A THERMODYNAMICS BASED MODEL FOR PREDICTING PISTON ENGINE PERFORMANCE FOR USE IN AVIATION VEHICLE DESIGN

A Dissertation
Presented to
The Academic Faculty

By

Justin L. Highley

In Partial Fulfillment
Of the Requirements for the Degree
Master of Science in Aerospace Engineering

Georgia Institute of Technology
April 2004

A THERMODYNAMICS BASED MODEL FOR PREDICTING PISTON ENGINE
PERFORMANCE FOR USE IN AVIATION VEHICLE DESIGN

Approved By:

Dr. Daniel P. Schrage, Advisor

Dr. Jimmy Tai

Mr. Mark Waters

Date Approved: April 2, 2004

ACKNOWLEDGEMENTS

I would like to thank Dr. Daniel Schrage for encouraging me to pursue a thesis in piston engine analysis, and for providing me with the academic resources necessary to complete the research.

I must also thank Dr. Jimmy Tai for working with me throughout the project and providing guidance and mentorship. Particularly in the early stages of research process, he assisted in the development of my specific thesis topic, and helped me create the road map that laid the foundation necessary for success.

Finally, I must thank Mr. Mark Waters. Without his constant assistance, suggestions, and sanity checks I would have never obtained the level of fidelity and quality I desired for this project. His eagerness to help and genuine interest in my work made this research much easier and more enjoyable.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

B	Bore
BDC	Bottom Dead Center
BMEP	Brake mean effective pressure
BSFC	Brake specific fuel consumption
C_P	Specific heat evaluated at constant pressure
C_v	Specific heat evaluated at constant volume
IMEP	Indicated mean effective pressure
L	Piston Stroke
m	Total fuel-air mixture mass
m_a	Air mass
m_c	Total mass in the cylinder
MEP	Mean Effective Pressure
m_r	Residual mass of the burned fuel-air mixture
N	Engine Speed in RPM
n_c	Number of piston cylinders
n_r	Number of crankshaft revolutions per power cycle
P	Power
$P_{ambient}$	Ambient pressure
P_B	Brake power
P_e	Pressure of the exhaust gases
P_i	Engine inlet pressure

P_1	Initial pressure of the fuel air mixture
P_2	Pressure following compression
P_3	Pressure following combustion
P_4	Pressure following expansion
Q_{hv}	Fuel heating value
Q^*	Specific internal energy loss during isothermal combustion
r_c	Compression ratio
s_e	Entropy of the exhaust gases
SFC	Specific Fuel Consumption
s_1	Initial entropy of the fuel-air mixture
s_2	Entropy following compression
s_3	Entropy following combustion
s_4	Entropy following expansion
TDC	Top Dead Center
T_e	Exhaust gas temperature
TFMEP	Total friction mean effective pressure
T_i	Engine inlet temperature
T_1	Initial temperature of the fuel air mixture
T_2	Temperature following compression
T_3	Temperature following combustion
T_4	Temperature following expansion
$u^{\circ}_{f,u}$	Internal energy of formation
u_s	Sensible internal energy

u_1	Initial internal energy of the fuel-air mixture
u_2	Internal energy following compression
u_3	Internal energy following combustion
u_4	Internal energy following expansion
V_C	Clearance Volume
V_D	Displaced volume
v_e	Specific volume of the exhaust gases
V_T	Total Volume
v_1	Initial specific volume of the fuel-air mixture
v_2	Specific volume following compression
v_3	Specific volume following combustion
v_4	Specific volume following expansion
W_B	Brake work
$W_{Compression}$	Compression work
$W_{Expansion}$	Expansion work
W_F	Friction work
W_I	Indicated work
W_M	Mechanical Work
W_{Net}	Net Work
W_P	Pumping Work
W_{Par}	Parasitic Work
$W_{1\text{-}2}$	Work from state point 1 to state point 2
$W_{3\text{-}4}$	Work from state point 3 to state point 4

x_b	Burned gas fraction
$\sum u^{\circ}_{f,u}$	Summation of the internal energies of formation
G	Ratio of specific heats, C_p/C_v
η_F	Fuel Conversion Efficiency
F	Fuel-air equivalence ratio
$\dot{m}_f$	Fuel mass flow rate
$\left(\dfrac{F}{A}\right)_s$	Stoichiometric fuel to air ratio

SUMMARY

The purpose of this thesis is to develop a thermodynamics based computer program to predict the performance of naturally aspirated spark ignition piston engines. Advances in piston engine technology, coupled with high costs of turbine engines have led many general aviation manufacturers to explore the use of piston engines in their smaller vehicles. However, very few engine models are available to analyze piston engine performance. Consequently, designers using vehicle synthesis programs are unable to accurately predict vehicle performance when piston engines are used. This thesis documents the development of a comprehensive, thermodynamics based performance model that meets that need.

The first part of this thesis details the basics of piston engine operation, including component geometry and the four stroke engine cycle. Next, the author analyzes the critical components of engine performance, including engine work and power. In developing the engine performance model the Ideal Engine Cycles are discussed. The cold air and fuel-air working fluid models are discussed, along with the types of combustion models, including the Otto Cycle, Diesel Cycle, and the Dual Cycle.

Two performance models are generated using the Constant Volume Ideal Engine Cycle: an *Ideal Gas Standard Cycle*, and a *Fuel-Air Cycle*. The *Ideal Gas Standard Cycle* is useful for parametric analysis but lacks the accuracy required for performance calculations. The *Fuel-Air Cycle*, however, more accurately models the engine cycle and is selected as the basis for the computer program.

In developing the computer program the thermodynamic charts used in the *Fuel-Air Cycle* calculations must be reproduced. To accomplish this, the NASA Chemical Equilibrium Application (CEA) program is integrated into a parent VBA based computer code to provide thermodynamic state point data. Finally, the computer program is correlated to the performance of an existing aviation engine to validate the model.

INTRODUCTION

The reciprocating engine is one of the most important inventions of the 19th century. Its versatility, low cost, and durability make it an indispensable part of today's mechanized society. It powers everything from cars, trains, and boats, to lawnmowers and generators, and currently there is a drive to use more piston engines in aviation applications. Presently, only smaller general aviation vehicles use piston engines for propulsion, while larger vehicles rely almost exclusively on turbine engines. However, general aviation manufacturers are beginning to use more piston engines in their larger vehicles. The biggest reason for this is cost: other than avionics, the system that contributes most to a vehicle's price is its propulsion system, and a turbine engine can cost up to five times more than a comparable piston engine.[1] Additionally, recent years have seen large technological advances in piston engine manufacturing, making them lighter, more powerful, and more efficient. Finally, piston engines are known to have greater flexibility with respect to transient power requirements than turbine engines, which not only increases safety, but also performance and efficiency.[2] Because of these factors, many general aviation manufacturers are beginning to pursue piston engines in an attempt to reduce vehicle price and increase the potential marketplace.

Unfortunately, most engine analysis tools in use today are based on turbine engines and cannot model piston engine performance. As a result, researchers oftentimes cannot accurately predict vehicle performance in synthesis programs such as GTPDP

---

[1]Schrage, Daniel P., AE8803 B Class Notes, Sep 11, 2003.
[2]Atzler, Frank. On the Future of the Piston Engine with Internal Combustion: An Overview. Marie Curie Fellowship Conference, May 2001, pg 1.

when piston engines are used. Based on that shortcoming, this thesis will develop a thermodynamics based model to predict piston engine performance for use in aerospace vehicle synthesis programs. The tool will use the thermodynamic cycles of a piston engine, and will create a performance table or engine deck for the engine based on user specified input parameters. This performance table will predict engine performance over a range of flight profiles as defined by altitude and ambient conditions.

SCOPE

Because of the numerous variables associated with engine type, design, and operation, this thesis will focus on four stroke engines only. Additionally, only naturally aspirated (i.e. non-supercharged) spark ignition engines are considered.

METHODOLOGY

In order to create the engine performance model a systematic methodology is required. The methodology developed for this thesis appears in Figure 1.
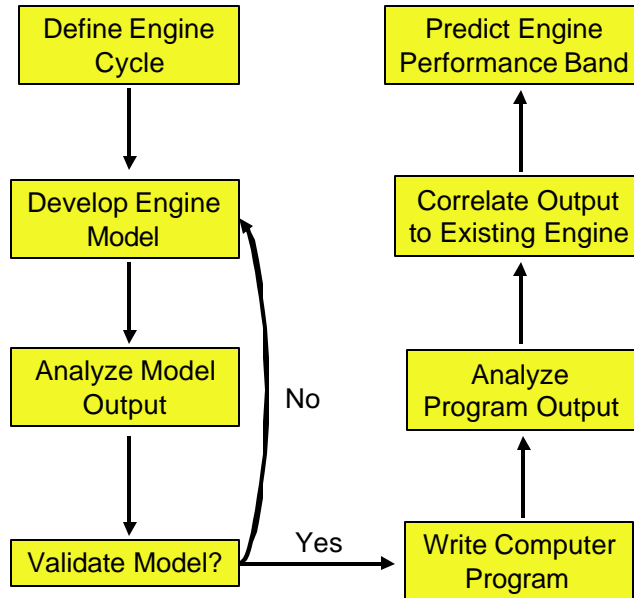
Figure 1: Methodology

This report will follow the steps outlined in the diagram. The initial portion of this report will define the four stroke engine cycle from a thermodynamic perspective. Next, the various engine models will be analyzed and subsequently used to develop a performance model that predicts the engine's power output. The results of this model will be compared to existing engine performance data to determine its accuracy. Once the model is validated, it will be translated to a computer code. The output of this code will be analyzed and correlated to an existing aviation piston engine. The final model will then be used to establish the performance band for specific ambient flight conditions.

PISTON ENGINE BACKGROUND

The first internal combustion engine was invented by Nicolaus Otto in 1876, and it quickly reshaped the world in which he lived. By the late 1880s carburetor and ignition improvements resulted in engine driven automobiles, and in the late 1890s 600 bhp engines were produced.[3] In the mid 20th century, the onset of green house gas effects and fuel shortages placed an emphasis on engine research and development to reduce emissions, increase fuel economy, and decrease costs. While advances in engine technology have helped to achieve these goals and increase performance, the fundamental thermodynamic principles behind the piston engine remain the same as in Otto's day.

Piston Engine Basics

This section addresses the fundamental concepts behind piston engines, including the basic operating principles, standard geometry, and the individual processes involved in the four stroke cycle.

Piston Engine Operation

All reciprocating engines are characterized by a piston that moves back and forth in a cylinder. This piston movement in turn drives a crankshaft, which transmits the

---

[3] Heywood, John B., <u>Internal Combustion Engine Fundamentals</u>, McGraw-Hill, 1988, pg 3.

power to a drive shaft or transmission of some type. The most important component of the engine is the piston / cylinder combination, which is the focus of the thermodynamic analysis. Although the piston is dependant on the crankshaft for movement during the non-power strokes, each piston operates independent of the others. For this reason, the thermodynamic analysis of reciprocating engines is not dependant on the number of cylinders or even engine geometry. While these parameters are extremely important from a structures and materials perspective, they are irrelevant in the performance analysis.

Piston Engine Geometry

The important aspects of piston/cylinder geometry are shown in Figure 2. Each cylinder contains a piston, which is connected to the crankshaft (not shown) via a connecting rod. The cylinder also contains two valves: an intake valve and an exhaust valve. These allow for the induction and expulsion of the fuel-air mixture.
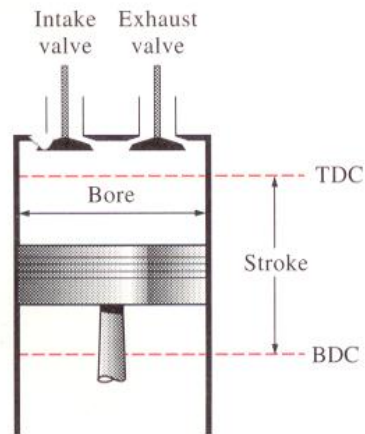
Figure 2: Cylinder/Piston Geometry

Other important aspects of the cylinder/piston geometry shown in Figure 2 are:

Top Dead Center (TDC): This is where the piston comes to rest at the highest point in the cylinder, and is associated with the minimum cylinder volume.

Bottom Dead Center (BDC): Similar to TDC, except this is the lowest piston position and results in the maximum cylinder volume.

Piston Stroke (L): The length the piston travels between BDC and TDC.

Bore (B): Cylinder width.

Based on these parameters one can define several volumes that are important to the engine's operation:

V_T: **Total Volume**. The maximum cylinder volume. Based on the distance between BDC and the cylinder valves.

6

V_C: Clearance Volume. The minimum cylinder volume, which is given by the distance between TDC and the cylinder valves.

V_D: Displaced volume. The amount of gas swept out of the cylinder, given by the difference between the V_T and V_C.

The ratio of the last two volumes, V_D and V_C, is the piston's compression ratio, r_c. These parameters are shown in Figure 3.

Compression Ratio, r_C

$$r_C = \frac{V_{max}}{V_{min}} = \frac{V_D + V_C}{V_C}$$

TDC

BDC

**Displacement
Volume V_D**

**Clearance
Volume V_C**
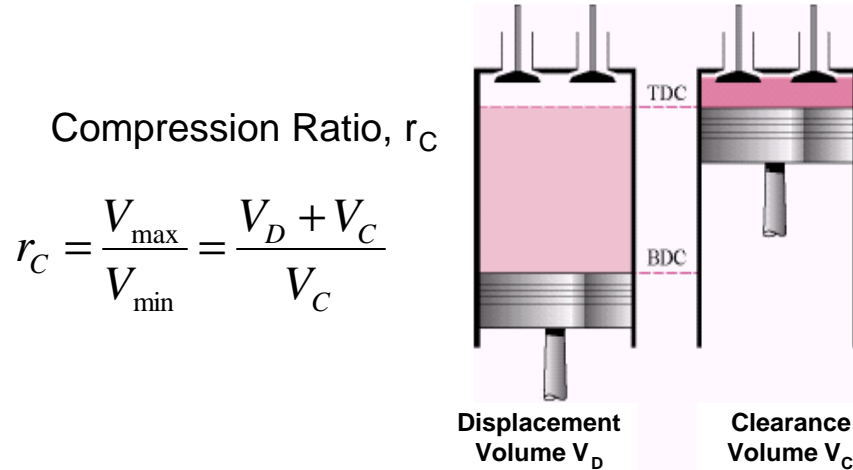
Figure 3: Piston Volumes

Four Stroke Cycle Analysis

Reciprocating engines are categorized by the number of piston strokes required to complete the engine cycle, which is either two or four. This thesis focuses on four stroke engines, as they are the most common engines encountered in aviation applications. The sequence of events that take place in a four stroke engine is shown in Figure 4.
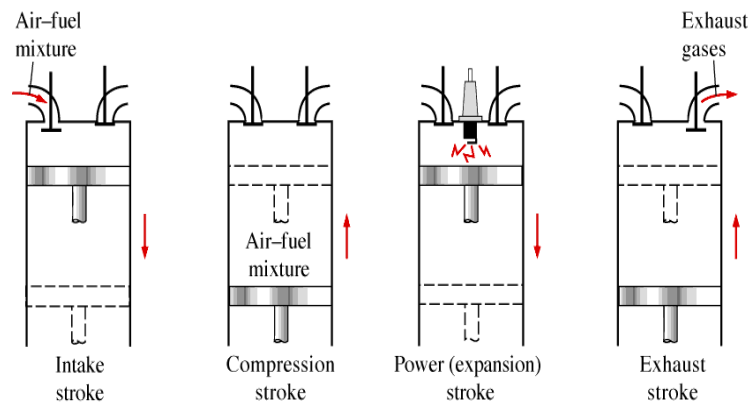
Figure 4: Four Stroke Cycle

As the name indicates, there are four discrete steps or strokes that occur within the cylinder.

Intake Stroke: During this step the intake valve is open. The piston starts at TDC and moves to BDC, which creates a vacuum and sucks the fuel-air mixture into the cylinder. To maximize the mass of the intake charge the intake valve normally opens before the cylinder reaches TDC and closes after BDC.

Compression Stroke: In this step both valves are closed, and the piston moves from BDC to TDC to compress the fuel-air mixture prior to combustion. As the piston nears TDC, combustion is initiated, causing a rapid rise in cylinder pressure.

Power Stroke: Also known as the expansion stroke. The piston begins at TDC and is forced down to BDC by the combustion of the intake mixture. In moving the piston, the high temperature, high pressure gases also rotate the crankshaft, providing compression work to the other cylinders. These gases exert approximately five times the amount of work on the piston as the piston exerted on the gas during the compression stroke.[4]

Exhaust Stroke: Here the exhaust gases are expelled through the exhaust valve, which opens at BDC. Since the cylinder is at a higher pressure than the exhaust outlet, the gases flow freely through the valve. Additional gases are pushed out as the piston travels to TDC. Just prior to TDC the intake valve opens again and the cycle starts over.

Measuring Engine Performance

The goal of an engine performance model is to predict the engine's power output. Since power is a function of work, the engine's work must first be calculated. An engine's work is grouped into two categories: positive work and negative work, which are used to define the engine's brake work. Each of these will be discussed in turn.

---

[4] Ibid, pg 10.

Calculating Engine Work

The positive work produced by the engine has only one component: W_G, which is the gross indicated work produced by the combustion process. The engine's negative work, however, is comprised of three separate components:

W_P: Pumping Work – losses caused by aspiration / expelling of gases.

W_M: Mechanical Work – losses from friction between engine parts.

W_{Par}: Parasitic Work – losses caused by engine driven accessories (generator, oil pump, etc).

The friction work, W_F, represents the engine's total negative work and is the summation of these three losses:

$$W_F = W_P + W_M + W_{Par}$$

Using these parameters one can calculate the net work produced by the engine, also known as its brake work, W_B.

$$W_B = W_G - W_F$$

Brake work is the net work measured at the engine's crankshaft and is normally used when referring to the engine's power output.

Another aspect of engine work is its total indicated work, W_I, which captures both the positive and negative aspects of work. It is obtained from the engine's P-V diagram, which is the most accurate way of finding the engine's work output. The engine's indicated work is the entire area enclosed by the P-V diagram:

$$W_I = \oint pdV = W_G + W_F$$

This concept is shown in Figure 5 below. While only the pumping losses, W_p, are explicitly shown, the other components of the total friction losses are captured by W_G.
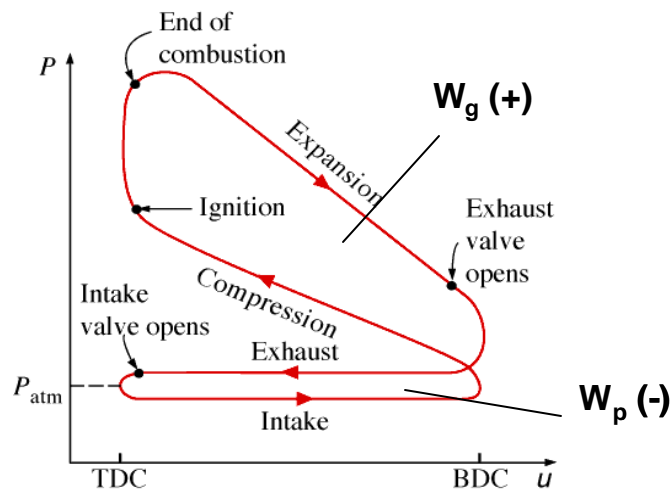
Figure 5: P-V Diagram

Obviously, testing every engine to obtain its P-V diagram is unrealistic. Therefore, one must quantitatively obtain the positive and negative work components in order to predict

engine performance. These work values are found by analyzing the thermodynamic processes that take place in the engine, and will be discussed in a later section.

Calculating Engine Power

As stated previously, the engine's power is a function of the work produced, which is expressed by the following relationship.

$$P = \frac{W N}{n_r}$$

where

W is the engine work, either W_G or W_B (giving P_G or P_B, respectively)

N is the engine speed in RPM

n_r is the number of crankshaft revolutions per power cycle (2 in a four stroke engine)

Based on this equation an engine's power output will theoretically increase with engine speed. In practice, however, engine losses tend to increase exponentially with speed, which serves to limit the available power.

Additional Engine Parameters

When defining engine performance, many parameters are used other than the work and power output. Several of these are defined now, and will be used extensively throughout this report.

Specific Fuel Consumption (SFC): The SFC is a measure of how efficiently the engine uses the supplied fuel to produce work:

$$\text{SFC} = \frac{\dot{m}_f}{P}$$

When power is expressed as the brake power, P_B, then the SFC becomes the brake specific fuel consumption, or BSFC. SFC can also be expressed in dimensionless terms using the Fuel Conversion (Thermal) Efficiency, $?_F$

$$?_{f=}\frac{P}{\dot{m}_f Q_{hv}} = \frac{W}{m_f Q_{hv}} = \frac{1}{sfcQ_{hv}}$$

where

Q_{hv} is the fuel heating value

The fuel conversion efficiency, η_f, can also be defined as the ratio of the work produced per cycle to the amount of energy that can be released into the combustion process per cycle.

