

University of Wales Trinity Saint David

School of Engineering

COMBUSTION AND EMISSION STUDY MODELLING FUELS AND NO_x

Powertrain and Sustainability

M6X01206

Ryan James 0751093

29/12/2019

Contents

1	Introduction.....	1
1.1	Aims.....	1
1.2	Objectives	1
2	Modelling App.....	2
2.1	All Fuels.....	2
2.2	Varying Fuel 1 Percentage.....	3
2.3	Other Reactant Values	3
2.4	Total of each Element in the Reaction.....	3
2.5	Balancing the Reaction	4
2.5.1	Calculating x	4
2.5.2	Calculating Exhaust Nitrogen	4
2.5.3	Fuel Equivalence Ratio	5
2.5.4	Balancing for FER.....	5
3	Task 1	7
3.1	(a) Calculate the mass percentages of methanol and isooctane in this blend	7
3.1.1	Molar Masses of Elements	7
3.1.2	Mass Percentage of Fuel	7
3.2	(b) Calculate the Air: Fuel mass ratio for stoichiometric combustion in air.....	8
3.2.1	Air Mass.....	8
3.2.2	Fuel Mass	8
3.2.3	Air Fuel Ratio	8
3.3	(c) From the enthalpy of formations of the fuel blend and products, calculate the lower heating value of the blend in MJ/kg.	8
3.3.1	Enthalpy of Fuels	8
3.3.2	Enthalpy of Exhaust Molecules	8
3.3.3	Lower Heating Value	9
3.4	(d) Explain enthalpy of formation.....	9

3.5	(e) Explain the difference between higher and lower heating values.	9
3.6	(f) Calculate the mass of CO ₂ produced per kg of fuel burned.	10
3.7	Carbon Monoxide (CO)	11
3.7.1	Calculation the percentage of CO	11
3.8	(g) Describe the properties of an “Ideal” fuel.	11
4	Task 2	11
4.1	Overview	11
4.2	NO _x Model for Isooctane Running on the Lean side of Stoichiometric.....	12
4.2.1	Enthalpy for reaction at a range of FER	12
4.2.2	Tables of Enthalpy’s for Balanced equation at all FER and range of EGR	12
4.2.3	Looking up Temperature from Table.....	13
4.2.4	Increasing Flame Temperature with Pressure.....	14
4.3	Merker’s Zeldovich model.....	16
4.3.1	Calculating alpha	16
4.3.2	Concentrations of oxygen and nitrogen in Combustion	17
4.3.3	NO _x Kmols/Degree Merker.....	17
4.3.4	NO _x PPM Merker	17
5	Conclusions.....	20
6	Extra Work	21
6.1	McAlister Zeldovich model	21
6.2	Heywood Zeldovich model	22
6.2.1	Graphs for Each NO _x model with 0 EGR against FER.....	22
6.3	Varying RPM	23
6.4	Varying Polytropic Index	23
6.5	Varying Gas Constant	25
6.6	Varying Compression Ratio (Combustion Volume)	25
6.7	Varying Volumetric Efficiency	26
6.8	Fuels Compared.....	28

7	References.....	28
8	Appendices.....	29
9	Script.....	29

1 Introduction

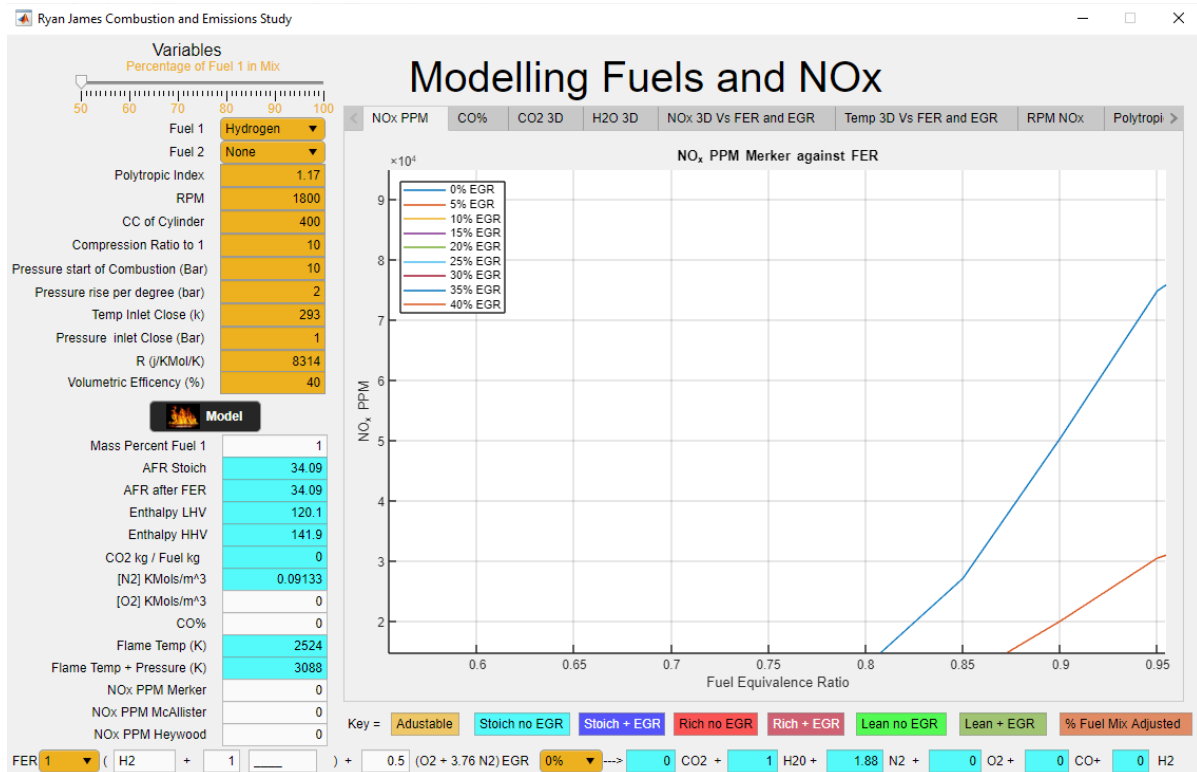
1.1 Aims

To perform a basic study on fuel combustion first through chemical balancing to work out masses of reactants then using enthalpy to estimate Combustion Flame Temperatures and finally the Zeldovich Model to approximate Nitrous Oxides (NO_x) Production and the effect Exhaust Gas Recirculation (EGR) has NO_x .

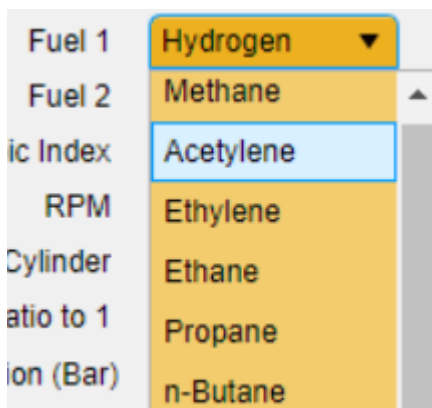
1.2 Objectives

To give a better understanding of the combustion process the study has been applied to MATLAB Apps. To make the App interactive for the end user variables were applied to calculations (using a full range of FER throughout to show NO_x and Carbon Monoxide (CO)).

2 Modelling App



2.1 All Fuels



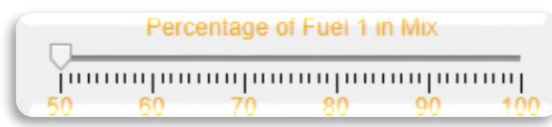
There are 2 drop downs to select different fuels when the app is opened it is pre-set to the brief part 1 fuels.

The following shortened script section shows how the fuels are selected to the variable Fuel_1 for calculations and the string is added to the balanced equation. Fuel 2 is done in a similar manner and has the option of none if performing a single fuel model.

```
% selects Fuel 1 Properties
Fuel_1 = app.Fuel1DropDown.Value;
if app.Fuel1DropDown.Value == 'a' % Methane
    Fuel_1 = [1;4;0;0;74.4];
%[carbon;hydrogen;oxygen;nitrogen;Enthalpy]
    app.EditField_4.Value = ('CH4'); % string for balanced
equation
elseif app.Fuel1DropDown.Value == 'b' % Acetylene
    Fuel_1 = [2;2;0;0;8.7];
    app.EditField_4.Value = ('C2H2');
```

Within the [] are the molar masses for elements that make up the fuel and the enthalpy ([Carbon;Hydrogen;Oxygen;Nitrogen;Enthalpy]).

2.2 Varying Fuel 1 Percentage



This sets the user adjustable percentage between 50 and 100% for Fuel_1 as a total blend if the user wants less than 50%, they should Swap Fuel 1 and 2 Selections.

```
a=app.PercentageofFuel1inMixSlider.Value; % a is the percentage of Fuel 1
selected on slider
```

b turns the percentage into a Ratio: 1 for Fuel_2.

```
% b = turns the percentage of fuel 1 (a) into a ratio
b = (100-a)/a;
```

2.3 Other Reactant Values

Other Compounds that are used each time that are not selected as follows (_L = lower heating value _H higher heating value).

```
air          = [0;0;2;7.52];
CO2          = [1;0;2;0;393.5];
CO           = [1;0;1;0;110.5];
H2O_L       = [0;1;2;0;241.8];
H2O_H       = [0;1;2;0;285.8];
```

2.4 Total of each Element in the Reaction

To Balance the chemical equation the total amount of each element unbalanced needs to be obtained.

```
% React totals the hydrogen carbons oxygens and
nitrogens going in
      % the combustion and keeps the totals of
elements in separate rows
      React =
      [(Fuel_1(1:4,:)), (b*Fuel_2(1:4,:)), air(1:4,:)];
```

2.5 Balancing the Reaction

The equation needs to be balanced as the laws of conservation of mass, say that the quantity of each element does not change through a reaction (Konicek *et al.*, 1973). This is the reaction for Task 1 Balanced from the App. the colours for the Balancing Coefficients (n_n) change dependant on if the equation is rich/ Lean/ Stoichiometric (when each side of the reaction has the correct number of molecules for complete combustion).

FER 1 (C8H18 + 1 CH4O) + 14 (O2 + 3.76 N2) EGR 0% → 9 CO2 + 11 H2O + 52.64 N2 + 0 O2 + 0 CO + 0 H2

In script this equation's nomenclature is:

FER*(Fuel_1+b*Fuel_2)+x(air)+EGR→nCO2(CO2)+nH2O(H2O)+nN2(N2)+nO2(O2)+nCO(CO)+nH2(H2)

2.5.1 Calculating x

The following balances the entire equation when FER =1 and EGR = 0%. As x will not change with FER and EGR only the n values change. nCO2 and nH2O need to be calculated first from the carbon and hydrogen in the fuel.

```
% to Calculate X the balance must be done at the stoichometric
% FER
nCO2 = (sum(React(1,:))); % balance of CO2 out
nH2O = (sum(React(2,:))/2); % balance of H2o Out
```

when x is calculated with the nCO2 and nH2O also any oxygen in the fuels need to be taken into consideration and the proportion of them (if a small percentage of Fuel_2 is selected) therefore fuel_2 is multiplied by b.

```
x = (((2*nCO2)+nH2O)-(Fuel_1 (3,1) + (Fuel_2(3,1)*b)))/2; % the amount of air balance
```

2.5.2 Calculating Exhaust Nitrogen

The balance for nitrogen only changes with fuel selected as FER changes fuel amount, not air in.

```
∴ nN2 = (x)*3.76; % Balance of Nitrogen
```

2.5.3 Fuel Equivalence Ratio

This is changing the ratio of Fuel/ Fuels to air from the calculated stoichiometric Balance. So, when keeping the Ratio of air, the same(x) increasing the FER by 10% (making it 10% Richer) will Give a FER of 1.1.

To view the effect on flame temperatures and emissions a range of FER has been taken from 0.6 to 1.24.

```
FER = [(0.6:0.05:0.9),(0.95:0.01:1),(1.02:0.02:1.24)];
% range of Phi in Brief plus rich range
```

2.5.4 Balancing for FER

2.5.4.1 nO2 with FER script

The following script works out the Balancing Coefficient for nO2 when FER is Lean and fills with zeros when rich.

```
nO2 = [(x.*(1-FER(1,1:13))),zeros(1,12)]; % Balance for O2 out when lean
```

2.5.4.2 Balancing nH2O, nCO2, nCO and nH2 when there is Carbon in the Fuel

```
% any but not only hydrogen
if ((sum(React(1,:)))>=0.01 % if there is Carbon

nH2O = (sum(React(2,:))/2).*FER; %Balance for H2O when rich and lean

nCO2= [((sum(React(1,:)).*FER(1,1:13)) ((2*x - ((nH2O(1,14:end)))-
...(sum(React(1,:)).*FER(1,14:25)))+(Fuel_1(3,1)+(Fuel_2(3,1)*b))*FER(1,14:
25)))]; % balance CO2 for Phi

nCO = abs([(zeros(1,13)), (nCO2(1,14:end)-
...((sum(React(1,:)).*FER(1,14:25)))]); % Balance For CO when rich

nH2 = (zeros(1,25)); % Balance for Hydrogen in exhaust
```

- nH2O is simply sum of the hydrogen divided by 2 multiplied by FER
- nCO2 is the sum of the carbon multiplied by FER when lean. When Rich it is 2*x minus the rich nH2O minus the rich carbon multiplied by FER plus the fuel oxygen content multiplied by FER this gives nCO2 less the CO
- nCO is simply nCO2 (original) multiplied by FER minus pre calculated nCO2 to leave nCO (when lean it is just zeros).

- All H_2 is used so when rich a lack of oxygen leads to CO, not exhaust $H_2 \therefore n_{H_2} = 0$

2.5.4.3 Balancing n_{H_2O} , n_{CO_2} , n_{CO} and n_{H_2} when the Fuel is only Hydrogen

```

                %hydrogen only
elseif ((sum(React(1,:)))==0 % if there is no carbon (just
hydrogen)
    nCO2= ((sum(React(1,:)).*FER); % balance CO2 for Phi
    nCO = abs((zeros(1,25))); % Balance For CO when rich
    nH2O = [((sum(React(2,:))/2).*FER(1,1:13)),(ones(1,12))];
%Balance for H2O when rich and lean
    nH2 = [(zeros(1,13)),(FER(1,14:25)-1)];
end

```

- If only hydrogen is selected only n_{H_2O} needs balancing when lean as there is no carbon
- n_{H_2} is the excess hydrogen when Rich

2.5.4.4 Exhaust Gas Recirculation (EGR) Vector

```

                                %range of EGR
EGR = (1:0.05:1.4);

```

- This is the range of EGR used in future calculations

2.5.4.5 Rebalancing to show the Effect of EGR

2.5.4.5.1 EGR Display Selector

```

% ===== w = selects EGR for results look up and display
w=app.EGRDropDown.Value;
if app.EGRDropDown.Value == 'a'
    w = 1;
elseif app.EGRDropDown.Value == 'b'
    w = 2;

```

- The 'if' statement for the variable 'w' does go to 9 for all instances of EGR selectable.
- All instances of EGR are calculated it's just for selected results table display and balancing coefficients.

2.5.4.5.2 Rebalancing for Displaying EGR

```
% Rebalances the Resultants for EGR to show in balanced equation
```

```
nN2_EGR = EGR(1,w)*nN2;
nCO2_EGR = EGR(1,w)*nCO2;
nH2O_EGR = EGR(1,w)*nH2O;
nCO_EGR = EGR(1,w)*nCO;
nO2_EGR = EGR(1,w)*nO2;
nH2_EGR = EGR(1,w)*nH2;
```

- When EGR is selected from the drop down the balancing result is calculated as above.

3 Task 1

3.1 (a) Calculate the mass percentages of methanol and isooctane in this blend

To calculate the mass percentage the molar masses listed in the periodic table for each element need to be used.

3.1.1 Molar Masses of Elements

```
%molar masses of the elements
```

```
masses = [12.0107,1.007,15.9994,14.0067];
```

The molar masses of the elements are added to a vector in the same order as the amount of each element is within its fuel vector.

3.1.2 Mass Percentage of Fuel

```
% Calculating mass Percentages of Fuel
```

```
M_P_Fuel =
sum(((Fuel_1(1:4,:).*(masses'))))/ (sum((b.*Fuel_2(1:4,:).*(masses')))+sum(((Fuel_1(1:4,:).*(masses'))))));
```

- The masses of the Fuel_1 is totalled then divided by the total mass of both fuels.
- For this example an iso octane methanol blend of 50 :50 molar composition has been selected and the result from the app is below.

Mass Percent Fuel 1	0.7809
---------------------	--------

3.2 (b) Calculate the Air: Fuel mass ratio for stoichiometric combustion in air.

3.2.1 Air Mass

```
% Calculating Air Mass
air_mass = x*(sum((air(1:4,:).*(masses'))));
```

- This sums the air mass for the balanced equation for any fuel selection.

3.2.2 Fuel Mass

```
%Calculating Fuel Mass
fuel_mass =
(sum((b*Fuel_2(1:4,:).*(masses')))+sum(((Fuel_1(1:4,:)).*(masses')
)));
```

- This sums the fuel mass for the balanced equation for any fuel selection.

3.2.3 Air Fuel Ratio

```
%Calculating AFR
AFR = air_mass/fuel_mass;
```

- This calculated the air fuel ration the results from the app below are for the brief iso octane methanol.

AFR Stoich	13.15
------------	-------

3.3 (c) From the enthalpy of formations of the fuel blend and products, calculate the lower heating value of the blend in MJ/kg.

3.3.1 Enthalpy of Fuels

```
% Enthalpy of Fuels
enthalpy_in = (Fuel_1(5,1) + b*Fuel_2(5,1))*FER;
```

- This totals the enthalpy of the Fuels for all FER.

3.3.2 Enthalpy of Exhaust Molecules

```
% Enthalpy of Reaction Result
enthalpy_out = (nCO2*CO2(5,1)) + (nH2O*H2O_L(5,1)) + (nCO*CO(5,1));
```

- This totals the enthalpy for the formed exhaust molecules (O₂ and N₂ pass straight through) for all balanced FER.

3.3.3 Lower Heating Value

```
% net Enthalpy Calculation
enthalpy_net = enthalpy_out-enthalpy_in;
% Lower Heating Value Calculation
LHV = enthalpy_net/fuel_mass;
```

- This Calculates the net enthalpy then divides it by the fuel mass to give per kg.

Enthalpy LHV MJ/Kg	39.25
--------------------	-------

3.4 (d) Explain enthalpy of formation.

- Enthalpy of formation is the change in enthalpy from the formation of a molecule from its primary elements(Nagrimanov *et al.*, 2019).
- $\Delta H^\ominus$ is the notation for the enthalpy of formation.
- The $\ominus$ Denotes the formation occurs under standard conditions (T=298k P=1atm).
- Enthalpy is measured in Joules, but formation is per mol as the substances are measured in moles(OXToby, 1987).
- The joules per mole can be related to temperature from pre-determined experiments if the data is available or Hess's law can be used for the reaction from individual element enthalpy's(the overall enthalpy will be the same no matter how you break down the reaction)(Maronglu *et al.*, 1988).

3.5 (e) Explain the difference between higher and lower heating values.

```
% Higher Heating Value Calculation

enthalpy_out_H = (nC02*C02(5,1)) + (nH2O*H2O_H(5,1));

enthalpy_net_H = enthalpy_out_H-enthalpy_in;

HHV = enthalpy_net_H/fuel_mass;
```

- This Calculates the net enthalpy then divides it by the fuel mass to give per kg but this time using the higher enthalpy of formation value.
- The table below show the difference between the Higher and lower values.

Enthalpy LHV MJ/Kg	39.25
Enthalpy HHV MJ/Kg	42.56

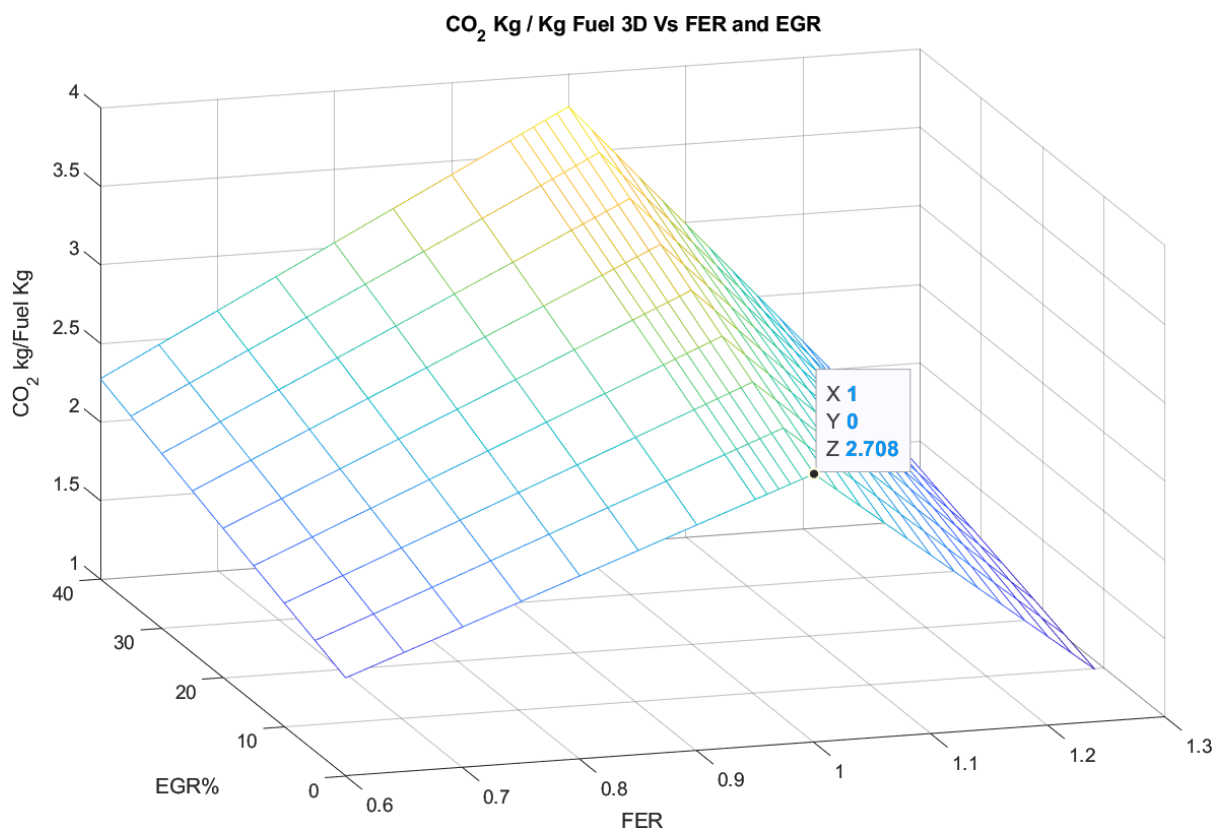
- The higher heating value is for when a reaction makes use of the full range of temperature e.g. the energy release when H_2O returns to liquid state as well as its formation and vapour energy release (the H_2O in a car's exhaust leaves the cylinder as vapour so cannot use this).
- The lower heating value only considers the enthalpy when at a vapour as the energy in a liquid state is not accounted for.
- The higher and lower values can be measured in a bomb calorimeter where the heat released is measured in the cooling liquid the lower value is taken as all heat released for the experiment above 150°C and the higher is the entire Range.

