*RPM^2+5*10^(-5)*RPM+.7088;
    CF=-8*10^(-9)*RPM^2+.000135*RPM+.31944;
    if Load<=.9
        CF=-8*10^(-9)*RPM^2+.000135*RPM+.31944;
        %CF = CF-(1-Load)/4;
    end
end
end
```

```
function [BSFC,T_ac,N_RPM,eta_f]=LOAD2(Load)
RPM = 1000; %Sets Starting Pt. For Loop
N_RPM = 10; %Fifteen RPM Data Sets
BSFC(1:N_RPM)=zeros; %Preallocate Array
T_ac(1:N_RPM)=zeros; %Preallocate Array
PPM_NO(1:N_RPM)=zeros;
for i = 1:N_RPM
    RPM = RPM+1000;
    [BSFC(i),T_ac(i),eta_f(i)]=BSFCAnnand(RPM,Load);
end
end
```

Egg Curve Function

```
%engine data for bsfc map
%L=142;                                      %Connecting rod length
%Vd38=0.0030;                                %Displacment volume
%Vd1c=0.0006;                                %Displacement volume per
cylinder
%Vc=6.660762856239030e-05;                   %Clearance volume at TDC
%B=0.0929;
%Upavg=10.84;                                %Avarage piston spd
%eta_mech=0.85;                              %mechanical efficiency cycle
%Nc=6 ;
%cst=60;                                     %for conversion
%a=0.5;                                      %constant
%Ap=0.006778;                                %For one cylindr
%Pin=183.26;                                 %inlet manifold pressure Kpa
%rc=9;                                       %compression ratio
```

```
%Tin=333;                                      %manifold inlet temp
%R=0.287;                                      %Global air const
%ma=0.0037;                                    %Air mass in
%mf=2.459542203338950e-04;                     %kg/cylinder/cycle
%Hu=44000;                                     %Kg/kj
%P1=560.26;
%P2=0.7404938;
%P3=1.021125e-3;
%omega=375:1:775;
%Te=(P1+(P2*omega)-(P3*omega.^2));             %engine torque
%N=3600:10:7600;                               %engine RPM

%Calculations
%for i=1:1:400
%    for j=1:1:400
%        Wb_dot(i,j)=(2*pi*N(i)*Te(j))*10^-5;
%        bmep(j)=((4*pi*Te(j))/Vd38)*0.001;
%        imep(j)=bmep(j)/eta_mech;
%        fmep(j)=imep(j)-bmep(j);
%        Wf_dot(j)=(1/4)*fmep(j)*Ap*Upavg*Nc;
%        Wb(j)=bmep(j)*Vd1c;
%        mf_dot=(mf*Nc*(N/cst)*a);
%        bsfc=((mf_dot/Wb_dot(i,j))*1000)/0.000277;
%    end
%end

%disp(bsfc);
%disp(Te);
%disp(N);

%jdata=xlsread('JeremyData');
%x = jdata(:,1);
%y = jdata(:,2);
%z = jdata(:,3);
%plot3(x,y,z,'.-')
%tri = delaunay(x,y);
%h = trisurf(tri, x, y, z);
%shading interp
%colorbar
%title('Brake Specific Fuel Consumption (g/kW-hr)')
%xlabel('RPM')
%ylabel('Torque (N*m)')
%zlabel('BSFC')
Load = 1.1;                  %Start High, Decreased By .1 Each Iteration
N_load = 7;                  %Six Loads
%bsfc(1:N_RPM,1:N_load)=zeros;
%Torque(1:N_RPM,1:N_load)=zeros;
%_______________________________________________________________________
%N_RPM=8;
Torque(1:10,1:7)=zeros;
%ETA_f(1:15,1:7)=zeros;
bsfc(1:10,1:7)=zeros;
for j = 1:N_load
    Load = Load-.1;                                                  %Decreases
Load With Each Iteration
    [BSFC,T_ac,N_RPM,eta_f]=LOAD2(Load);                    %Feeds "Load" Into
Load Function
    BSFC=BSFC';                                                      %Transposes
Vector
    bsfc(1:N_RPM,j)=BSFC;                                            %Creates
Matrix Out of BSFC Vectors
```

```
    T_ac=T_ac';                                                    %Transposes
Torque Vector
    Torque(1:N_RPM,j)=T_ac';                                       %Creates
Matrix Out of Torque Vectors
    eta_f=eta_f';
    ETA_f(1:N_RPM,j)=eta_f;
end

%_______________________________________________________________________

%Create RPM Matrix
RPM(1:N_RPM,1:N_load)=zeros;
RPM(1,1)=1000;
for i = 2:N_RPM
    RPM(i,1) = RPM(i-1,1)+1000;
end
for j = 2:N_load
    RPM(1:N_RPM,j)=RPM(1:N_RPM,j-1);
end

%for u=1:15
%     for y=1:7
%         abcde(u,y)=bsfc(u,y)
%     end
%end

rtyu(1,1:7)=10000;
rtyu(2,1:7)=9000;
rtyu(3,1:7)=8000;
rtyu(4,1:7)=7000;
rtyu(5,1:7)=6000;
rtyu(6,1:7)=5000;
rtyu(7,1:7)=4000;
rtyu(8,1:7)=3000;
rtyu(9,1:7)=2000;
rtyu(10,1:7)=1000;
%_______________________________________________________________________
%Plot Statements
% figure(1)
% grid
% contourf(RPM,Torque,bsfc,75)
% xlabel('RPM')
% ylabel('Torque [N*m]')
% title('Nissan GTR BSFC [g/kW-h] MAP')
% colorbar
% colormap jet
% caxis([250 700])
%
figure(2)
grid
contourf(rtyu,Torque,bsfc,50)
xlabel('RPM')
ylabel('Torque [N*m]')
title('HONDA CR250F BSFC [g/kW-h] MAP')
colorbar
colormap jet
caxis([250 700])
%figure(4)
%contourf(RPM,Torque,ETA_f,100)
%xlabel('RPM')
```

```
%ylabel('Torque [N*m]')
%title('Nissan GTR HC Fuel Efficiency Map [%]')
%colorbar
%caxis([0 0.3])
disp(RPM);
```