Mean Effective Pressure (MEP): The Mean Effective Pressure is the pressure that must be exerted on the piston to produce the same amount of work as the engine cycle:

$$MEP = \frac{W}{V_{max} - V_{min}}$$

Although not an actual engine operating parameter that can be measured, MEP is important because it represents the engine's normalized work and is used to compare engines of different sizes and speeds. As in SFC, using the brake work in the equation yields the brake mean effective pressure, BMEP. Likewise, using the engine's indicated work results in the IMEP. Both of these parameters will be used in the engine performance models developed later in this report.

Fuel-Air Equivalence Ratio: The final parameter is the fuel-air equivalence ratio, Φ, which identifies the unburned mixture's composition. Changes to the engine's fuel-air ratio (lean or rich) have a significant impact on the composition of the combustion products, and a simple ratio of fuel to air is insufficient for describing the properties of the mixture. Therefore, a more robust parameter is required to define the fuel-air

mixture, which leads to the introduction of the equivalence ratio. The equivalence ratio is defined as the ratio of the actual fuel-air ratio to the stoichiometric fuel-air ratio:

$$f = \frac{(F/A)_{actual}}{(F/A)_{stoichiometric}} \qquad \text{(EQ 1)}$$

For fuel lean mixtures, F is < 1, while fuel rich mixtures have F > 1. When F = 1 the mixture is said to be stoichiometric.

IDEAL ENGINE CYCLES

As discussed in the previous section, the best way to measure engine work is through the use of a P-V diagram. However, in vehicle design this is not a viable option and the designer must predict the engine's performance based on a few critical engine parameters. This is done through the use of ideal engine cycles. By dividing the engine operating cycle into a sequence of separate processes (compression, combustion, expansion, and exhaust) and modeling each process, the designer can simulate the complete engine cycle. These simulated engine cycles in turn allow the user to estimate the engine's performance.

When analyzing an ideal cycle, it is important to note that a piston engine is not a closed system, and therefore cannot be considered a heat engine as defined in classical thermodynamics. Rather, a piston engine is an open system that exchanges heat and work with it's environment (the atmosphere). The two reactants in this system are the

fuel/air mixture, which flows into the system, and the exhaust gas byproducts, which flow out.[5] Therefore, the ideal cycles discussed here are a sequence of engine processes wherein the working fluid is analyzed, and are not thermodynamic cycles per se. However, the analyses used within the individual processes are based on thermodynamic principles.

Working Fluid Models

When developing the ideal cycles the designer must determine which model to use for the working fluid within the cylinder. By defining the fluid's thermodynamic properties, the cycle can be simplified using various assumptions. The simplest fluid model uses the cold air standard assumptions, or CASA. Ideal engine models combined with CASA are known as *Ideal Gas Standard Cycles*, and are useful for obtaining analytical results. Another commonly used fluid model consists of a fuel-air mixture whose unburned components are a mixture of frozen Ideal Gases, and whose burned mixture is in chemical equilibrium. This model more accurately represents the actual fluid properties and therefore results in a more reliable engine model. By combining this fluid model with an ideal cycle one obtains a *Fuel-Air Cycle*. This thesis employs both *Ideal Gas Standard Cycles* and *Fuel-Air Cycles.*

The assumptions applicable to the two working fluid models are summarized below.

---

[5] Ibid, pg 162.

Cold Air Standard Assumptions.[6]

- The working fluid in the cylinder is air, which behaves as an Ideal Gas

- All processes within the cycle are internally reversible

- The combustion process is modeled by heat addition

- The exhaust process is modeled by heat rejection

- The working fluid (air) has constant specific heats

Fuel-Air Assumptions.[7,8]

- Unburned Fuel-Air mixture is frozen (no reactions between the fuel and air)

- Burned mixture is in chemical equilibrium above 1700° K; mixture composition is frozen below 1700° K.

- Each species in the mixture behave as an Ideal Gas

- Thermodynamic properties (T, v, u) are obtained from Gas Tables

Combustion Models

Ideal cycles are categorized based on the method used to model the combustion process. The three most common models are Constant Volume (Otto Cycle), Constant Pressure (Diesel Cycle) and Limited Pressure (Dual Cycle). In each cycle, the processes

---

[6] Cengel, Yunus A. and Boles, Michael A., Thermodynamics: An Engineering Approach, McGraw-Hill, 1989, pp. 279-380.
[7] Heywood, pg 113.
[8] Ibid, pg 116.

other than the combustion process remain the same. The assumptions associated with the various cycles are listed here by process.[9]

Ideal Cycle Assumptions by Process:

Compression (1-2) 1. Adiabatic and Reversible (hence isentropic)

Combustion (2-3) 1. Adiabatic

2. Combustion occurs at

a. Constant Volume (Otto Cycle) or

b. Constant Pressure (Diesel Cycle) or

c. Part at constant volume and part at constant pressure (Dual Cycle)

3. Complete (no unburned gases)

Expansion (3-4) 1. Adiabatic and Reversible (isentropic)

Exhaust (4-1) 1. Adiabatic

2. Valve events occur at BDC

3. No changes in cylinder volume as pressure differences across open valves drops to zero

4. Exhaust pressures are constant

5. Velocity effects are negligible

Constant Volume Cycle (Otto Cycle)

This cycle represents the case where the combustion occurs at constant volume, and is therefore infinitely fast. Complete combustion occurs at TDC.

---

[9] Heywood 163
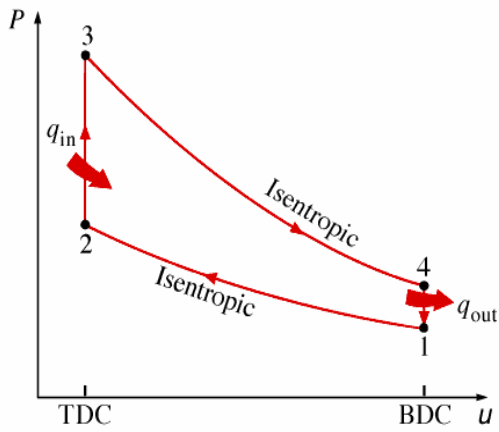
Figure 6: Constant Volume Ideal Cycle

Constant Pressure Cycle (Diesel Cycle)

In this cycle the combustion is modeled as slow and late (continues past TDC).

(a) P-υ diagram
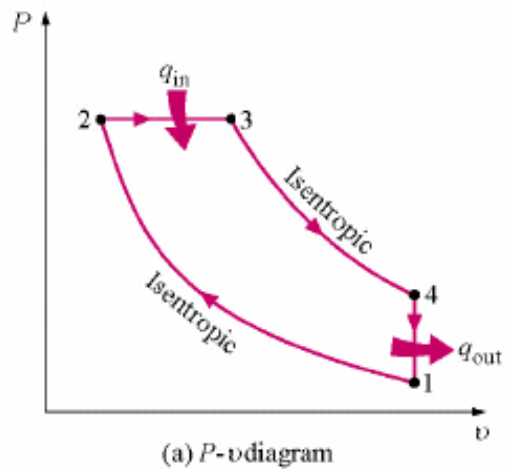
Figure 7: Constant Pressure Ideal Cycle

Limited Pressure Cycle (Dual Cycle)

In the limited pressure cycle, combustion occurs partly at constant volume and partly at constant pressure. It is a combination of the previous two models.
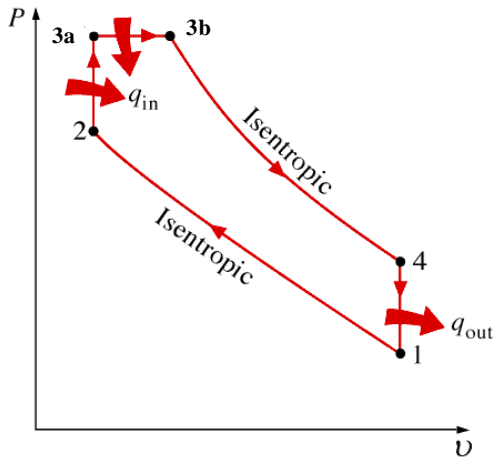
Figure 8: Limited Pressure Ideal Cycle

Comparison of Actual and Ideal Cycles

The differences between the actual and ideal 4-stroke engine cycles appear in Figure 9. For comparison purposes a Constant Volume Ideal Cycle is used.
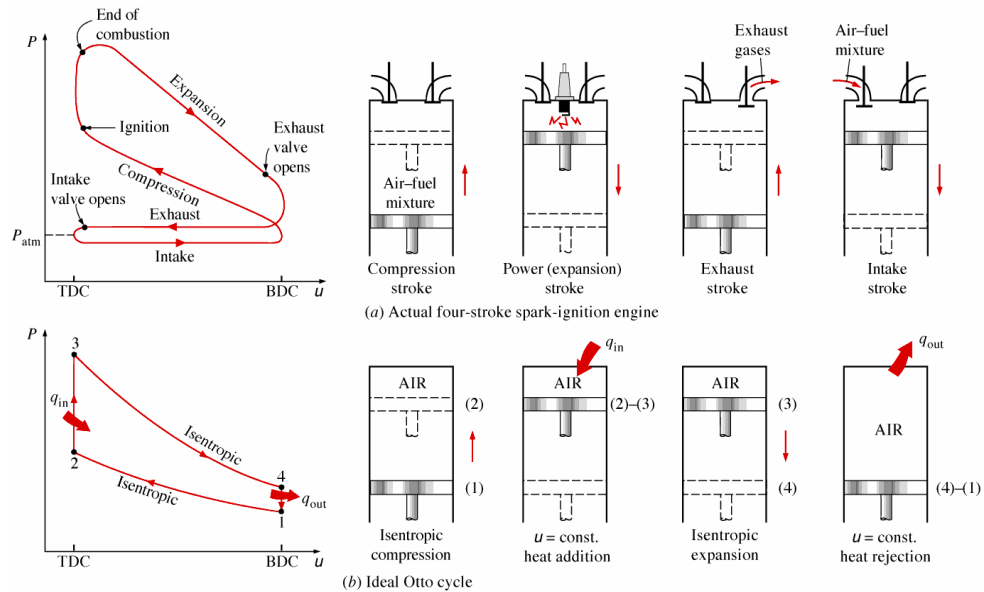
Figure 9: Comparison of Actual and Ideal Engine Cycles

The largest difference between the two diagrams is the simplification of the intake and exhaust strokes in the ideal cycle. For this reason, the ideal cycle does not accurately predict the pumping work, W_P, of the engine. This, combined with the simplifying assumptions made during the modeling process (no heat transfer, complete combustion, etc), lead to the fact that the enclosed area of the P-V diagram for an actual engine is only 0.8 the size of the area enclosed by the P-V diagram of the ideal cycle.[10] In other words, the ideal engine cycle will overestimate the power produced by the actual engine by 25%. This correction will be taken into consideration when obtaining results using these models.

---

[10] Ibid, pg 194.

IDEAL GAS STANDARD CYCLE **PERFORMANCE MODEL**

In order to demonstrate the application of the ideal cycles described here, and to establish a methodology for calculating engine performance, the first performance model uses an *Ideal Gas Standard Cycle*. Specifically, a Constant Volume Ideal Cycle is used. The model provides quantitative results and can predict engine performance based on calculated engine parameters. The methodology used in this analysis follows the working fluid's changes of state through each phase of the engine's operating cycle. By analyzing the state of the fluid, its corresponding thermodynamic properties can be tracked throughout the cycle. These properties, in turn, can be used to calculate the engine's performance.

Methodology

When describing the characteristics of an engine operating cycle, one of the most important operating parameters is the mean effective pressure. In this instance, the indicated mean effective pressure (IMEP) is used, and is calculated with the following equation:

$$IMEP = \left(\frac{Q^*}{C_v T_1}\right)\left(\frac{1}{g-1}\right)\left(\frac{m}{m_a}\right)h_f\, p_1 \qquad \text{(EQ 1)}$$

where

Q* is the specific internal energy produced during isothermal combustion

η_f is the fuel conversion efficiency

m_a is the air mass

m is the total fuel-air mixture mass

and γ is the ratio of specific heats of the fluid, C_p/C_v.

These terms are discussed below.

Specific Internal Energy Loss: The parameter Q* used in this model is the specific internal energy produced during isothermal combustion per unit mass of working fluid.[11] Q* is a function of the fuel heating value, the mass of inducted air, and the fuel mass:

$$Q^* = \left(\frac{m_f}{m_a}\right) Q_{LHV} \left(\frac{m_a}{m}\right) \qquad \text{(EQ 2)}$$

If it is assumed that fresh air fills the displaced volume during the cycle, and that the residual gas fills the clearance volume at the same density, m_a/m can be approximated as

$$\frac{m_a}{m} = \frac{r_c - 1}{r_c} \qquad \text{(EQ 3)}$$

Furthermore, if the fuel-air mixture is assumed to be stoichiometric and the fuel is iso-octane, then

---

[11] Ibid, pg 170.

$$Q^* = 2.75 \frac{r_c - 1}{r_c} \qquad \text{(EQ 4)}$$

where Q* has units of MJ/kg air.

Fuel Conversion Efficiency: As discussed previously, the fuel conversion efficiency, $?_f$, is the ratio of the work produced to the amount of energy supplied to the engine through the combustion process. For a Constant Volume Ideal Cycle, this parameter is a function of the compression ratio and the ratio of specific heats:

$$?_f = 1 - \frac{1}{r_c^{g-1}} \qquad \text{(EQ 5)}$$

Using the relationships in EQ 2 through EQ 5, IMEP can be rewritten as follows:

$$IMEP = \left(\frac{2.75 \frac{r_c - 1}{r_c}}{C_v T_1} \right) \left(\frac{1}{g-1} \right) \left(\frac{r_c}{r_c - 1} \right) \left(1 - \frac{1}{r_c^{g-1}} \right) P_1 \qquad \text{(EQ 6)}$$

To convert this value to brake mean effective pressure, BMEP, one must subtract the friction losses:

$$BMEP = IMEP - TFMEP \qquad \text{(EQ 7)}$$

where TFMEP represents the total friction losses expressed as a mean effective pressure. An estimate of these losses is given by the equation[12]

$$\text{TFMEP} = 0.97 + 0.15\left(\frac{N}{1000}\right) + 0.05\left(\frac{N}{1000}\right)^2 \qquad \text{(EQ 8)}$$

where N is the engine RPM.

The result of EQs 7 and 8 is that engine's BMEP varies with engine speed, N. In order to convert BMEP to power the piston cylinder geometry must be known. Additionally, since BMEP is per cylinder, multiplying by the number of cylinders, n_c, yields the engine's total power output.

$$P = \frac{BMEP \cdot V_d\, n_c}{2} \qquad \text{(EQ 9)}$$

where V_d is the displaced volume of the cylinder, $\dfrac{n_c\boldsymbol{p}\; B^2 L}{4}$

Therefore, given the initial ambient conditions, compression ratio, and cylinder geometry, one can predict piston engine performance using the *Ideal Gas Standard Cycle* model and the equations listed above.

---

[12] Ibid, pg 722.

Model Results

As shown in Figure 1, once the model is developed, the next step in the methodology is to analyze its output and validate the model. This is done by applying the *Ideal Gas Standard Cycle* model to an existing engine and comparing the predicted performance to the actual engine data. To do this, an Excel spreadsheet is developed using the equations established in this section. The test engine data and ambient conditions used in this model are summarized in Table 1 below.

Table 1: Test Case Data

Parameter	Value	Parameter	Value
P_1 (PSI)	14.7	n_c	4
T_1 (°F)	100	?	1.4035088
T_1 (°R)	559.67	Q_{LHV} (BTU/lb)	1268
r_c	8.5	(Stoichiometric Iso-Octane)	
C_p (BTU/lb °R)	0.24	B/L	0.9615385
C_v (BTU/lb °R)	0.171	B (in)	3.32

Based on this data, the Excel model calculates the engine's power output, and graphs it versus the engine speed. The results appear in Table 2 and shown graphically in Figure 10.

Table 2: Ideal Gas Standard Cycle Model Results

RPM	TFMEP (PSI)	TFMEP (HP)	BMEP (PSI)	Power Output (HP)	Net Power (HP)	Target Value (HP)	
1000	68.80	2.60	824.47	33.71	31.12		
2000	86.44	6.52	806.83	67.43	60.90		
2500	97.46	9.20	795.81	84.28	75.09		
3000	109.96	12.45	783.31	101.14	88.69		
3500	123.92	16.37	769.35	117.99	101.63		
4000	139.36	21.04	753.91	134.85	113.81		
4500	156.26	26.54	737.01	151.71	125.17		% Higher
5000	174.64	32.95	718.63	168.56	135.61	83.97	61.51
5500	194.48	40.37	698.79	185.42	145.05		
6000	215.80	48.87	677.47	202.28	153.41		

Figure 10: Ideal Gas Standard Cycle Power Output

The target value shown on this graph is the maximum power output of the actual engine. In this instance, the predicted power is 62% greater than the actual power. This is due to the CASA assumptions and the associated fluid model, which make the *Ideal Gas Standard Cycle* too simplistic for accurate performance calculations. This model is, however, useful for conducting parametric analysis of the engine and determining the

effects of varying parameters on engine performance (compression ratio, Temperature, Pressure, etc).

Based on these results a more accurate representation of the working fluid is required. Therefore, the next model will be based on a *Fuel-Air Cycle*. The Excel spreadsheet used for the performance calculations of the *Ideal Gas Standard Cycle* performance model appear in Appendix A.

FUEL-AIR CYCLE **PERFORMANCE MODEL**

As mentioned previously, combining an ideal engine cycle with a fuel-air fluid model results in a *Fuel-Air Cycle*. Because the working fluid in this model is more accurately represented, these cycles are generally more precise than the *Ideal Gas Standard Cycle*. In the *Fuel-Air Cycle* the unburned fuel-air mixture is frozen in composition and the burned mixture is in chemical equilibrium. Additionally, each species in the mixture behaves as an Ideal Gas. However, obtaining the thermodynamic properties of the fuel-air mixture is much more difficult than in the *Ideal Gas Standard Cycle*, and requires the use of tables for both the burned and unburned gases.

The *Fuel-Air Cycle* is subject to the following assumptions, listed by process.[13]

Compression (1-2): Isentropic Compression of a mixture of air, fuel vapor, and residual gas without change to the chemical composition.

---

[13] Ibid, pg 177.

Combustion (2-3): Complete, adiabatic combustion to burned gases in chemical equilibrium at either Constant Volume, Constant Pressure, or Limited Pressure, depending on the Ideal Cycle selected.

Expansion (3-4): Isentropic expansion of burned gases which remain in chemical equilibrium.

Exhaust (4-1): Ideal adiabatic exhaust blowdown and displacement of burned gases that are frozen in chemical composition.

Methodology

As in the *Ideal Standard Gas Cycle*, the *Fuel-Air Cycle* is analyzed using a stepwise process, identifying the states of the working fluid at each point in the engine cycle. The objective of this procedure is to calculate the net work produced by combustion based on the following relationship:

$$W_{Net} = W_{Expansion} - W_{Compression} = W_{3-4} - W_{1-2} \quad (EQ\ 10)$$

This will then be converted to the engine's brake work, by subtracting the losses:

$$W_B = W_{Net} - W_{Mech} - W_{Par} \qquad (EQ\ 11)$$

Once the brake work is found, the engine's power output is calculated using the same formula as in the previous model:

$$P = \frac{W_B N}{2} \qquad \text{(EQ 12)}$$

In order to obtain the individual work components in EQ 10, the state of the working fluid must be known at all points in the engine cycle (compression, combustion, etc). Therefore, the individual processes must be analyzed based on the selected ideal cycle. In this instance, a Constant Volume cycle is used since it is the most common. The following sections document the changes in the fuel-air mixture that occur during the various ideal engine cycle processes. These changes are then used to calculate the required individual work expressions. The analysis detailed here uses gas tables to obtain the thermodynamic properties.

Before beginning the individual engine processes, a very important parameter must be introduced. The burned gas fraction, x_b, is the ratio of the residual mass of the burned fuel-air mixture (m_r) left over from the previous cycle to the total mass in the cylinder (m_c).

$$x_b = \frac{m_r}{m_c} \qquad \text{(EQ 13)}$$

The residual mass, m_r, is the burned fuel-air mixture leftover following the blowdown process that occurs during the isentropic expansion from P_4 to $P_{exhaust}$. If the state of the fluid is known during the exhaust process, the gas fraction becomes:

$$x_b = \frac{v_2}{v_e} \qquad\qquad \text{(EQ 14)}$$

where v_e is the fluid volume of the exhausted mixture.

Since the residual gas is recycled into the fresh fuel-air mixture, decreasing the amount of usable fuel for combustion, the engine's work output is decreased. The residual mass depends on many factors, including valve timing and overlap, intake and exhaust pressure, and valve sealing. Therefore, x_b is normally estimated prior to the analysis and then validated once the calculations are complete. Using an iterative process, the calculations are repeated until the estimated and calculated x_bs are equal.

Process 1-2: Isentropic Compression

The compression process is considered to be adiabatic and reversible and therefore isentropic. The major variables required for input into this step are the inlet temperature and pressure (T_1, P_1), fuel-air equivalence ratio (F), and the compression ratio (r_c). These parameters, in conjunction with the thermodynamic gas tables, will define the fuel-air mixture's properties during the compression process.