3.6 (f) Calculate the mass of CO_2 produced per kg of fuel burned.

% CO_2 Mass Per Kg of Fuel

$$\text{CO2_mass_kg} = \text{nCO2} * (\text{sum}(\text{masses} * \text{CO2}(1:4, :))) / \text{fuel_mass};$$

- This is the balance coefficient of nCO_2 multiplied by it's masses over the previously calculated fuel mass.

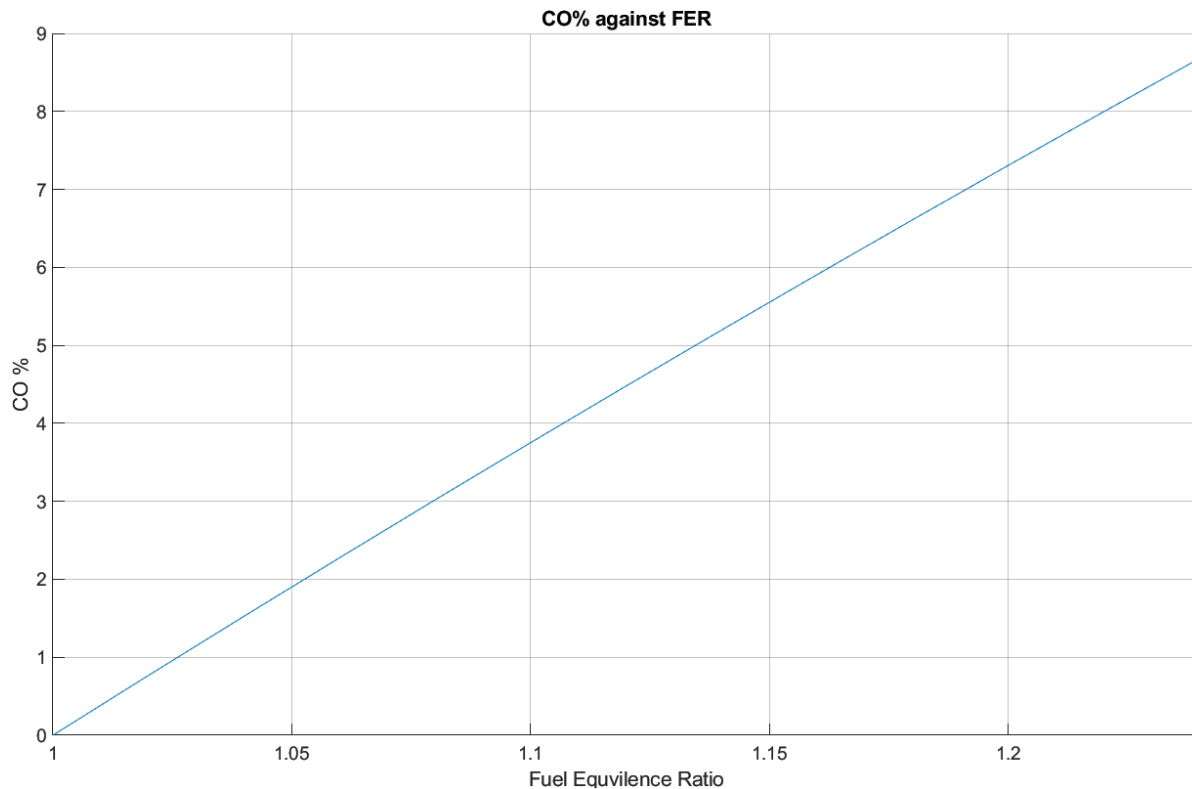


- The effect of EGR does not produce more CO_2 from combustion so EGR axis values can be ignored but the recycled CO_2 is re balanced and this is what the graph shows as well.
- CO_2 drops when rich as CO is formed.

3.7 Carbon Monoxide (CO)

3.7.1 Calculation the percentage of CO

$$\text{\% Percentage of CO} \\ p_{\text{CO}} = ((n_{\text{CO}}) ./ (n_{\text{CO}_2} + n_{\text{CO}} + n_{\text{H}_2\text{O}} + n_{\text{N}_2})) * 100;$$



- This shows the production of CO increases linearly.

3.8 (g) Describe the properties of an “Ideal” fuel.

- An ideal fuel should have no contaminants (sulphur etc). as combustion of these produces unwanted emissions.
- The energy Density should be high.
- The flash point should be above room temperature for stability and safety.
- Combustion should not create harmful emissions.
- Requires little processing for creation (natural gas).
- Easy to transport and store (not hydrogen).

4 Task 2

4.1 Overview

The items Following Table were given to be used as constants in the brief. In the app they can be changed by the user and the results can be calculated to see their effect.

Polytropic Index	1.17	% Task 2 Constants made user editable
RPM	1800	poly = app.PolytropicIndexEditField.Value;
CC of Cylinder	400	rpm = app.RPMEditField.Value;
Compression Ratio to 1	10	v = app.CCofCylinderEditField.Value;
Pressure start of Combustion (Bar)	10	cr = app.CompressionRatioto1EditField.Value;
Pressure rise per degree (bar)	2	pc = app.PressurestartofCombustionBarEditField.Value;
Temp Inlet Close (k)	293	pr = app.PressureriseperdegreebarEditField.Value;
Pressure inlet Close (Bar)	1	fmixT = app.TempInletClosekEditField.Value;
R (J/KMol/K)	8314	ps = app.PressureinletCloseBarEditField.Value;
Volumetric Efficiency (%)	40	R = app.RiKMolKEditField.Value
		u = app.VolumetricEfficiencyEditField.Value/100;

4.2 NO_x Model for Isooctane Running on the Lean side of Stoichiometric

4.2.1 Enthalpy for reaction at a range of FER

Each balanced molecule has its own enthalpy and due to the balancing, its proportion of enthalpy in the reaction changes with FER. The following equation works out the enthalpy for the 25 different FER's Rich and Lean.

```
% Enthalpy of the Reaction
enrL = ((nCO2).*(CO2(5,1)))+(nH2O).*(H2O_L(5,1)))+(nCO).*(CO(5,1))-
((Fuel_1(5,1)).*FER)+(b*Fuel_2(5,1)).*FER);
```

4.2.2 Tables of Enthalpy's for Balanced equation at all FER and range of EGR

Each molecule has its own temperature that corresponds with the enthalpy but as previously discussed the enthalpy for the reaction has various amounts of each molecule in it. Enthalpy values from tables that have temperature for each molecule have been assigned to the following Variables.

```
temp_r = (Enth_Tables(:,1));
O2_Tab = (Enth_Tables(:,3));
N2_Tab = (Enth_Tables(:,8));
H2O_Tab = (Enth_Tables(:,13));
CO2_Tab = (Enth_Tables(:,18));
CO_Tab = (Enth_Tables(:,23));
```

temp_r has the Temperature in Degrees Kelvin. The other variables have the corresponding enthalpies for those temperatures.

These enthalpy tables are multiplied by their Balance Coefficients and EGR. This puts 3 variables into 1 equation which would not work. To simplify the code the enthalpy tables are multiplied by the balance coefficients creating a matrix which is saved in a cell and

the calculation is done 9 times for each EGR saving each matrix to a different Cell. Stacking matrix within Cell structures is often referred to as a 4D matrix.

```
Fl_Enthalpy_EGR = cell(1,9);
for G = (1:9)

Fl_Enthalpy_EGR{G} =
(nCO2.*(EGR(G))).*CO2_Tab+(nH2O.*(EGR(G))).*H2O_Tab+(nO2.*(EGR(G))).*O2_Tab+
(nN2.*(EGR(G))).*N2_Tab+ (nCO.*(EGR(G))).*CO_Tab;

end
```

increasing the amount of balance coefficient by multiplying by the EGR value effectively shifts the final look up tables enthalpies higher for a given temperature due to the added mass. The ideal gas law shows that if you increase the number of moles to have the same pressure and volume the temperature must decrease $pV = nRT$.

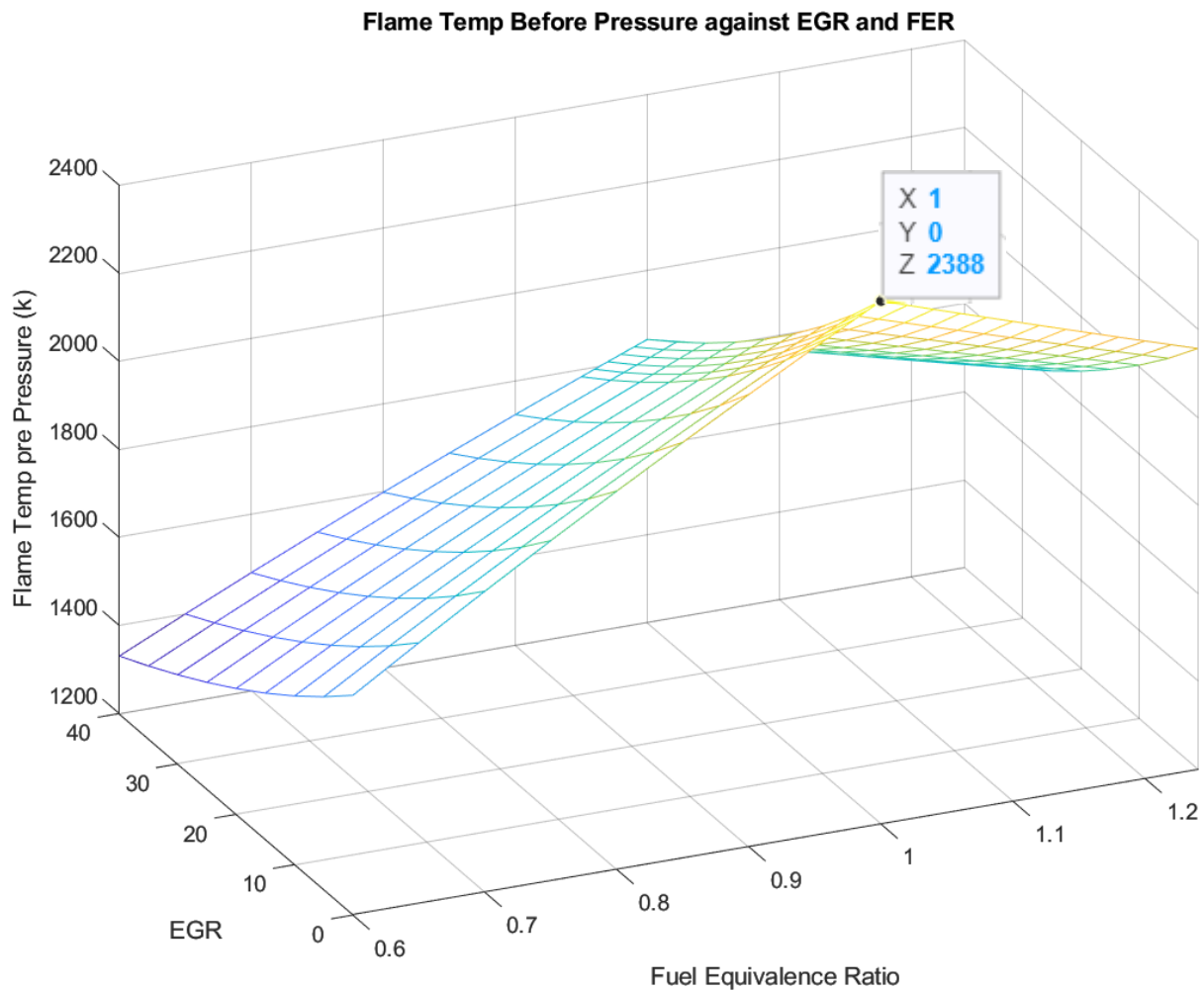
4.2.3 Looking up Temperature from Table

To calculate flame temperature an accurate shape preserving interpolating look up method was used that made the use of creating curve fit equations redundant. This meant the original enthalpy tables could be put into the equation to create a temperature look up table.

```
Flame_Temp_EGR =nan(1,25);

for f=1:9
for i=1:25
Flame_Temp_EGR(f,i)
=interp1(Fl_Enthalpy_EGR{1,f}(:,i),temp_r',enrL(1,i), 'pchip');
end
end
```

A nested loop was used to look up the temperatures for the 25 FER's first then the repeated for the 9 EGR ranges. The results were put into a 9by 25 matrix.



- This shows the highest temperatures were at FER 1.
- As FER is Lean more non-combusted mass passes through the reaction reducing temperature and it reduces the amount of Fuel to release energy.
- As FER is rich the production of CO_2 Decreases which has the highest enthalpy release as the H_2O and CO production increase with lower enthalpy release.
- The predicted combustion does not include hydrocarbons unburnt and many other reactions skewing the temperatures.

4.2.4 Increasing Flame Temperature with Pressure

4.2.4.1 Pressure Rise

from the user selectable variables, a pressure per degree vector is created. This pressure is then offset to create two vectors, one for P1 (from the original), the offset data representing P2.

```
pressure = (((pc)*1e5):(pr*1e5):((15*(pr*1e5))+(pc)*1e5)); %pressure vector pascals
p2 = circshift(pressure,[0 -1])'; %p2
```

The matrix of flame temperatures for FER and EGR first need to be transposed (Rotated 90°) split into vectors for each EGR.

```
Flame_Temp_EGR = (Flame_Temp_EGR'); % transpose flame temp  
  
Flame_Temp_EGR_0 = Flame_Temp_EGR(:,1);  
Flame_Temp_EGR_5 = Flame_Temp_EGR(:,2);
```

4.2.4.2 Increasing Temperature with pressure

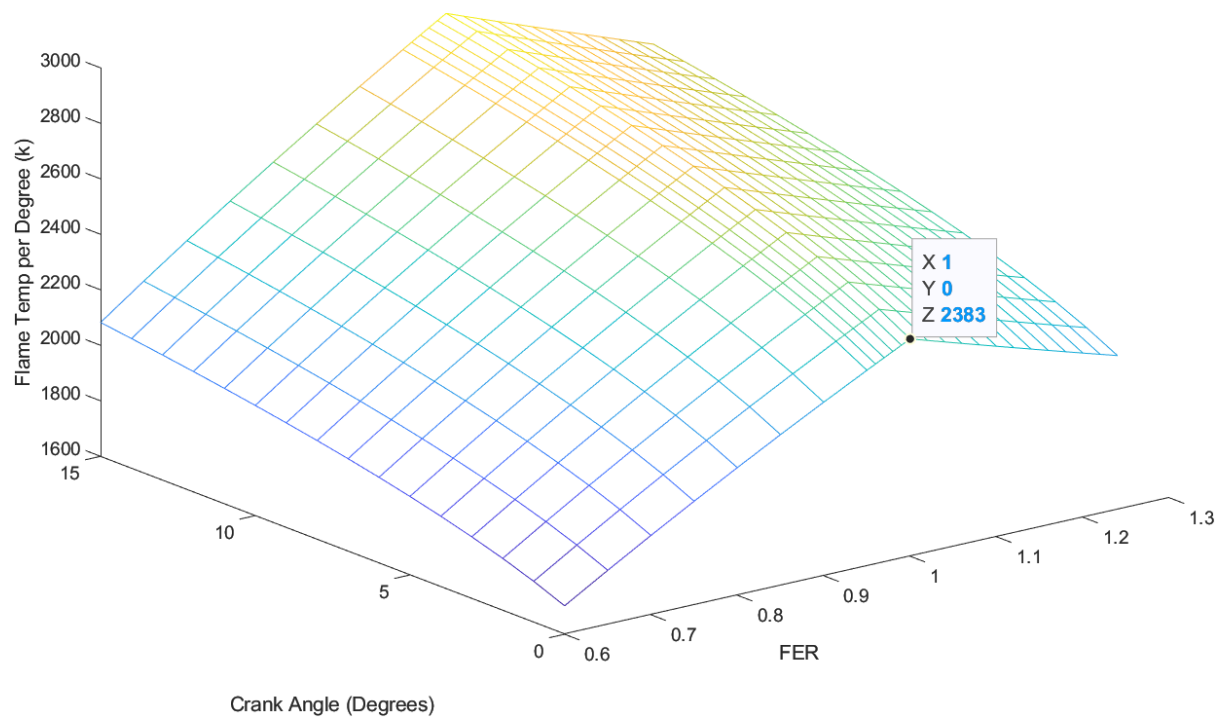
The formula to calculate pressure rise per degree was used in a loop for each initial temperature.

```
for i=1:1:16  
  
    Flame_Temp_EGR_0(:,i+1) =  
    Flame_Temp_EGR_0(:,i).*(((p2(i))./pressure(i)).^((poly-1)/poly));  
  
    Flame_Temp_EGR_5(:,i+1) =  
    Flame_Temp_EGR_5(:,i).*(((p2(i))./pressure(i)).^((poly-1)/poly));  
  
end  
Flame_Temp_EGR_0(:,17) = [];  
Flame_Temp_EGR_5(:,17) = [];
```

The polytropic index is the rate of change from a PV diagram this does change per degree in reality but a average constant can be assumed for this model.

The equation used is derived from boyles law and the ideal gas law.

Flame Temp Rising with Pressure without EGR against FER and Pressure



this shows the temperature increasing per degree as expected.

4.3 Merker's Zeldovich model

4.3.1 Calculating alpha

Alpha is the number of kmols for each side of the reaction for every variable of FER. The pressure at the start and the volume of the cylinder need to be in S.I. units. The volume is multiplied by the volumetric efficiency.

$$\alpha = \frac{((p_s * 1e5) * ((v / 1e6) * u)) / (R * f_{mix} * T))}{((FER) + (x * (1 + 3.76)))};$$

4.3.2 Concentrations of oxygen and nitrogen in Combustion

```
o2conc = ((nO2.*alpha)./((v/1e6)/cr))';  
  
n2conc = ((nN2.*alpha)./((v/1e6)/cr))';
```

4.3.3 NO_x Kmols/Degree Merker

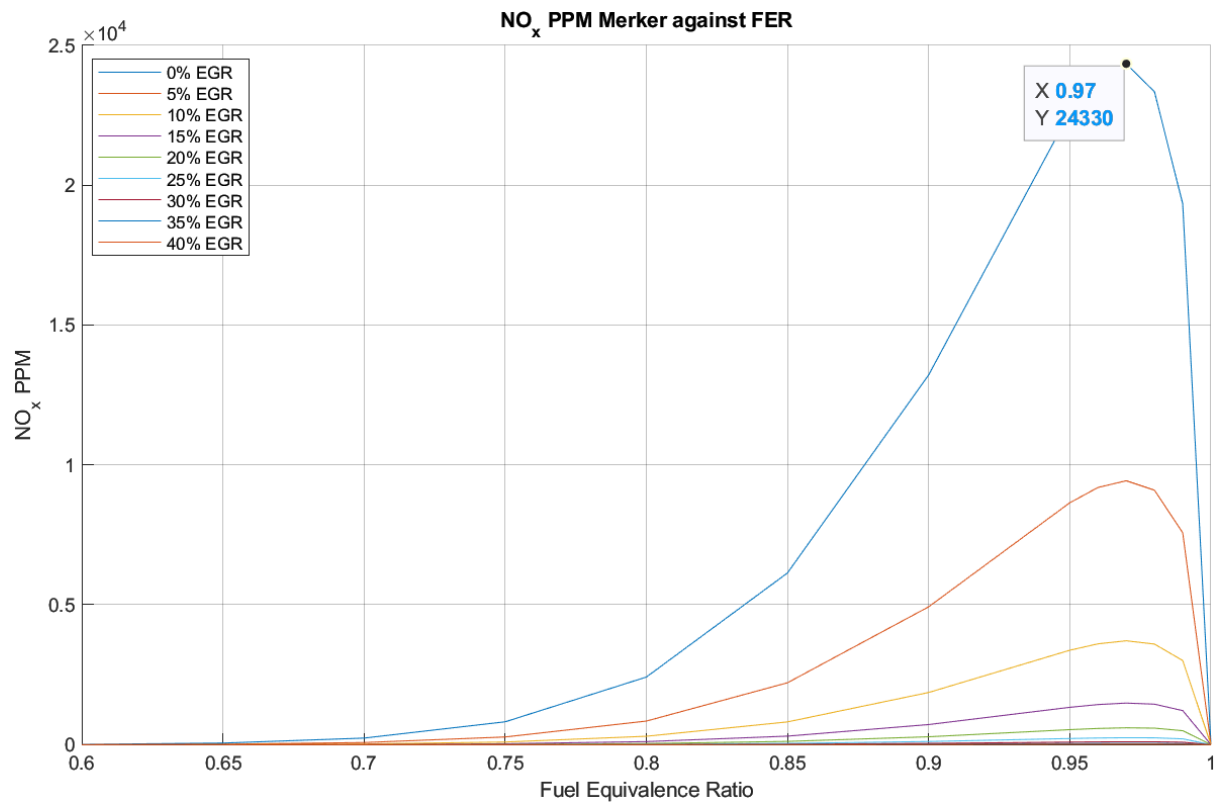
```
Nox_kmol_0 =  
(4.7e13.*(n2conc.*(o2conc).^5).*exp(67837./Flame_Temp_EGR_0).*((v/1e6)/cr))  
./(rpm*6);  
  
Nox_kmol_5 =  
(4.7e13.*(n2conc.*(o2conc).^5).*exp(7837./Flame_Temp_EGR_5).*((v/1e6)/cr))  
./(rpm*6);
```

This Gives kmols per m³ as S.I units used. This also gives the kmols per m³ per degree. It has been done for with and without EGR.