In addition to the CASA used in the previous analysis, one of the inaccuracies was the method used to determine the initial temperature of the mixture, T_1. Previously, it was assumed that T_1 was equal to the temperature within the engine inlet, T_i (normally 20-30° F higher than ambient). While this is a fairly common simplification, it is not

31

accurate. Indeed, the residual gases from the combustion process and the high cylinder wall temperatures work to make the temperature of the fuel-air mixture substantially higher. To account for this, the following equation is used to calculate the initial fuel-air mixture temperature:[14]

$$T_1 = \frac{1 - x_b}{1 - \dfrac{1}{\mathbf{g}}\left[\dfrac{P_e}{P_i} + (\mathbf{g} - 1)\right]} T_i \qquad \text{(EQ 15)}$$

where

P_e/P_i is the ratio of exhaust pressure to inlet pressure.

In performance calculations the ratio P_e/P_i quantifies the engine's induction process, and has a normal range of associated with it. Furthermore, the exhaust pressure, P_e, is normally assumed to be equal to the ambient pressure, P_{amb}, which allows the designer to define P_e. In naturally aspirated engines, when the inlet pressure is less than the ambient pressure ($P_i < P_{amb}$), the engine is said to be throttled. In this case the ratio of $P_{amb}/P_i = P_e/P_i = 2.$[15] If the inlet pressure is equal to the ambient pressure ($P_i = P_{amb}$), the engine is said to be operating at full throttle, and the ratios P_{amb}/P_i and P_e/P_i are equal to 1.[16] Generally, the full throttle setting also produces the maximum power output. In this analysis the engine is assumed to be operating at full throttle, and a P_e/P_i of 1 is used in EQ 15.

---

[14] Ibid, pg 172.
[15] Ibid, pg 175.
[16] Taylor, Charles F., The Internal Combustion Engine in Theory and Practice, M.I.T. Press, 1966, pg 76.

During the initial calculation of T_1, an estimate of the burned gas fraction, x_b, is normally used to begin the analysis. Once the performance calculations are complete, the estimated value of x_b (and therefore T_1) are checked against the new values. If they do not match then another iteration of the model is conducted using the new values. This process is explained in detail in a later section.

Once T_1 is found, the next step is to find the fuel-air mixtures temperature at the end of compression, T_2. This is done by using an isentropic compression chart, which appears in Figure 11.
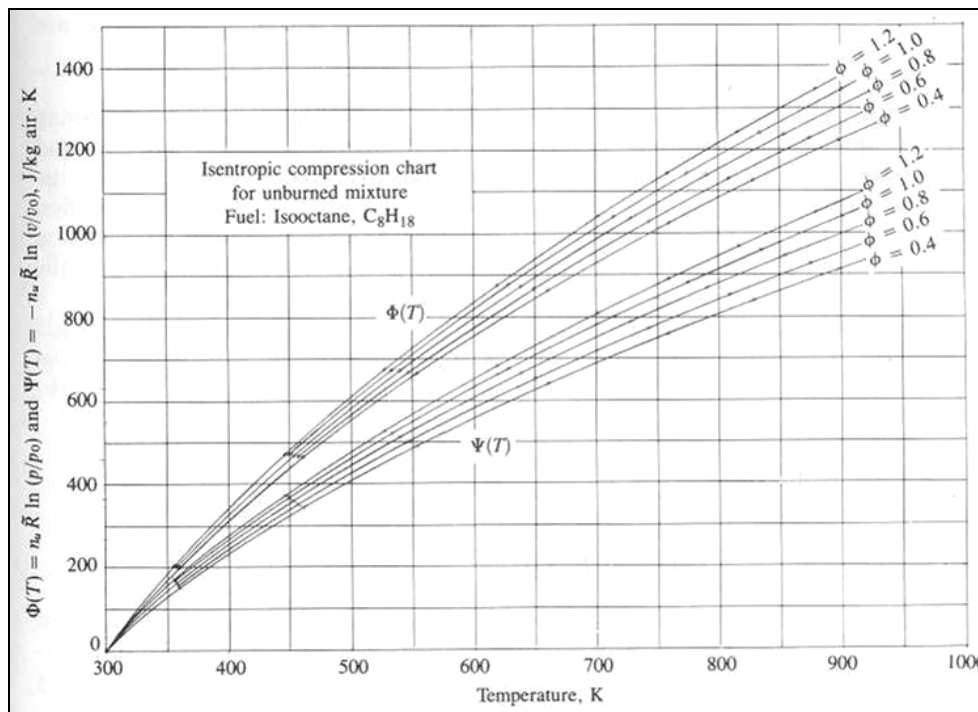
Figure 11: Isentropic Compression Chart

To use this chart, the equation for T_2 is

$$\Psi(T_2) = \Psi(T_1) - n_u R \ln\left(\frac{v2}{v1}\right) \qquad \text{(EQ 16)}$$

where ?(T) is an integral function used to construct the chart in Figure 11. The quantity

$n_u R$ refers to the composition of the unburned mixture, which is a function of the

equivalence ratio. Using a least squares regression of tabular data, the equation for this

value is

$$n_u R = 287.89 + 4F - 3.57(F-0.8)^2 \qquad \text{(EQ 17)}$$

in units of J/kg air.

Using the Ideal Gas relationships one can find v_1, v_2, and P_2.

$$v_1 = \frac{n_u \tilde{R} T_1}{P_1} \qquad \text{(EQ 18)}$$

$$p_2 = P_1 \frac{T_2}{T_1} r_c \qquad \text{(EQ 19)}$$

$$v_2 = \frac{v_1}{r_c} \qquad \text{(EQ 20)}$$

At this point, if the initial conditions of the fuel-air mixture are known (T_1, P_1, F,

and r_c), all of its properties can be calculated for the compression process. Then, based

on the temperatures T_1 and T2, the internal energies of the fluid are found using the internal energy table in Figure 12.
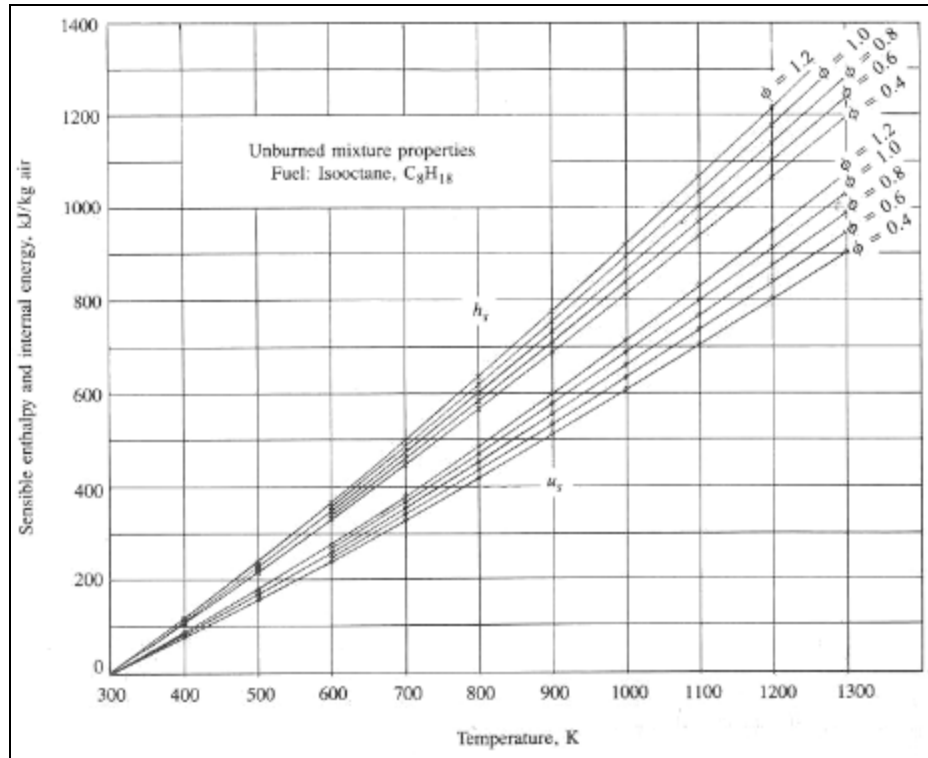
Figure 12: Unburned Mixture Internal Energy

Since the compression process is adiabatic, the work is simply the differences in the internal energies of the fuel-air mixture at the two temperatures:

$$W_{Comp} = W_{1\text{-}2} = u_2 - u_1 \qquad \text{(EQ 21)}$$

Obviously, because work is being done on the system (fuel-air mixture) during the compression process, W_{1-2} is a negative quantity.

Process 2-3: Constant Volume Adiabatic Combustion

In the *Ideal Gas Standard Cycle*, the combustion process is modeled by heat addition (Q*), while a *Fuel-Air Cycle* models combustion based on the fuel-air mixture's thermodynamic properties. Burning the fuel-air mixture results in chemical transformations that change its temperature and pressure (assuming constant volume combustion). These changes lead to the subsequent expansion of gases and therefore more accurately quantify the engine's power output if properly captured.

The key to analyzing the combustion process is to link the properties of the unburned mixture to those of the burned mixture. The intent is to define the state of the unburned mixture after combustion for a given T_2, P_2, and v_2 (the state of the mixture following isentropic compression). To do this, the unburned and burned mixtures are assigned a zero datum for measuring internal energy and enthalpy. Unburned mixtures normally assume zero internal energy at 298.15° K, and the internal energy relative to this datum is called the sensible internal energy, u_s.[17] Using this convention, changes in internal energy are a result of temperature changes from the zero datum, and ignore changes due to chemical reactions.[18]

The burned fuel-air mixture's datum is different than the unburned datum insofar as only certain species within the burned mixture are assigned zero enthalpy at 298.15°K.

---

[17] Ibid, pg 113.
[18] Ibid, pg 113.

Specifically, the species O_2, N_2, H_2, and C have zero enthalpy at that datum.[19] The differences in internal energy between these species in the two datum is called the unburned mixture's internal energy of formation, $u^\circ_{f,u}$. Therefore, the summation of $u^\circ_{f,u}$ for all the aforementioned species, annotated by $?u^\circ_{f,u}$, represents the change in the internal energy between the burned and unburned mixtures. Consequently, the internal energy of the unburned mixture, u_u is the sum of the sensible internal energy and the summation of the internal energies of formation:

$$u_u = u_{s,u} + ?\,u^\circ_{f,u} \qquad\qquad \text{(EQ 22)}$$

The internal energy of formation for a stoichiometric ($F=1$) fuel-air mixture is a function of the burned gas fraction:[20]

$$?\,u^\circ_{f,u} = -118.2 - 2956x_b \qquad\qquad \text{(EQ 23)}$$

where $?\,u^\circ_{f,u}$ has units of J/kg air.

Additionally, since the unburned gases in both the compression and combustion processes use a datum of 298.15° K, the sensible internal energy equals the internal energy at the end of compression: $u_{s,u} = u_2$. Therefore, in a constant volume adiabatic combustion process, the burned and unburned gases are related as follows:

---

[19] Ibid, pg 123.
[20] Ibid, pg 124.

$$u_b = u_u = u_3 = u_2 + ?\,u^{\circ}_{f,u} \qquad \text{(EQ 24)}$$

$$v_b = v_u = v_2 = v_3 \qquad \text{(EQ 25)}$$

Using these relationships, u_3 and v_3 are found, thereby fixing the state of the working fluid following the combustion process. As a result, the remaining properties (T_3, P_3) can be obtained using thermodynamic gas tables as shown in Figure 13.
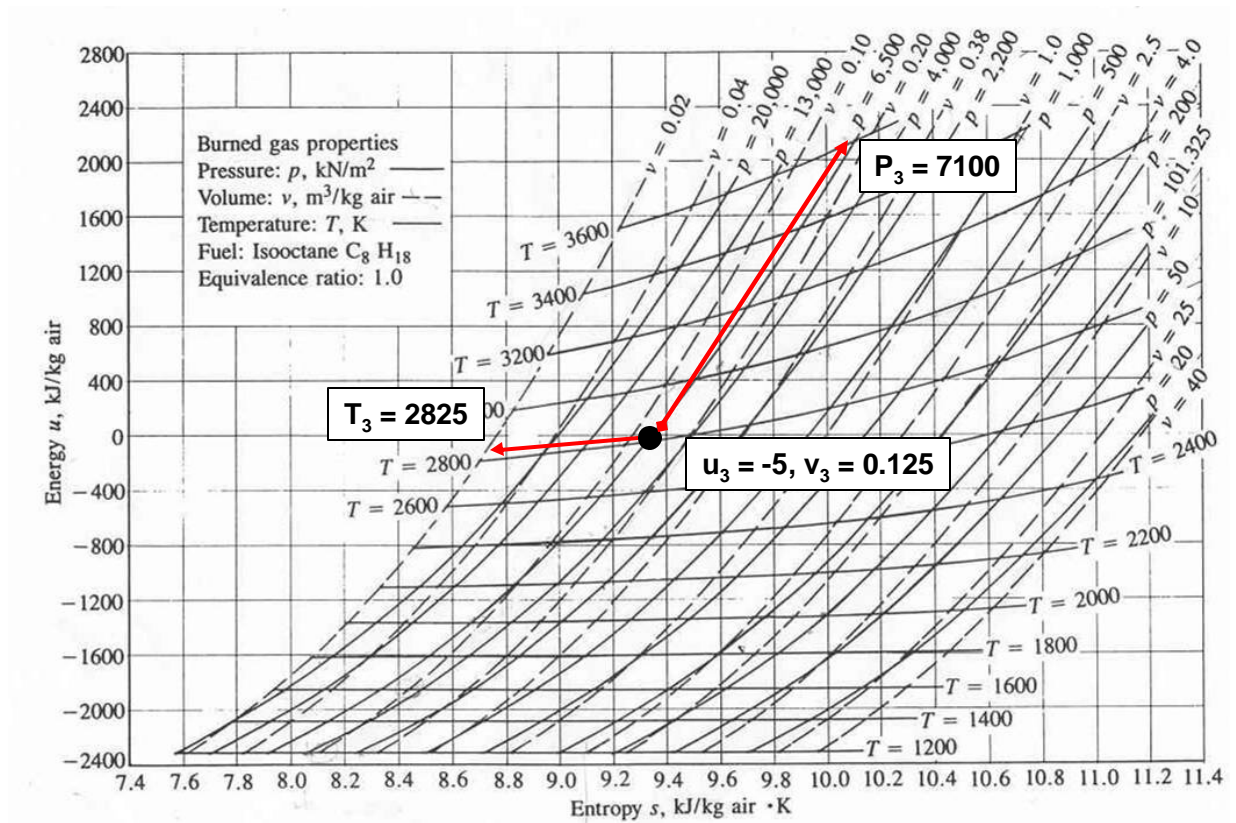
Figure 13: Constant Volume Adiabatic Combustion

For illustrative purposes, u_3 and v_3 are assigned hypothetical values of -5 kJ/kg air and 0.125 m³/kg air, respectively. By following the lines of constant pressure p_3 is found to

be 7100 kN/m^3. Likewise, T_3 is shown to be 2825° K. Once the burned fuel-air mixture's properties are defined, the next step is to analyze the expansion process.

Process 3-4: Isentropic Expansion

Modeling the expansion of the burned fuel-air mixture requires extensive use of the thermodynamic gas tables. The first step is to calculate the volume at completion of the expansion process. For a Constant Volume cycle, $v_1 = v_4$ and $v_2 = v_3$.

On the thermodynamic gas charts, the same initial state point (u_3, v_3) as before is used, and expanded isentropically to the final volume, v_4. This process is shown in Figure 14.
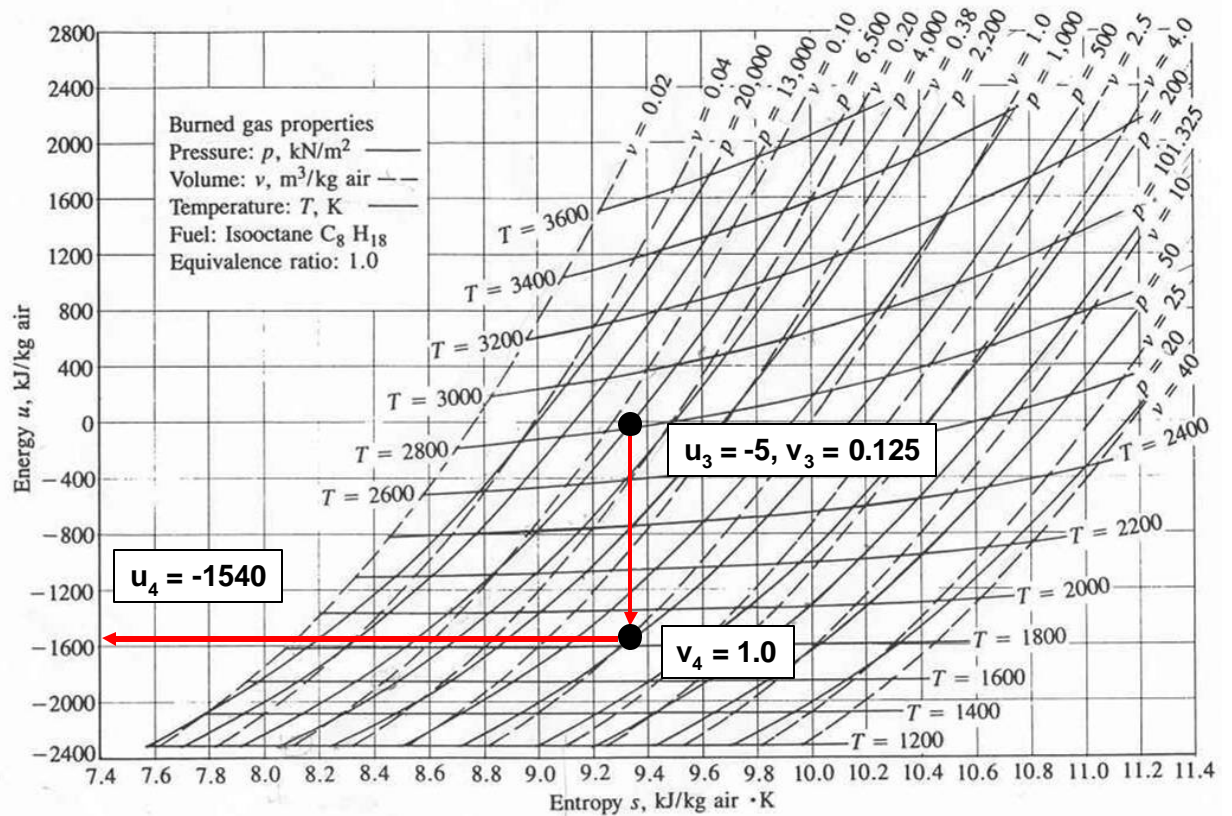
Figure 14: Isentropic Expansion

Again, for illustrative purposes example numbers are used. While all of the fuel-air mixture's properties can be defined at this point, the most important parameter is its internal energy. As in the compression process, the expansion work is the difference between the internal energy prior to expansion (u_3) and the internal energy following expansion (u_4):

$$W_{Exp} = W_{3-4} = u_3 - u_4 \qquad \text{(EQ 26)}$$

Verification of Burned Gas Fraction

Before calculating the engine's net work, the burned gas fraction, x_b, must be verified. When analyzing the combustion process x_b is used in EQ 15 to find T_1, and again in EQ 23 to find $?\mathfrak{u}_{f,u}$. However, as mentioned previously, x_b is normally estimated in the beginning of the model, and calculated after the analysis is complete. If the calculated and estimated values are different, the performance calculations are repeated until the estimated and calculated x_bs converge. As shown in EQ 14, the burned gas fraction is

$$x_b = \frac{v_2}{v_e}$$

To find v_e, the burned gas chart in Figure 14 is used once again. The burned gases are expanded isentropically to the exhaust pressure, P_e, which is equal to P_i when operating at full open throttle. From this new state point, v_e is found from the lines of constant volume. An example of this process is shown in Figure 15. Since v_2 is already known, the burned gas fraction is easily calculated. If this new value does not correspond to the initial estimate of x_b, the calculations are repeated using the new x_b. Typically, one to two iterations are required before the values converge.
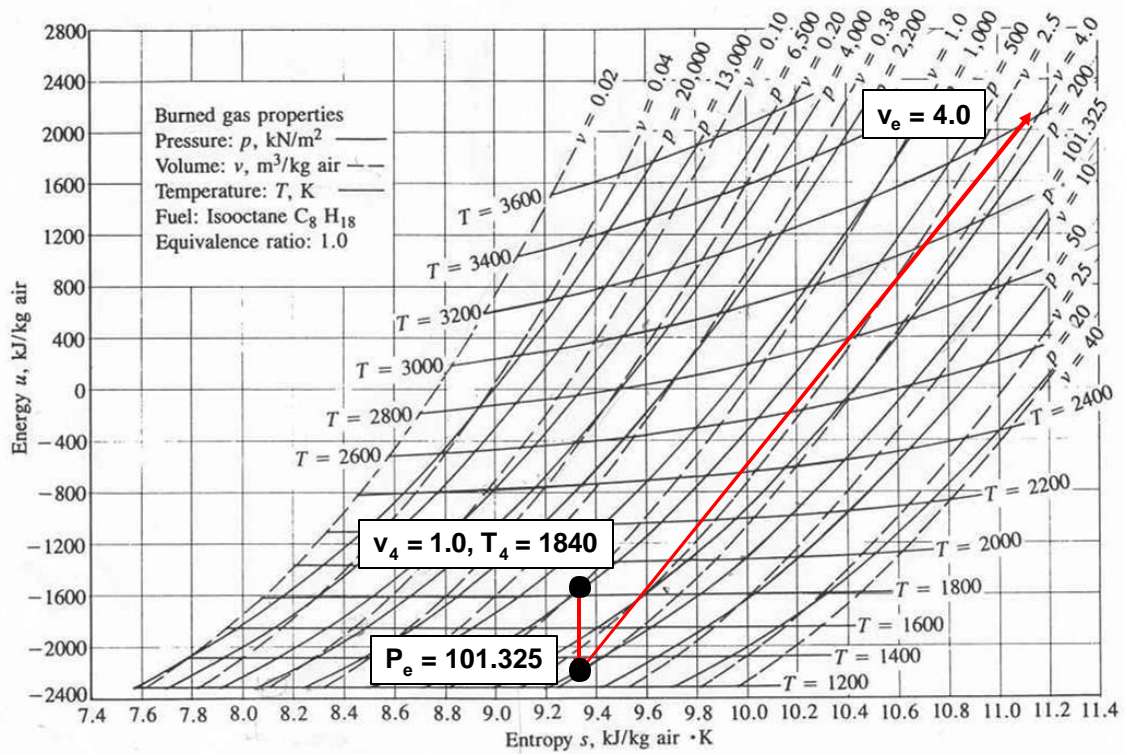
Figure 15: Isentropic Expansion of Exhaust Gases

Engine Net Work

At this point, the individual work components used in EQ 10 are known, and the net work of the engine is calculated. The next step is to convert the net work to the engine's Indicated Mean Effective Pressure.

$$IMEP = \frac{W_{Net}}{V_{max} - V_{Min}} = \frac{W_{Net}}{v_1 - v_2} \qquad \text{(EQ 27)}$$

To find BMEP, the same loss model used in the *Ideal Gas Standard Cycle* model is applied:

$$TFMEP = 0.97 + 0.15\left(\frac{N}{1000}\right) + 0.05\left(\frac{N}{1000}\right)^2$$

and

$$BMEP = IMEP - TFMEP$$

Once the BMEP is known, the engine's power is calculated using EQ 12, and the major performance calculations are complete.