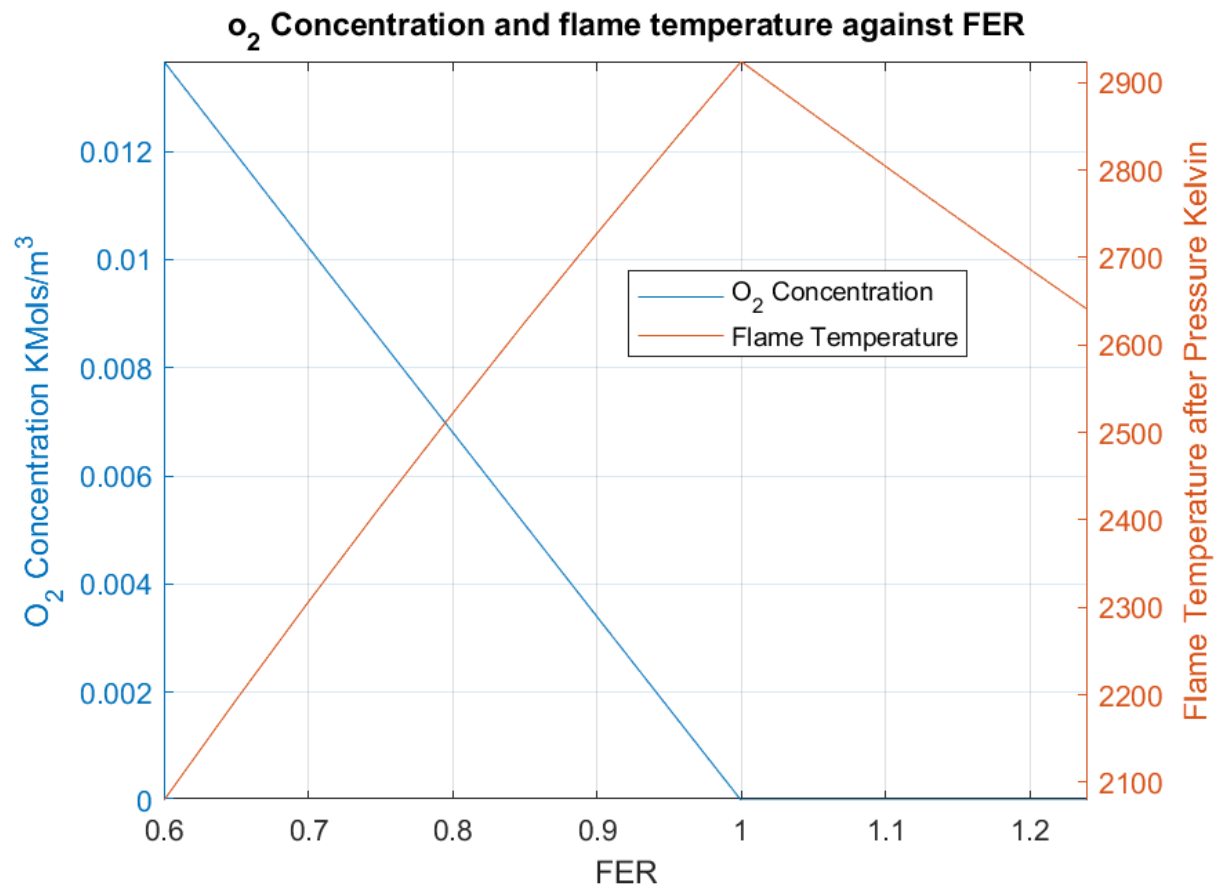
4.3.4 NO_x PPM Merker

```
ppm(1,:) = ((sum(Nox_kmol_0,2)')*1e6)./(alpha.*(nCO2+nH2O+nN2+nO2));  
ppm(2,:) = ((sum(Nox_kmol_5,2)')*1e6)./(alpha.*(nCO2+nH2O+nN2+nO2));
```

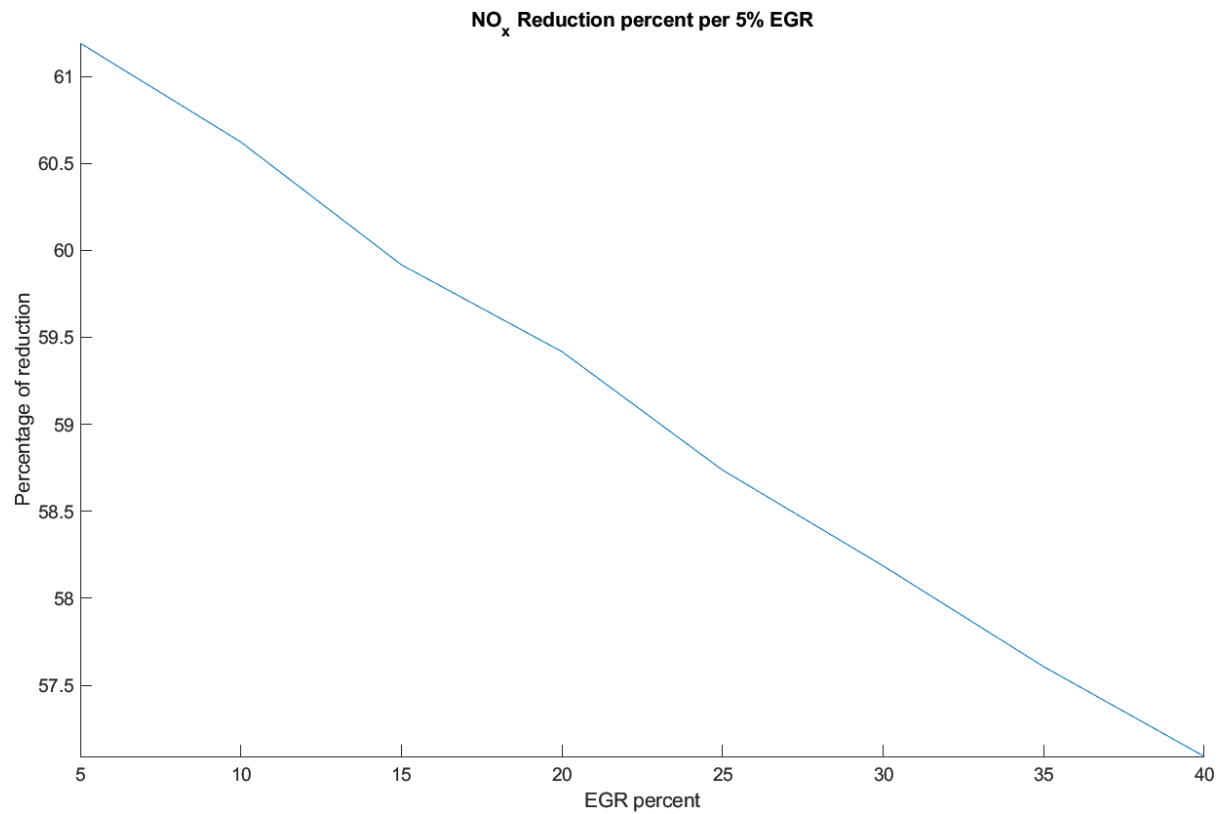
Totalling the Kmols per degree then multiplying by 1million and dividing by the total Kmols in the exhaust gives parts per million.



- Maximum NO_x is seen at 0.97 FER as this is when the flame temperature is high and the O₂ concentration is high enough to form ideal conditions for NO_x (Magina *et al.*, 2015).
- This shows increasing the mass reduces the temperature enough to reduce NO_x.



- This Graph shows as the flame temperature decreases O₂ increases.
- This shows that even though neither of the driving factors are at their maximum 0.97 is the point where there is the enough of each.



- The above graph shows the percentage reduction in NO_x for each 5% increment of EGR
- The reduction is always around 60% per 5%
- It reduces its effectiveness as the temperature drops slowing formation.

5 Conclusions

The Zeldovich model does used with the assumptions given does show the trends for NO_x

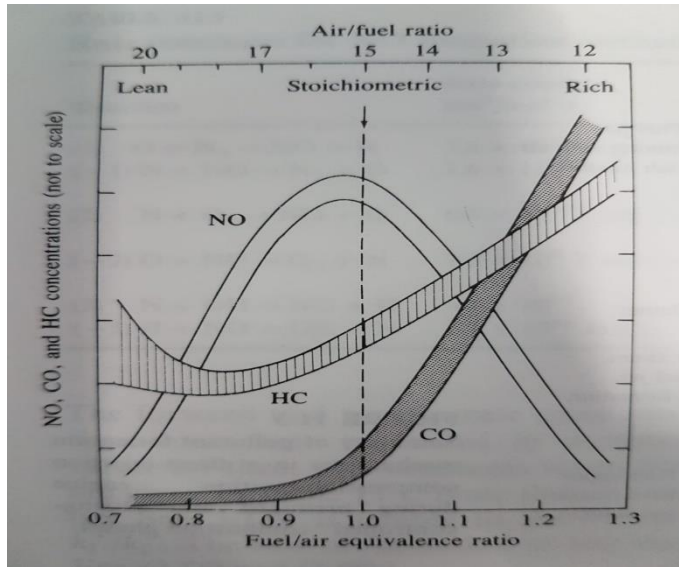


Figure 1 Heywood

As the above Graphic shows, NO and HC occur across the FER Range. This has not been accounted for in the simple balancing model.

- Dissociation and forward and backwards reaction have not been accounted for only a time-based assumption has been made.
- The fuels used do not represent pump fuels as there are many other reactions due to varying hydrocarbon sizes and contaminants.
- It is assumed to be isochoric but, the piston is constantly moving.
- Its assumed to be adiabatic but, heat transfer would occur.

6 Extra Work

6.1 McAlister Zeldovich model

```
%McAlister method
Mc_Nox_kmol_0 = (1.476e15.*((n2conc.*(o2conc).^5)).*exp(-
67520./Flame_Temp_EGR_0).*((v/1e6)/cr))./(rpm*6);

Mc_Nox_kmol_5 = (1.476e15.*((n2conc.*(o2conc).^5)).*exp(-
67520./Flame_Temp_EGR_5).*((v/1e6)/cr))./(rpm*6);

%
Mc_ppm(1,:) = ((sum(Mc_Nox_kmol_0,2)')*1e4)./(alpha.*(nC02+nH2O+nN2+nO2));
Mc_ppm(2,:) = ((sum(Mc_Nox_kmol_5,2)')*1e4)./(alpha.*(nC02+nH2O+nN2+nO2));
```

- This uses different constants and its units are in Kmol per mm³ per degree this means when summing it has been multiplied by 10000 to give ppm.

6.2 Heywood Zeldovich model

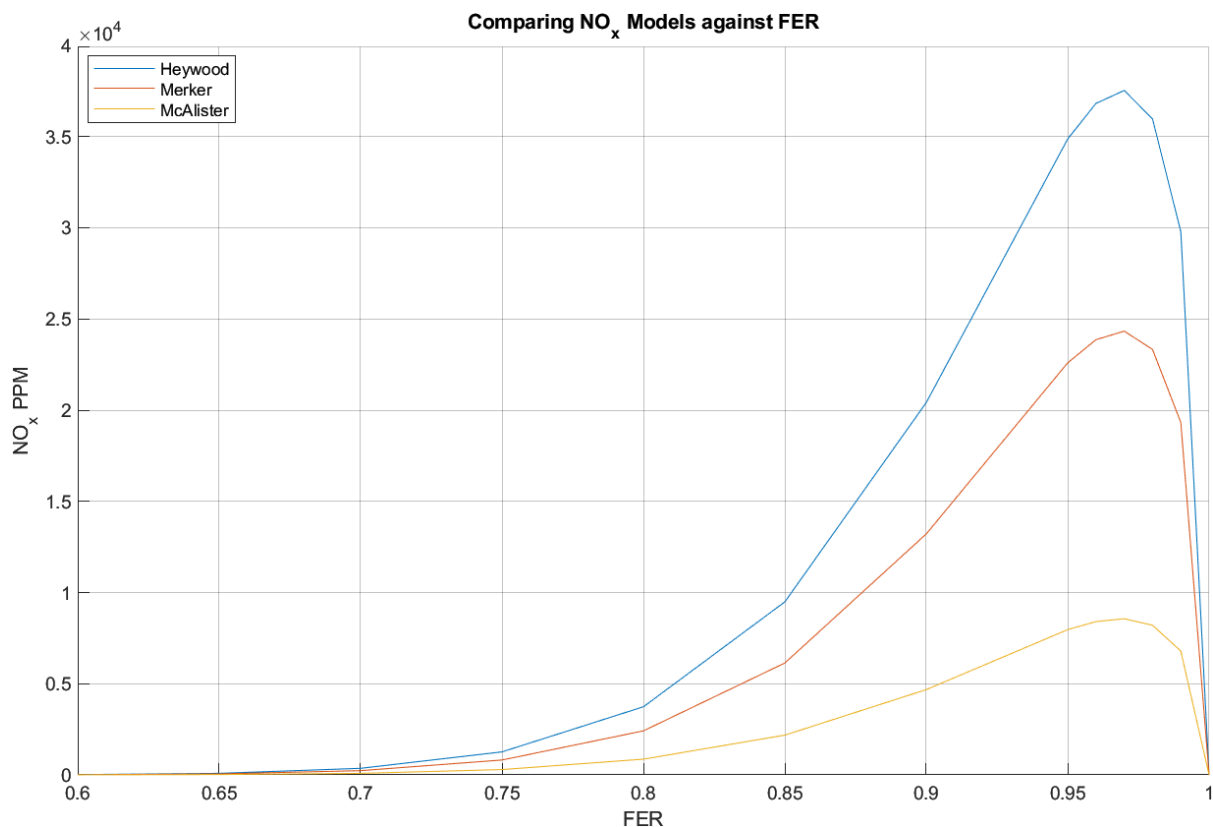
```
% Heywood Method
H_NoX_kmol_0 =
(((6e16)./(Flame_Temp_EGR_0.^5)).*((n2conc.*(o2conc).^5)).*exp(-
69090./Flame_Temp_EGR_0).*((v/1e6)/cr))./(rpm*6);

H_NoX_kmol_5 =
(((6e16)./(Flame_Temp_EGR_5.^5)).*((n2conc.*(o2conc).^5)).*exp(-
69090./Flame_Temp_EGR_5).*((v/1e6)/cr))./(rpm*6);

%
H_ppm(1,:) = ((sum(H_NoX_kmol_0,2)')*1e5)./(alpha.*(nC02+nH2O+nN2+nO2));
H_ppm(2,:) = ((sum(H_NoX_kmol_5,2)')*1e5)./(alpha.*(nC02+nH2O+nN2+nO2));
```

- This uses different constants and its units are in Kmol per cm³ per degree this means when summing it has been multiplied by 100000 to give ppm.

6.2.1 Graphs for Each NO_x model with 0 EGR against FER



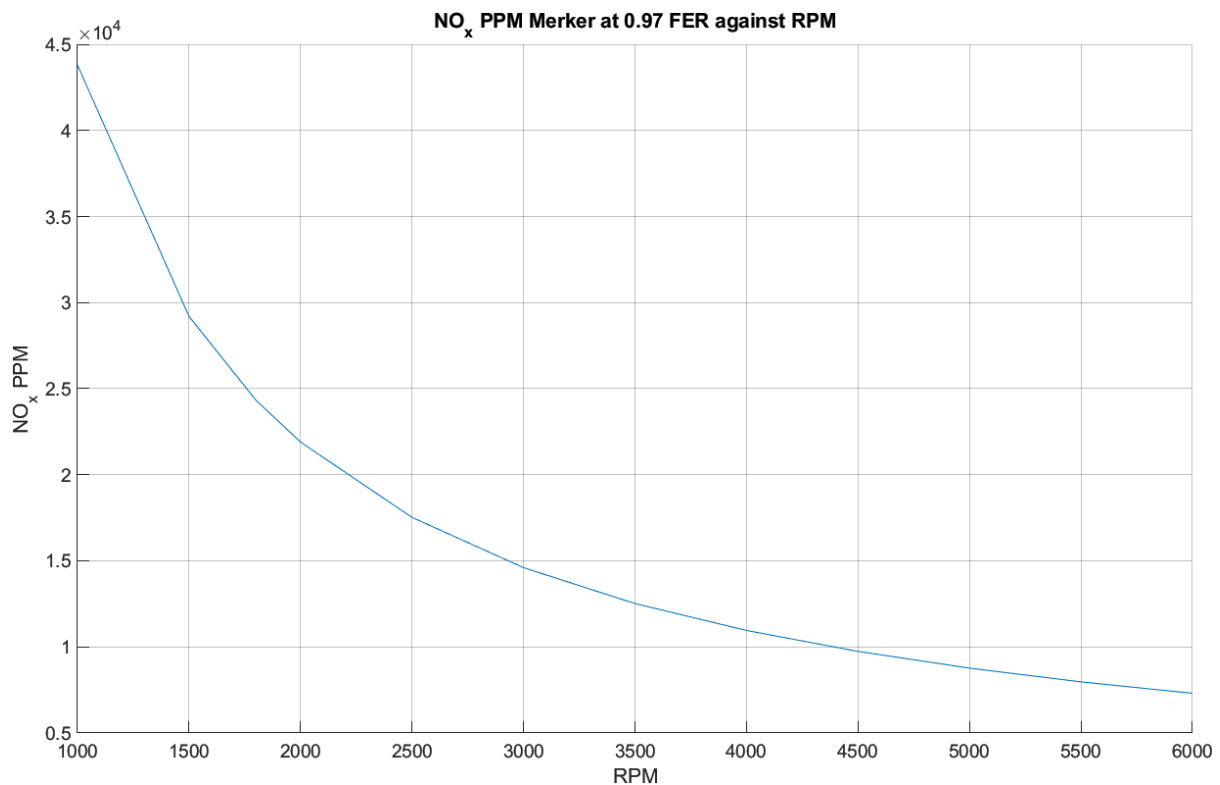
- The Graphs show the large variations in prediction models this is probably due to the different engines used to validate their models.

6.3 Varying RPM

Looking at the Merker model for 0.97 FER the RPM was varied whilst keeping all other parameters constant using the following formula to see its effect.

```
%===== Varying the RPM=====
RPM = [1000; 1500 ;1800 ;2000; 2500 ;3000; 3500; 4000; 4500; 5000; 5500; 6000];
noxkmRpm = (4.7e13.*(n2conc(10,1).*(o2conc(10,1)).^.5).*exp(-
67837./Flame_Temp_EGR_0(10,:)).*((v/1e6)/cr))./(RPM*6);%shit just kmol

ppmRPM =
((sum(noxkmRpm,2)')*1e6)./(alpha(1,10).*(nC02(1,10)+nH2O(1,10)+nN2+nO2(1,10))));
```



- As expected, the RPM is inversely proportional to PPM production as it reduces reaction time.
- This shows its exponentially inversely proportional.
- This also show higher RPM vehicles like motorcycles will produce less NO_x for the same temperature.

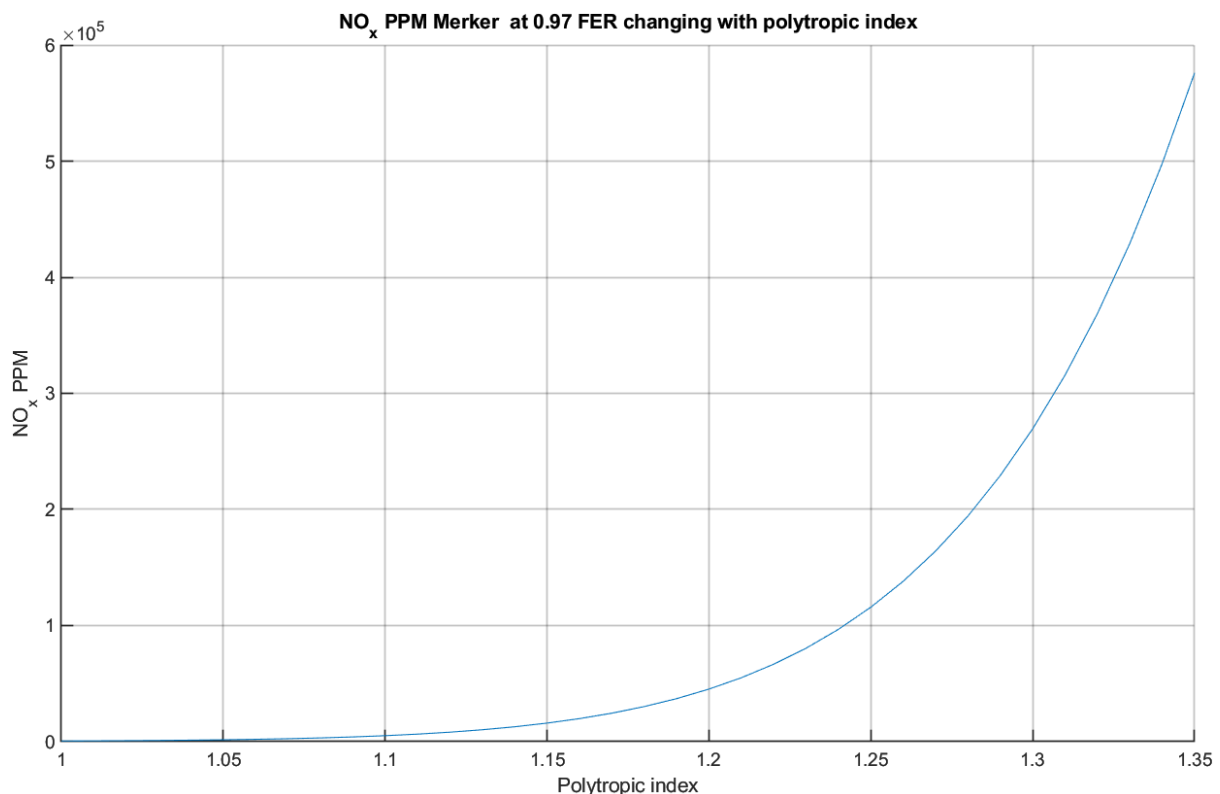
6.4 Varying Polytrropic Index

Looking at the Merker model for 0.97 FER the Polytrropic Index was varied whilst keeping all other parameters constant using the following formula to see its effect.

```

% ===== Varying the Polytropic index =====
polyV = (1:0.01:1.35);
Flame_Temp_EGR_n= cell(1,36);
for j=1:36
    Flame_Temp_EGR_n{j} = Flame_Temp_EGR(10,1);
end
for G = (1:36)
    for i=1:1:16
        Flame_Temp_EGR_n{G}(:,i+1) =
            Flame_Temp_EGR_n{G}(:,i).*(((p2(i))./pressure(i))).^((polyV(G)-1)./polyV(G)));
    end
end
Nox_kmol_n= cell(1,36);
for G = (1:36)
    Nox_kmol_n{G}= (4.7e13.*(n2conc(10,1).*(o2conc(10,1)).^.5).*exp(-
        67837./Flame_Temp_EGR_n{G}(1,1:16)).*((v/1e6)/cr))./(rpm*6);
end
ppm_n_d= cell(1,36);
for G = (1:36)
    ppm_n_d{G} =
        ((sum(Nox_kmol_n{G},2)')*1e6)./(alpha(1,10).*(nCO2(1,10)+nH2O(1,10)+nN2+nO2(1,10)));
end
ppm_n = cell2mat(ppm_n_d);

```



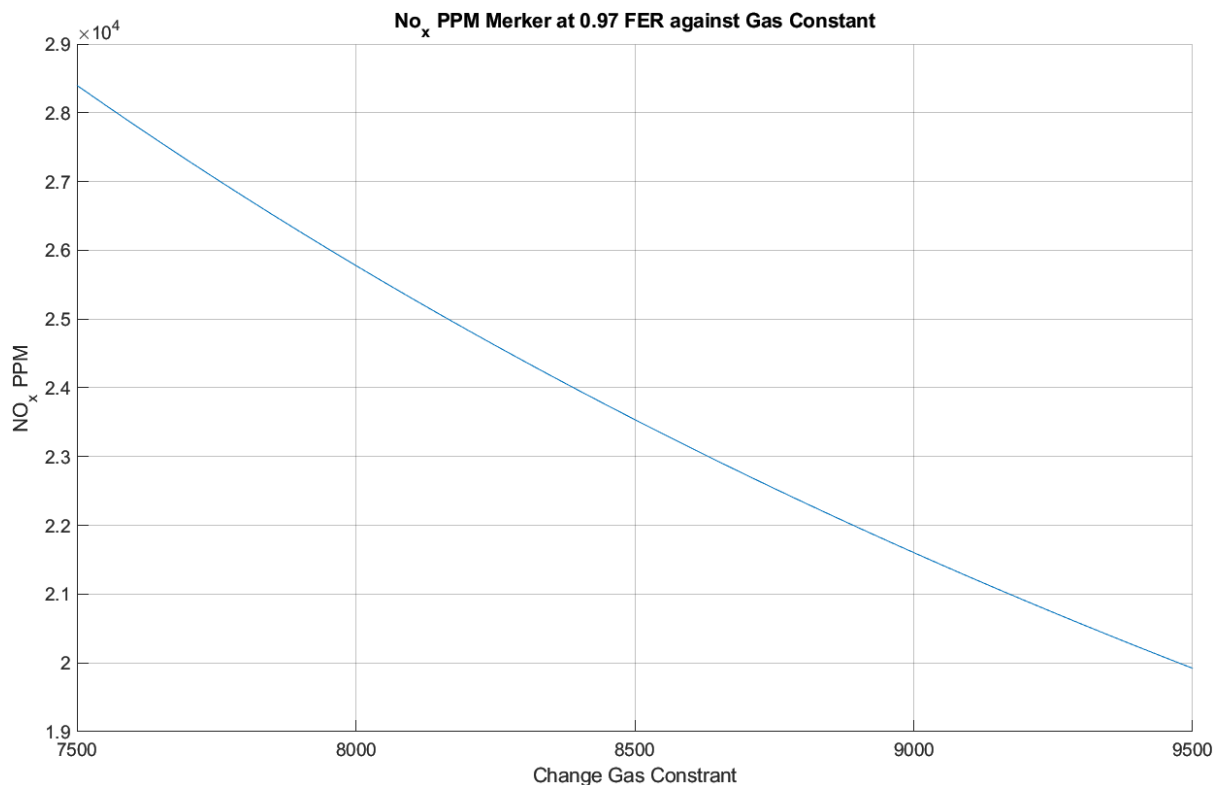
- The graph shows increasing the polytropic index is exponentially proportional to the production of NO_x.

- The polytropic index changes per degree this would have to be taken into account in future models.

6.5 Varying Gas Constant

Looking at the Merker model for 0.97 FER the Gas Constant (R) was varied whilst keeping all other parameters constant using the following formula to see its effect.

```
%===== varying the gas constant R=====
Rv =(7500:50:9500)';
nnv = ((ps*1e5)*((v/1e6)*u))./(Rv*fmixT);
alphav = nnv./((FER)+(x*(1+3.76)));
o2concv = ((nO2.*alphav)./((v/1e6)/cr))';
n2concv = ((nN2.*alphav)./((v/1e6)/cr))';
Nox_kmol_0R = (4.7e13.*((n2concv(10,:))'.*(o2concv(10,:)).^0.5).*exp(-
67837./Flame_Temp_EGR_0(10,:)).*((v/1e6)/cr))./(rpm*6);
ppmR(1,:) =
```

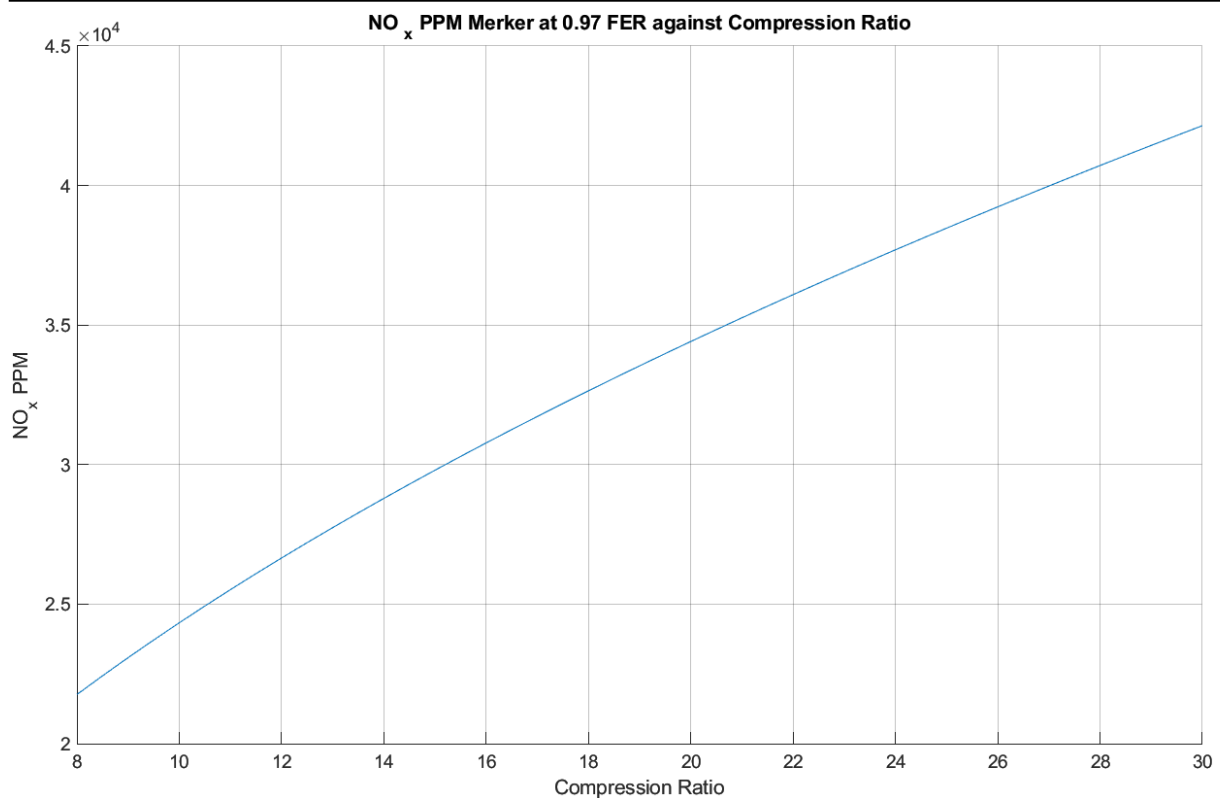


- This shows the gas constant to be nearly linearly inversely proportional.
- As there is never a point of constant pressure or volume in a cylinder this would also vary per degree.

6.6 Varying Compression Ratio (Combustion Volume)

Looking at the Merker model for 0.97 FER the Compression Ratio. was varied whilst keeping all other parameters constant using the following formula to see its effect.

```
%===== varying the Compression Ratio=====
CRv =(8:0.5:30)';
vol_combustcrv = (v/1e6)./CRv; % (v/1e6) of combustion Zone
o2conccrv = ((nO2(1,10)).*alpha(1,10))./vol_combustcrv';
n2conccrv = ((nN2.*alpha(1,10))./vol_combustcrv)';
Nox_kmol_0crv = (4.7e13.*((n2conccrv').*(o2conccrv').^.5).*exp(-
67837./Flame_Temp_EGR_0(10,:)).*vol_combustcrv)./(rpm*6);
ppmcrv =
((sum(Nox_kmol_0crv,2)')*1e6)./(alpha(1,10).*(nC02(1,10)+nH2O(1,10)+nN2+nO2(1,10))));
```

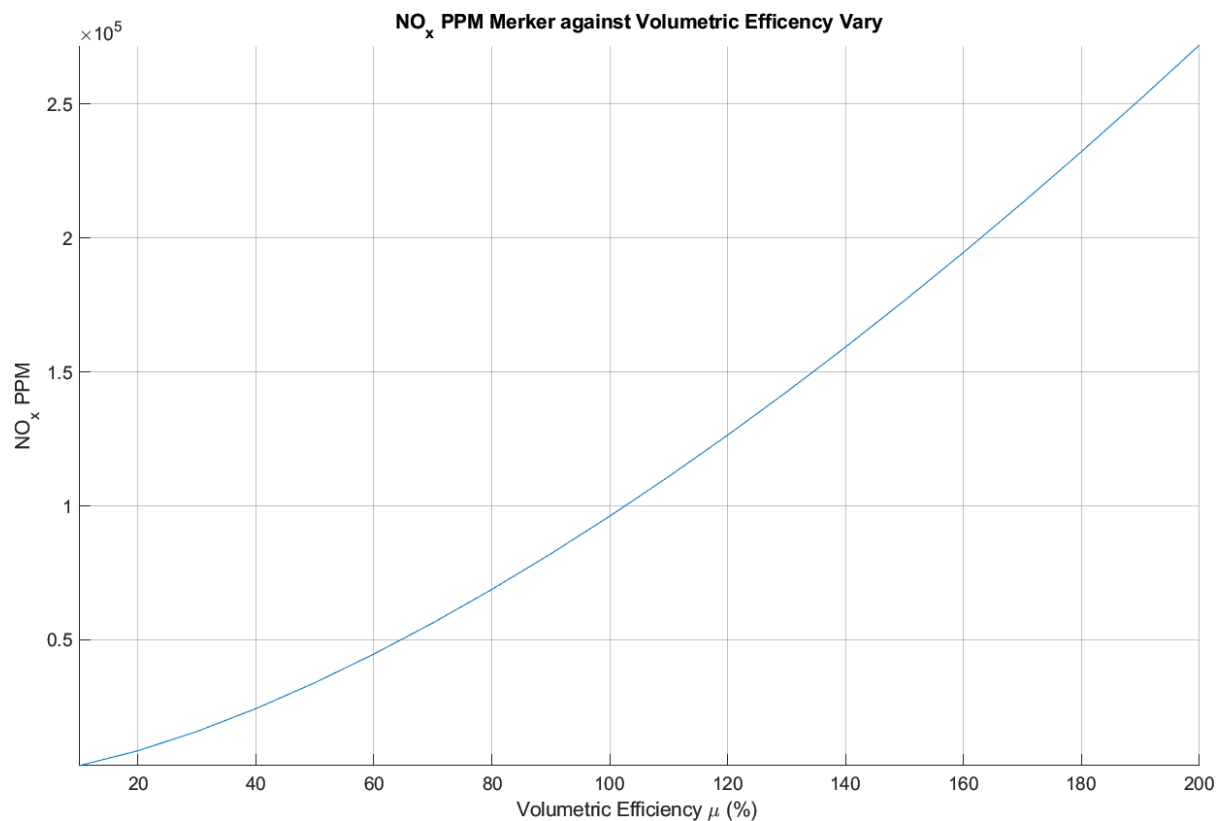


- As expected, a reduced combustion zone volume in relation to total increases the concentration of O₂ increasing NO_x.

6.7 Varying Volumetric Efficiency

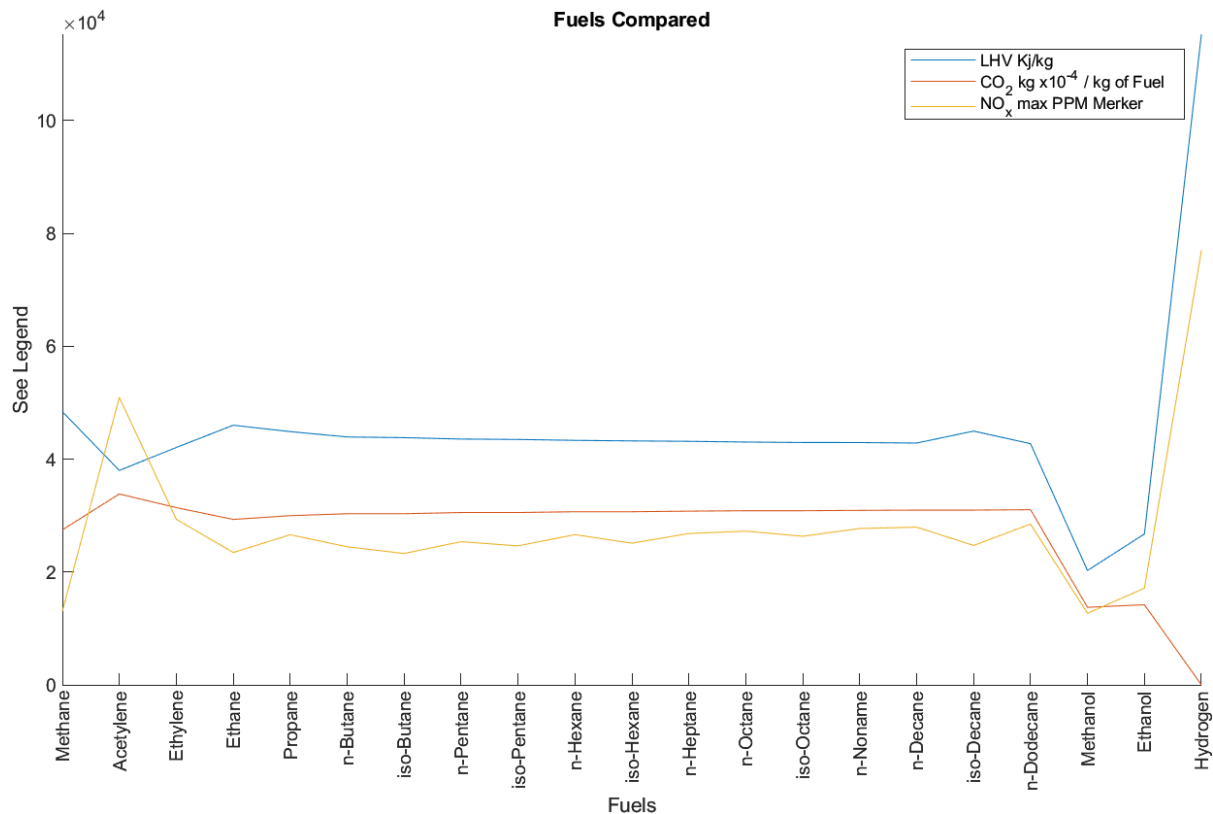
Looking at the Merker model for 0.97 FER the Volumetric Efficiency was varied whilst keeping all other parameters constant using the following formula to see its effect.

```
% ===== Vary efficiency =====
uv = (0.1:0.1:2)';
nuv = ((ps*1e5)*((v/1e6)*uv))./(R*fmixT);
alphauv = nuv./((FER)+(x*(1+3.76)));
o2concuv = ((nO2.*alphauv)./((v/1e6)/cr))';
n2concuv = ((nN2.*alphauv)./((v/1e6)/cr))';
Nox_kmol_0u = (4.7e13.*((n2concuv(10,:))'.*(o2concuv(10,:)).^5).*exp(-
67837./Flame_Temp_EGR_0(10,:)).*((v/1e6)/cr))./(rpm*6);
ppmu(1,:) =
((sum(Nox_kmol_0u,2)')*1e6)./(alpha(1,10).*(nC02(1,10)+nH2O(1,10)+nN2+nO2(1,10)));
```



- This shows forced induction would increase NO_x formation.

6.8 Fuels Compared



- This shows hydrocarbons always produce around 3kg of CO_2 per kg of Fuel.
- Excluding acetylene, the lower heating value can be used to predict if a fuel will produce NO_x than another as this is proportional to flame temperature (Acetylene has an abundance of oxygen in lean conditions for its high burn temperature).

7 References

Konicek, J. *et al.* (1973) 'Design and Testing of a Vaporisation Calorimeter. Enthalpies of Vaporisation of Some Alkyl Cyanides.', *Acta Chemica Scandinavica*. Danish Chemical Society, 27, pp. 1496–1502. doi: 10.3891/acta.chem.scand.27-1496.

Magina, N. *et al.* (2015) 'Propagation, dissipation, and dispersion of disturbances on harmonically forced, non-premixed flames', *Proceedings of the Combustion Institute*. Elsevier Ltd, 35(1), pp. 1097–1105. doi: 10.1016/j.proci.2014.07.050.

Maronglu, B. *et al.* (1988) 'Thermodynamics of Binary Mixtures Containing Ethers or Acetals. 1. Excess Enthalpies of Linear Ethers or Acetals + Heptane or + Cyclohexane Mixtures', *Journal of Chemical and Engineering Data*, 33(2), pp. 118–122. doi: 10.1021/je00052a017.

Nagrimanov, R. N. *et al.* (2019) 'Improving the method of solution calorimetry for evaluation of the enthalpies of phase transitions and condensed state enthalpies of formation', *Journal of Chemical Thermodynamics*. Academic Press, 128, pp. 141–147. doi: 10.1016/j.jct.2018.08.011.

OXToby, D. W. (1987) *Principles of Modern Chemistry, Journal of Organometallic Chemistry*. Brooks/Cole Cengage Learning. doi: 10.1016/0022-328X(87)80173-0.

8 Appendices

Initially the original tables of enthalpies against temperature for exhaust molecules was interpolated to give enthalpy per degree k. this was done using 'fit' 'poly9' to obtain an equation then multiplying the equation by the range of temperatures. This gave a result that MATLAB predicted had only 95% accuracy.

Then.

```
O2_fit = fit((Enth_Tables(:,1)), (Enth_Tables(:,3)), 'pchipinterp');
```

The above method was used, and this method uses the same technique that is used to plot the original Graph. This gives a shape preserving interpolation which has no change from the original. This became redundant as a shape preserving look up method for the original enthalpy tables can be used saving this step.

9 Script

```
classdef PowertrainApp_Ryan < matlab.apps.AppBase
```

```
% Properties that correspond to app components
```

```
properties (Access = public)
```

```
    Ryan_James_Powertrain_app
```

```
    Fuel1Label
```

```
    Fuel1DropDown
```

```
    Fuel2DropDownLabel
```

```
    Fuel2DropDown
```

```
    ModelButton
```

```
    Label_3
```

```
    EditField_2
```

```
    EditField_4
```

```
    EditField
```

```
    matlab.ui.Figure
```

```
    matlab.ui.control.Label
```

```
    matlab.ui.control.DropDown
```

```
    matlab.ui.control.Label
```

```
    matlab.ui.control.DropDown
```

```
    matlab.ui.control.Button
```

```
    matlab.ui.control.Label
```

```
    matlab.ui.control.NumericEditField
```

```
    matlab.ui.control.EditField
```

```
    matlab.ui.control.EditField
```

Label	matlab.ui.control.Label
EditField_3	matlab.ui.control.NumericEditField
CO2Label	matlab.ui.control.Label
CO2EditField	matlab.ui.control.NumericEditField
H2OLabel	matlab.ui.control.Label
H2OEditField	matlab.ui.control.NumericEditField
O2376N2Label	matlab.ui.control.Label
FERDropDownLabel	matlab.ui.control.Label
FERDropDown	matlab.ui.control.DropDown
EGRDropDownLabel	matlab.ui.control.Label
EGRDropDown	matlab.ui.control.DropDown
N2Label_2	matlab.ui.control.Label
N2EditField	matlab.ui.control.NumericEditField
O2Label	matlab.ui.control.Label
O2EditField	matlab.ui.control.NumericEditField
ModellingFuelsandNOxLabel	matlab.ui.control.Label
PercentageofFuelinMixSliderLabel	matlab.ui.control.Label
PercentageofFuelinMixSlider	matlab.ui.control.Slider
Label_5	matlab.ui.control.Label
EditField_5	matlab.ui.control.NumericEditField
Label_6	matlab.ui.control.Label
TabGroup	matlab.ui.container.TabGroup
NOxPPMTab	matlab.ui.container.Tab
UIAxes	matlab.ui.control.UIAxes
COTab	matlab.ui.container.Tab
UIAxes8	matlab.ui.control.UIAxes
CO23DTab	matlab.ui.container.Tab
UIAxes9	matlab.ui.control.UIAxes
H2O3DTab	matlab.ui.container.Tab
UIAxes10	matlab.ui.control.UIAxes
NOx3DVsfERandEGRTab	matlab.ui.container.Tab
UIAxes2	matlab.ui.control.UIAxes
Temp3DVsfERandEGRTab	matlab.ui.container.Tab
UIAxes3	matlab.ui.control.UIAxes
TemperaturePerDegree3DTab	matlab.ui.container.Tab
UIAxes14	matlab.ui.control.UIAxes
ComparedNOxModelsTab	matlab.ui.container.Tab
UIAxes11	matlab.ui.control.UIAxes
RPMNOxTab	matlab.ui.container.Tab
UIAxes4	matlab.ui.control.UIAxes
PolytropicindexTab	matlab.ui.container.Tab
UIAxes5	matlab.ui.control.UIAxes
RTab	matlab.ui.container.Tab
UIAxes6	matlab.ui.control.UIAxes
CRTab	matlab.ui.container.Tab
UIAxes7	matlab.ui.control.UIAxes
EfficiencyTab	matlab.ui.container.Tab
UIAxes13	matlab.ui.control.UIAxes
FuelsComparedTab	matlab.ui.container.Tab
UIAxes12	matlab.ui.control.UIAxes
EffectofEGRTab	matlab.ui.container.Tab
UIAxes15	matlab.ui.control.UIAxes
RPMEditFieldLabel	matlab.ui.control.Label