Specific Fuel Consumption

One of the most important operating parameters of an engine is its efficiency, particularly it's efficiency in converting the supplied fuel to useful work. One of the best metrics for measuring this efficiency is the Specific Fuel Consumption (SFC). The SFC is defined as the fuel flow rate per unit power output:

$$SFC = \frac{\dot{m}_f}{P}$$

As described earlier, an engine's SFC can also be expressed in dimensionless terms using the Fuel Conversion (Thermal) Efficiency, $?_F$

$$?_f = \frac{P}{\dot{m}_f Q_{hv}} = \frac{1}{sfcQ_{hv}} \qquad \text{(EQ 28)}$$

or

$$?_f = \frac{W_{Net}}{m_f Q_{hv}} \qquad \text{(EQ 29)}$$

At this point in the model, all elements of EQ 29 are known except for the mass of the fuel, m_f. However, if a stoichiometric mixture is assumed, m_f becomes

$$m_f = \left(\frac{F}{A}\right)_s (1 - x_b) \qquad \text{(EQ 30)}$$

where

$\left(\dfrac{F}{A}\right)_s$ is the stoichiometric fuel to air ratio of the fuel (0.0661 for Iso-octane).

Once $?_f$ is found using EQ 29, the engine's SFC is calculated after simple manipulation of EQ 28.

$$SFC = \frac{1}{h_f Q_{HV}} \qquad \text{(EQ 31)}$$

44

Model Results

As with the *Ideal Gas Standard Cycle*, the outlined *Fuel-Air Cycle* performance model is now applied to an existing engine. The same data used in the *Ideal Gas Standard Cycle* is used in this analysis. The input data is summarized in Table 3, and is exactly the same as the previous analysis except for the elimination of unnecessary data (C_p, Q_{LHV}, etc).

Table 3: Test Case Data

Parameter	Value	Parameter	Value
P_1 (PSI)	14.7	n_c	4
P_1 (kPA)	101.36	B/L	0.96
T_{inlet} (°F)	100	B (in)	3.32
T_{inlet} (°K)	310.93	B (dm)	0.84
x_b (Initial)	0.029	V_d (dm^3)	0.49
?	1.40	V_d (in^3)	29.89
T_1 (°K)	342.17	?	1
r_c	8.5	p_e/p_i	1

Once again, an Excel spreadsheet is used to calculate and plot the engine's performance as a function of engine speed. The results appear in Table 4 and are plotted in Figure 16.

Table 4: Fuel-Air Cycle Model Results

RPM	TFMEP	TFMEP	BMEP	Power Output	Net Power	Target Value	
	(PSI)	(HP)	(PSI)	(HP)	(HP)	(HP)	
1000	68.80	2.60	601.86	25.31	22.71		
2000	86.44	6.52	584.22	50.62	44.10		
2500	97.46	9.20	573.20	63.28	54.08		
3000	109.96	12.45	560.70	75.93	63.48		
3500	123.92	16.37	546.74	88.59	72.22		
4000	139.36	21.04	531.30	101.24	80.21		
4500	156.26	26.54	514.40	113.90	87.36		% Higher
5000	174.64	32.95	496.02	126.56	93.60	83.97	11.48
5500	194.48	40.37	476.18	139.21	98.84		
6000	215.80	48.87	454.86	151.87	103.00		

Figure 16: Fuel-Air Cycle Power Output

Immediately, one notices that the *Fuel-Air Cycle* model is much more accurate than the *Ideal Standard Gas Cycle*. Indeed, this analysis yields an 11% error, whereas the previous example had a 62% overage. Obviously, the combination of a revised starting temperature (T_1) and a more accurate combustion model make the *Fuel-Air Cycle* a much better model for predicting the performance of the actual engine cycle. While an 11% error is not acceptable for performance calculations, consideration was given to the

error inherent in graphical interpolation of the thermodynamic properties. Based on this consideration, the results presented here are sufficiently accurate to validate the model and begin developing the computer program. The Excel spreadsheet with the supporting performance calculations appears in Appendix B.

COMPUTER PROGRAM DEVELOPMENT

Once the *Fuel-Air Cycle* performance model is validated, the next step is to develop the computer code to automate the procedures. The biggest challenge is reproducing the data presented in the thermodynamic tables used in the *Fuel-Air Cycle* performance calculations. Toward this end, the author employs a thermodynamic equilibrium program developed by NASA: the "Computer Program for Calculation of Complex Chemical Equilibrium Composition and Applications." Also known as the NASA CEA (Chemical Equilibrium with Applications) program, it was developed at the NASA Glenn Research Center, and is well documented and readily available. It is extremely powerful, with uses in analyzing thermodynamic states, Chapman-Jouguet detonations, rocket performance, and shock-tube parameters for incident and reflected shocks.[21] A substantial portion of the programming efforts required for this computer model are dedicated to integrating the CEA software.

For the governing performance program, an Excel based Visual Basic (VBA) code is selected for its user friendly interface and relative simplicity. Since the NASA

---

[21] NASA Glenn Research Center CEA Homepage: http://www.grc.nasa.gov/WWW/CEAWeb/

CEA program is written in FORTRAN, an interface between it and the primary VBA code must be developed. The intent is that the VBA parent program will perform all the performance calculations based on the thermodynamic properties generated by the CEA program. Essentially, the NASA program will serve as a subroutine for the VBA code, generating thermodynamic data. Therefore, when conducting performance analysis the CEA program runs simultaneously. The integration of the CEA program into the overall computer code will be discussed at length, including an overview of its uses and the specific applications required in the performance calculations.

The final computer model is divided into four distinct modules or sections: Input, State Points and Work, Performance, and the Engine Deck. The latter three sections rely exclusively on the CEA software for the thermodynamic calculations. Each of these four modules will be discussed in detail. The complete computer code, divided into the separate VBA Modules, appears in Appendix C. A user's manual for the final program appears in Appendix D.

CEA Overview

As previously stated, the parent computer program is an Excel based VBA model that uses the NASA CEA code for its thermodynamic calculations. The major advantages of the VBA model is that it is based in a well known interface in Excel, and the language stems largely from the built in functions of Excel. However, the NASA CEA program is a critical part of the overall computer code used in the *Fuel-Air Cycle* model and must be properly integrated. Because of the differences in computer

languages between the Excel Visual Basic Model and the FORTRAN-based CEA code, a series of Visual Basic macros are needed to automate the CEA program and obtain specific state point data. As shown in the development of the *Fuel-Air Cycle* performance model, thermodynamic tables play a critical role in the cycle analysis. Specifically, seven of the parameters used in the calculations were found using thermodynamic tables: u_1, T_2, u_2, T_3, P_3, u_4, and v_4. The remaining parameters were found through other means, including the Ideal Gas Law and the internal energy of formation. At a minimum, the seven parameters previously listed must now be provided by the CEA program. To increase accuracy and maintain consistency, the other state points will be obtained from the CEA program when possible.

As described earlier, the CEA program is extremely powerful and can perform a multitude of tasks. Indeed, the program defines nine specific problem types or paths, each of which can be applied to the problem based on the user's requirements. However, for the purposes of this thesis only four of the nine are required. The specific paths used for each process in the engine cycle are listed below in Table 5:

Table 5: CEA Problem Types

Engine Process	Problem Type	Problem Description
Initial Condition	PT	Assigned Pressure and Temperature
1-2: Isentropic Compression	SV	Assigned Entropy and Volume
2-3: CV Combustion	UV	Assigned Internal Energy and Volume
3-4: Isentropic Expansion	SV	Assigned Entropy and Volume
4-Exh: Exhaust Expansion	SP	Assigned Entropy and Pressure

Each problem type requires the user to specify the assigned parameters, as well as the reactants, which in this case are iso-octane fuel and air. Based on this information,

the program calculates the mixture's thermodynamic properties, including pressure, temperature, density, enthalpy, internal energy, entropy, and specific heat.

CEA Application

The first step in integrating the equilibrium program into the overall computer code is to reproduce the data used in the *Fuel-Air Cycle* performance model. This will serve to further validate the *Fuel-Air Cycle* analysis and develop familiarity with the CEA software. Table 5 shows that each process is associated with its own problem type, and is therefore treated as an independent problem by CEA. However, because the engine processes are modeled as a cycle, the data produced by the individual problems is not truly independent. Indeed, the output of one state becomes the input of the next, creating a continuity within the separate problems.

As shown in Table 5, the first use of the CEA program is to define the thermodynamic properties at the initial conditions of the mixture. In this case the primary parameters of interest are the internal energy and entropy, which are found using the "PT" problem type. Given the values of T_1, P_1, Equivalence Ratio, and the fuel type, CEA calculates the remaining properties. The entropy at this point, s_1, and the volume, v_1, are then used in the isentropic compression process, which is the "SV" problem path. The volume at state 2, v_2, is found from the relation of $v_2 = v_1/r_c$. The results of this problem are the state points at the end of compression: T_2, P_2, and u_2. Therefore, the compression work, W_{1-2}, can be found using the equation $W_{1-2} = u_2 - u_1$.

Modeling the combustion process requires using the "UV" path. This problem calculates the thermodynamic properties based on a fixed internal energy and volume, which are characteristics of the Constant Volume combustion process employed in the *Fuel-Air* methodology. However, the internal energy used here is not the absolute internal energy obtained at the end of compression. Rather, it is the relative internal energy based on the internal energy of formation. To compute this, another "SV" problem is run from state point 1 using the zero datum for temperature, 298.15° K, and is labeled state point 2 Standard. The difference between this standardized internal energy, u_{2_std}, and the original u_2 becomes the internal energy at state 3, u_3, and the input for the "UV" problem. Since this is a constant volume process, $v_3 = v_2$.

To calculate the isentropic expansion of the burned mixture, another "SV" problem is used. Up to now the reactants used in the CEA program have been fuel and air, and their mixture properties were based on an equivalence ratio, ? . At this point, however, the fuel-air mixture has been exploded into its different constituents. Fortunately, the output of the "UV" combustion problem includes the exploded chemical composition of the fuel-air mixture and the relative mole fractions. One of the features of CEA is that it enables the user to establish the minimum quantity for trace products. For the purposes of this thesis, the trace amount is set at E-3. Combustion products whose mole fractions are less than this are not displayed, with no effect on accuracy.

These compounds and their respective quantities now become the input of the "SV" problem. The input parameters are known, since $s_4 = s_3$ and $v_4 = v_1$, and the output provides the internal energy after expansion, u_4. The expansion work, W_{3-4}, is found

from the relation $W_{3-4} = u_3 - u_4$. The net work of the cycle is then found as before: $W_{Net} = W_{3-4} - W_{1-2}$.

The final step is to find the volume of the burned gas after its expansion. This is done using the "SP" path of the CEA program. Since this is an isentropic expansion continued from the previous state point, the entropy is already known. Also, the expansion will continue until atmospheric pressure is reached, so all required parameters are known, and the specific volume can be found.

The processes outlined above not only calculate the net work, W_{Net}, but in doing so also calculate the seven parameters that were previously found using thermodynamic tables. To provide a basis of comparison, the CEA program was applied to the *Fuel-Air Cycle* performance model example using the same initial data, and then compared to the previous output. The results appear below in Table 6. In this table, all the state point parameters listed under the CEA program were found using the program itself. The seven parameters previously found in tables are highlighted in red for comparison.

Table 6: State Point Comparison: Thermodynamic Tables vs. CEA

	1	CEA (PT)	2	CEA (SV)	3	CEA (UV)
Temp (°K)	342.15	342.15	690.00	718.96	2900.00	2996.47
Pressure (kPA)	101.33	101.33	1737.33	1810.67	6700.00	7726.94
Volume (m³/kg)	0.9857	0.9907	0.1160	0.1165	0.1160	0.1165
u form (kJ/kg)			-203.924	351.60		
u (kJ/kg)	50.00	-2945.49	370	-2628.39	166.076	351.60
s (kJ/kg °K)		7.13		7.13		8.83

	4	CEA (SV)	Exh	CEA(SP)
Temp (°K)		2028.00		1424.79
Pressure (kPA)		601.63	101.33	101.33
Volume (m³/kg)	0.9857	0.9907	4.00	4.07
u form (kJ/kg)				
u (kJ/kg)	-1410.00	-1222.06		-1925.49
s (kJ/kg °K)		8.83		8.83

	Tables	CEA	% Diff
W_{1-2} (kJ/kg air)	320.00	317.10	0.91%
W_{3-4} (kJ/kg air)	1576.08	1573.66	0.15%
W_{Net} (kJ/kg air)	1257.96	1256.56	0.11%

The blocked out sections are those where data either did not apply, or was not required for the previous example problem and therefore not obtained from the tables (i.e. T_4, P_4). With the exception of P_3 and internal energy, all calculated parameters have a less than 4% deviation from the table values. The internal energy varies significantly due to a difference in the datum used by the CEA program and the datum used for the thermodynamic tables. However, it is the relative differences between the values that are important in calculating the engine work, and not the values themselves. Indeed, the engine work calculated by the CEA program is almost identical to the work values obtained from the thermodynamic tables, with an error less than 1%. These results lead one to the conclusion that the NASA CEA program is sufficiently accurate to use in the performance calculations.

CEA Integration

After reproducing the desired results using the CEA program, the next step is to automate the calculations and integrate the program into the Excel-based model. This is done through the use of three separate Visual Basic macros. The first macro creates the cea.inp file based on the user's input. The second executes the CEA program proper, and the third extracts the data from the resulting cea.out file. This procedure is repeated for each process in the engine cycle (compression, combustion, expansion, exhaust expansion), and is depicted in Figure 17.
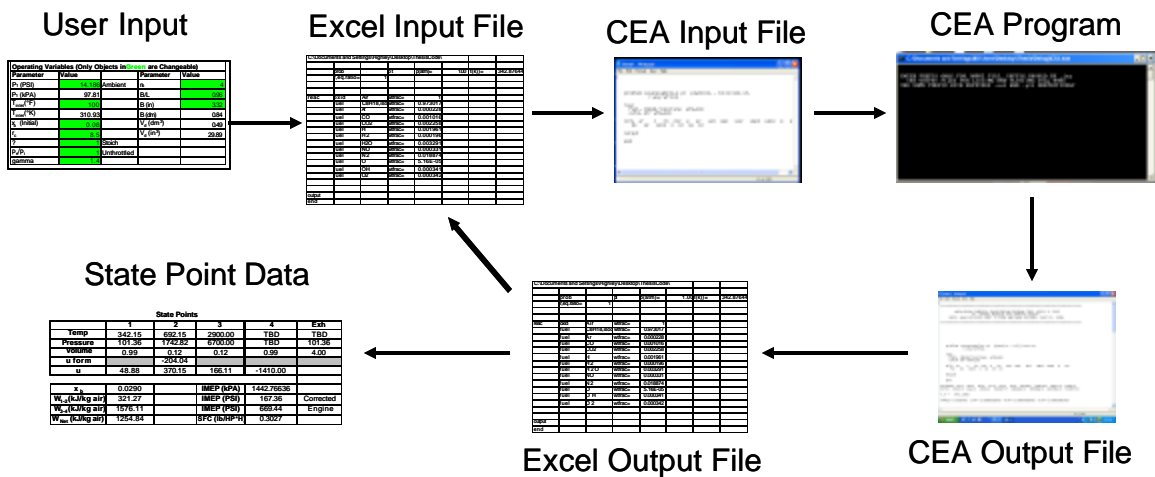
Figure 17: CEA Integration

In the base Excel program, a separate worksheet is assigned to the input and output of each engine process, for a total of 12 worksheets (including the standardized State 2). Each input sheet lists the appropriate problem type (i.e. PT, SV, etc), and the generic items that CEA requires for the input file. Other than the temperature of the State

2 Standard input sheet, which is fixed at 298.15° K, all of the parameters change based on the user input. Some parameters change directly from the input module, such as P_1 and $?$. Others, like T_2 and P_2, change indirectly through the output of another process. Within the program each parameter is linked to the appropriate source, and is updated each time CEA runs. Once the input sheet is complete, the macro writes the information to the input file, cea.inp. It is important to note that only one input file is used by the program at any given time; cea.inp is rewritten for each state point as the program is executed.

The second macro executes the CEA program proper using the newly created input file. Since the CEA program normally requires user interaction to enter the file name, the base code was modified to automatically execute the program using the cea.inp file present in the working directory. This directory is listed on the first input sheet of the calculations (State 1 Input) and must be updated anytime the program is transferred to another computer. The program runs once for each engine process, for a total of six times per cycle. Fortunately the CEA program runs quickly, so computational time is minimal.

The final macro used to integrate CEA into the overall VBA program reads the output file, cea.out, and imports the data to the appropriate Excel worksheet. As with the input macro, each state point has its own worksheet. Once the data is placed into the worksheet, additional macros and functions search the data and extract the desired parameters which then become inputs to other engine processes, or are used in the performance calculations. However, the output file is not produced instantaneously, which can cause an error when reading and importing the output file. To overcome this

obstacle, the VBA code employs a time delay between execution of the CEA program and retrieval of the output file. Presently this delay is set at two seconds.

Although not one of the main three macros, another important function built into the VBA code reintroduces the combustion products into the compression and combustion processes. These products are read from the combustion process in the form of mole fractions, which are converted to weight fractions based on the burned gas fraction, x_b. They are then added to the input files of the compression and combustion steps, decreasing the amount of fresh fuel and air that enters the cylinder. The result is a more accurate representation of these engine processes, and a better estimate of the work produced by the engine.

Parent Computer Program

The overall methodology employed by the computer model was shown in Figure 17 above, depicting the interaction of VBA with CEA. The program itself is not extremely complex, and essentially automates the *Fuel-Air Cycle* performance model. The main difference is that the computer model iterates to find the engine's burned gas fraction, x_b. The calculations begin with an initial estimate of the gas fraction, x_{bi}. Based on this value, the performance calculations are completed, ultimately yielding a new value of x_b. If this value and the initial estimate are not within a specified tolerance, currently set at 0.0005, the initial value is set to the calculated value ($x_{bi} = x_b$) and the calculations are repeated. Each time the calculations are repeated, all three of the macros described above are executed for each of the six state points, which can lead to

considerable computational time. Fortunately, most problems converge within three iterations, regardless of the initial estimate and ambient conditions. Once the burned gas fraction is finalized all the work and MEP calculations are completed and the data is sent to its respective modules.

The user interface consists of four separate modules or sections. Each module plays a critical role in the overall program, and are designed to be simple to use and easily understandable. Each of these modules is discussed in turn.

Input Module

The first of the four sections of the power program, the input module provides the user with a simple interface for specifying the critical engine parameters and the ambient conditions. This section appears in Figure 18.

Operating Variables (Only Objects in Green are Changeable)					
Parameter	Value	Parameter	Value	Parameter	Value
P_1 (PSI)	14.696	T_{inlet} (°F)	100	r_c	8.5
P_1 (kPA)	101.33	T_{inlet} (°K)	310.93	n_c	4
$T_{ambient}$ (°F)	59	x_b (Initial)	0.025	B/L	0.96
$T_{ambient}$ (°K)	288.15	?	1	B (in)	3.32
? T (°F)	41	?	1.3667	V_d (in^3)	29.94
? T (°K)	22.78				

Reset Input   Calculate   Clear Output

Figure 18: Input Module

The ambient conditions are entered in English units, which are automatically converted to SI units for the calculations. The major inputs to the program are as follows:

- P_1. This is the ambient pressure conditions of the static engine.