RPMEditField	matlab.ui.control.NumericEditField
VariablesLabel	matlab.ui.control.Label
PolytropicIndexEditFieldLabel	matlab.ui.control.Label
PolytropicIndexEditField	matlab.ui.control.NumericEditField
CCofCylinderEditFieldLabel	matlab.ui.control.Label
CCofCylinderEditField	matlab.ui.control.NumericEditField
CompressionRatioto1EditFieldLabel	matlab.ui.control.Label
CompressionRatioto1EditField	matlab.ui.control.NumericEditField
PressureriseperdeegreebarEditFieldLabel	matlab.ui.control.Label
PressureriseperdeegreebarEditField	matlab.ui.control.NumericEditField
PressureinletCloseBarEditFieldLabel	matlab.ui.control.Label
PressureinletCloseBarEditField	matlab.ui.control.NumericEditField
RjKMolKEditFieldLabel	matlab.ui.control.Label
RjKMolKEditField	matlab.ui.control.NumericEditField
COLabel	matlab.ui.control.Label
nH2EditField_2	matlab.ui.control.NumericEditField
H2Label	matlab.ui.control.Label
TempInletClosekEditFieldLabel	matlab.ui.control.Label
TempInletClosekEditField	matlab.ui.control.NumericEditField
PressurestartofCombustionBarEditFieldLabel	matlab.ui.control.Label
PressurestartofCombustionBarEditField	
matlab.ui.control.NumericEditField	
MassPercentFuel1EditFieldLabel	matlab.ui.control.Label
MassPercentFuel1EditField	matlab.ui.control.NumericEditField
AFRafterFEREditFieldLabel	matlab.ui.control.Label
AFRafterFEREditField	matlab.ui.control.NumericEditField
AFRStoichEditFieldLabel	matlab.ui.control.Label
AFRStoichEditField	matlab.ui.control.NumericEditField
EnthalpyLHVMJKgEditFieldLabel	matlab.ui.control.Label
EnthalpyLHVMJKgEditField	matlab.ui.control.NumericEditField
EnthalpyHHVMJKgEditFieldLabel	matlab.ui.control.Label
EnthalpyHHVMJKgEditField	matlab.ui.control.NumericEditField
CO2kgFuelkgEditFieldLabel	matlab.ui.control.Label
CO2kgFuelkgEditField	matlab.ui.control.NumericEditField
FlameTempPressureKEditFieldLabel	matlab.ui.control.Label
FlameTempPressureKEditField	matlab.ui.control.NumericEditField
NOxPPMMcAllisterEditFieldLabel	matlab.ui.control.Label
NOxPPMMcAllisterEditField	matlab.ui.control.NumericEditField
NOxPPMHeywoodEditFieldLabel	matlab.ui.control.Label
NOxPPMHeywoodEditField	matlab.ui.control.NumericEditField
N2KMolsm3EditFieldLabel	matlab.ui.control.Label
N2KMolsm3EditField	matlab.ui.control.NumericEditField
O2KMolsm3EditFieldLabel	matlab.ui.control.Label
O2KMolsm3EditField	matlab.ui.control.NumericEditField
VolumetricEfficiencyEditFieldLabel	matlab.ui.control.Label
VolumetricEfficiencyEditField	matlab.ui.control.NumericEditField
FlameTempKEditFieldLabel	matlab.ui.control.Label
FlameTempKEditField	matlab.ui.control.NumericEditField
COEditFieldLabel	matlab.ui.control.Label
COEditField	matlab.ui.control.NumericEditField
NOxPPMMerkerEditFieldLabel	matlab.ui.control.Label
NOxPPMMerkerEditField	matlab.ui.control.NumericEditField
KeyEditFieldLabel	matlab.ui.control.Label

```

KeyEditField          matlab.ui.control.EditField
EditField_6           matlab.ui.control.EditField
EditField_7           matlab.ui.control.EditField
EditField2            matlab.ui.control.EditField
EditField2_2          matlab.ui.control.EditField
EditField_8           matlab.ui.control.EditField
EditField2_3          matlab.ui.control.EditField
EditField_9           matlab.ui.control.EditField
end

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

% Button pushed function: ModelButton
function ModelButtonPushed(app, event)
    close all % closes any open Figures

    % selects Fuel 1 Properties
    Fuel_1 = app.Fuel1DropDown.Value;
    if app.Fuel1DropDown.Value == 'a' % Methane
        Fuel_1 = [1;4;0;0;74.4];
%[carbon;hydrogen;oxygen;nitrogen;Enthalpy]
        app.EditField_4.Value = ('CH4'); % string for balanced
equation
    elseif app.Fuel1DropDown.Value == 'b' % Acetylene
        Fuel_1 = [2;2;0;0;8.7];
        app.EditField_4.Value = ('C2H2');
    elseif app.Fuel1DropDown.Value == 'c' % Ethylene
        Fuel_1 = [2;4;0;0;52.4];
        app.EditField_4.Value = ('C2H4');
    elseif app.Fuel1DropDown.Value == 'd' % Ethane
        Fuel_1 = [2;6;0;0;83.8];
        app.EditField_4.Value = ('C2H6');
    elseif app.Fuel1DropDown.Value == 'e' % Propane
        Fuel_1 = [3;8;0;0;104.7];
        app.EditField_4.Value = ('C3H8');
    elseif app.Fuel1DropDown.Value == 'f' % n-Butane
        Fuel_1 = [4;10;0;0;146.6];
        app.EditField_4.Value = ('C4H10');
    elseif app.Fuel1DropDown.Value == 'g' % iso-Butane
        Fuel_1 = [4;10;0;0;153.5];
        app.EditField_4.Value = ('C4H10');
    elseif app.Fuel1DropDown.Value == 'h' % n-Pentane
        Fuel_1 = [5;12;0;0;173.5];
        app.EditField_4.Value = ('C5H12');
    elseif app.Fuel1DropDown.Value == 'i' % iso-Pentane
        Fuel_1 = [5;12;0;0;178.5];
        app.EditField_4.Value = ('C5H12');
    elseif app.Fuel1DropDown.Value == 'j' % n-Hexane
        Fuel_1 = [6;14;0;0;198.7];
        app.EditField_4.Value = ('C6H14');
    elseif app.Fuel1DropDown.Value == 'k' % iso-Hexane

```

```

    Fuel_1 = [6;14;0;0;207.4];
    app.EditField_4.Value = ('C6H14');
elseif app.Fuel1DropDown.Value == 'l' % n_Heptane
    Fuel_1 = [7;16;0;0;224.2];
    app.EditField_4.Value = ('C7H16');
elseif app.Fuel1DropDown.Value == 'm' % n-Octane
    Fuel_1 = [8;18;0;0;250.1];
    app.EditField_4.Value = ('C8H18');
elseif app.Fuel1DropDown.Value == 'n' % iso-Octane
    Fuel_1 = [8;18;0;0;259.2];
    app.EditField_4.Value = ('C8H18');
elseif app.Fuel1DropDown.Value == 'o' % n-Nonane
    Fuel_1 = [9;20;0;0;274.7];
    app.EditField_4.Value = ('C9H20');
elseif app.Fuel1DropDown.Value == 'p' % n-Decane
    Fuel_1 = [10;22;0;0;300.9];
    app.EditField_4.Value = ('C10H22');
elseif app.Fuel1DropDown.Value == 'q' % iso-Decane
    Fuel_1 = [10;22;0;0;320];
    app.EditField_4.Value = ('C10H22');
elseif app.Fuel1DropDown.Value == 'r' % n-Dodecane
    Fuel_1 = [12;26;0;0;350.9];
    app.EditField_4.Value = ('C12H26');
elseif app.Fuel1DropDown.Value == 's' % Methanol
    Fuel_1 = [1;4;1;0;201.5];
    app.EditField_4.Value = ('CH4O');
elseif app.Fuel1DropDown.Value == 't' % Ethanol
    Fuel_1 = [2;6;1;0;235.1];
    app.EditField_4.Value = ('C2H6O');
elseif app.Fuel1DropDown.Value == 'u' % Hydrogen
    Fuel_1 = [0;2;0;0;0];
    app.EditField_4.Value = ('H2');
end

% Fuel 2 Properties
Fuel_2 = app.Fuel2DropDown.Value;
if app.Fuel2DropDown.Value == 'a' % Methane
    Fuel_2 = [1;4;0;0;74.4];
    app.EditField.Value = ('CH4');
elseif app.Fuel2DropDown.Value == 'b' % Acetylene
    Fuel_2 = [2;2;0;0;8.7];
    app.EditField.Value = ('C2H2');
elseif app.Fuel2DropDown.Value == 'c' % Ethylene
    Fuel_2 = [2;4;0;0;52.4];
    app.EditField.Value = ('C2H4');
elseif app.Fuel2DropDown.Value == 'd' % Ethane
    Fuel_2 = [2;6;0;0;83.8];
    app.EditField.Value = ('C2H6');
elseif app.Fuel2DropDown.Value == 'e' % Propane
    Fuel_2 = [3;8;0;0;104.7];
    app.EditField.Value = ('C3H8');
elseif app.Fuel2DropDown.Value == 'f' % n-Butane
    Fuel_2 = [4;10;0;0;146.6];

```

```

        app.EditField.Value = ('C4H10');
elseif app.Fuel2DropDown.Value == 'g' % iso_Butane
    Fuel_2 = [4;10;0;0;153.5];
    app.EditField.Value = ('C4H10');
elseif app.Fuel2DropDown.Value == 'h' % n_Pentane
    Fuel_2 = [5;12;0;0;173.5];
    app.EditField.Value = ('C5H12');
elseif app.Fuel2DropDown.Value == 'i' % iso_Pentane
    Fuel_2 = [5;12;0;0;178.5];
    app.EditField.Value = ('C5H12');
elseif app.Fuel2DropDown.Value == 'j' % n_Hexane
    Fuel_2 = [6;14;0;0;198.7];
    app.EditField.Value = ('C6H14');
elseif app.Fuel2DropDown.Value == 'k' % iso_Hexane
    Fuel_2 = [6;14;0;0;207.4];
    app.EditField.Value = ('C6H14');
elseif app.Fuel2DropDown.Value == 'l' % n_Heptane
    Fuel_2 = [7;16;0;0;224.2];
    app.EditField.Value = ('C7H16');
elseif app.Fuel2DropDown.Value == 'm' % n-Octane
    Fuel_2 = [8;18;0;0;250.1];
    app.EditField.Value = ('C8H18');
elseif app.Fuel2DropDown.Value == 'n' % iso-Octane
    Fuel_2 = [8;18;0;0;259.2];
    app.EditField.Value = ('C8H18');
elseif app.Fuel2DropDown.Value == 'o' % n_Nonane
    Fuel_2 = [9;20;0;0;274.7];
    app.EditField.Value = ('C9H20');
elseif app.Fuel2DropDown.Value == 'p' % n-Decane
    Fuel_2 = [10;22;0;0;300.9];
    app.EditField.Value = ('C10H22');
elseif app.Fuel2DropDown.Value == 'q' % iso-Decane
    Fuel_2 = [10;22;0;0;0];
    app.EditField.Value = ('C10H22');
elseif app.Fuel2DropDown.Value == 'r' % n_Dodecane
    Fuel_2 = [12;26;0;0;350.9];
    app.EditField.Value = ('C12H26');
elseif app.Fuel2DropDown.Value == 's' % Methanol
    Fuel_2 = [1;4;1;0;201.5];
    app.EditField.Value = ('CH4O');
elseif app.Fuel2DropDown.Value == 't' % Ethanol
    Fuel_2 = [2;6;1;0;235.1];
    app.EditField.Value = ('C2H6O');
elseif app.Fuel2DropDown.Value == 'u' % Hydrogen
    Fuel_2 = [0;2;0;0;0];
    app.EditField.Value = ('H2');
elseif app.Fuel2DropDown.Value == 'v' % None
    Fuel_2 = [0;0;0;0;0];
    app.EditField.Value = ('____');
end
a=app.PercentageofFuel1inMixSlider.Value; % a is the percentage of
Fuel 1 selected on slider

```

```

% b = turns the percentage of fuel 1 (a) into a ratio
b = (100-a)/a;

% Values used in all reactions that dont need to be selected
air      = [0;0;2;7.52];
CO2      = [1;0;2;0;393.5];
CO       = [1;0;1;0;110.5];
H2O_L    = [0;1;2;0;241.8];
H2O_H    = [0;1;2;0;285.8];

% React totals the hydrogen carbons oxygens and nitrogens going in
% the combustion and keeps the totals of elements in separate rows
React = [(Fuel_1(1:4,:)),(b*Fuel_2(1:4,:)),air(1:4,:)];

% to Calculate X the balance must be done at the stoichometric
% FER
nCO2 = (sum(React(1,:))); % balance of CO2 out
nH2O = (sum(React(2,:))/2); % balance of H2o Out
x = (((2*nCO2)+nH2O)-(Fuel_1(3,1) + (Fuel_2(3,1)*b)))/2; % the
amount of air balance
nN2 = (x)*3.76; % Balance of Nitrogen

FER = [(0.6:0.05:0.9),(0.95:0.01:1),(1.02:0.02:1.24)]; % range of
Phi in Brief plus rich range

nO2 = [(x.*(1-FER(1,1:13))),zeros(1,12)]; % Balance for O2 out
when lean

% any but not only hydrogen
if ((sum(React(1,:)))>=0.01 % if there is Carbon
    nH2O = (sum(React(2,:))/2).*FER; %Balance for H2O when rich
and lean
    nCO2= [((sum(React(1,:))).*FER(1,1:13)) ((2*x -
((nH2O(1,14:end)))-
(sum(React(1,:)).*FER(1,14:25)))+(Fuel_1(3,1)+(Fuel_2(3,1)*b))*FER(1,14:25))
)]; % balance CO2 for Phi
    nCO = abs([(zeros(1,13)), (nCO2(1,14:end)-
((sum(React(1,:)).*FER(1,14:25))))]); % Balance For CO when rich
    nH2 = (zeros(1,25));
    %hydrogen only
elseif ((sum(React(1,:)))=0 % if there is no carbon (just
hydrogen)
    nCO2= ((sum(React(1,:))).*FER); % balance CO2 for Phi
    nCO = abs((zeros(1,25))); % Balance For CO when rich
    nH2O = [((sum(React(2,:))/2).*FER(1,1:13)),(ones(1,12))];
%Balance for H2O when rich and lean
    nH2 = [(zeros(1,13)),(FER(1,14:25)-1)];
end

% molar masses of the elements
masses = [12.0107,1.007,15.9994,14.0067];

```

```

    % Calculating mass Percentages of Fuel
    M_P_Fuel =
sum((((Fuel_1(1:4,:).*(masses'))))/((sum((b.*Fuel_2(1:4,:).*(masses')))+sum((((Fuel_1(1:4,:).*(masses')))))));

    % Calculating Air Mass
    air_mass = x*(sum((air(1:4,:).*(masses'))));

    %Calculating Fuel Mass
    fuel_mass =
(sum((b.*Fuel_2(1:4,:).*(masses')))+sum((((Fuel_1(1:4,:).*(masses')))))));

    %Calculating AFR
    AFR = air_mass/fuel_mass;

    % Enthalpy of Fuels
    enthalpy_in = Fuel_1(5,1) + b*Fuel_2(5,1);
    % Enthalpy of Reaction Result
    enthalpy_out = (nC02*CO2(5,1)) + (nH2O*H2O_L(5,1));

    % net Enthalpy Calculation
    enthalpy_net = enthalpy_out-enthalpy_in;
    % Lower Heating Value Calculation
    LHV = enthalpy_net/fuel_mass;
    % Higher Heating Value Calculation
    enthalpy_out_H = (nC02*CO2(5,1)) + (nH2O*H2O_H(5,1));
    enthalpy_net_H = enthalpy_out_H-enthalpy_in;
    HHV = enthalpy_net_H/fuel_mass;

    % CO2 Mass Per Kg of Fuel
    CO2_mass_kg = nC02*(sum( masses*CO2(1:4,:)))/fuel_mass;

    % Percentage of CO
    pCO = ((nC0)./(nC02+nC0+nH2O+nN2))*100;

    load
('polnon.mat','CO2_Tab','H2O_Tab','O2_Tab','N2_Tab','CO_Tab','temp_r','afg');

    % Task 2 Constants made user editable
    poly = app.PolytropicIndexEditField.Value;
% polytropic index for burned zone to peak pressure
    rpm = app.RPMEditField.Value;
    v = app.CCofCylinderEditField.Value;
    cr = app.CompressionRatioto1EditField.Value;
    pc = app.PressurestartofCombustionBarEditField.Value;
    pr = app.PressurerisepdegreebarEditField.Value;
    fmixT = app.TempInletClosekEditField.Value;
    ps = app.PressureinletCloseBarEditField.Value;
    R = app.RjKMolKEditField.Value; %
j/KMol/Kelvin lake
    u = app.VolumetricEfficiencyEditField.Value/100;
% (v/1e6) efficiency

```

```

%range of EGR
EGR = (1:0.05:1.4);

Fl_Enthalpy_EGR = cell(1,9);
for G = (1:9)

    Fl_Enthalpy_EGR{G} =
    (nCO2.*(EGR(G))).*CO2_Tab+(nH2O.*(EGR(G))).*H2O_Tab+(nO2.*(EGR(G))).*O2_Tab+(n
    N2.*(EGR(G))).*N2_Tab+ (nCO.*(EGR(G))).*CO_Tab;

end

% Enthalpy of the Reaction
enrL =
((nCO2).*(CO2(5,1)))+(nH2O).*(H2O_L(5,1)))+(nCO).*(CO(5,1))-
(((Fuel_1(5,1)).*FER)+(b*Fuel_2(5,1)).*FER));

Flame_Temp_EGR =nan(1,25);

for f=1:9
    for i=1:25
        Flame_Temp_EGR(f,i) =
interp1(Fl_Enthalpy_EGR{1,f}(:,i),temp_r',enrL(1,i),'pchip'); % gets the
starting temps
    end
end

Flame_Temp_EGR = (Flame_Temp_EGR'); % transpose flame temp

Flame_Temp_EGR_0 = Flame_Temp_EGR(:,1);
Flame_Temp_EGR_5 = Flame_Temp_EGR(:,2);
Flame_Temp_EGR_10 = Flame_Temp_EGR(:,3);
Flame_Temp_EGR_15 = Flame_Temp_EGR(:,4);
Flame_Temp_EGR_20 = Flame_Temp_EGR(:,5);
Flame_Temp_EGR_25 = Flame_Temp_EGR(:,6);
Flame_Temp_EGR_30 = Flame_Temp_EGR(:,7);
Flame_Temp_EGR_35 = Flame_Temp_EGR(:,8);
Flame_Temp_EGR_40 = Flame_Temp_EGR(:,9);

pressure = (((pc)*1e5):(pr*1e5):((15*(pr*1e5))+(pc)*1e5));
%pressure vector pascals
p2 = circshift(pressure,[0 -1]'); %p2

for i=1:1:16

    Flame_Temp_EGR_0(:,i+1) =
    Flame_Temp_EGR_0(:,i).*(((p2(i))./pressure(i))).^((poly-1)/poly));
    Flame_Temp_EGR_5(:,i+1) =
    Flame_Temp_EGR_5(:,i).*(((p2(i))./pressure(i))).^((poly-1)/poly));

```

```

        Flame_Temp_EGR_10(:,i+1) =
Flame_Temp_EGR_10(:,i).*(((p2(i))./pressure(i))).^((poly-1)/poly));
        Flame_Temp_EGR_15(:,i+1) =
Flame_Temp_EGR_15(:,i).*(((p2(i))./pressure(i))).^((poly-1)/poly));
        Flame_Temp_EGR_20(:,i+1) =
Flame_Temp_EGR_20(:,i).*(((p2(i))./pressure(i))).^((poly-1)/poly));
        Flame_Temp_EGR_25(:,i+1) =
Flame_Temp_EGR_25(:,i).*(((p2(i))./pressure(i))).^((poly-1)/poly));
        Flame_Temp_EGR_30(:,i+1) =
Flame_Temp_EGR_30(:,i).*(((p2(i))./pressure(i))).^((poly-1)/poly));
        Flame_Temp_EGR_35(:,i+1) =
Flame_Temp_EGR_35(:,i).*(((p2(i))./pressure(i))).^((poly-1)/poly));
        Flame_Temp_EGR_40(:,i+1) =
Flame_Temp_EGR_40(:,i).*(((p2(i))./pressure(i))).^((poly-1)/poly));

end

% the folloowing chops off the extra colum created instead of
% using 2 counters from the i+1
Flame_Temp_EGR_0(:,17) = [];
Flame_Temp_EGR_5(:,17) = [];
Flame_Temp_EGR_10(:,17) = [];
Flame_Temp_EGR_15(:,17) = [];
Flame_Temp_EGR_20(:,17) = [];
Flame_Temp_EGR_25(:,17) = [];
Flame_Temp_EGR_30(:,17) = [];
Flame_Temp_EGR_35(:,17) = [];
Flame_Temp_EGR_40(:,17) = [];

% ===== Flame temp with Pressure matrix for all EGR (final temp)
=====
Flame_Temp_EGR_p(1,:) = Flame_Temp_EGR_0(:,16);
Flame_Temp_EGR_p(2,:) = Flame_Temp_EGR_5(:,16);
Flame_Temp_EGR_p(3,:) = Flame_Temp_EGR_10(:,16);
Flame_Temp_EGR_p(4,:) = Flame_Temp_EGR_15(:,16);
Flame_Temp_EGR_p(5,:) = Flame_Temp_EGR_20(:,16);
Flame_Temp_EGR_p(6,:) = Flame_Temp_EGR_25(:,16);
Flame_Temp_EGR_p(7,:) = Flame_Temp_EGR_30(:,16);
Flame_Temp_EGR_p(8,:) = Flame_Temp_EGR_35(:,16);
Flame_Temp_EGR_p(9,:) = Flame_Temp_EGR_40(:,16);

alpha = (((ps*1e5)*((v/1e6)*u))/(R*fmixT))./(((FER)+(x*(1+3.76))));

o2conc = ((nO2.*alpha)./((v/1e6)/cr))';

n2conc = ((nN2.*alpha)./((v/1e6)/cr))';

% Merker Method
Nox_kmol_0 = (4.7e13.*(n2conc.*(o2conc).^5).*exp(-
67837./Flame_Temp_EGR_0).*((v/1e6)/cr))./(rpm*6);
Nox_kmol_5 = (4.7e13.*(n2conc.*(o2conc).^5).*exp(-
67837./Flame_Temp_EGR_5).*((v/1e6)/cr))./(rpm*6);

```

```

        Nox_kmol_10 = (4.7e13.*(n2conc.*(o2conc).^5).*exp(-
67837./Flame_Temp_EGR_10).*((v/1e6)/cr))./(rpm*6);
        Nox_kmol_15 = (4.7e13.*(n2conc.*(o2conc).^5).*exp(-
67837./Flame_Temp_EGR_15).*((v/1e6)/cr))./(rpm*6);
        Nox_kmol_20 = (4.7e13.*(n2conc.*(o2conc).^5).*exp(-
67837./Flame_Temp_EGR_20).*((v/1e6)/cr))./(rpm*6);
        Nox_kmol_25 = (4.7e13.*(n2conc.*(o2conc).^5).*exp(-
67837./Flame_Temp_EGR_25).*((v/1e6)/cr))./(rpm*6);
        Nox_kmol_30 = (4.7e13.*(n2conc.*(o2conc).^5).*exp(-
67837./Flame_Temp_EGR_30).*((v/1e6)/cr))./(rpm*6);
        Nox_kmol_35 = (4.7e13.*(n2conc.*(o2conc).^5).*exp(-
67837./Flame_Temp_EGR_35).*((v/1e6)/cr))./(rpm*6);
        Nox_kmol_40 = (4.7e13.*(n2conc.*(o2conc).^5).*exp(-
67837./Flame_Temp_EGR_40).*((v/1e6)/cr))./(rpm*6);

```

```

        ppm(1,:) =
((sum(Nox_kmol_0,2)')*1e6)./(alpha.*(nC02+nH2O+nN2+nO2));
        ppm(2,:) =
((sum(Nox_kmol_5,2)')*1e6)./(alpha.*(nC02+nH2O+nN2+nO2));
        ppm(3,:) =
((sum(Nox_kmol_10,2)')*1e6)./(alpha.*(nC02+nH2O+nN2+nO2));
        ppm(4,:) =
((sum(Nox_kmol_15,2)')*1e6)./(alpha.*(nC02+nH2O+nN2+nO2));
        ppm(5,:) =
((sum(Nox_kmol_20,2)')*1e6)./(alpha.*(nC02+nH2O+nN2+nO2));
        ppm(6,:) =
((sum(Nox_kmol_25,2)')*1e6)./(alpha.*(nC02+nH2O+nN2+nO2));
        ppm(7,:) =
((sum(Nox_kmol_30,2)')*1e6)./(alpha.*(nC02+nH2O+nN2+nO2));
        ppm(8,:) =
((sum(Nox_kmol_35,2)')*1e6)./(alpha.*(nC02+nH2O+nN2+nO2));
        ppm(9,:) =
((sum(Nox_kmol_40,2)')*1e6)./(alpha.*(nC02+nH2O+nN2+nO2));