- $T_{ambient}$. The ambient temperature of the static engine

- ?T. The difference between the ambient temperature and the inlet temperature. Primarily a design feature, this enables the designer to account for variances in the engine inlet placement and possible temperature increases from radiant heat.

- T_{inlet}. This is the engine inlet temperature, given by $T_{ambient}$ + ?T.

- x_b (initial). The initial guess for the burned gas constant. Values normally range from 1-10%. This serves only as the initial estimate - the program will iterate to find the actual value.

- ? . The fuel-air equivalence ratio. This parameter is used to model either a lean (? < 1) or rich (? > 1) mixture.

- ? – The ratio of specific heats. This value is not actually an input, but calculated by the CEA program.

- r_c. The engine's compression ratio.

The remaining parameters model the engine's cylinder geometry. These values enable the program to convert the engine's power from a MEP to an actual horsepower. It is important to note that these parameters consider only the number and dimensions of the piston cylinders, and do not account for overall engine geometry (V, radial, etc).

These parameters will become critical in the development and design of engines in conjunction with the vehicle synthesis programs.

- n_c. This is the number of cylinders.

- B/L. The bore to stroke ratio. For design purposes this can be set to 1.

- B. The bore of the cylinder.

- V_d. This is the engine's displaced volume and is calculated based on the geometry and number of cylinders.

At the very bottom of the input section are three macro buttons. The "Reset Input" button automatically resets the input parameters to enable the user to start a new engine model. The "Calculate" button begins program execution once the user has set all the parameters. Finally, the "Clear Output" button erases all previous output data.

State Points and Work Module

The next section of the program is where the thermodynamic analysis takes place, and its output appears in Figure 19.

State Points					
	1	**2**	**3**	**4**	**Exh**
Temp (°K)	342.38	719.30	2997.54	2028.58	1421.34
Pressure (kPA)	101.33	1809.40	7720.30	601.36	101.32
Volume (m³/kg)	0.9788	0.1151	0.1151	0.9788	4.0632
u (kJ/kg)	-2946.50	-2629.20	351.99	-1222.13	-1930.10
s (kJ/kg °K)	7.13	7.13	8.83	8.83	8.83
x_b	0.028339		**IMEP (kPA)**	1455.31	
W_{1-2} (kJ/kg air)	317.30		**IMEP (PSI)**	168.82	Corrected
W_{3-4} (kJ/kg air)	-1574.12		**IMEP (PSI)**	675.26	Engine
W_{Net} (kJ/kg air)	1256.82		**SFC (lb/HP*hr)**	0.3198	

Figure 19: State Points and Work Module

This module tracks the thermodynamic properties of the fuel-air mixture as it proceeds through the five (including exhaust expansion) processes of the engine cycle. It provides the user with a point by point synopsis of the engine processes and the resulting thermodynamic changes. It also summarizes the engine's work output and IMEP. State 1 refers to the fuel-air mixture prior to the compression process, and sets the mixture's initial conditions. State 2 gives the fluid's properties after the isentropic compression, while the results of the combustion process are listed in State 3. Next, State 4 gives the burned mixture's properties after the isentropic expansion following the combustion process. Finally, the EXH or exhaust state gives the fluid's properties after the isentropic

60

expansion to the atmospheric pressure as it enters the exhaust valve. This information is used in when calculating the final burned gas fraction, x_b. The specific parameters are:

- x_b. This is the final burned gas fraction calculated by the program.

- W_{1-2}. The compression work, based on EQ 22.

- W_{3-4}. This is the expansion work, as determined by EQ 27.

- W_{Net}. The net work produced by the engine, based on EQ 13.

- IMEP. This is the engine's indicated mean effective pressure.

- IMEP (Corrected). This is the engine's IMEP after the 0.8 empirical correction is applied.

- IMEP (Engine). The total engine IMEP (IMEP * n_c).

- SFC. The base Specific Fuel Consumption of the engine, based on the engine's indicated work. Because it is indicated, this value is substantially lower than the actual values.

For simplicity the data at State point 2 Standard is not listed. The data at this point is used only in the calculation of u_3, and is not otherwise useful for tracking thermodynamic changes to the fuel-air mixture. Furthermore, the State point data listed in Figure 19 is mostly a compendium of the data obtained from the output worksheets. If more detailed information is required the complete CEA output files are captured in the respective State point output worksheets. However, since the worksheets are rewritten each time the program executes and a new output file is created, the data listed is only for the final iteration of the performance calculations.

Power Output Module

The third module calculates the engine's power output as a function of engine speed. The user then has the option of plotting the data and/or creating an engine deck. The power output table appears in Figure 20.

Engine Speed	Gross Power	Losses	Net Power	SFC
(RPM)	(HP)	(HP)	(HP)	(lb/HP*hr)
1000	25.53	2.60	22.93	0.4450
1500	38.29	4.36	33.93	0.4511
2000	51.05	6.53	44.52	0.4584
2500	63.81	9.21	54.60	0.4671
3000	76.58	12.47	64.11	0.4775
3500	89.34	16.40	72.95	0.4895
4000	102.10	21.07	81.03	0.5036
4500	114.87	26.58	88.29	0.5200
5000	127.63	33.01	94.62	0.5391
5500	140.39	40.43	99.96	0.5614

Plot Results   Engine Deck

Figure 20: Power Output Module

Based on the IMEP, the engine's gross power is calculated. The friction losses are then calculated and subtracted from the gross power to obtain the engine's net power output. Finally, the engine's specific fuel consumption is calculated using this net power value. These parameters are discussed in more detail below.

- Gross Power. The engine's indicated or gross power output, without accounting for any losses. This is given by EQ 12.

- Losses. These are the pumping, mechanical, and parasitic losses seen by the engine, as given by the TFMEP found in EQ 11.

- Net Power. The engine's brake power – the actual output at the crankshaft.

- BSFC. The engine's Brake Specific Fuel Consumption, which is based on the engine's BMEP.

This module also gives the user the option of plotting the power output and Specific Fuel Consumption versus the engine's speed by clicking the "Plot Results" button. The graphical depiction of the engine's major performance parameters enables the user to visualize the performance trends as a function of engine speed and will help when designing engines to meet specified design points. An example chart is shown below in Figure 21.
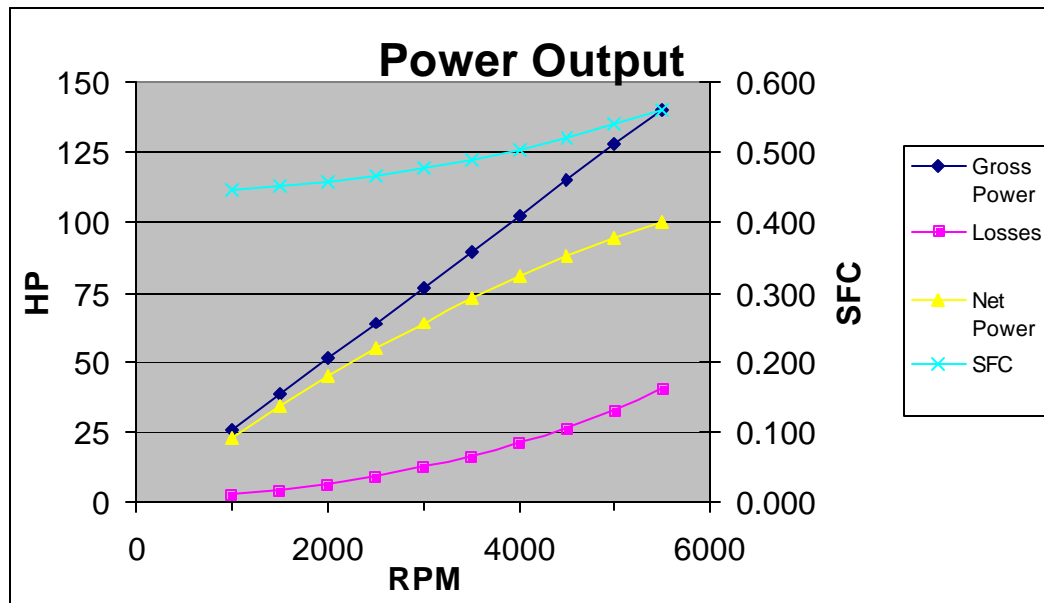
Figure 21: Engine Power Curve

Finally, an "Engine Deck" button is available in this section should the user want to see the engine's performance at different flight conditions. This feature is discussed in the next section.

Engine Deck

An engine's power output is a function of the ambient conditions in which it operates. Because altitude and temperature changes will affect the mass flow rate of the air inducted into an engine, its power output will also change. In aerospace applications, accounting for these changes is a fundamental requirement of engine selection and design. To this end, the final module within this program creates the engine deck, which determines the engine's power as a function of both altitude and engine speed. This enables the designer to determine the engine's performance in a specific flight regime and under certain atmospheric conditions. The engine deck module allows the user to input a maximum altitude and then calculates the pressure and temperature at increments of 1000 ft. These values are then used as P_1 and $T_{ambient}$ for the power calculations, and the performance calculations described previously are repeated for all altitudes. As before, the IMEP is calculated and used to determine the engine's gross and net power output as a function of RPM. The results are listed by engine speed (1000 to 5500 RPM) per 1000 feet of altitude, up to the limit established by the user. For the calculations, all inputs other than temperature and pressure are taken from the input module.

To calculate the temperature and pressure at altitude, the following model is used:[22]

$$T_{meas} = 59 - .00356\,h \qquad \text{(EQ 32)}$$

$$P_{meas} = 2116\left[\frac{T_{meas} + 456.7}{518.6}\right]^{5.256} \qquad \text{(EQ 33)}$$

where

h is the altitude in feet, T_{meas} is in ° F, and P_{meas} is in lb/ft^2.

The model in EQ 32 is based on standard day conditions, and may not represent the flight conditions the designer wishes to replicate. Therefore, the program provides the user with the ability to input a temperature deviation from the standard day. This is done through an input box similar to the one used to enter the maximum altitude.

The next step is to use T_{meas} in EQ 15 as T_{inlet}, and find the new T_1. Likewise, P_{meas} becomes P_1. Using these new values of T_1 and P_1, the power calculations are repeated using the same computer model. The results of these calculations are the engine's new work and IMEP, which lead to the gross power as a function of engine speed. When finding the net power output the total friction losses are assumed to be constant with respect to ambient conditions, and therefore the same TFMEP model used previously is applied. This results in a net power curve with the same shape as that found in the *Power Output* section.

---

[22] "Earth Atmosphere Model," http://www.grc.nasa.gov/WWW/K-12/airplane/atmos.html

Using the new net power value, the engine's brake specific fuel consumption is calculated. To do this the power is converted to BMEP, which in turn is used to find the brake work, W_B. Using this value, the program calculates the fuel conversion efficiency, $?_f$, which is converted to the BSFC by EQ 31.

Once all the individual parameters (gross power, TFMEP, net power, BSFC) are consolidated and displayed to create the engine deck, an example of which appears in Figure 22.

Altitude	Temp	Press	Engine Speed	Power	BSFC
(Ft)	(°F)	(Atm)	(RPM)	(HP)	(lb/HP*hr)
1000	55.44	0.965	1000	26.32	0.4300
1000	55.44	0.965	1500	39.02	0.4350
1000	55.44	0.965	2000	51.30	0.4412
1000	55.44	0.965	2500	63.09	0.4485
1000	55.44	0.965	3000	74.29	0.4570
1000	55.44	0.965	3500	84.82	0.4670
1000	55.44	0.965	4000	94.60	0.4785
1000	55.44	0.965	4500	103.55	0.4918
1000	55.44	0.965	5000	111.59	0.5071
1000	55.44	0.965	5500	118.62	0.5247
2000	51.88	0.930	1000	25.52	0.4311
2000	51.88	0.930	1500	37.81	0.4364
2000	51.88	0.930	2000	49.70	0.4427
2000	51.88	0.930	2500	61.08	0.4503
2000	51.88	0.930	3000	71.88	0.4591
2000	51.88	0.930	3500	82.01	0.4695
2000	51.88	0.930	4000	91.39	0.4815
2000	51.88	0.930	4500	99.94	0.4953
2000	51.88	0.930	5000	107.57	0.5113
2000	51.88	0.930	5500	114.20	0.5298

Figure 22: Engine Deck

It is important to note that at each altitude the program will execute the full calculation cycle, iterating until x_b converges to itself. Depending on the maximum altitude specified by the user, the computation time can become substantial. However, repeating the full program for each altitude ensures the consistency and accuracy of the results.

Computer Program Results

Once the program is complete it must be validated against the manual *Fuel-Air Cycle* results. Using the same test engine and ambient conditions as in the manual calculations, the computer program is executed and compared to the base test engine. The results appear in Table 7 and Figure 23.

Table 7: Computer Program Results

Engine Speed	Gross Power	Losses	Net Power	SFC	Target Value	% Error
(RPM)	(HP)	(HP)	(HP)	(lb/HP*hr)		
1000	25.53	2.60	22.93	0.4450		
1500	38.29	4.36	33.93	0.4511		
2000	51.05	6.53	44.52	0.4584		
2500	63.81	9.21	54.60	0.4671		
3000	76.58	12.47	64.11	0.4775		
3500	89.34	16.40	72.95	0.4895		
4000	102.10	21.07	81.03	0.5036		
4500	114.87	26.58	88.29	0.5200		
5000	127.63	33.01	94.62	0.5391	83.97	12.69%
5500	140.39	40.43	99.96	0.5614		
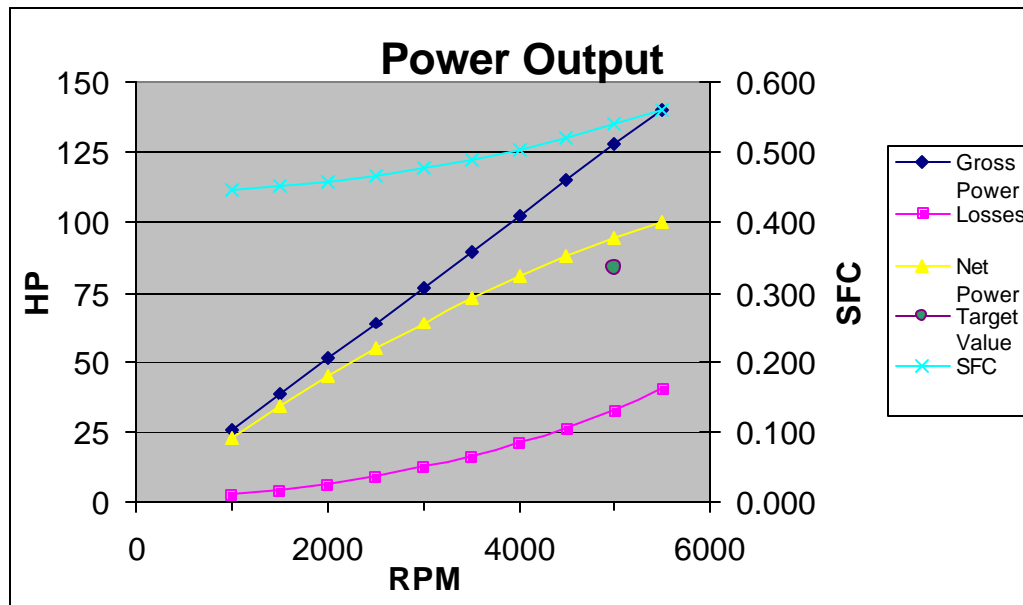
Figure 23: Computer Program Power Output

The original *Fuel-Air Cycle* performance model calculated 93.6 HP, and the program predicts 94.62 HP, which is a deviation of only 1%. However, the original model used an inlet temperature of 100° F, which was used in the input module for this model. If T_i is set to standard day conditions (59° F), the computer model's error

increases significantly to 36%. For these reasons, the ?T correction should be used to account for inlet design and radiant heat to ensure more accurate results.

Computer Program Summary

The finalized computer program used to model the *Fuel-Air Cycle*, henceforth called the Piston Engine Performance Program (PEPP), is an Excel based Visual Basic model that integrates the NASA CEA program to reproduce the data previously obtained from thermodynamic tables. The CEA program, though working as a subroutine of the parent VBA program, performs the bulk of the calculations and provides the thermodynamic State point data. The program is executed for each state point, with Visual Basic Macros writing the input files, executing the program, and reading the output files. The data from these files is then used to calculate the burned gas fraction, and the program iterates until the value converges to itself based on a user supplied initial estimate. Once this occurs, the engine's work and IMEP are found using the final thermodynamic data, and the program generates the power output table and charts. Finally, the program enables the user to create an engine deck to predict performance at different flight conditions. Using a standard atmospheric model, T_i and P_1 are calculated and the performance calculations are completed.

Using the same initial data as the manual Fuel-Air Cycle calculations, the program yields nearly the same results, with only a 1% deviation in results. However, the model is sensitive to inlet temperature deviations, and when standard day conditions are

used with no accountability to radiant heat near the inlet, the program overestimates engine performance.

PERFORMANCE MODEL CORRELATION

Once the computer model is complete, the final step is to correlate it to the performance of a known engine. For this correlation a Lycoming O-320-E2A engine is selected. The intent is to validate the computer model by reproducing the performance curves of the O-320-E2A engine at various flight conditions.

O-320-E2A Engine Specifications

The Lycoming O-320-E2A aircraft engine is a four cylinder, naturally aspirated, spark ignition, direct drive engine. It uses a float type carburetor, and is found mostly on Piper aircraft. The main parameters needed for the comparison are listed below in Table 8.

Table 8: Lycoming O-320-E2A Engine Data

Parameter	Value
r_c	7
n_c	4
B/L	1.32
B (in)	5.13

All performance data for the engine is based on an ideal fuel air mixture, which for the purposes of this analysis is assumed to be stoichiometric. Additionally, the engine is operating at wide open throttle. Both of these conditions are consistent with the computer model.

Model Modifications

The first step is to run PEPP using the data in Table 8 and compare the results to the Lycoming performance curves. For these calculations standard day ambient conditions are used: T = 59° F and P = 14.696 lb/in^2. The results are listed below, along with the Lycoming Data.

Table 9: Predicted vs. Actual Power (Unmodified)

Engine Speed	Predicted	Actual	Error
(RPM)	(HP)	(HP)	%
2000	127.62	114.50	11.45
2100	133.55	121.04	10.34
2200	139.44	127.78	9.12
2300	145.26	134.00	8.40
2400	151.03	139.50	8.26
2500	156.73	143.00	9.60
2600	162.38	147.78	9.88
2700	167.95	151.94	10.54

Avg Error	9.70

Based on the consistency of the error values, the slopes of two curves are very similar, which is verified by a plot. Based on this observation, one can conclude that the current friction loss model does not accurately predict the O-320-E2A's losses and must

therefore be modified. A plot of the two power curves, along with the friction losses appears in Figure 24.
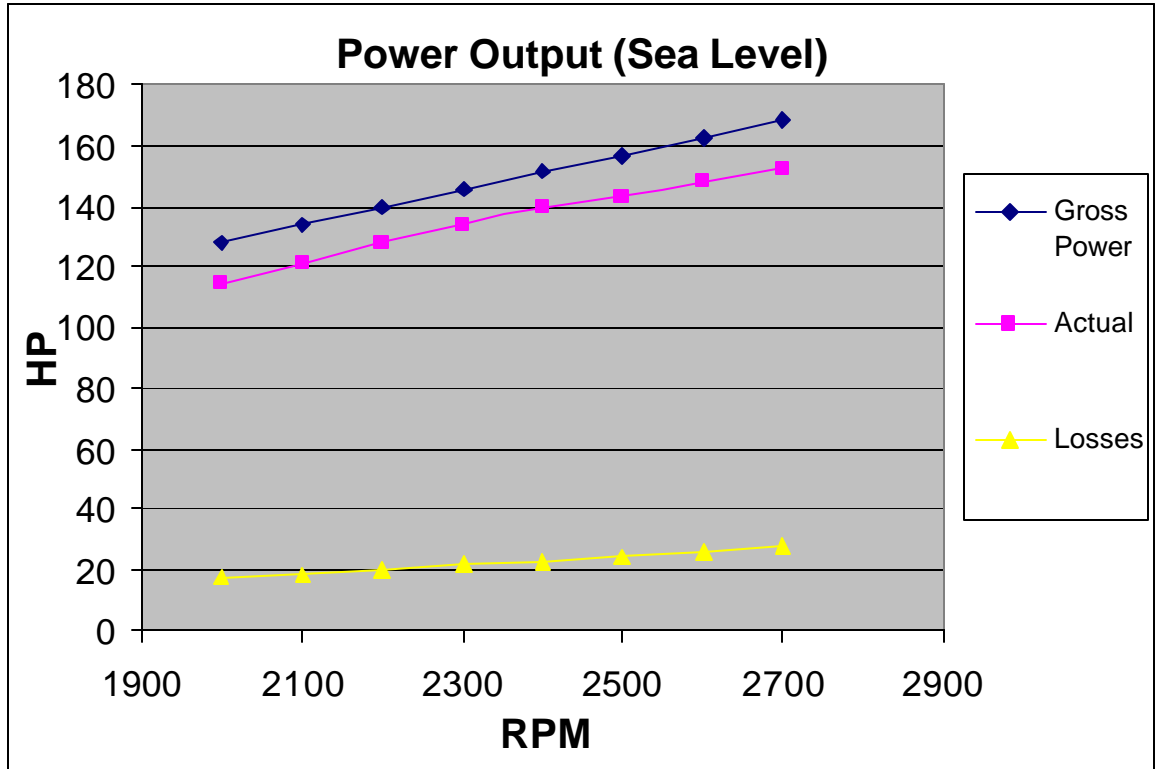
Figure 24: Predicted vs. Actual Performance (Unmodified)

From this plot, the general trend of the loss curve is correct, but is simply too low to accurately portray the O-320-E2A's losses. The simplest way to adjust this is to shift the entire loss curve up, which can be done by adjusting the y-intercept of the TFMEP model. Recall that the TFMEP equation is

$$\text{TFMEP} = 0.97 + 0.15\left(\frac{N}{1000}\right) + 0.05\left(\frac{N}{1000}\right)^2$$

In this model 0.97 is the y intercept, which must be increased in order to decrease the difference between the two sets of data. Indeed, increasing the y intercept to 1.9074 decreases the average error to 0%. The results are shown in Table 10 and Figure 25.

Table 10: Predicted vs. Actual Performance (Modified)

RPM	Predicted	Actual	Error
2000	116.48	114.50	1.70%
2100	121.86	121.04	0.67%
2200	127.18	127.78	-0.47%
2300	132.45	134.00	-1.17%
2400	137.66	139.50	-1.34%
2500	142.81	143.00	-0.13%
2600	147.89	147.78	0.08%
2700	152.92	151.94	0.64%

Avg Error	0.00%
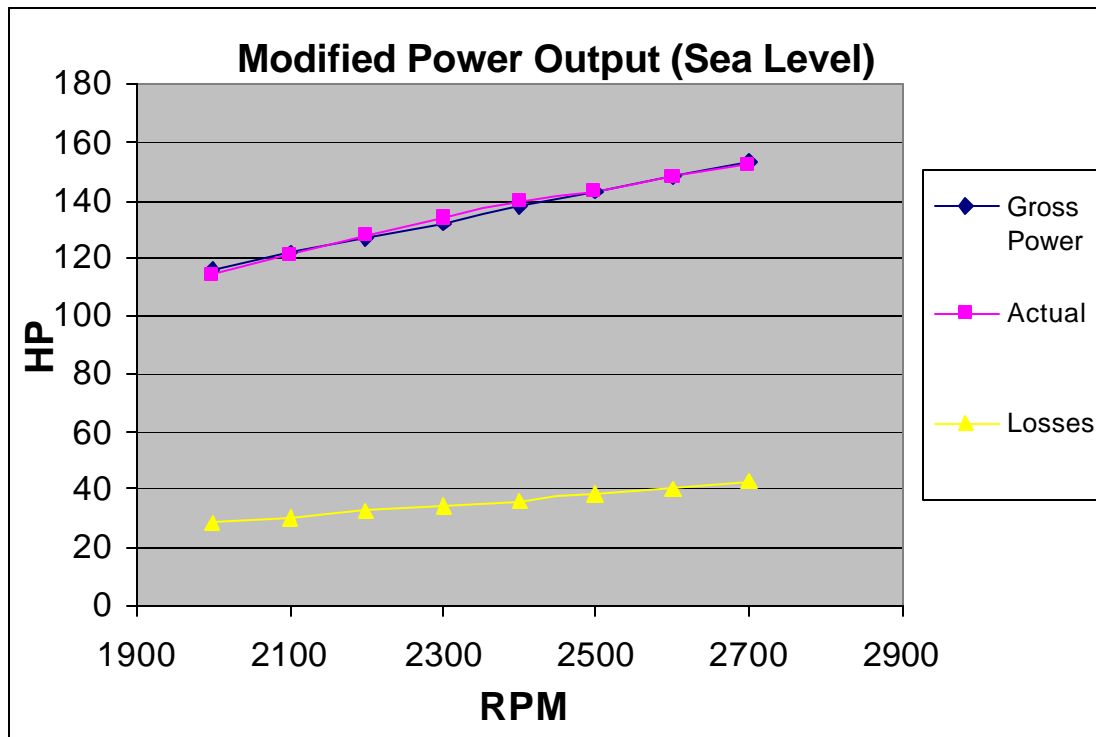
Figure 25: Predicted vs. Actual Performance (Modified)

Based on these results, one can conclude that the PEPP model is very well correlated to the O-320-E2A at sea level conditions.

The next step is to predict the engine's performance at various flight conditions. Using the program's Engine Deck feature, the engine's power output is predicted at altitude and compared to the manufacturer's specifications. In this instance the engine was evaluated at altitudes of 1000, 2000, 5000, 10000, and 15000 ft. When creating the deck, the same modified TFMEP model is used for the engine's losses. The results appear below in Table 11.

Table 11: Predicted vs. Actual Performance at Various Altitudes

RPM	1000 ft			2000 ft		
	Predicted	Actual	Error	Predicted	Actual	Error
2000	115.50	110.69	4.16%	112.33	107.08	4.67%
2100	120.83	117.08	3.10%	117.51	113.26	3.61%
2200	126.11	123.61	1.98%	122.62	119.65	2.42%
2300	131.32	130.56	0.59%	127.68	126.04	1.29%
2400	136.49	135.69	0.58%	132.69	130.69	1.50%
2500	141.58	138.96	1.86%	137.63	134.65	2.16%
2600	146.62	142.57	2.76%	142.51	137.92	3.22%
2700	151.60	147.43	2.75%	147.32	142.08	3.56%

		Avg Error	2.22%		Avg Error	2.80%

RPM	5000 ft			10000 ft			15000 ft		
	Predicted	Actual	Error	Predicted	Actual	Error	Predicted	Actual	Error
2000	104.10	96.39	7.41%	89.76	71.04	20.85%	74.86	67.08	10.39%
2100	108.86	102.43	5.91%	93.80	85.69	8.64%	78.16	71.04	9.11%
2200	113.57	108.13	4.79%	97.79	90.69	7.26%	81.41	75.00	7.87%
2300	118.22	113.82	3.72%	101.72	95.35	6.27%	84.60	78.61	7.07%
2400	122.81	118.13	3.81%	105.60	98.96	6.29%	87.72	81.74	6.83%
2500	127.34	121.39	4.67%	109.41	101.39	7.33%	90.79	83.96	7.53%
2600	131.81	124.65	5.43%	113.16	104.31	7.82%	93.80	86.39	7.90%
2700	136.21	128.26	5.83%	116.85	107.43	8.06%	96.74	88.96	8.04%

	Avg Error	5.20%		Avg Error	9.07%		Avg Error	8.09%

	Avg Total Error	5.48%

Based on the results in Table 11, the model is not as accurate at altitudes. Indeed, it seems that in general, increasing altitude increases the error. This may be in part due to changes in the friction losses at altitude, which are not accounted for in the TFMEP model. However, the average error for the specified flight conditions is 5.48%, which makes the results acceptable for design purposes.

CONCLUSIONS

The objective of this thesis was to develop a thermodynamic computer program to model a naturally aspirated, spark ignition piston engine. This has been accomplished. Using a stepwise methodology, two engine models were analyzed and considered as a basis for the computer program. First, an *Ideal Gas Standard Cycle* was developed, but proved inaccurate for useful performance calculations. Therefore, a *Fuel-Air Cycle* performance model was created, which greatly increased accuracy. This model then became the basis for the computer program.

One of the challenges in developing the Piston Engine Performance Model (PEPP) was reproducing the data in the thermodynamic charts used by the *Fuel-Air Cycle*. To this end, the NASA Chemical Equilibrium with Applications (CEA) program was integrated into the parent Visual Basic Model. The result is a comprehensive VBA model that incorporates the NASA CEA code as a subroutine to provide thermodynamic data for the performance analysis.

PEPP's output correlates closely to the manual *Fuel-Air Cycle* calculations. However, the program is very sensitive to the engine's inlet temperature, and lower

temperature increase the error between actual and predicted performance. To mitigate this error, the model provides the user with the ability to input an inlet temperature difference to simulate engine layout and inlet placement.

Finally, by adjusting the engine loss equation, PEPP was correlated to the performance of a Lycoming engine. The model is very well correlated at sea level conditions, but errors increase with altitude, with an average error of 5.48% over the range of the test data. However, this error is low enough to use the model to predict vehicle performance with an acceptable degree of accuracy.

FUTURE WORK

While this thesis produced a working performance model in PEPP, there are a few modifications that could be made to increase its accuracy and utility. While these modifications are currently beyond the scope of this project, they would prove extremely useful for improving PEPP's output in the future.

The first major modification would be the development of a comprehensive friction loss model for predicting the engine losses. This improved loss model would address many of PEPP's inaccuracies. The current TFMEP model is relatively simple, and is a function of engine speed only. However, there are many contributing factors that affect engine losses, including piston speed, ambient conditions, and engine geometry. By taking friction data and performing a multivariate regression or surface plot, one could obtain a loss model that was a function of all these parameters. This would greatly increase the accuracy of the current model.

The next topic entails validating the 0.8 empirical correction used to account for the differences between the actual and ideal engine cycles. While this correction is cited in Heywood, comparing the results of an actual engine's work to that of the ideal cycle may help refine the value for specific engine parameters (intake/exhaust pressures, number of cylinders, engine geometry, etc). By calculating the positive and negative work of an engine through a P-V diagram, and comparing the results to the W_{Net} of the ideal cycle, a more accurate correction factor could be found. By conducting this analysis based on a series of engines parameters (number of cylinders, geometry, volume,

etc), an engine specific correction factor could be developed and applied to the engines that fit those parameters. This would result in more accurate performance calculations.

Another possible improvement would be modifying PEPP to model compression ignition (Diesel) engines, which are becoming more prevalent within the piston engine marketplace. Most of the modifications required to model a diesel engine would take place in the CEA portion of PEPP. Specifically, the combustion process would need to be changed, as would v_2 and v_3. The parent VBA program would remain largely unchanged.

Along with changing the combustion process, the code could be modified to model supercharged engines. This requires changing the P_e/P_i ratio to a value less than one, depending on the amount of supercharging. The volume within the cylinder would have to be increased based on the degree of supercharging, which would provide the increased work production. However, superchargers use substantial amounts of power at higher RPMs, so the loss model would have to again be modified.

The final potential improvement would be to provide a user input for the type of fuel to be used. CEA is capable of modeling dozens of liquid fuels, each of which could be used as an input for the thermodynamic calculations.

APPENDIX A

Ideal Gas Standard Cycle Performance Model Calculations

Variables							
Parameter	**Value**		**Parameter**	**Value**	**Normal Range**		
P_1 (PSI)	14.7		n_c	4			
T_1 (°F)	100		?	1.403508772			
T_1 (°R)	559.67		Q HV (BTU/	1268	Stoichiometric		
r_c	8.5		Q lhv (BTU/	1268	Isooctane		
C_p (BTU/lb °R)	0.24		B/L	0.961538462			
C_v (BTU/lb °R)	0.171		B (in)	3.32	0.7 - 1.2		
f	1						
Calculations							
Q* (BTU/lbm)	1118.82	EQ 4					
$?_f =$	0.57833	EQ 5					
$?_{f\ corr} =$	0.46267	EQ 5 * 0.8					
IMEP (PSI)	223.317	EQ 6	Per Cylinder				
IMEP (PSI)	893.269	IMEP * n_c	Engine				
Estimate V_d							
V_d (in^3)	29.89						
Displacement	119.56	V_d * n_c					
RPM	**TFMEP**	**TFMEP**	**BMEP**	**Power Output**	**Net Power**	**Target Value**	
	(PSI)	(HP)	(PSI)	(HP)	(HP)	(HP)	
1000	68.80	2.60	824.47	33.71	31.12		
2000	86.44	6.52	806.83	67.43	60.90		
2500	97.46	9.20	795.81	84.28	75.09		
3000	109.96	12.45	783.31	101.14	88.69		
3500	123.92	16.37	769.35	117.99	101.63		
4000	139.36	21.04	753.91	134.85	113.81		
4500	156.26	26.54	737.01	151.71	125.17		**% Higher**
5000	174.64	32.95	718.63	168.56	135.61	83.97	61.51
5500	194.48	40.37	698.79	185.42	145.05		
6000	215.80	48.87	677.47	202.28	153.41		

Power Output

APPENDIX B

Fuel-Air Cycle Performance Model Calculations

Operating Variables			Engine Geometry			
Parameter	Value		Parameter	Value		
P_1 (PSI)	14.7	Ambient	n_c	4		
P_1 (kPA)	101.36		B/L	0.96		
T_{inlet} (°F)	100		B (in)	3.32		
T_{inlet} (°K)	310.93		B (dm)	0.84		
x_b (Initial)	0.029		V_d (dm^3)	0.49		
?	1.40		V_d (in^3)	29.89		
T_1 (°K)	342.17	*EQ 15*				
r_c	8.5					
?	1	Stoich				
p_e/p_i	1	Unthrottled				
Process 1-2 Isentropic Compression						
n_uR (J/kg air)	292	*EQ 17*		v_1 (m^3/kg air)	0.98	*EQ 18*
?(T_1)	145	Table		p_2 (kPA)	1737.33	*EQ 19*
?(T_2) (J/kg air)	769.358	*EQ 16*		v_2 (m^3/kg air)	0.12	*EQ 20*
T_2 (°K)	690	Table				
u_1 (kJ/kg air)	50	Table				
u_2 (kJ/kg air)	370	Table				
Compression Work						
W_{1-2} (kJ/kg air)	320	*EQ 21*				
Process 2-3 CV Combustion						
$u_{f,u}$ (kJ/kg air)	-203.92	*EQ 23*	P_3 (kPA)	6700	Table	
u_3 (kJ/kg air)	166.076	*EQ 24*	T_3 (°K)	2900	Table	
v_3 (m^3/kg air)	0.11587	equals v_2 (CV)				
Process 3-4 Isentropic Expansion						
v_4 (m^3/kg air)	0.9849	equals v_1 (CV)				
u_4 (kJ/kg air)	-1410	Table				
Expansion Work						
W_{3-4} (kJ/kg air)	1576.08	*EQ 26*				

Net Work							
W_{Net} (kJ/kg air)	1257.96	*EQ 10*					
Verify x_b							
v_5 (m^3/kg air)	4	Table, assuming P_5 equals P_1 (unthrottled)					
x_b	0.02897	*EQ 14*					
T_5	1380	Table					
Calculate MEP							
IMEP (kPA)	1445.38	*EQ 27*					
IMEP (PSI)	167.664	Corrected, Per Cylinder					
IMEP (PSI)	670.656	Engine					
Calculate Power							
RPM	TFMEP	TFMEP	BMEP	Power Output	Net Power	Target Value	
	(PSI)	(HP)	(PSI)	(HP)	(HP)	(HP)	
1000	68.80	2.60	601.86	25.31	22.71		
2000	86.44	6.52	584.22	50.62	44.10		
2500	97.46	9.20	573.20	63.28	54.08		
3000	109.96	12.45	560.70	75.93	63.48		
3500	123.92	16.37	546.74	88.59	72.22		
4000	139.36	21.04	531.30	101.24	80.21		
4500	156.26	26.54	514.40	113.90	87.36		% Higher
5000	174.64	32.95	496.02	126.56	93.60	83.97	11.48
5500	194.48	40.37	476.18	139.21	98.84		
6000	215.80	48.87	454.86	151.87	103.00		

APPENDIX C

PEPP Computer Code

Create Input File Module

```
' This sub routine writes the data on the input sheet to the .INP file
' used by CEA.

Sub ExportInputFile(counter)

    Dim currentdir As String
    Dim inputfile As String

' Find the folder containing CEA

    currentdir = Worksheets(3).Range("A1").Value

    inputfile = currentdir + "cea.inp"

' Open destination file for output.

    DestFile = inputfile
    Open DestFile For Output As #1    'FileNum

    Worksheets(counter).Activate

        Range("A2:J28").Select

' Loop for each row in selection.

        For RowCount = 1 To Selection.Rows.count

' Loop for each column in selection.

            For ColumnCount = 1 To Selection.Columns.count

' Write current cell's text to file with quotation marks.

                Print #1, Selection.Cells(RowCount, _
                  ColumnCount).Text;

' Check if cell is in last column. If so, then write a blank line
' otherwise write a comma.

                    If ColumnCount = Selection.Columns.count Then
                        Print #1,
                    Else
                        Print #1, " ";
                    End If

' Start next iteration of ColumnCount loop.

        Next ColumnCount

' Start next iteration of RowCount loop.
```

```vb
    Next RowCount

' Close destination file.

    Close #1

    Range("A1").Select

End Sub
```

Run CEA Module

```
' This sub routine runs the CEA application.  It uses whatever cea.inp
' file is in the destination folder at the time.  It then writes the
' cea.out file.  This process is repeated for every state point.

Sub RunApp()

    Dim executable As String
    Dim outputfile As String
    Dim currentdir As String

    currentdir = Worksheets(3).Range("A1").Value

    executable = currentdir + "cea.exe"
    outputfile = currentdir + "cea.out"

'Change active directory

    ChDir currentdir

    Dim Myapp

' Sets Myapp variable equal to the Shell statement.

    Myapp = Shell(executable, 1)

    ' Executes the shell statement.

End Sub
```

Read Output File Module

```vba
' This sub routine extracts information from the *.OUT files
' and writes it to the output sheet.

Sub ImportOutputFile(counter)


    Dim DestBook As Workbook, SourceBook As Workbook
    Dim currentdir As String
    Dim outputfile As String

    Worksheets(counter).Range("A1:M300").ClearContents

' This finds the folder where the files will be written

    currentdir = Worksheets(3).Range("A1").Value

    outputfile = currentdir + "cea.out"

    Set DestBook = ActiveWorkbook

' This actually reads the file data

    Workbooks.OpenText Filename:= _
        outputfile, Origin:= _
        xlWindows, Startrow:=1, DataType:=xlDelimited, TextQualifier:=
_
        xlDoubleQuote, ConsecutiveDelimiter:=True, Tab:=False,
Semicolon:=False, _
        Comma:=False, Space:=True, Other:=False,
FieldInfo:=Array(Array(1, 1), _
        Array(2, 1), Array(3, 1), Array(4, 1), Array(5, 1), Array(6,
1), Array(7, 1), Array(8, 1), _
        Array(9, 1), Array(10, 1), Array(11, 1), Array(12, 1),
Array(13, 1))


' Set an object variable for the workbook containing the text file.

    Set SourceBook = ActiveWorkbook

' Copy the contents of the entire sheet containing the text file.

    Range(Range("A2"), Range("A2").SpecialCells(xlLastCell)).Copy

' Activate the destination workbook and paste special the values
' from the text file.

    DestBook.Activate

    Worksheets(counter).Range("A1").PasteSpecial Paste:=xlValues
```

```vba
' Clear clipboard

    Application.CutCopyMode = False

' Close the book containing the text file.

    SourceBook.Close False

    Range("A1").Select

End Sub
```