```

%McAlister Method

```

        Mc_NoX_kmol_0 = (1.476e15.*((n2conc.*(o2conc).^5)).*exp(-
67520./Flame_Temp_EGR_0).*((v/1e6)/cr))./(rpm*6);
        Mc_NoX_kmol_5 = (1.476e15.*((n2conc.*(o2conc).^5)).*exp(-
67520./Flame_Temp_EGR_5).*((v/1e6)/cr))./(rpm*6);
        Mc_NoX_kmol_10 = (1.476e15.*((n2conc.*(o2conc).^5)).*exp(-
67520./Flame_Temp_EGR_10).*((v/1e6)/cr))./(rpm*6);
        Mc_NoX_kmol_15 = (1.476e15.*((n2conc.*(o2conc).^5)).*exp(-
67520./Flame_Temp_EGR_15).*((v/1e6)/cr))./(rpm*6);
        Mc_NoX_kmol_20 = (1.476e15.*((n2conc.*(o2conc).^5)).*exp(-
67520./Flame_Temp_EGR_20).*((v/1e6)/cr))./(rpm*6);
        Mc_NoX_kmol_25 = (1.476e15.*((n2conc.*(o2conc).^5)).*exp(-
67520./Flame_Temp_EGR_25).*((v/1e6)/cr))./(rpm*6);
        Mc_NoX_kmol_30 = (1.476e15.*((n2conc.*(o2conc).^5)).*exp(-
67520./Flame_Temp_EGR_30).*((v/1e6)/cr))./(rpm*6);
        Mc_NoX_kmol_35 = (1.476e15.*((n2conc.*(o2conc).^5)).*exp(-
67520./Flame_Temp_EGR_35).*((v/1e6)/cr))./(rpm*6);
        Mc_NoX_kmol_40 = (1.476e15.*((n2conc.*(o2conc).^5)).*exp(-
67520./Flame_Temp_EGR_40).*((v/1e6)/cr))./(rpm*6);

```

```

%
Mc_ppm(1,:) =
((sum(Mc_Nox_kmol_0,2)')*1e4)./(alpha.*(nCO2+nH2O+nN2+nO2));
Mc_ppm(2,:) =
((sum(Mc_Nox_kmol_5,2)')*1e4)./(alpha.*(nCO2+nH2O+nN2+nO2));
Mc_ppm(3,:) =
((sum(Mc_Nox_kmol_10,2)')*1e4)./(alpha.*(nCO2+nH2O+nN2+nO2));
Mc_ppm(4,:) =
((sum(Mc_Nox_kmol_15,2)')*1e4)./(alpha.*(nCO2+nH2O+nN2+nO2));
Mc_ppm(5,:) =
((sum(Mc_Nox_kmol_20,2)')*1e4)./(alpha.*(nCO2+nH2O+nN2+nO2));
Mc_ppm(6,:) =
((sum(Mc_Nox_kmol_25,2)')*1e4)./(alpha.*(nCO2+nH2O+nN2+nO2));
Mc_ppm(7,:) =
((sum(Mc_Nox_kmol_30,2)')*1e4)./(alpha.*(nCO2+nH2O+nN2+nO2));
Mc_ppm(8,:) =
((sum(Mc_Nox_kmol_35,2)')*1e4)./(alpha.*(nCO2+nH2O+nN2+nO2));
Mc_ppm(9,:) =
((sum(Mc_Nox_kmol_40,2)')*1e4)./(alpha.*(nCO2+nH2O+nN2+nO2));

% Heywood Method
H_Nox_kmol_0 =
(((6e16)./(Flame_Temp_EGR_0.^5)).*((n2conc.*(o2conc).^5)).*exp(-
69090./Flame_Temp_EGR_0).*((v/1e6)/cr))./(rpm*6);
H_Nox_kmol_5 =
(((6e16)./(Flame_Temp_EGR_5.^5)).*((n2conc.*(o2conc).^5)).*exp(-
69090./Flame_Temp_EGR_5).*((v/1e6)/cr))./(rpm*6);
H_Nox_kmol_10 =
(((6e16)./(Flame_Temp_EGR_10.^5)).*((n2conc.*(o2conc).^5)).*exp(-
69090./Flame_Temp_EGR_10).*((v/1e6)/cr))./(rpm*6);
H_Nox_kmol_15 =
(((6e16)./(Flame_Temp_EGR_15.^5)).*((n2conc.*(o2conc).^5)).*exp(-
69090./Flame_Temp_EGR_15).*((v/1e6)/cr))./(rpm*6);
H_Nox_kmol_20 =
(((6e16)./(Flame_Temp_EGR_20.^5)).*((n2conc.*(o2conc).^5)).*exp(-
69090./Flame_Temp_EGR_20).*((v/1e6)/cr))./(rpm*6);
H_Nox_kmol_25 =
(((6e16)./(Flame_Temp_EGR_25.^5)).*((n2conc.*(o2conc).^5)).*exp(-
69090./Flame_Temp_EGR_25).*((v/1e6)/cr))./(rpm*6);
H_Nox_kmol_30 =
(((6e16)./(Flame_Temp_EGR_30.^5)).*((n2conc.*(o2conc).^5)).*exp(-
69090./Flame_Temp_EGR_30).*((v/1e6)/cr))./(rpm*6);
H_Nox_kmol_35 =
(((6e16)./(Flame_Temp_EGR_35.^5)).*((n2conc.*(o2conc).^5)).*exp(-
69090./Flame_Temp_EGR_35).*((v/1e6)/cr))./(rpm*6);
H_Nox_kmol_40 =
(((6e16)./(Flame_Temp_EGR_40.^5)).*((n2conc.*(o2conc).^5)).*exp(-
69090./Flame_Temp_EGR_40).*((v/1e6)/cr))./(rpm*6);
%
H_ppm(1,:) =
((sum(H_Nox_kmol_0,2)')*1e5)./(alpha.*(nCO2+nH2O+nN2+nO2));

```

```

        H_ppm(2,:) =
        ((sum(H_Nox_kmol_5,2)')*1e5)./(alpha.*(nCO2+nH2O+nN2+nO2));
        H_ppm(3,:) =
        ((sum(H_Nox_kmol_10,2)')*1e5)./(alpha.*(nCO2+nH2O+nN2+nO2));
        H_ppm(4,:) =
        ((sum(H_Nox_kmol_15,2)')*1e5)./(alpha.*(nCO2+nH2O+nN2+nO2));
        H_ppm(5,:) =
        ((sum(H_Nox_kmol_20,2)')*1e5)./(alpha.*(nCO2+nH2O+nN2+nO2));
        H_ppm(6,:) =
        ((sum(H_Nox_kmol_25,2)')*1e5)./(alpha.*(nCO2+nH2O+nN2+nO2));
        H_ppm(7,:) =
        ((sum(H_Nox_kmol_30,2)')*1e5)./(alpha.*(nCO2+nH2O+nN2+nO2));
        H_ppm(8,:) =
        ((sum(H_Nox_kmol_35,2)')*1e5)./(alpha.*(nCO2+nH2O+nN2+nO2));
        H_ppm(9,:) =
        ((sum(H_Nox_kmol_40,2)')*1e5)./(alpha.*(nCO2+nH2O+nN2+nO2));

%===== Varying the RPM=====
        RPM = [1000; 1500 ;1800 ;2000; 2500 ;3000; 3500; 4000; 4500; 5000;
5500; 6000];
        noxkmRpm = (4.7e13.*(n2conc(10,1).*(o2conc(10,1)).^5).*exp(-
67837./Flame_Temp_EGR_0(10,:)).*((v/1e6)/cr))./(RPM*6);% kmol
        ppmRPM =
        ((sum(noxkmRpm,2)')*1e6)./(alpha(1,10).*(nCO2(1,10)+nH2O(1,10)+nN2+nO2(1,10)))
;

% ===== Varying the Polytropic index =====
        polyV = (1:0.01:1.35);
        Flame_Temp_EGR_n= cell(1,36);
        for j=1:36
            Flame_Temp_EGR_n{j} = Flame_Temp_EGR(10,1);
        end
        for G = (1:36)
            for i=1:1:16
                Flame_Temp_EGR_n{G}(:,i+1) =
                Flame_Temp_EGR_n{G}(:,i).*(((p2(i))./pressure(i))).^((polyV(G)-
1)./polyV(G)));
            end
        end
        Nox_kmol_n= cell(1,36);
        for G = (1:36)
            Nox_kmol_n{G}=
            (4.7e13.*(n2conc(10,1).*(o2conc(10,1)).^5).*exp(-
67837./Flame_Temp_EGR_n{G}(1,1:16)).*((v/1e6)/cr))./(rpm*6);
        end
        ppm_n_d= cell(1,36);
        for G = (1:36)
            ppm_n_d{G} =
            ((sum(Nox_kmol_n{G},2)')*1e6)./(alpha(1,10).*(nCO2(1,10)+nH2O(1,10)+nN2+nO2(1,
10))));

```

```

end
ppm_n = cell2mat(ppm_n_d);

%===== varying the gas constant R=====
Rv =(7500:50:9500)';
nnv = ((ps*1e5)*((v/1e6)*u))./(Rv*fmixT);
alphav = nnv./(((FER)+(x*(1+3.76))));
o2concv = ((nO2.*alphav)./((v/1e6)/cr))';
n2concv = ((nN2.*alphav)./((v/1e6)/cr))';
Nox_kmol_0R =
(4.7e13.*((n2concv(10,:))'.*(o2concv(10,:))'.^5).*exp(-
67837./Flame_Temp_EGR_0(10,:)).*((v/1e6)/cr))./(rpm*6);
ppmR(1,:) =
((sum(Nox_kmol_0R,2)')*1e6)./(alpha(1,10).*(nCO2(1,10)+nH2O(1,10)+nN2+nO2(1,10
))));

% ===== Vary efficiency =====
uv =(0.1:0.1:2)';
nuv = ((ps*1e5)*((v/1e6)*uv))./(R*fmixT);
alphauv = nuv./(((FER)+(x*(1+3.76))));
o2concuuv = ((nO2.*alphauv)./((v/1e6)/cr))';
n2concuuv = ((nN2.*alphauv)./((v/1e6)/cr))';
Nox_kmol_0u =
(4.7e13.*((n2concuuv(10,:))'.*(o2concuuv(10,:))'.^5).*exp(-
67837./Flame_Temp_EGR_0(10,:)).*((v/1e6)/cr))./(rpm*6);
ppmu(1,:) =
((sum(Nox_kmol_0u,2)')*1e6)./(alpha(1,10).*(nCO2(1,10)+nH2O(1,10)+nN2+nO2(1,10
))));

%===== varying the Compression Ratio=====
CRv =(8:0.5:30)';
vol_combustcrv = (v/1e6)./CRv; % (v/1e6) of combustion Zone
o2conccrv = ((nO2(1,10).*alpha(1,10))./vol_combustcrv)';
n2conccrv = ((nN2.*alpha(1,10))./vol_combustcrv)';
Nox_kmol_0crv = (4.7e13.*((n2conccrv)'.*(o2conccrv')^5).*exp(-
67837./Flame_Temp_EGR_0(10,:)).*vol_combustcrv)./(rpm*6);
ppmcrv =
((sum(Nox_kmol_0crv,2)')*1e6)./(alpha(1,10).*(nCO2(1,10)+nH2O(1,10)+nN2+nO2(1,
10))));

% ===== w = selects EGR for results look up and display
w=app.EGRDropDown.Value;
if app.EGRDropDown.Value == 'a'
    w = 1;
elseif app.EGRDropDown.Value == 'b'
    w = 2;
elseif app.EGRDropDown.Value == 'c'
    w = 3;
elseif app.EGRDropDown.Value == 'd'
    w = 4;
elseif app.EGRDropDown.Value == 'e'
    w = 5;
elseif app.EGRDropDown.Value == 'f'

```

```
        w = 6;
elseif app.EGRDropDown.Value == 'g'
        w = 7;
elseif app.EGRDropDown.Value == 'h'
        w = 8;
elseif app.EGRDropDown.Value == 'i'
        w = 9;
end

% ===== z = selects FER for results look up and display
if app.FERDropDown.Value == 'a'
    z=1;
elseif app.FERDropDown.Value == 'b'
    z=2;
elseif app.FERDropDown.Value == 'c'
    z=3;
elseif app.FERDropDown.Value == 'd'
    z=4;
elseif app.FERDropDown.Value == 'e'
    z=5;
elseif app.FERDropDown.Value == 'f'
    z=6;
elseif app.FERDropDown.Value == 'g'
    z=7;
elseif app.FERDropDown.Value == 'h'
    z=8;
elseif app.FERDropDown.Value == 'i'
    z=9;
elseif app.FERDropDown.Value == 'j'
    z=10;
elseif app.FERDropDown.Value == 'k'
    z=11;
elseif app.FERDropDown.Value == 'l'
    z=12;
elseif app.FERDropDown.Value == 'm'
    z=13;
elseif app.FERDropDown.Value == 'n'
    z=14;
elseif app.FERDropDown.Value == 'o'
    z=15;
elseif app.FERDropDown.Value == 'p'
    z=16;
elseif app.FERDropDown.Value == 'q'
    z=17;
elseif app.FERDropDown.Value == 'r'
    z=18;
elseif app.FERDropDown.Value == 's'
    z=19;
elseif app.FERDropDown.Value == 't'
    z=20;
elseif app.FERDropDown.Value == 'u'
    z=21;
elseif app.FERDropDown.Value == 'v'
```

```

        z=22;
    elseif app.FERDropDown.Value == 'w'
        z=23;
    elseif app.FERDropDown.Value == 'x'
        z=24;
    elseif app.FERDropDown.Value == 'y'
        z=25;
    end

    % Rebalances the Resultants for EGR to show in balanced equation
    nN2_EGR = EGR(1,w)*nN2;
    nCO2_EGR = EGR(1,w)*nCO2;
    nH2O_EGR = EGR(1,w)*nH2O;
    nCO_EGR = EGR(1,w)*nCO;
    nO2_EGR = EGR(1,w)*nO2;
    nH2_EGR = EGR(1,w)*nH2;

    % egr effect
    p_ch = [(((ppm(2,10))-(ppm(1,10)))/(ppm(1,10))*100),(((ppm(3,10))-
(ppm(2,10)))/(ppm(2,10))*100),(((ppm(4,10))-
(ppm(3,10)))/(ppm(3,10))*100),(((ppm(5,10))-
(ppm(4,10)))/(ppm(4,10))*100),(((ppm(6,10))-
(ppm(5,10)))/(ppm(5,10))*100),(((ppm(7,10))-
(ppm(6,10)))/(ppm(6,10))*100),(((ppm(8,10))-
(ppm(7,10)))/(ppm(7,10))*100),(((ppm(9,10))-(ppm(8,10)))/(ppm(8,10))*100)]*-1;
    p_egr = [5 10 15 20 25 30 35 40];

%=====Displaying results=====
    app.EditField_2.Value = x;
    app.H2OEditField.Value = (nN2_EGR);
    app.MassPercentFuel1EditField.Value = (M_P_Fuel);
    app.AFRStoichEditField.Value = (AFR);
    app.EditField_3.Value = nCO2_EGR(1,z);
    app.CO2EditField.Value = nH2O_EGR(1,z);
    app.O2EditField.Value = nCO_EGR(1,z);
    app.N2EditField.Value = nO2_EGR(1,z);
    app.EnthalpyLHVMJKgEditField.Value = LHV(1,z);
    app.EnthalpyHHVMJKgEditField.Value = HHV(1,z);
    app.CO2kgFuelkgEditField.Value = CO2_mass_kg(1,z);
    app.AFRafterFEREditField.Value = (AFR)/FER(1,z);
    app.FlameTempKEditField.Value = Flame_Temp_EGR(z,w);
    app.FlameTempPressureKEditField.Value = Flame_Temp_EGR_p(w,z);
    app.NOxPPMMerkerEditField.Value = ppm(w,z);
    app.COEditField.Value = pCO(1,z);
    app.O2KMolsm3EditField.Value = o2conc(z,1);
    app.N2KMolsm3EditField.Value = n2conc(z,1);
    app.EditField_5.Value = b;
    app.nH2EditField_2.Value = nH2_EGR(1,z);
    app.NOxPPMMcAllisterEditField.Value = Mc_ppm(w,z);
    app.NOxPPMHeywoodEditField.Value = H_ppm(w,z);

%===== Vectors for Axis =====

```

```

    EGR_per = [0 5 10 15 20 25 30 35 40];
    xa = (1:21);
    UV = (10:10:200);
    ca = (0:15);

%=====Tabbed Plots =====
    plot(app.UIAxes,FER(1:13),ppm(:,1:13));
    plot(app.UIAxes8,FER(13:25),pCO(:,13:25));
    legend(app.UIAxes, 'Location', 'northwest')
    legend(app.UIAxes, '0% EGR', '5% EGR', '10% EGR', '15% EGR', '20%
EGR', '25% EGR', '30% EGR', '35% EGR', '40% EGR');
    mesh(app.UIAxes2,FER,EGR_per,ppm)
    app.UIAxes2.ZLabel.String='NO_{x} PPM';
    mesh(app.UIAxes3,FER,EGR_per,(Flame_Temp_EGR'))
    mesh(app.UIAxes14,FER,ca,(Flame_Temp_EGR_0'))
    app.UIAxes14.ZLabel.String='Flame Temp per Degree (k)';
    app.UIAxes3.ZLabel.String='Flame Temp pre Pressure (k)';
    plot(app.UIAxes4,RPM,ppmRPM)
    plot(app.UIAxes5,polyV,ppm_n)
    plot(app.UIAxes6,Rv,ppmR)
    plot(app.UIAxes7,CRv,ppmcrv)
    mesh(app.UIAxes9,FER,EGR_per,((nCO2.*(EGR')).*(sum(
masses*CO2(1:4,:)))/fuel_mass)) %nCO2*(sum( masses*CO2(1:4,:)))/fuel_mass
    mesh(app.UIAxes10,FER,EGR_per,((nH2O.*(EGR')).*(sum(
masses*H2O_L(1:4,:)))/fuel_mass))
    app.UIAxes9.ZLabel.String='CO_{2} kg/Fuel Kg';
    app.UIAxes10.ZLabel.String='H_{2}O kg/Fuel Kg';
    plot(app.UIAxes11,FER(1:13),H_ppm(1,1:13));
    hold (app.UIAxes11, 'on')
    plot(app.UIAxes11,FER(1:13),ppm(1,1:13));
    plot(app.UIAxes11,FER(1:13),Mc_ppm(1,1:13));
    hold (app.UIAxes11, 'off')
    legend(app.UIAxes11, 'Location', 'northwest')
    legend(app.UIAxes11, 'Heywood', 'Merker', 'McAlister')
    plot(app.UIAxes12,xa,(afg(:,6)*1000))%LHV
    hold (app.UIAxes12, 'on')
    plot(app.UIAxes12,xa,(afg(:,4)*10000)) %co2
    plot(app.UIAxes12,xa,(afg(:,5))) %nox
    hold (app.UIAxes12, 'off')
    legend(app.UIAxes12, 'LHV Kj/kg ', 'CO_{2} kg x10^{-4} / kg of Fuel
', 'NO_{x} max PPM Merker ')
    axis(app.UIAxes12, 'tight')
    plot(app.UIAxes13,UV,ppmu)
    axis(app.UIAxes13, 'tight')
    plot(app.UIAxes15,p_egr,p_ch)
    axis(app.UIAxes15, 'tight')

%===== External Figures =====
    figure ('name', 'O2 concentration and Flame
Temperature', 'numbertitle', 'off')
    grid on
    title ('O_{2} Concentration and flame temperature against FER')
    yyaxis('left')
    plot(FER,o2conc)
    ylabel ('O_{2} Concentration KMols/m^{3}')

```

```

yyaxis('right')
plot(FER,Flame_Temp_EGR_0(:,16))
ylabel ('Flame Temperature after Pressure Kelvin')
axis tight % we all like it tight
xlabel ('FER')
legend ('O_{2} Concentration','Flame Temperature')

figure ('name','NOx PPM and CO%','numbertitle','off')
hold on
grid on
title ('NO_{x} PPM and Carbon Monoxide against FER')
yyaxis('left')
plot(FER,ppm(1,:), '-','color','k')
plot(FER,ppm(2,:), '-','color','b')
plot(FER,ppm(3,:), '-','color','g')
plot(FER,ppm(4,:), '-','color','r')
plot(FER,ppm(5,:), '-','color','c')
plot(FER,ppm(6,:), '-','color','m')
plot(FER,ppm(7,:), '-','color','y')
plot(FER,ppm(8,:), '--','color','k')
plot(FER,ppm(9,:), '--','color','b')
ylabel ('NO_{x} PPM')
yyaxis('right')
plot(FER,pCO, '--','color','r')
ylabel ('CO%')
axis tight % we all like it tight
xlabel ('FER')
legend ('PPM NO_{X} 0% EGR','PPM NO_{X} 5% EGR','PPM NO_{X} 10%
EGR','PPM NO_{X} 15% EGR','PPM NO_{X} 20% EGR','PPM NO_{X} 25% EGR','PPM
NO_{X} 30% EGR','PPM NO_{X} 35% EGR','PPM NO_{X} 40% EGR','CO%');
legend('Location','northwest')
hold off

% =====Adusting Background Colours of Results
=====

if b<1
    app.MassPercentFuel1EditField.BackgroundColor=('[0.85 0.33
0.10]');
    app.EditField_5.BackgroundColor=('[0.85 0.33 0.10]');
elseif b==1
    app.MassPercentFuel1EditField.BackgroundColor=('w');
    app.EditField_5.BackgroundColor=('w');
end

app.AFRStoichEditField.BackgroundColor=('c');

if w==1 && z>13
    app.EditField_3.BackgroundColor=('r');
    app.CO2EditField.BackgroundColor=('r');
    app.H2OEditField.BackgroundColor=('r');
    app.N2EditField.BackgroundColor=('r');
    app.O2EditField.BackgroundColor=('r');

```

```

app.nH2EditField_2.BackgroundColor=('r');
app.FlameTempPressureKEditField.BackgroundColor=('r');
app.FlameTempKEditField.BackgroundColor=('r');
app.NOxPPMMerkerEditField.BackgroundColor=('r');
app.NOxPPMMcAllisterEditField.BackgroundColor=('r');
app.NOxPPMHeywoodEditField.BackgroundColor=('r');
app.COEditField.BackgroundColor=('r'); %no egr effect
app.N2KMolsm3EditField.BackgroundColor=('r'); %no egr effect
app.O2KMolsm3EditField.BackgroundColor=('w'); %no egr effect
app.AFRafterFEREditField.BackgroundColor=('r'); %no egr effect
app.EnthalpyLHVMJKgEditField.BackgroundColor=('r'); %no egr
effect
app.EnthalpyHHVMJKgEditField.BackgroundColor=('r'); %no egr
effect

app.CO2kgFuelkgEditField.BackgroundColor=('r'); %no egr effect
app.EditField_3.FontColor=('k');
app.CO2EditField.FontColor=('k');
app.H2OEditField.FontColor=('k');
app.N2EditField.FontColor=('k');
app.O2EditField.FontColor=('k');
app.nH2EditField_2.FontColor=('k');
app.FlameTempPressureKEditField.FontColor=('k');
app.FlameTempKEditField.FontColor=('k');
app.NOxPPMMerkerEditField.FontColor=('k');
app.NOxPPMMcAllisterEditField.FontColor=('k');
app.NOxPPMHeywoodEditField.FontColor=('k');
elseif w>1 && z>13
    app.EditField_3.BackgroundColor=('[0.74 0.08 0.18]');
    app.CO2EditField.BackgroundColor=('[0.74 0.08 0.18]');
    app.H2OEditField.BackgroundColor=('[0.74 0.08 0.18]');
    app.N2EditField.BackgroundColor=('[0.74 0.08 0.18]');
    app.O2EditField.BackgroundColor=('[0.74 0.08 0.18]');
    app.nH2EditField_2.BackgroundColor=('[0.74 0.08 0.18]');
    app.FlameTempPressureKEditField.BackgroundColor=('[0.74 0.08
0.18]');
    app.FlameTempKEditField.BackgroundColor=('[0.74 0.08 0.18]');
    app.NOxPPMMerkerEditField.BackgroundColor=('[0.74 0.08
0.18]');
    app.NOxPPMMcAllisterEditField.BackgroundColor=('[0.74 0.08
0.18]');
    app.NOxPPMHeywoodEditField.BackgroundColor=('[0.74 0.08
0.18]');
    app.EditField_3.FontColor=('w');
    app.CO2EditField.FontColor=('w');
    app.H2OEditField.FontColor=('w');
    app.N2EditField.FontColor=('w');
    app.O2EditField.FontColor=('w');
    app.nH2EditField_2.FontColor=('w');
    app.FlameTempPressureKEditField.FontColor=('w');
    app.FlameTempKEditField.FontColor=('w');
    app.NOxPPMMerkerEditField.FontColor=('w');
    app.NOxPPMMcAllisterEditField.FontColor=('w');
    app.NOxPPMHeywoodEditField.FontColor=('w');

```

```

app.COEditField.BackgroundColor=('r'); %no egr effect
app.N2KMolsm3EditField.BackgroundColor=('r'); %no egr effect
app.O2KMolsm3EditField.BackgroundColor=('w'); %no egr effect
app.AFRafterFEREditField.BackgroundColor=('r'); %no egr effect
app.EnthalpyLHVMJKgEditField.BackgroundColor=('r'); %no egr

effect

effect

app.EnthalpyHHVMJKgEditField.BackgroundColor=('r'); %no egr

app.CO2kgFuelkgEditField.BackgroundColor=('r'); %no egr effect
elseif w==1 && z == 13
    app.EditField_3.BackgroundColor=('c');
    app.CO2EditField.BackgroundColor=('c');
    app.H2OEditField.BackgroundColor=('c');
    app.N2EditField.BackgroundColor=('c');
    app.O2EditField.BackgroundColor=('c');
    app.nH2EditField_2.BackgroundColor=('c');
    app.FlameTempPressureKEditField.BackgroundColor=('c');
    app.FlameTempKEditField.BackgroundColor=('c');
    app.NOxPPMMerkerEditField.BackgroundColor=('w');
    app.NOxPPMAllisterEditField.BackgroundColor=('w');
    app.NOxPPMHeywoodEditField.BackgroundColor=('w');
    app.COEditField.BackgroundColor=('w'); %no egr effect
    app.N2KMolsm3EditField.BackgroundColor=('c'); %no egr effect
    app.O2KMolsm3EditField.BackgroundColor=('w'); %no egr effect
    app.AFRafterFEREditField.BackgroundColor=('c'); %no egr effect
    app.EnthalpyLHVMJKgEditField.BackgroundColor=('c'); %no egr

effect

effect

app.EnthalpyHHVMJKgEditField.BackgroundColor=('c'); %no egr

app.CO2kgFuelkgEditField.BackgroundColor=('c'); %no egr effect
app.EditField_3.FontColor=('k');
app.CO2EditField.FontColor=('k');
app.H2OEditField.FontColor=('k');
app.N2EditField.FontColor=('k');
app.O2EditField.FontColor=('k');
app.nH2EditField_2.FontColor=('k');
app.FlameTempPressureKEditField.FontColor=('k');
app.FlameTempKEditField.FontColor=('k');
app.NOxPPMMerkerEditField.FontColor=('k');
app.NOxPPMAllisterEditField.FontColor=('k');
app.NOxPPMHeywoodEditField.FontColor=('k');
elseif w>1 && z == 13
    app.EditField_3.BackgroundColor=('b');
    app.CO2EditField.BackgroundColor=('b');
    app.H2OEditField.BackgroundColor=('b');
    app.N2EditField.BackgroundColor=('b');
    app.O2EditField.BackgroundColor=('b');
    app.nH2EditField_2.BackgroundColor=('b');
    app.FlameTempPressureKEditField.BackgroundColor=('b');
    app.FlameTempKEditField.BackgroundColor=('b');
    app.NOxPPMMerkerEditField.BackgroundColor=('w');
    app.NOxPPMAllisterEditField.BackgroundColor=('w');
    app.NOxPPMHeywoodEditField.BackgroundColor=('w');

```

```

app.COEditField.BackgroundColor=('w'); %no egr effect
app.N2KMolsm3EditField.BackgroundColor=('c'); %no egr effect
app.O2KMolsm3EditField.BackgroundColor=('w'); %no egr effect
app.AFRafterFEREditField.BackgroundColor=('c'); %no egr effect
app.EnthalpyLHVMJKgEditField.BackgroundColor=('c'); %no egr
effect
effect

app.EnthalpyHHVMJKgEditField.BackgroundColor=('c'); %no egr

app.CO2kgFuelkgEditField.BackgroundColor=('c'); %no egr effect
app.EditField_3.FontColor=('w');
app.CO2EditField.FontColor=('w');
app.H2OEditField.FontColor=('w');
app.N2EditField.FontColor=('w');
app.O2EditField.FontColor=('w');
app.nH2EditField_2.FontColor=('w');
app.FlameTempPressureKEditField.FontColor=('w');
app.FlameTempKEditField.FontColor=('w');
app.NOxPPMMerkerEditField.FontColor=('k');
app.NOxPPMMcAllisterEditField.FontColor=('k');
app.NOxPPMHeywoodEditField.FontColor=('k');
elseif w==1 && z<13
    app.EditField_3.BackgroundColor=('g');
    app.CO2EditField.BackgroundColor=('g');
    app.H2OEditField.BackgroundColor=('g');
    app.N2EditField.BackgroundColor=('g');
    app.O2EditField.BackgroundColor=('g');
    app.nH2EditField_2.BackgroundColor=('g');
    app.FlameTempPressureKEditField.BackgroundColor=('g');
    app.FlameTempKEditField.BackgroundColor=('g');
    app.NOxPPMMerkerEditField.BackgroundColor=('g');
    app.NOxPPMMcAllisterEditField.BackgroundColor=('g');
    app.NOxPPMHeywoodEditField.BackgroundColor=('g');
    app.COEditField.BackgroundColor=('w'); %no egr effect
    app.N2KMolsm3EditField.BackgroundColor=('c'); %no egr effect
    app.O2KMolsm3EditField.BackgroundColor=('g'); %no egr effect
    app.AFRafterFEREditField.BackgroundColor=('g'); %no egr effect
    app.EnthalpyLHVMJKgEditField.BackgroundColor=('g'); %no egr
effect
effect

app.EnthalpyHHVMJKgEditField.BackgroundColor=('g'); %no egr

app.CO2kgFuelkgEditField.BackgroundColor=('g'); %no egr effect
app.EditField_3.FontColor=('k');
app.CO2EditField.FontColor=('k');
app.H2OEditField.FontColor=('k');
app.N2EditField.FontColor=('k');
app.O2EditField.FontColor=('k');
app.nH2EditField_2.FontColor=('k');
app.FlameTempPressureKEditField.FontColor=('k');
app.FlameTempKEditField.FontColor=('k');
app.NOxPPMMerkerEditField.FontColor=('k');
app.NOxPPMMcAllisterEditField.FontColor=('k');
app.NOxPPMHeywoodEditField.FontColor=('k');
elseif w>1 && z<13

```

```

app.EditField_3.BackgroundColor=('[0.47 0.67 0.19']);
app.CO2EditField.BackgroundColor=('[0.47 0.67 0.19']);
app.H2OEditField.BackgroundColor=('[0.47 0.67 0.19']);
app.N2EditField.BackgroundColor=('[0.47 0.67 0.19']);
app.O2EditField.BackgroundColor=('[0.47 0.67 0.19']);
app.nH2EditField_2.BackgroundColor=('[0.47 0.67 0.19']);
app.FlameTempPressureKEditField.BackgroundColor=('[0.47 0.67
0.19']);

app.FlameTempKEditField.BackgroundColor=('[0.47 0.67 0.19']);
app.NOxPPMMerkerEditField.BackgroundColor=('[0.47 0.67
0.19']);

app.NOxPPMMcAllisterEditField.BackgroundColor=('[0.47 0.67
0.19']);

app.NOxPPMHeywoodEditField.BackgroundColor=('[0.47 0.67
0.19']);

app.COEditField.BackgroundColor=('w'); %no egr effect
app.N2KMolsm3EditField.BackgroundColor=('c'); %no egr effect
app.O2KMolsm3EditField.BackgroundColor=('g'); %no egr effect
app.AFRafterFEREditField.BackgroundColor=('g'); %no egr effect
app.EnthalpyLHVMJKgEditField.BackgroundColor=('g'); %no egr
effect

app.EnthalpyHHVMJKgEditField.BackgroundColor=('g'); %no egr
effect

app.CO2kgFuelkgEditField.BackgroundColor=('g'); %no egr effect
app.EditField_3.FontColor=('k');
app.CO2EditField.FontColor=('k');
app.H2OEditField.FontColor=('k');
app.N2EditField.FontColor=('k');
app.O2EditField.FontColor=('k');
app.nH2EditField_2.FontColor=('k');
app.FlameTempPressureKEditField.FontColor=('k');
app.FlameTempKEditField.FontColor=('k');
app.NOxPPMMerkerEditField.FontColor=('k');
app.NOxPPMMcAllisterEditField.FontColor=('k');
app.NOxPPMHeywoodEditField.FontColor=('k');
end
end
end

% Component initialization
methods (Access = private)

% Create UIFigure and components
function createComponents(app)

% Create Ryan_James_Powertrain_app and hide until all components
are created
app.Ryan_James_Powertrain_app = uifigure('Visible', 'off');
app.Ryan_James_Powertrain_app.AutoResizeChildren = 'off';
app.Ryan_James_Powertrain_app.Color = [0.9412 0.9412 0.9412];

```

```

        app.Ryan_James_Powertrain_app.Colormap = [0.2431 0.149
0.6588;0.251 0.1647 0.7059;0.2588 0.1804 0.7529;0.2627 0.1961 0.7961;0.2706
0.2157 0.8353;0.2745 0.2353 0.8706;0.2784 0.2549 0.898;0.2784 0.2784
0.9216;0.2824 0.302 0.9412;0.2824 0.3216 0.9569;0.2784 0.3451 0.9725;0.2745
0.3686 0.9843;0.2706 0.3882 0.9922;0.2588 0.4118 0.9961;0.2431 0.4353 1;0.2196
0.4588 0.9961;0.1961 0.4863 0.9882;0.1843 0.5059 0.9804;0.1804 0.5294
0.9686;0.1765 0.549 0.9529;0.1686 0.5686 0.9373;0.1529 0.5922 0.9216;0.1451
0.6078 0.9098;0.1373 0.6275 0.898;0.1255 0.6471 0.8902;0.1098 0.6627
0.8745;0.0941 0.6784 0.8588;0.0706 0.6941 0.8392;0.0314 0.7098 0.8157;0.0039
0.7216 0.7922;0.0078 0.7294 0.7647;0.0431 0.7412 0.7412;0.098 0.749
0.7137;0.1412 0.7569 0.6824;0.1725 0.7686 0.6549;0.1922 0.7765 0.6235;0.2157
0.7843 0.5922;0.2471 0.7922 0.5569;0.2902 0.7961 0.5176;0.3412 0.8
0.4784;0.3922 0.8039 0.4353;0.4471 0.8039 0.3922;0.5059 0.8 0.349;0.5608
0.7961 0.3059;0.6157 0.7882 0.2627;0.6706 0.7804 0.2235;0.7255 0.7686
0.1922;0.7725 0.7608 0.1647;0.8196 0.749 0.1529;0.8627 0.7412 0.1608;0.902
0.7333 0.1765;0.9412 0.7294 0.2118;0.9725 0.7294 0.2392;0.9961 0.7451
0.2353;0.9961 0.7647 0.2196;0.9961 0.7882 0.2039;0.9882 0.8118 0.1882;0.9804
0.8392 0.1765;0.9686 0.8627 0.1647;0.9608 0.8902 0.1529;0.9608 0.9137
0.1412;0.9647 0.9373 0.1255;0.9686 0.9608 0.1059];
        app.Ryan_James_Powertrain_app.Position = [100 100 1134 704];
        app.Ryan_James_Powertrain_app.Name = 'Ryan James Combustion and
Emissions Study';
        app.Ryan_James_Powertrain_app.Resize = 'off';

% Create Fuel1Label
app.Fuel1Label = uilabel(app.Ryan_James_Powertrain_app);
app.Fuel1Label.HorizontalAlignment = 'right';
app.Fuel1Label.Position = [149 605 39 22];
app.Fuel1Label.Text = 'Fuel 1';

% Create Fuel1DropDown
app.Fuel1DropDown = uidropdown(app.Ryan_James_Powertrain_app);
app.Fuel1DropDown.Items = {'Methane', 'Acetylene', 'Ethylene',
'Ethane', 'Propane', 'n-Butane', 'iso-Butane', 'n-Pentane', 'iso-Pentane', 'n-
Hexane', 'iso-Hexane', 'n-Heptane', 'n-Octane', 'iso-Octane', 'n-Nonane', 'n-
Decane', 'iso-Decane', 'n-Dodecane', 'Methanol', 'Ethanol', 'Hydrogen'};
app.Fuel1DropDown.ItemsData = {'a', 'b', 'c', 'd', 'e', 'f', 'g',
'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u'};
app.Fuel1DropDown.BackgroundColor = [0.9294 0.6941 0.1255];
app.Fuel1DropDown.Position = [203 605 100 22];
app.Fuel1DropDown.Value = 'n';

% Create Fuel2DropDownLabel
app.Fuel2DropDownLabel = uilabel(app.Ryan_James_Powertrain_app);
app.Fuel2DropDownLabel.HorizontalAlignment = 'right';
app.Fuel2DropDownLabel.Position = [149 583 39 22];
app.Fuel2DropDownLabel.Text = 'Fuel 2';

% Create Fuel2DropDown
app.Fuel2DropDown = uidropdown(app.Ryan_James_Powertrain_app);

```

```

        app.Fuel2DropDown.Items = {'Methane', 'Acetylene', 'Ethylene',
'Ethane', 'Propane', 'n-Butane', 'iso-Butane', 'n-Pentane', 'iso-Pentane', 'n-
Hexane', 'iso-Hexane', 'n-Heptane', 'n-Octane', 'iso-Octane', 'n-Nonane', 'n-
Decane', 'iso-Decane', 'n-Dodecane', 'Methanol', 'Ethanol', 'Hydrogen',
'None'};
        app.Fuel2DropDown.ItemsData = {'a', 'b', 'c', 'd', 'e', 'f', 'g',
'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v'};
        app.Fuel2DropDown.BackgroundColor = [0.9294 0.6941 0.1255];
        app.Fuel2DropDown.Position = [203 583 100 22];
        app.Fuel2DropDown.Value = 's';

% Create ModelButton
app.ModelButton = uibutton(app.Ryan_James_Powertrain_app, 'push');
app.ModelButton.ButtonPushedFcn = createCallbackFcn(app,
@ModelButtonPushed, true);
app.ModelButton.Icon = 'combustion-1-638.jpg';
app.ModelButton.BackgroundColor = [0.149 0.149 0.149];
app.ModelButton.FontWeight = 'bold';
app.ModelButton.FontColor = [1 1 1];
app.ModelButton.Tooltip = {'Click this to show the results of any
changed drop downs or variables'};
app.ModelButton.Position = [136 329 106 30];
app.ModelButton.Text = 'Model';

% Create Label_3
app.Label_3 = uilabel(app.Ryan_James_Powertrain_app);
app.Label_3.HorizontalAlignment = 'right';
app.Label_3.Position = [305 7 33 22];
app.Label_3.Text = ') + ';

% Create EditField_2
app.EditField_2 = uieditfield(app.Ryan_James_Powertrain_app,
'numeric');
app.EditField_2.Eitable = 'off';
app.EditField_2.Position = [337 7 46 22];

% Create EditField_4
app.EditField_4 = uieditfield(app.Ryan_James_Powertrain_app,
'text');
app.EditField_4.Eitable = 'off';
app.EditField_4.Position = [102 7 59 22];

% Create EditField
app.EditField = uieditfield(app.Ryan_James_Powertrain_app,
'text');
app.EditField.Eitable = 'off';
app.EditField.Position = [230 7 65 22];

```

```
% Create Label
app.Label = uilabel(app.Ryan_James_Powertrain_app);
app.Label.HorizontalAlignment = 'right';
app.Label.Position = [554 7 38 22];
app.Label.Text = ' ---> ';

% Create EditField_3
app.EditField_3 = uieditfield(app.Ryan_James_Powertrain_app,
'numeric');
app.EditField_3.Editable = 'off';
app.EditField_3.Position = [588 7 47 22];

% Create CO2Label
app.CO2Label = uilabel(app.Ryan_James_Powertrain_app);
app.CO2Label.HorizontalAlignment = 'right';
app.CO2Label.Position = [634 7 51 22];
app.CO2Label.Text = 'CO2 + ';

% Create CO2EditField
app.CO2EditField = uieditfield(app.Ryan_James_Powertrain_app,
'numeric');
app.CO2EditField.Editable = 'off';
app.CO2EditField.Position = [684 7 48 22];

% Create H2OLabel
app.H2OLabel = uilabel(app.Ryan_James_Powertrain_app);
app.H2OLabel.HorizontalAlignment = 'right';
app.H2OLabel.Position = [731 7 48 22];
app.H2OLabel.Text = 'H2O + ';

% Create H2OEditField
app.H2OEditField = uieditfield(app.Ryan_James_Powertrain_app,
'numeric');
app.H2OEditField.Editable = 'off';
app.H2OEditField.Position = [778 7 54 22];

% Create O2376N2Label
app.O2376N2Label = uilabel(app.Ryan_James_Powertrain_app);
app.O2376N2Label.Position = [388 7 85 22];
app.O2376N2Label.Text = '(O2 + 3.76 N2)';

% Create FERDropDownLabel
app.FERDropDownLabel = uilabel(app.Ryan_James_Powertrain_app);
app.FERDropDownLabel.HorizontalAlignment = 'right';
app.FERDropDownLabel.Position = [1 7 29 22];
app.FERDropDownLabel.Text = 'FER';
```

```

% Create FERDropDown
app.FERDropDown = uidropdown(app.Ryan_James_Powertrain_app);
app.FERDropDown.Items = {'0.6', '0.65', '0.7', '0.75', '0.8',
'0.85', '0.9', '0.95', '0.96', '0.97', '0.98', '0.99', '1', '1.02', '1.04',
'1.06', '1.08', '1.1', '1.12', '1.14', '1.16', '1.18', '1.2', '1.22', '1.24'};
app.FERDropDown.ItemsData = {'a', 'b', 'c', 'd', 'e', 'f', 'g',
'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v',
'w', 'x', 'y', 'z'};
app.FERDropDown.BackgroundColor = [0.9294 0.6941 0.1255];
app.FERDropDown.Position = [31 7 58 22];
app.FERDropDown.Value = 'm';

% Create EGRDropDownLabel
app.EGRDropDownLabel = uilabel(app.Ryan_James_Powertrain_app);
app.EGRDropDownLabel.HorizontalAlignment = 'right';
app.EGRDropDownLabel.Position = [464 7 32 22];
app.EGRDropDownLabel.Text = 'EGR';

% Create EGRDropDown
app.EGRDropDown = uidropdown(app.Ryan_James_Powertrain_app);
app.EGRDropDown.Items = {'0%', '5%', '10%', '15%', '20%', '25%',
'30%', '35%', '40%'};
app.EGRDropDown.ItemsData = {'a', 'b', 'c', 'd', 'e', 'f', 'g',
'h', 'i'};
app.EGRDropDown.BackgroundColor = [0.9294 0.6941 0.1255];
app.EGRDropDown.Position = [506 7 60 22];
app.EGRDropDown.Value = 'a';

% Create N2Label_2
app.N2Label_2 = uilabel(app.Ryan_James_Powertrain_app);
app.N2Label_2.HorizontalAlignment = 'right';
app.N2Label_2.Position = [831 7 45 22];
app.N2Label_2.Text = 'N2 + ';

% Create N2EditField
app.N2EditField = uieditfield(app.Ryan_James_Powertrain_app,
'numeric');
app.N2EditField.Editable = 'off';
app.N2EditField.Position = [875 7 50 22];

% Create O2Label
app.O2Label = uilabel(app.Ryan_James_Powertrain_app);
app.O2Label.HorizontalAlignment = 'right';
app.O2Label.Position = [924 7 42 22];
app.O2Label.Text = 'O2 + ';

% Create O2EditField

```

```

        app.O2EditField = uieditfield(app.Ryan_James_Powertrain_app,
'numeric');
        app.O2EditField.Editable = 'off';
        app.O2EditField.Position = [966 7 42 22];

        % Create ModellingFuelsandNOxLabel
        app.ModellingFuelsandNOxLabel =
uilabel(app.Ryan_James_Powertrain_app);
        app.ModellingFuelsandNOxLabel.FontSize = 40;
        app.ModellingFuelsandNOxLabel.Position = [382 630 505 71];
        app.ModellingFuelsandNOxLabel.Text = 'Modelling Fuels and NOx';

        % Create PercentageofFuel1inMixSliderLabel
        app.PercentageofFuel1inMixSliderLabel =
uilabel(app.Ryan_James_Powertrain_app);
        app.PercentageofFuel1inMixSliderLabel.HorizontalAlignment =
'right';
        app.PercentageofFuel1inMixSliderLabel.FontColor = [0.9294 0.6941
0.1255];
        app.PercentageofFuel1inMixSliderLabel.Position = [37 664 223 22];
        app.PercentageofFuel1inMixSliderLabel.Text = 'Percentage of Fuel 1
in Mix';

        % Create PercentageofFuel1inMixSlider
        app.PercentageofFuel1inMixSlider =
uislider(app.Ryan_James_Powertrain_app);
        app.PercentageofFuel1inMixSlider.Limits = [50 100];
        app.PercentageofFuel1inMixSlider.MajorTicks = [50 60 70 80 90
100];
        app.PercentageofFuel1inMixSlider.MajorTickLabels = {'50', '60',
'70', '80', '90', '100'};
        app.PercentageofFuel1inMixSlider.FontColor = [0.9294 0.6941
0.1255];
        app.PercentageofFuel1inMixSlider.Position = [71 658 230 3];
        app.PercentageofFuel1inMixSlider.Value = 50;

        % Create Label_5
        app.Label_5 = uilabel(app.Ryan_James_Powertrain_app);
        app.Label_5.HorizontalAlignment = 'right';
        app.Label_5.Position = [160 7 25 22];
        app.Label_5.Text = '+ ';

        % Create EditField_5
        app.EditField_5 = uieditfield(app.Ryan_James_Powertrain_app,
'numeric');
        app.EditField_5.Editable = 'off';
        app.EditField_5.Position = [187 7 35 22];