Performance Calculations Module

```
' This is the overall routine that calculates the IMEP of the
' engine.  It first calculates the initial value of T1 and then
' writes the CEA input file.  Next it runs the CEA application and
' writes the output file.  It repeats this for all 5 engine processes
' (Initial Conditions, Compression, Combustion, Expansion,
' Exhaust Expansion).  Based on the information in the exhaust output
' file it calculates the burned gas fraction.  If this value is
' different from the initial guess it iterates until they converge.
' Once the gas fraction is correct it calculates all the work and
' performance parameters.

Const pe_pi = 1
Const Q_LHV = 44.4 'Heating Value of Isooctane


Sub Power_Calculations()

    Application.ScreenUpdating = False

    Worksheets("Power Calculations").Activate

' Pull in parameters from Input Module

    Ti = Range("D4"): r_c = Range("F3"): P1 = Range("B4")
    x_bi = Range("D5"): n_c = Range("F4"): V_d = Range("F7")

    Worksheets("State 1 Input").Range("F3").Value = P1 / 101.325

   count = 1
gamma = 1.4   ' Initial guess for gamma
10

' Find T1

    T1 = (1 - x_bi) * Ti / (1 - (1 / gamma / r_c) * _
    (pe_pi + (gamma - 1)))

' Write T1 and fuel Wt fraction to the State 1 input file

    Worksheets("State 1 Input").Range("H3").Value = T1
    Worksheets("State 1 Input").Range("E8").Value = 1 - x_bi
    Worksheets("State 1 Input").Range("E7").Value = 1 - x_bi

' Use a For Next Loop to run CEA for each State spreadsheet

    For Index = 3 To 13 Step 2

        ExportInputFile (Index)  ' Write the .INP file

        RunApp    ' Run CEA
```

```
' This builds in a time delay to allow the .OUT file to be written

        newHour = Hour(Now())
        newMinute = Minute(Now())
        newSecond = Second(Now()) + 2
        waitTime = TimeSerial(newHour, newMinute, newSecond)
        Application.Wait waitTime

        ImportOutputFile (Index + 1)  ' Read the .OUT file

' Once the "State 1 Output" file is written, the resultant volume
' is used to calculate v2

        If Index = 3 Then

            Rng = Worksheets("State 1 Output").Range("B1:G250")
            Rho1 = Application.WorksheetFunction.VLookup("RHO,", Rng,
4, False)
            Exp1 = Application.WorksheetFunction.VLookup("RHO,", Rng,
5, False)

            If Right(Rho1, 2) = "-1" Then
                v1 = 1 / (Left(Rho1, 6) * 0.1)
            ElseIf Right(Rho1, 2) = "-2" Then
                v1 = 1 / (Left(Rho1, 6) * 0.01)
            ElseIf Right(Rho2, 2) = "-3" Then
                v1 = 1 / (Left(Rho1, 6) * 0.001)
            Else: v1 = 1 / Left(Rho1, 6)
            End If

            If Exp1 = "1" Then
                v1 = 1 / (Left(Rho1, 6) * 10)
            ElseIf Exp1 = "2" Then
                v1 = 1 / (Left(Rho1, 6) * 100)
            ElseIf Exp1 = "3" Then
                v1 = 1 / (Left(Rho1, 6) * 1000)
            End If

            v2 = v1 / r_c

            Worksheets("State 2 Input").Range("H3").Value = v2

        End If


' After completing the "State 3 Output" sheet this calls a sub
' procedure to extract the combustion products.  These products
' are then used as inputs on the other sheets.


        If Index = 9 Then


            CombustionProd

        End If
```

```vba
    Next Index

' This pulls in the v2 and v5 values from the output sheets and
' calculates the burned gas fraction, x_b.  Since only the specific
' density is given, it takes the inverse.  However, when CEA writes
' exponentials, it doesn't use an E, only the exponent.  Therefore
' the value must be converted to a real number first.


    Rng = Worksheets("State 5 Output").Range("B1:G250")
    Rho5 = Application.WorksheetFunction.VLookup("RHO,", Rng, 4, False)
    Exp5 = Application.WorksheetFunction.VLookup("RHO,", Rng, 5, False)

    If Right(Rho5, 2) = "-1" Then
        v5 = 1 / (Left(Rho5, 6) * 0.1)
    ElseIf Right(Rho5, 2) = "-2" Then
        v5 = 1 / (Left(Rho5, 6) * 0.01)
    ElseIf Right(Rho5, 2) = "-3" Then
        v5 = 1 / (Left(Rho5, 6) * 0.001)
    Else: v5 = 1 / Left(Rho5, 6)
    End If


        If Exp5 = "1" Then
            v5 = 1 / (Left(Rho5, 6) * 10)
            ElseIf Exp5 = "2" Then
                v5 = 1 / (Left(Rho5, 6) * 100)
            ElseIf Exp5 = "3" Then
                v5 = 1 / (Left(Rho5, 6) * 1000)
            End If

    x_b = v2 / v5

' This checks to see how far the user's initial estimate deviates
' from the actual value.  If the value is off, the estimate is
' replaced with the new value and the calculations are done again.

    If Abs(x_b - x_bi) < 0.0005 Or count = 5 Then GoTo 20

        x_bi = x_b

' Since CEA calculates gamma, the value is used in T1

    Rng = Worksheets("State 1 Output").Range("B1:G250")
    gamma = Application.WorksheetFunction.VLookup("GAMMAs", Rng, 2,
False)

    count = count + 1

' This prevents the program from going into an infinite loops
' if the input parameters create unusable results.

    If count = 6 Then

        MsgBox ("Values Would Not Converge")
```

```
      GoTo 20

    End If

    GoTo 10   ' Go back to the beginning

20


' Once x_b is finalized, pull in parameters from the output sheets
' and calculate the various components of work.


' PROCESS 1-2:  ISENTROPIC COMPRESSION

' State 1 parameters

    Rng = Worksheets("State 1 Output").Range("B1:G250")

    u1 = Application.WorksheetFunction.VLookup("U,", Rng, 3, False)
    s1 = Application.WorksheetFunction.VLookup("S,", Rng, 3, False)
    Fuel_Air = 1 / Application.WorksheetFunction.VLookup _
    ("O/F", Rng, 3, False)

' State 2 parameters

    Rng = Worksheets("State 2 Output").Range("B1:G250")

    u2 = Application.WorksheetFunction.VLookup("U,", Rng, 3, False)
    s2 = Application.WorksheetFunction.VLookup("S,", Rng, 3, False)
    T2 = Application.WorksheetFunction.VLookup("T,", Rng, 3, False)
    P2 = Application.WorksheetFunction.VLookup("P,", Rng, 3, False) *
100


        Work_1to2 = u2 - u1  'Compression Work


' PROCESS 2-3:  CONSTANT VOLUME COMBUSTION

    v3 = v2

' Get the baseline u2 at 298.15 deg K

    Rng = Worksheets("State 2 std Output").Range("B1:G250")

    u2_std = Application.WorksheetFunction.VLookup("U,", Rng, 3, False)

' u3 is the difference between u2 and u2 std

        u3 = u2 - u2_std

    Rng = Worksheets("State 3 Output").Range("B1:G250")

    s3 = Application.WorksheetFunction.VLookup("S,", Rng, 3, False)
    T3 = Application.WorksheetFunction.VLookup("T,", Rng, 3, False)
```

```vba
    P3 = Application.WorksheetFunction.VLookup("P,", Rng, 3, False) *
100


'PROCESS 3-4:   ISENTROPIC EXPANSION

    v4 = v1

    Rng = Worksheets("State 4 Output").Range("B1:G250")
    u4 = Application.WorksheetFunction.VLookup("U,", Rng, 3, False)
    s4 = Application.WorksheetFunction.VLookup("S,", Rng, 3, False)
    T4 = Application.WorksheetFunction.VLookup("T,", Rng, 3, False)
    P4 = Application.WorksheetFunction.VLookup("P,", Rng, 3, False) *
100


        Work_3to4 = u4 - u3 'Expansion Work


' Calculate the Engine's Net Work

        Work_net = Abs(Work_3to4 + Work_1to2)

'PROCESS 4-5:   ISENTROPIC EXHAUST EXPANSION

    Rng = Worksheets("State 5 Output").Range("B1:G250")
    u5 = Application.WorksheetFunction.VLookup("U,", Rng, 3, False)
    s5 = Application.WorksheetFunction.VLookup("S,", Rng, 3, False)
    T5 = Application.WorksheetFunction.VLookup("T,", Rng, 3, False)
    P5 = Application.WorksheetFunction.VLookup("P,", Rng, 3, False) *
100


'Calculate IMEP

    IMEP = Work_net / (v1 - v2)      'EQ 28

    IMEP_corr = IMEP * 0.145 * 0.8  'Convert to PSI and apply .8 CF

    IMEP_total = IMEP_corr * n_c  'Multiply by number of cylinders


' Calculate Base SFC using indicated fuel conversion efficiency

    Eta_fi = Work_net / Fuel_Air / (1 - x_b) / Q_LHV / 1000

    SFC_Base = 3600 / Eta_fi / Q_LHV / 608.3


' Output results to spreadsheet

    Worksheets("Power Calculations").Activate

    Range("B14:F18").ClearContents
    Range("B20:B23").ClearContents
    Range("E20:E23").ClearContents
```

```
    Range("A27:E36").ClearContents

    Range("B15").Value = P1: Range("B14").Value = T1
    Range("B16").Value = v1: Range("B17").Value = u1
    Range("B18").Value = s1: Range("D7").Value = gamma

    Range("C14").Value = T2: Range("C15").Value = P2
    Range("C16").Value = v2: Range("C17").Value = u2
    Range("C18").Value = s2

    Range("D14").Value = T3: Range("D15").Value = P3
    Range("D16").Value = v3: Range("D17").Value = u3
    Range("D18").Value = s3

    Range("E14").Value = T4: Range("E15").Value = P4
    Range("E16").Value = v4: Range("E17").Value = u4
    Range("E18").Value = s4

    Range("F14").Value = T5: Range("F15").Value = P5
    Range("F16").Value = v5: Range("F17").Value = u5
    Range("F18").Value = s5

    Range("B20").Value = x_b: Range("B21").Value = Work_1to2
    Range("B22").Value = Work_3to4: Range("B23").Value = Work_net

    Range("E20").Value = IMEP: Range("E21").Value = IMEP_corr
    Range("E22").Value = IMEP_total: Range("E23").Value = SFC_Base


' Create Power Table

    Dim i As Integer

    RPM = 500

    RowRange = Worksheets(1).Range("A1", "A50")
    Rownum = Application.WorksheetFunction.Match _
    ("(RPM)", RowRange, 0)

    For i = 1 To 10

        Speed = RPM + 500 * i
        Cells(Rownum + i, 1).Value = Speed

        Gross_Power = Speed * V_d * IMEP_total / 2 / 396000
        Cells(Rownum + i, 2).Value = Gross_Power

        TFMEP = (0.97 + 0.15 * Speed / 1000 + 0.05 * (Speed / 1000) ^
2) * 14.7 * n_c * V_d * Speed / 2 / 396000
        Cells(Rownum + i, 3).Value = TFMEP


        Net_Power = Gross_Power – TFMEP
        Cells(Rownum + i, 4).Value = Net_Power


' Calculate SFC as a function of RPM.  Find BMEP, convert to
```

```
' Work and then Efficiency.

        BMEP = Net_Power * 2 * 396000 / V_d / Speed / 0.145 / n_c
        W_brake = BMEP * (v1 - v2)

        Eta_f = W_brake / Fuel_Air / (1 - x_b) / Q_LHV / 1000
        SFC_ = 3600 / Eta_f / Q_LHV / 608.3

        Cells(Rownum + i, 5).Value = SFC_

    Next i


End Sub
```

Combustion Products Module

```
' This subroutine finds the combustion products and writes them to
' the state 4 input file for use in the Isentropic Expansion.  In
' the output files the products all begin with a *, which causes an
' error if used in an input file.  Once the program finds the products
' it removes the *.  Also, the values are in exponential form.
' Since CEA does not use "E" when writing exponents the values are
' converted to the approriate real numbers.

Sub CombustionProd()

    Worksheets(11).Range("B9:B21").ClearContents
    Worksheets(11).Range("E9:E21").ClearContents

    Worksheets(10).Activate

' This finds the location of the combustion products, which varies
' based on the input parameters.

    RowRange = Worksheets(10).Range("B1", "B250")
    Rownum = Application.WorksheetFunction.Match("MOLE", RowRange, 0)


' Once the location is found, the individual components are extracted.

    i = 2

    Do Until IsEmpty(Cells(Rownum + i, 2).Value)

        component = Cells(Rownum + i, 2).Value

        Length = Len(component) - 1

' Some components don't begin with * (i.e. H20 )

            If Left(component, 1) <> "*" Then
                Length = Len(component)
            End If

        Moles = Cells(Rownum + i, 3).Value

            If Right(Moles, 2) = "-1" Then
                Molenum = Left(Moles, 6) * 0.1
            ElseIf Right(Moles, 2) = "-2" Then
                Molenum = Left(Moles, 6) * 0.01
            ElseIf Right(Moles, 2) = "-3" Then
                Molenum = Left(Moles, 6) * 0.001
            Else: Molenum = Left(Moles, 6)

            End If
```

```
        Worksheets(11).Cells(i + 7, 5).Value = Molenum
        Worksheets(11).Cells(i + 7, 2).Value = Right(component, Length)

        i = i + 1

    Loop


End Sub
```

Power Chart Module

```vba
' This module generates the power output chart based on the results
' of the power calculations and the table it generated.

Sub Powerchart()

'Application.ScreenUpdating = False
Dim Power_Plot As Chart

' Delete any present graphs

ActiveSheet.ChartObjects.Delete

' Add the chart to the sheet


Set Power_Plot = Charts.Add
Set Power_Plot = Power_Plot.Location(Where:=xlLocationAsObject,
Name:="Power Calculations")

    With Power_Plot

        .ChartType = xlXYScatterSmooth
        .SetSourceData Source:=Sheets("Power
Calculations").Range("A27:E36"), PlotBy _
        :=xlColumns
        .HasTitle = True
        .ChartTitle.Text = "Power Output"
        .ChartTitle.Font.Size = 12
        .SeriesCollection(4).AxisGroup = 2

'Set the location of the chart

        With .Parent

            .Top = Range("A41").Top
            .Left = Range("A41").Left
            .Width = Range("A41:F58").Width
            .Height = Range("A41:F58").Height

        End With

' Add Axis Titles

        .Axes(xlCategory, xlPrimary).HasTitle = True
        .Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = "RPM"
        .Axes(xlValue, xlPrimary).HasTitle = True
        .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "HP"
        .Axes(xlValue, xlSecondary).HasTitle = True
        .Axes(xlValue, xlSecondary).AxisTitle.Characters.Text = "SFC"
```

```
' Format the Y axis

        With .Axes(xlValue)

        .AxisTitle.Font.Size = 11
        .TickLabels.Font.Size = 10
        .TickLabels.NumberFormat = "General"
        .MajorUnit = 25

        End With


' Format the Secondary Y axis

        With .Axes(xlValue, xlSecondary)

        .AxisTitle.Font.Size = 11
        .TickLabels.Font.Size = 10
        .TickLabels.NumberFormat = "0.000"

        End With

' Format the X axis

        With .Axes(xlPrimary)

        .AxisTitle.Font.Size = 11
        .TickLabels.Font.Size = 10
        .MinorUnit = 1000
        .MinorTickMark = xlOutside

        End With

 'Resize the chart area

        With .PlotArea

            .Width = 290
            .Top = 18
            .Height = 195
            .Left = 15

        End With


' Label the legend

        .SeriesCollection(1).Name = "='Power Calculations'!R25C2"
        .SeriesCollection(2).Name = "='Power Calculations'!R25C3"
        .SeriesCollection(3).Name = "='Power Calculations'!R25C4"
        .SeriesCollection(4).Name = "='Power Calculations'!R25C5"

' Resize the legend

        .Legend.Width = 55
        .Legend.Left = 335
        .Legend.Top = 52
```

```
            .Legend.Height = 101

            .Legend.Font.Size = 8


     End With

Range("A50").Select

End Sub
```

Engine Deck Module

```
' This Module Creates the Engine Deck.  It asks for a
' maximum altitude and then calculates the temperature and pressure
' for those altitudes.  These values are then used in the power
' calculations as T1 and P1.  The main power calculation module
' is used here for each value of T1 and P1.


'These are the constants used in the calculations.
Const Q_LHV = 44.4 'Isooctane
Const pe_pi = 1

Sub Engine_Deck()

    Application.ScreenUpdating = False

    Sheets("Engine Deck").Select

    Range("A3:F166").ClearContents

     ' Retrieve the input parameters

    Worksheets("Power Calculations").Activate

    delta_T = Range("B8"): V_d = Range("F7")
    r_c = Range("F3"): x_bi = Range("D5")
    n_c = Range("F4"): gamma = Range("D7")

' This queries the user for a maximum altitude

    alt = InputBox("Enter the Maximum Altitude in feet")

10

    If alt = "" Then alt = 5000
    If alt <= 999 Then alt = InputBox("Altitude must be at least 1000
ft"): _
    GoTo 10
    If alt > 20000 Then alt = InputBox("That is too high - Try Again"):
_
    GoTo 10

    std_diff = InputBox("Enter the Temperature above Standard Day (deg
F)")

    If std_diff = "" Then std_diff = 0

    T_diff = std_diff / 1.8


' This creates the actual charts
```

```
    counter = 2

    For i = 1 To alt / 1000

        Altitude = 1000 * i
        Worksheets(2).Cells(i + counter, 1).Value = Altitude

'Find the new Ti and P1 to use in the power calculations

' Calculate the Temperature at altitude

        T_meas = 59 - 0.00356 * (Altitude)

' Calculate the Pressure at altitude (Atm)

        P_meas = 2116 * ((T_meas + 459.67) / 518.6) ^ 5.256


        P1 = P_meas * 47.88 / 101.325 / 1000 ' Convert to atm

        Ti = (T_meas + 459.67) / 1.8 + delta_T + T_diff

' Output the Temp and Press

        Worksheets(2).Cells(i + counter, 2).Value = (Ti * 1.8 - 459.67)
- delta_T * 1.8
        Worksheets(2).Cells(i + counter, 3).Value = P1
' The power calculations procedures are now performed for
' each altitude

20

' Find T1

    T1 = (1 - x_bi) * Ti / (1 - (1 / gamma / r_c) * _
    (pe_pi + (gamma - 1)))

' Write T1, P1, and fuel Wt fraction to the State 1 input file

    Worksheets("State 1 Input").Range("H3").Value = T1
    Worksheets("State 1 Input").Range("F3").Value = P1
    Worksheets("State 1 Input").Range("E8").Value = 1 - x_bi

' Use a For Next Loop to run CEA for each State spreadsheet

    For Index = 3 To 13 Step 2

        ExportInputFile (Index)  ' Write the .INP file

        RunApp    ' Run CEA

' This builds in a time delay to allow the .OUT file to be written

        newHour = Hour(Now())
        newMinute = Minute(Now())
        newSecond = Second(Now()) + 2
```

```
        waitTime = TimeSerial(newHour, newMinute, newSecond)
        Application.Wait waitTime

        ImportOutputFile (Index + 1)   ' Read the .OUT file

' Once the "State 1 Output" file is written, the resultant volume
' is used to calculate v2

        If Index = 3 Then

            Rng = Worksheets("State 1 Output").Range("B1:G250")
            Rho1 = Application.WorksheetFunction.VLookup("RHO,", Rng,
4, False)
            Exp1 = Application.WorksheetFunction.VLookup("RHO,", Rng,
5, False)

            If Right(Rho1, 2) = "-1" Then
                v1 = 1 / (Left(Rho1, 6) * 0.1)
            ElseIf Right(Rho1, 2) = "-2" Then
                v1 = 1 / (Left(Rho1, 6) * 0.01)
            ElseIf Right(Rho2, 2) = "-3" Then
                v1 = 1 / (Left(Rho1, 6) * 0.001)
            Else: v1 = 1 / Left(Rho1, 6)
            End If

            If Exp1 = "1" Then
                v1 = 1 / (Left(Rho1, 6) * 10)
            ElseIf Exp1 = "2" Then
                v1 = 1 / (Left(Rho1, 6) * 100)
            ElseIf Exp1 = "3" Then
                v1 = 1 / (Left(Rho1, 6) * 1000)
            End If

            v2 = v1 / r_c

            Worksheets("State 2 Input").Range("H3").Value = v2

        End If
' After completing the "State 3 Output" sheet this calls a sub
' procedure to extract the combustion products.  These products
' are then used as inputs on the other sheets.

        If Index = 9 Then

            CombustionProd

        End If

    Next Index

' This uses the v2 and v5 values from the output sheets to
' calculate the burned gas fraction, x_b.  Since only the specific
' density is given, it takes the inverse.


    Rng = Worksheets("State 5 Output").Range("B1:G250")
```

```vbnet
    Rho5 = Application.WorksheetFunction.VLookup("RHO,", Rng, 4, False)
    Exp5 = Application.WorksheetFunction.VLookup("RHO,", Rng, 5, False)

    If Right(Rho5, 2) = "-1" Then
        v5 = 1 / (Left(Rho5, 6) * 0.1)
    ElseIf Right(Rho5, 2) = "-2" Then
        v5 = 1 / (Left(Rho5, 6) * 0.01)
    ElseIf Right(Rho5, 2) = "-3" Then
        v5 = 1 / (Left(Rho5, 6) * 0.001)
    Else: v5 = 1 / Left(Rho5, 6)
    End If


    If Exp5 = "1" Then
        v5 = 1 / (Left(Rho5, 6) * 10)
    ElseIf Exp5 = "2" Then
        v5 = 1 / (Left(Rho5, 6) * 100)
    ElseIf Exp5 = "3" Then
        v5 = 1 / (Left(Rho5, 6) * 1000)
    End If

    x_b = v2 / v5

' This checks the initial estimate of x_b against the calculated value
' and iterates as required.

    If Abs(x_b - x_bi) < 0.005 Then GoTo 30

        x_bi = x_b

' Since CEA calculates gamma, the value is used in T1

    Rng = Worksheets("State 1 Output").Range("B1:G250")
    gamma = Application.WorksheetFunction. _
    VLookup("GAMMAs", Rng, 2, False)

    GoTo 20  ' Go back to the beginning

30

' Once x_b is finalized, pull in parameters from the output sheets
' and calculate the various components of work.  Since most of the
' parameters are not used for output in the deck, they are not called.


' PROCESS 1-2:  ISENTROPIC COMPRESSION

' State 1 parameters

    Rng = Worksheets("State 1 Output").Range("B1:G250")

    u1 = Application.WorksheetFunction.VLookup("U,", Rng, 3, False)
    Fuel_Air = 1 / Application.WorksheetFunction.VLookup _
    ("O/F", Rng, 3, False)

' State 2 parameters
```

```vba
    Rng = Worksheets("State 2 Output").Range("B1:G250")

    u2 = Application.WorksheetFunction.VLookup("U,", Rng, 3, False)

        Work_1to2 = u2 - u1  'Compression Work


' PROCESS 2-3:  CONSTANT VOLUME COMBUSTION

' Get the baseline u2 at 298.15 deg K

    Rng = Worksheets("State 2 std Output").Range("B1:G250")

    u2_std = Application.WorksheetFunction.VLookup("U,", Rng, 3, False)

' u3 is the difference between u2 and u2 std

        u3 = u2 - u2_std

    Rng = Worksheets("State 3 Output").Range("B1:G250")

'PROCESS 3-4:  ISENTROPIC EXPANSION

    Rng = Worksheets("State 4 Output").Range("B1:G250")
    u4 = Application.WorksheetFunction.VLookup("U,", Rng, 3, False)

        Work_3to4 = u4 - u3 'Expansion Work


' Calculate the Engine's Net Work

        Work_net = Abs(Work_3to4 + Work_1to2)

'Calculate IMEP

    IMEP = Work_net / (v1 - v2)      'EQ 28

    IMEP_corr = IMEP * 0.145 * 0.8  'Convert to PSI and apply .8 CF

    IMEP_total = IMEP_corr * n_c   'Multiply by number of cylinders

' Based on the IMEP, calculate the power output as a function
' of engine RPM.

  Worksheets("Engine Deck").Activate

  Dim j As Integer

          For j = 1 To 10

            RPM = 500 + j * 500
            Cells(j + counter + i - 1, 4).Value = RPM

            Gross_Power = RPM * V_d * IMEP_total / 2 / 396000

            TFMEP = (0.97 + 0.15 * RPM / 1000 + 0.05 * (RPM / 1000) ^
2) * 14.7 * n_c * V_d * RPM / 2 / 396000
```

```
            Net_Power = Gross_Power - TFMEP
            Cells(j + counter + i - 1, 5).Value = Net_Power

' Calculate SFC as a function of RPM.  Find BMEP, convert to Work
' and then Efficiency.

            BMEP = Net_Power * 2 * 396000 / V_d / RPM / 0.145 / n_c
            W_brake = BMEP * (v1 - v2)

            Eta_f = W_brake / Fuel_Air / (1 - x_b) / Q_LHV / 1000
            SFC_ = 3600 / Eta_f / Q_LHV / 608.3

            Cells(j + counter + i - 1, 6).Value = SFC_

        Next j

        counter = counter + 11

    Next i

    Range("A1").Select

End Sub
```

Miscellaneous Functions Module

```
Sub Clear_Output()

' Delete any present graphs

ActiveSheet.ChartObjects.Delete
Range("B14:F18").ClearContents
Range("B20:B23").ClearContents
Range("E20:E23").ClearContents
Range("A27:E36").ClearContents

Worksheets(2).Range("A3:F166").ClearContents


End Sub

Sub Clear_Input()
Range("B3").ClearContents
Range("B5").ClearContents
Range("B7").ClearContents
Range("D5:D7").ClearContents
Range("F3:F6").ClearContents

End Sub
```

APPENDIX D

Piston Engine Performance Program (PEPP) User's Manual

Overview

The Piston Engine Performance Program (PEPP) is an Excel based engine analysis program that predicts piston engine performance. Presently, PEPP works only for naturally aspirated, spark ignition engines. The purpose of PEPP is to provide aerospace vehicle designers with the ability to model piston engine performance in a myriad of flight conditions. It uses a Constant Volume (Otto) Ideal Cycle combined with a Fuel-Air fluid model for the calculations. The fuel is iso-octane (C_8H_{18}) and the engine is assumed to be running at wide open throttle. The calculations iterate to find the burned gas fraction, x_b, which is the ratio of burned fuel that gets recycled into the cylinder to the total fuel-air volume. An initial guess is required to begin the calculations, which are repeated until x_b converges to itself. Normal values of x_b range from 0.01 to 0.1; the initial x_b value does not influence the final calculations.

Setup

PEPP useS a thermodynamic equilibrium program called CEA. In order to run PEPP, CEA.exe and the accompanying files must be present. These files are all in the "PEPP Code" folder that comes with the parent code. Use only these CEA files as they have been modified specifically for use by the PEPP code. To minimize computational time it is recommended that these files first be placed on the hard drive. Because CEA writes output files during the calculations, the program will not run from a CD.

Prior to running the code, the location of the "PEPP Code" folder must be entered into PEPP. Enter the folder location in worksheet "State 1 Input" in cell A1. See Figure 26 below. Once the folder's location is entered, PEPP is ready to execute.
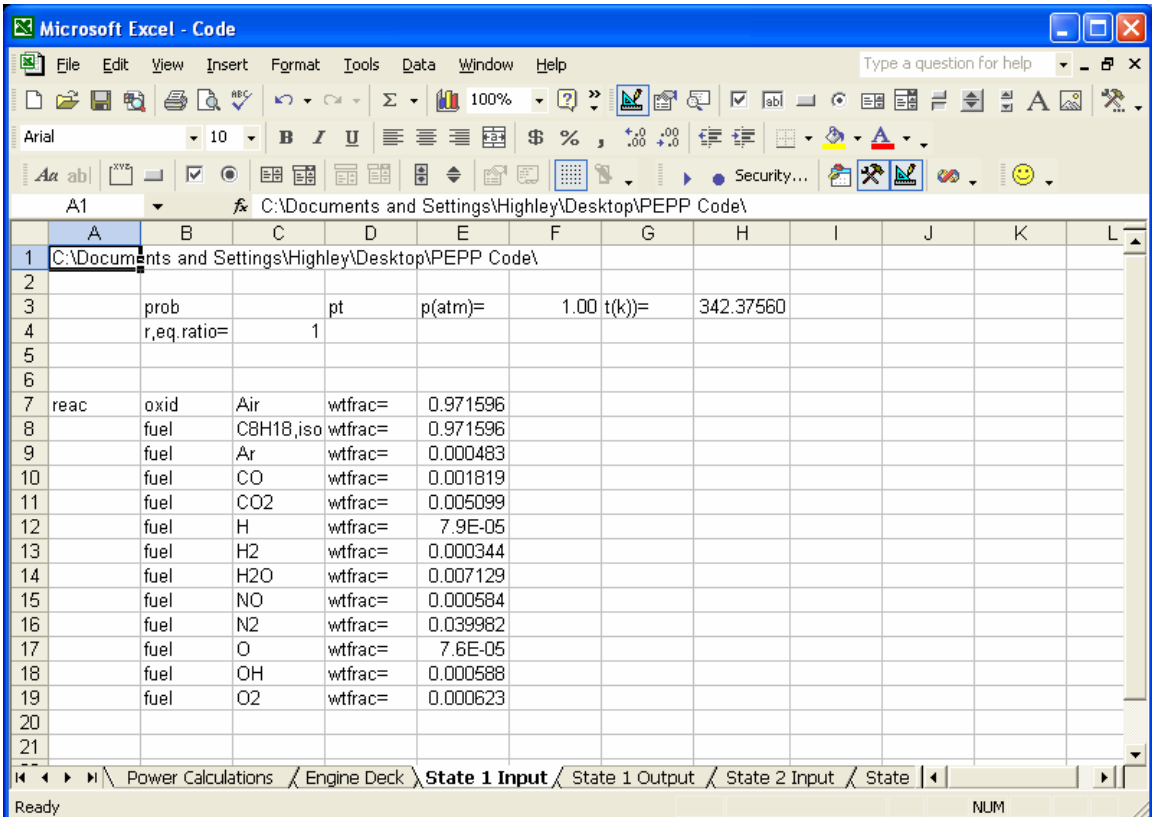

Figure 26: PEPP Code Location

Using PEPP

When using PEPP, only the first two worksheets are used ("Performance Calculations" and "Engine Deck," respectively). The remainder provide state point data

for the performance calculations and are not otherwise useful. PEPP is divided into four separate modules: *Input, State Points and Work, Power Output*, and *Engine Deck*. All modules except *Engine Deck* are found on the "Performance Calculations" worksheet.

The *Input* module is the user interface, and is where all the engine parameters are entered. This section is seen below in Figure 27.

Operating Variables (Only Objects in Green are Changeable)					
Parameter	Value	Parameter	Value	Parameter	Value
P_1 (PSI)	14.696	T_{inlet} (°F)	100	r_c	8.5
P_1 (kPA)	101.33	T_{inlet} (°K)	310.93	n_c	4
$T_{ambient}$ (°F)	59	x_b (Initial)	0.025	B/L	0.96
$T_{ambient}$ (°K)	288.15	?	1	B (in)	3.32
? T (°F)	41	?	1.3667	V_d (in^3)	29.94
? T (°K)	22.78				

Reset Input   Calculate   Clear Output

Figure 27: Input Module

Only the green sections can be changed by the user. All other sections are locked. The major ambient parameters are entered in English units, which are automatically converted to SI units. The input parameters are described below:

- P_1. This is the ambient pressure conditions of the static engine.

- $T_{ambient}$. The ambient temperature of the static engine

- ?T. The difference between the ambient temperature and the inlet temperature. Primarily a design feature, this accounts for variances in the engine inlet placement and possible temperature increases from radiant heat.

112

- T_{inlet}. This is the engine inlet temperature, given by $T_{ambient} + ?T$.

- x_b (initial). The initial guess for the burned gas constant

- ? . The fuel-air equivalence ratio. This parameter is used to model either a lean (? < 1) or rich (? > 1) mixture.

- ? – The ratio of specific heats. This value is not actually an input, but calculated by the CEA program.

- r_c. The engine's compression ratio.

- n_c. The number of cylinders in the engine.

- B/L. The bore to stroke ratio. For design purposes this can be set to 1.

- B. The bore of the cylinder.

- V_d. This is the engine's displaced volume and is calculated based on the geometry and number of cylinders.

At the very bottom of the input section are three macro buttons. The "Reset Input" button automatically resets the input parameters to start a new engine model. The "Calculate" button begins program execution once all the parameters have been set. Finally, the "Clear Output" button erases all previous output data.

Once the "Calculate" button is pressed, PEPP begins the performance calculations, and will iterate until the burned gas fraction, x_b, converges to itself. The results of the calculations are sent to the *State Points and Work* and *Power Output* modules. The *State Point and Work* module appears below in Figure 28.

State Points

	1	2	3	4	Exh
Temp (°K)	342.38	719.30	2997.54	2028.58	1421.34
Pressure (kPA)	101.33	1809.40	7720.30	601.36	101.32
Volume (m³/kg)	0.9788	0.1151	0.1151	0.9788	4.0632
u (kJ/kg)	-2946.50	-2629.20	351.99	-1222.13	-1930.10
s (kJ/kg °K)	7.13	7.13	8.83	8.83	8.83
x_b	0.028339		IMEP (kPA)	1455.31	
W_{1-2} (kJ/kg air)	317.30		IMEP (PSI)	168.82	Corrected
W_{3-4} (kJ/kg air)	-1574.12		IMEP (PSI)	675.26	Engine
W_{Net} (kJ/kg air)	1256.82		SFC (lb/HP*hr)	0.3198	

Figure 28: State Points and Work Module

State 1 refers to the fuel-air mixture prior to the compression process, and sets the mixture's initial conditions. State 2 gives the fluid's properties after the isentropic compression, while the results of the combustion process are listed in State 3. Next, State 4 gives the burned mixture's properties after the isentropic expansion following the combustion process. Finally, the EXH or exhaust state gives the fluid's properties after the isentropic expansion to the atmospheric pressure as it enters the exhaust valve. The additional parameters are:

- x_b. This is the final burned gas fraction calculated by the program.

- W_{1-2}. The engine's compression work.

- W_{3-4}. This is the engine's expansion work.

- W_{Net}. The net work produced by the engine.

- IMEP. This is the engine's indicated mean effective pressure.

- IMEP (Corrected). This is the engine's IMEP after a 0.8 empirical correction is applied to account for differences between the actual cycle and the ideal cycle used for the power calculations.

- IMEP (Engine). The total engine IMEP (IMEP * n_c).

- SFC. The base Specific Fuel Consumption of the engine, based on the engine's indicated work. Because it is indicated, this value is substantially lower than the actual values.

If more detailed thermodynamic information is required the complete CEA output files are captured in the respective state point output worksheets.

PEPP's third module calculates the engine's power output as a function of engine speed. The user then has the option of plotting the data and/or creating an engine deck. The power output table appears in Figure 29.

Engine Speed	Gross Power	Losses	Net Power	SFC
(RPM)	(HP)	(HP)	(HP)	(lb/HP*hr)
1000	25.53	2.60	22.93	0.4450
1500	38.29	4.36	33.93	0.4511
2000	51.05	6.53	44.52	0.4584
2500	63.81	9.21	54.60	0.4671
3000	76.58	12.47	64.11	0.4775
3500	89.34	16.40	72.95	0.4895
4000	102.10	21.07	81.03	0.5036
4500	114.87	26.58	88.29	0.5200
5000	127.63	33.01	94.62	0.5391
5500	140.39	40.43	99.96	0.5614

Plot Results	Engine Deck

Figure 29: Power Output Module

Based on the IMEP, the engine's gross power is calculated. The friction losses are then calculated and subtracted from the gross power to obtain the engine's net power output. Finally, the engine's specific fuel consumption is calculated. These parameters are discussed in more detail below.

- Gross Power. The engine's indicated or gross power output, without accounting for any losses

- Losses. The pumping, mechanical, and parasitic losses seen by the engine. These are given by an empirical equation and based on engine speed only.

- Net Power. The engine's brake power – the actual output at the crankshaft.

- BSFC. The engine's Brake Specific Fuel Consumption

At the bottom of the module are two macro buttons. The "Plot Results" button plots the power output and Specific Fuel Consumption versus the engine's speed. An example chart is shown below in Figure 30.
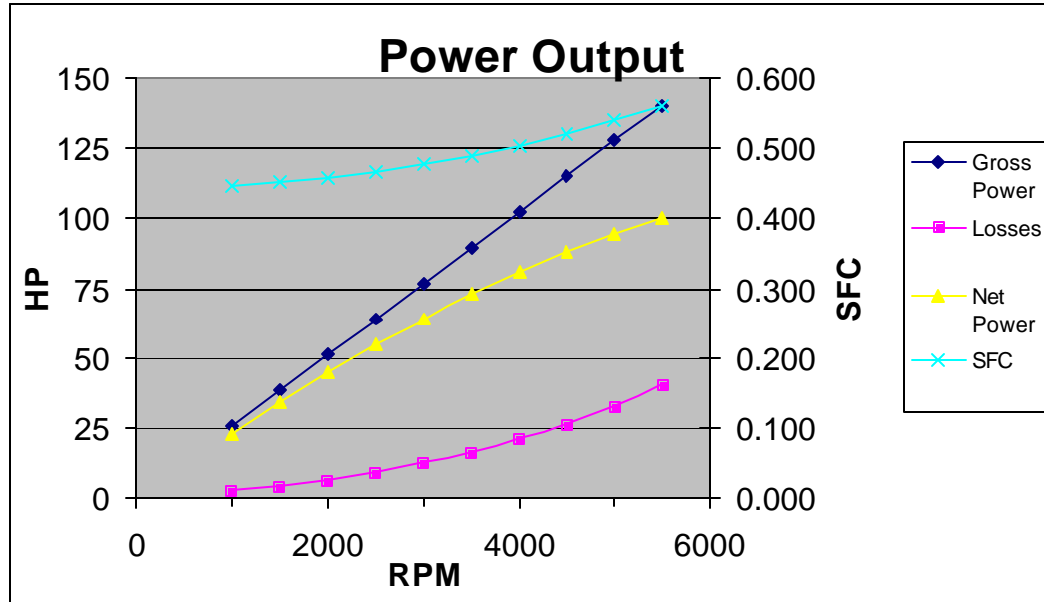
Figure 30: Engine Power Curve

The second button is the "Engine Deck" button. This prompts the user for the *Engine Deck* parameters, which can also be accessed through the "Engine Deck" worksheet. The *Engine Deck* enables the designer to determine the engine's performance in a specific flight regime and under certain atmospheric conditions by calculating the engine's power as a function of both altitude and engine speed. All parameters other than atmospheric data come from the *Input* module.

The *Engine Deck* is accessed two ways. The first is on the main worksheets; "Performance Calculations." The "Engine Deck" button in the power output section starts the engine deck calculations. Secondly, on the "Engine Deck" worksheet clicking the "Create Deck" button will also begin the analysis. Once the calculations begin, the following prompt appears:
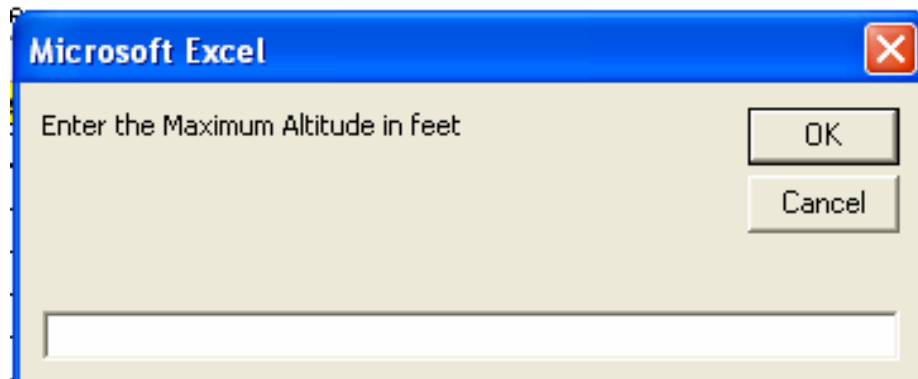
Figure 31: Altitude Prompt

Enter the maximum altitude for the engine deck calculations. If nothing is entered, the default value is 5000 ft. Once the altitude is entered, another input box is displayed, allowing for a temperature deviation:
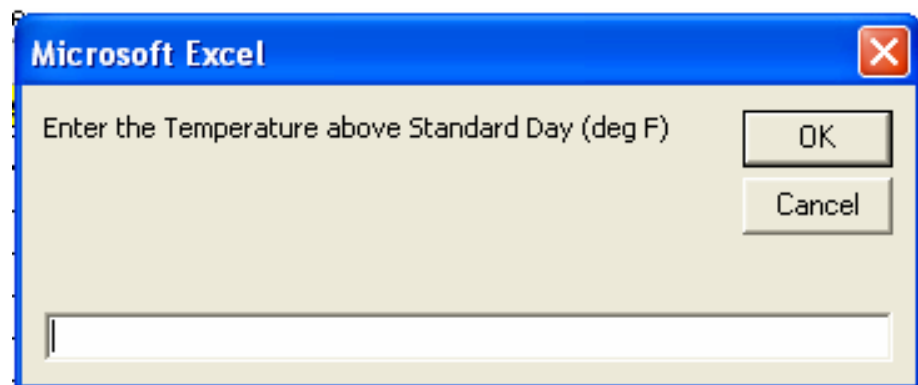

Figure 32: Temperature Deviation Prompt

The engine deck uses a standard atmospheric model to calculate temperature data at altitude (59° F at SL). If the desired performance band of the engine is at a higher

temperature, the specific deviation must be entered here. If nothing is entered, the default

value is 0. The engine deck output appears as follows:

Altitude	Temp	Press	Engine Speed	Power	BSFC
(Ft)	(°F)	(Atm)	(RPM)	(HP)	(lb/HP*hr)
1000	55.44	0.965	1000	26.32	0.4300
1000	55.44	0.965	1500	39.02	0.4350
1000	55.44	0.965	2000	51.30	0.4412
1000	55.44	0.965	2500	63.09	0.4485
1000	55.44	0.965	3000	74.29	0.4570
1000	55.44	0.965	3500	84.82	0.4670
1000	55.44	0.965	4000	94.60	0.4785
1000	55.44	0.965	4500	103.55	0.4918
1000	55.44	0.965	5000	111.59	0.5071
1000	55.44	0.965	5500	118.62	0.5247
2000	51.88	0.930	1000	25.52	0.4311
2000	51.88	0.930	1500	37.81	0.4364
2000	51.88	0.930	2000	49.70	0.4427
2000	51.88	0.930	2500	61.08	0.4503
2000	51.88	0.930	3000	71.88	0.4591
2000	51.88	0.930	3500	82.01	0.4695
2000	51.88	0.930	4000	91.39	0.4815
2000	51.88	0.930	4500	99.94	0.4953
2000	51.88	0.930	5000	107.57	0.5113
2000	51.88	0.930	5500	114.20	0.5298

Figure 33: Engine Deck

Because the program executed the full calculation cycle at each altitude, the

computation time increases with a high maximum altitude. The data in Figure 33 is the

only output of the engine deck. No state point or work data is displayed on the "Power

Calculations" page.

REFERENCES

A Propulsion Study of Horizontally-Opposed-Piston Internal-Combustion Engines for General Aircraft, Avco Lycoming Division, 1970.

Atzler, Frank. *On the Future of the Piston Engine with Internal Combustion: An Overview*, Marie Curie Fellowship Conference, May 2001.

Cengel, Yunus A. and Boles, Michael A., *Thermodynamics: An Engineering Approach*, McGraw-Hill, 1989.

Gordon, Sanford and McBride, Bonnie, *Computer Program for Calculation of Complex Chemical Equilibrium Compositions and Applications*, NASA Reference Publication 1311, 1994.

Heywood, John B., *Internal Combustion Engine Fundamentals*, McGraw-Hill, 1988.

NASA Glenn Research Center CEA Homepage: http://www.grc.nasa.gov/WWW/CEAWeb

"Earth Atmosphere Model," http://www.grc.nasa.gov/WWW/K-12/airplane/atmos.html

Schrage, Daniel P., AE8803 B Class Notes, Fall 2003.

Taylor, Charles F., *The Internal Combustion Engine in Theory and Practice*, M.I.T. Press, 1966.