```

```
% Create Label_6
app.Label_6 = uilabel(app.Ryan_James_Powertrain_app);
app.Label_6.Position = [82 7 25 22];
app.Label_6.Text = ' (';

% Create TabGroup
app.TabGroup = uitabgroup(app.Ryan_James_Powertrain_app);
app.TabGroup.AutoResizeChildren = 'off';
app.TabGroup.Position = [320 74 797 564];

% Create NOxPPMTab
app.NOxPPMTab = uitab(app.TabGroup);
app.NOxPPMTab.AutoResizeChildren = 'off';
app.NOxPPMTab.Title = 'NOx PPM';
app.NOxPPMTab.BackgroundColor = 'none';

% Create UIAxes
app.UIAxes = uiaxes(app.NOxPPMTab);
title(app.UIAxes, 'NO_{x} PPM Merker against FER')
xlabel(app.UIAxes, 'Fuel Equivalence Ratio')
ylabel(app.UIAxes, 'NO_{x} PPM')
app.UIAxes.PlotBoxAspectRatio = [1.61495317476932 1 1];
app.UIAxes.XLim = [0.6 1];
app.UIAxes.XGrid = 'on';
app.UIAxes.YGrid = 'on';
app.UIAxes.Position = [12 9 773 521];

% Create COTab
app.COTab = uitab(app.TabGroup);
app.COTab.Title = 'CO%';

% Create UIAxes8
app.UIAxes8 = uiaxes(app.COTab);
title(app.UIAxes8, 'CO% against FER')
xlabel(app.UIAxes8, 'Fuel Equivalence Ratio')
ylabel(app.UIAxes8, 'CO %')
app.UIAxes8.PlotBoxAspectRatio = [1.63419124140751 1 1];
app.UIAxes8.XLim = [1 1.24];
app.UIAxes8.XGrid = 'on';
app.UIAxes8.YGrid = 'on';
app.UIAxes8.Position = [1 10 786 513];

% Create CO23DTab
app.CO23DTab = uitab(app.TabGroup);
app.CO23DTab.Title = 'CO2 3D';

% Create UIAxes9
```

```

app.UIAxes9 = uiaxes(app.CO23DTab);
title(app.UIAxes9, 'CO_{2} Kg / Kg Fuel 3D Vs FER and EGR')
xlabel(app.UIAxes9, 'FER')
ylabel(app.UIAxes9, 'EGR%')
app.UIAxes9.XGrid = 'on';
app.UIAxes9.YGrid = 'on';
app.UIAxes9.ZGrid = 'on';
app.UIAxes9.Position = [9 11 764 521];

% Create H2O3DTab
app.H2O3DTab = uitab(app.TabGroup);
app.H2O3DTab.Title = 'H2O 3D';

% Create UIAxes10
app.UIAxes10 = uiaxes(app.H2O3DTab);
title(app.UIAxes10, 'H_{2}O Kg / Kg Fuel vs EGR and FER')
xlabel(app.UIAxes10, 'FER')
ylabel(app.UIAxes10, 'EGR%')
app.UIAxes10.XGrid = 'on';
app.UIAxes10.YGrid = 'on';
app.UIAxes10.ZGrid = 'on';
app.UIAxes10.Position = [1 1 795 531];

% Create NOx3DVsfERandEGRTab
app.NOx3DVsfERandEGRTab = uitab(app.TabGroup);
app.NOx3DVsfERandEGRTab.AutoResizeChildren = 'off';
app.NOx3DVsfERandEGRTab.Title = 'NOx 3D Vs FER and EGR';

% Create UIAxes2
app.UIAxes2 = uiaxes(app.NOx3DVsfERandEGRTab);
title(app.UIAxes2, 'NO_{x} PPM Merker against EGR and FER')
xlabel(app.UIAxes2, 'Fuel Equivalence Ratio ')
ylabel(app.UIAxes2, 'EGR %')
app.UIAxes2.PlotBoxAspectRatio = [1.64651159110457 1 1];
app.UIAxes2.XLim = [0.6 1.24];
app.UIAxes2.XGrid = 'on';
app.UIAxes2.YGrid = 'on';
app.UIAxes2.ZGrid = 'on';
app.UIAxes2.Position = [1 10 783 529];

% Create Temp3DVsfERandEGRTab
app.Temp3DVsfERandEGRTab = uitab(app.TabGroup);
app.Temp3DVsfERandEGRTab.Title = 'Temp 3D Vs FER and EGR';

% Create UIAxes3
app.UIAxes3 = uiaxes(app.Temp3DVsfERandEGRTab);
title(app.UIAxes3, 'Flame Temp Before Pressure against EGR and
FER')

```

```

xlabel(app.UIAxes3, 'Fuel Equivalence Ratio')
ylabel(app.UIAxes3, 'EGR')
app.UIAxes3.PlotBoxAspectRatio = [1.6173120367432 1 1];
app.UIAxes3.XLim = [0.6 1.24];
app.UIAxes3.XGrid = 'on';
app.UIAxes3.YGrid = 'on';
app.UIAxes3.ZGrid = 'on';
app.UIAxes3.Position = [1 1 795 538];

% Create TemperaturePerDegree3DTab
app.TemperaturePerDegree3DTab = uitab(app.TabGroup);
app.TemperaturePerDegree3DTab.Title = 'Temperature Per Degree 3D';

% Create UIAxes14
app.UIAxes14 = uiaxes(app.TemperaturePerDegree3DTab);
title(app.UIAxes14, 'Flame Temp Rising with Pressure without EGR
against FER and Pressure')
xlabel(app.UIAxes14, 'FER')
ylabel(app.UIAxes14, 'Crank Angle (Degrees)')
app.UIAxes14.Position = [1 1 795 538];

% Create ComparedNOxModelsTab
app.ComparedNOxModelsTab = uitab(app.TabGroup);
app.ComparedNOxModelsTab.Title = 'Compared NOx Models';

% Create UIAxes11
app.UIAxes11 = uiaxes(app.ComparedNOxModelsTab);
title(app.UIAxes11, 'Comparing NO_{x} Models against FER')
xlabel(app.UIAxes11, 'FER')
ylabel(app.UIAxes11, 'NO_{x} PPM')
app.UIAxes11.XGrid = 'on';
app.UIAxes11.YGrid = 'on';
app.UIAxes11.Position = [1 1 795 538];

% Create RPMNOxTab
app.RPMNOxTab = uitab(app.TabGroup);
app.RPMNOxTab.Title = 'RPM NOx';

% Create UIAxes4
app.UIAxes4 = uiaxes(app.RPMNOxTab);
title(app.UIAxes4, 'NO_{x} PPM Merker at 0.97 FER against RPM')
xlabel(app.UIAxes4, 'RPM')
ylabel(app.UIAxes4, 'NO_{x} PPM')
app.UIAxes4.PlotBoxAspectRatio = [1.62325581395349 1 1];
app.UIAxes4.XGrid = 'on';
app.UIAxes4.YGrid = 'on';
app.UIAxes4.Position = [8 1 788 531];

```

```

% Create PolytropicindexTab
app.PolytropicindexTab = uitab(app.TabGroup);
app.PolytropicindexTab.Title = 'Polytropic index';

% Create UIAxes5
app.UIAxes5 = uiaxes(app.PolytropicindexTab);
title(app.UIAxes5, 'NO_{x} PPM Merker at 0.97 FER changing with
polytropic index')
xlabel(app.UIAxes5, 'Polytropic index')
ylabel(app.UIAxes5, 'NO_{x} PPM')
app.UIAxes5.PlotBoxAspectRatio = [1.63023266919031 1 1];
app.UIAxes5.XLim = [1 1.35];
app.UIAxes5.LineWidth = 0.8;
app.UIAxes5.XGrid = 'on';
app.UIAxes5.YGrid = 'on';
app.UIAxes5.Position = [1 1 795 531];

% Create RTab
app.RTab = uitab(app.TabGroup);
app.RTab.Title = 'R';

% Create UIAxes6
app.UIAxes6 = uiaxes(app.RTab);
title(app.UIAxes6, 'No_{x} PPM Merker at 0.97 FER against Gas
Constant')
xlabel(app.UIAxes6, 'Change Gas Constrant')
ylabel(app.UIAxes6, 'NO_{x} PPM')
app.UIAxes6.PlotBoxAspectRatio = [1.62453531598513 1 1];
app.UIAxes6.XLim = [7500 9500];
app.UIAxes6.XGrid = 'on';
app.UIAxes6.YGrid = 'on';
app.UIAxes6.Position = [1 1 795 531];

% Create CRTab
app.CRTab = uitab(app.TabGroup);
app.CRTab.Title = 'CR';

% Create UIAxes7
app.UIAxes7 = uiaxes(app.CRTab);
title(app.UIAxes7, 'NO_{x} PPM Merker at 0.97 FER against
Compression Ratio')
xlabel(app.UIAxes7, 'Compression Ratio')
ylabel(app.UIAxes7, 'NO_{x} PPM')
app.UIAxes7.PlotBoxAspectRatio = [1.61538461538462 1 1];
app.UIAxes7.XLim = [8 30];
app.UIAxes7.XGrid = 'on';
app.UIAxes7.YGrid = 'on';
app.UIAxes7.Position = [1 1 795 537];

```

```

% Create EfficiencyTab
app.EfficiencyTab = uitab(app.TabGroup);
app.EfficiencyTab.Title = 'Efficiency';

% Create UIAxes13
app.UIAxes13 = uiaxes(app.EfficiencyTab);
title(app.UIAxes13, 'NO_{x} PPM Merker against Volumetric
Efficiency Vary')
xlabel(app.UIAxes13, 'Volumetric Efficiency \mu (%) ')
ylabel(app.UIAxes13, 'NO_{x} PPM')
app.UIAxes13.XGrid = 'on';
app.UIAxes13.YGrid = 'on';
app.UIAxes13.Position = [1 1 795 538];

% Create FuelsComparedTab
app.FuelsComparedTab = uitab(app.TabGroup);
app.FuelsComparedTab.Title = 'Fuels Compared';

% Create UIAxes12
app.UIAxes12 = uiaxes(app.FuelsComparedTab);
title(app.UIAxes12, 'Fuels Compared')
xlabel(app.UIAxes12, 'Fuels')
ylabel(app.UIAxes12, 'See Legend')
app.UIAxes12.XTick = [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
19 20 21];
app.UIAxes12.XTickLabelRotation = 90;
app.UIAxes12.XTickLabel = {'Methane'; 'Acetylene'; 'Ethylene';
'Ethane'; 'Propane'; 'n-Butane'; 'iso-Butane'; 'n-Pentane'; 'iso-Pentane'; 'n-
Hexane'; 'iso-Hexane'; 'n-Heptane'; 'n-Octane'; 'iso-Octane'; 'n-Nonane'; 'n-
Decane'; 'iso-Decane'; 'n-Dodecane'; 'Methanol'; 'Ethanol'; 'Hydrogen'};
app.UIAxes12.Position = [1 1 795 537];

% Create EffectofEGRTab
app.EffectofEGRTab = uitab(app.TabGroup);
app.EffectofEGRTab.Title = 'Effect of EGR';

% Create UIAxes15
app.UIAxes15 = uiaxes(app.EffectofEGRTab);
title(app.UIAxes15, 'NO_{x} Reduction percent per 5% EGR')
xlabel(app.UIAxes15, 'EGR percent')
ylabel(app.UIAxes15, 'Percentage of reduction')
app.UIAxes15.Position = [5 1 791 531];

% Create RPMEditFieldLabel
app.RPMEditFieldLabel = uilabel(app.Ryan_James_Powertrain_app);
app.RPMEditFieldLabel.HorizontalAlignment = 'right';
app.RPMEditFieldLabel.Position = [156 539 32 22];
app.RPMEditFieldLabel.Text = 'RPM';

```

```

% Create RPMEditField
app.RPMEditField = uieditfield(app.Ryan_James_Powertrain_app,
'numeric');
app.RPMEditField.BackgroundColor = [0.9294 0.6941 0.1255];
app.RPMEditField.Position = [203 539 100 22];
app.RPMEditField.Value = 1800;

% Create VariablesLabel
app.VariablesLabel = uilabel(app.Ryan_James_Powertrain_app);
app.VariablesLabel.FontSize = 16;
app.VariablesLabel.Tooltip = {'Edit these then click model to see
how they effect nox etc.'};
app.VariablesLabel.Position = [139 679 71 22];
app.VariablesLabel.Text = 'Variables';

% Create PolytropicIndexEditFieldLabel
app.PolytropicIndexEditFieldLabel =
uilabel(app.Ryan_James_Powertrain_app);
app.PolytropicIndexEditFieldLabel.HorizontalAlignment = 'right';
app.PolytropicIndexEditFieldLabel.Position = [97 561 91 22];
app.PolytropicIndexEditFieldLabel.Text = 'Polytropic Index';

% Create PolytropicIndexEditField
app.PolytropicIndexEditField =
uieditfield(app.Ryan_James_Powertrain_app, 'numeric');
app.PolytropicIndexEditField.BackgroundColor = [0.9294 0.6941
0.1255];
app.PolytropicIndexEditField.Position = [203 561 100 22];
app.PolytropicIndexEditField.Value = 1.17;

% Create CCoFylinderEditFieldLabel
app.CCoFylinderEditFieldLabel =
uilabel(app.Ryan_James_Powertrain_app);
app.CCoFylinderEditFieldLabel.HorizontalAlignment = 'right';
app.CCoFylinderEditFieldLabel.Position = [105 517 83 22];
app.CCoFylinderEditFieldLabel.Text = 'CC of Cylinder';

% Create CCoFylinderEditField
app.CCoFylinderEditField =
uieditfield(app.Ryan_James_Powertrain_app, 'numeric');
app.CCoFylinderEditField.BackgroundColor = [0.9294 0.6941
0.1255];
app.CCoFylinderEditField.Position = [203 517 100 22];
app.CCoFylinderEditField.Value = 400;

% Create CompressionRatioto1EditFieldLabel

```

```

        app.CompressionRatioto1EditFieldLabel =
uilabel(app.Ryan_James_Powertrain_app);
        app.CompressionRatioto1EditFieldLabel.HorizontalAlignment =
'right';
        app.CompressionRatioto1EditFieldLabel.Position = [57 495 131 22];
        app.CompressionRatioto1EditFieldLabel.Text = 'Compression Ratio to
1';

        % Create CompressionRatioto1EditField
        app.CompressionRatioto1EditField =
uieditfield(app.Ryan_James_Powertrain_app, 'numeric');
        app.CompressionRatioto1EditField.BackgroundColor = [0.9294 0.6941
0.1255];
        app.CompressionRatioto1EditField.Position = [203 495 100 22];
        app.CompressionRatioto1EditField.Value = 10;

        % Create PressureriseperdeegreebarEditFieldLabel
        app.PressureriseperdeegreebarEditFieldLabel =
uilabel(app.Ryan_James_Powertrain_app);
        app.PressureriseperdeegreebarEditFieldLabel.HorizontalAlignment =
'right';
        app.PressureriseperdeegreebarEditFieldLabel.Position = [22 450 166
22];
        app.PressureriseperdeegreebarEditFieldLabel.Text = 'Pressure rise
per degree (bar)';

        % Create PressureriseperdeegreebarEditField
        app.PressureriseperdeegreebarEditField =
uieditfield(app.Ryan_James_Powertrain_app, 'numeric');
        app.PressureriseperdeegreebarEditField.BackgroundColor = [0.9294
0.6941 0.1255];
        app.PressureriseperdeegreebarEditField.Position = [203 451 100 22];
        app.PressureriseperdeegreebarEditField.Value = 2;

        % Create PressureinletCloseBarEditFieldLabel
        app.PressureinletCloseBarEditFieldLabel =
uilabel(app.Ryan_James_Powertrain_app);
        app.PressureinletCloseBarEditFieldLabel.HorizontalAlignment =
'right';
        app.PressureinletCloseBarEditFieldLabel.Position = [42 407 146
22];
        app.PressureinletCloseBarEditFieldLabel.Text = 'Pressure inlet
Close (Bar)';

        % Create PressureinletCloseBarEditField
        app.PressureinletCloseBarEditField =
uieditfield(app.Ryan_James_Powertrain_app, 'numeric');
        app.PressureinletCloseBarEditField.BackgroundColor = [0.9294
0.6941 0.1255];

```

```

app.PressureinletCloseBarEditField.Position = [203 407 100 22];
app.PressureinletCloseBarEditField.Value = 1;

% Create RjKMolKEditFieldLabel
app.RjKMolKEditFieldLabel =
uilabel(app.Ryan_James_Powertrain_app);
app.RjKMolKEditFieldLabel.HorizontalAlignment = 'right';
app.RjKMolKEditFieldLabel.Position = [118 385 70 22];
app.RjKMolKEditFieldLabel.Text = 'R (j/KMol/K)';

% Create RjKMolKEditField
app.RjKMolKEditField = ueditfield(app.Ryan_James_Powertrain_app,
'numeric');
app.RjKMolKEditField.BackgroundColor = [0.9294 0.6941 0.1255];
app.RjKMolKEditField.Position = [203 385 100 22];
app.RjKMolKEditField.Value = 8314;

% Create COLabel
app.COLabel = uilabel(app.Ryan_James_Powertrain_app);
app.COLabel.HorizontalAlignment = 'right';
app.COLabel.Position = [1007 7 31 22];
app.COLabel.Text = 'CO+';

% Create nH2EditField_2
app.nH2EditField_2 = ueditfield(app.Ryan_James_Powertrain_app,
'numeric');
app.nH2EditField_2.Editable = 'off';
app.nH2EditField_2.Position = [1049 7 35 22];

% Create H2Label
app.H2Label = uilabel(app.Ryan_James_Powertrain_app);
app.H2Label.Position = [1092 7 25 22];
app.H2Label.Text = 'H2';

% Create TempInletClosekEditFieldLabel
app.TempInletClosekEditFieldLabel =
uilabel(app.Ryan_James_Powertrain_app);
app.TempInletClosekEditFieldLabel.HorizontalAlignment = 'right';
app.TempInletClosekEditFieldLabel.Position = [76 429 112 22];
app.TempInletClosekEditFieldLabel.Text = 'Temp Inlet Close (k)';

% Create TempInletClosekEditField
app.TempInletClosekEditField =
ueditfield(app.Ryan_James_Powertrain_app, 'numeric');
app.TempInletClosekEditField.BackgroundColor = [0.9294 0.6941
0.1255];
app.TempInletClosekEditField.Position = [203 429 100 22];

```

```
app.TempInletClosekEditField.Value = 293;

% Create PressurestartofCombustionBarEditFieldLabel
app.PressurestartofCombustionBarEditFieldLabel =
uilabel(app.Ryan_James_Powertrain_app);
app.PressurestartofCombustionBarEditFieldLabel.HorizontalAlignment
= 'right';
app.PressurestartofCombustionBarEditFieldLabel.Position = [-1 472
192 22];
app.PressurestartofCombustionBarEditFieldLabel.Text = 'Pressure
start of Combustion (Bar)';

% Create PressurestartofCombustionBarEditField
app.PressurestartofCombustionBarEditField =
uieditfield(app.Ryan_James_Powertrain_app, 'numeric');
app.PressurestartofCombustionBarEditField.BackgroundColor =
[0.9294 0.6941 0.1255];
app.PressurestartofCombustionBarEditField.Position = [203 473 100
22];
app.PressurestartofCombustionBarEditField.Value = 10;

% Create MassPercentFuel1EditFieldLabel
app.MassPercentFuel1EditFieldLabel =
uilabel(app.Ryan_James_Powertrain_app);
app.MassPercentFuel1EditFieldLabel.HorizontalAlignment = 'right';
app.MassPercentFuel1EditFieldLabel.Position = [75 305 115 22];
app.MassPercentFuel1EditFieldLabel.Text = 'Mass Percent Fuel 1';

% Create MassPercentFuel1EditField
app.MassPercentFuel1EditField =
uieditfield(app.Ryan_James_Powertrain_app, 'numeric');
app.MassPercentFuel1EditField.Editable = 'off';
app.MassPercentFuel1EditField.Position = [205 305 100 22];

% Create AFRafterFEREditFieldLabel
app.AFRafterFEREditFieldLabel =
uilabel(app.Ryan_James_Powertrain_app);
app.AFRafterFEREditFieldLabel.HorizontalAlignment = 'right';
app.AFRafterFEREditFieldLabel.Position = [106 263 84 22];
app.AFRafterFEREditFieldLabel.Text = 'AFR after FER';

% Create AFRafterFEREditField
app.AFRafterFEREditField =
uieditfield(app.Ryan_James_Powertrain_app, 'numeric');
app.AFRafterFEREditField.Editable = 'off';
app.AFRafterFEREditField.Position = [205 263 100 22];
```

```
% Create AFRStoichEditFieldLabel
app.AFRStoichEditFieldLabel =
uicontrol(app.Ryan_James_Powertrain_app);
app.AFRStoichEditFieldLabel.HorizontalAlignment = 'right';
app.AFRStoichEditFieldLabel.Position = [124 284 66 22];
app.AFRStoichEditFieldLabel.Text = 'AFR Stoich';

% Create AFRStoichEditField
app.AFRStoichEditField =
uicontrol(app.Ryan_James_Powertrain_app, 'numeric');
app.AFRStoichEditField.Eitable = 'off';
app.AFRStoichEditField.Position = [205 284 100 22];

% Create EnthalpyLHVMJkgEditFieldLabel
app.EnthalpyLHVMJkgEditFieldLabel =
uicontrol(app.Ryan_James_Powertrain_app);
app.EnthalpyLHVMJkgEditFieldLabel.HorizontalAlignment = 'right';
app.EnthalpyLHVMJkgEditFieldLabel.Position = [74 242 116 22];
app.EnthalpyLHVMJkgEditFieldLabel.Text = 'Enthalpy LHV MJ/Kg';

% Create EnthalpyLHVMJkgEditField
app.EnthalpyLHVMJkgEditField =
uicontrol(app.Ryan_James_Powertrain_app, 'numeric');
app.EnthalpyLHVMJkgEditField.Eitable = 'off';
app.EnthalpyLHVMJkgEditField.Position = [205 242 100 22];

% Create EnthalpyHHVMJkgEditFieldLabel
app.EnthalpyHHVMJkgEditFieldLabel =
uicontrol(app.Ryan_James_Powertrain_app);
app.EnthalpyHHVMJkgEditFieldLabel.HorizontalAlignment = 'right';
app.EnthalpyHHVMJkgEditFieldLabel.Position = [73 221 118 22];
app.EnthalpyHHVMJkgEditFieldLabel.Text = 'Enthalpy HHV MJ/Kg';

% Create EnthalpyHHVMJkgEditField
app.EnthalpyHHVMJkgEditField =
uicontrol(app.Ryan_James_Powertrain_app, 'numeric');
app.EnthalpyHHVMJkgEditField.Eitable = 'off';
app.EnthalpyHHVMJkgEditField.Position = [205 221 100 22];

% Create CO2kgFuelkgEditFieldLabel
app.CO2kgFuelkgEditFieldLabel =
uicontrol(app.Ryan_James_Powertrain_app);
app.CO2kgFuelkgEditFieldLabel.HorizontalAlignment = 'right';
app.CO2kgFuelkgEditFieldLabel.Position = [88 199 96 22];
app.CO2kgFuelkgEditFieldLabel.Text = 'CO2 kg / Fuel kg';

% Create CO2kgFuelkgEditField
```

```

        app.CO2kgFuelkgEditField =
uieditfield(app.Ryan_James_Powertrain_app, 'numeric');
        app.CO2kgFuelkgEditField.Editable = 'off';
        app.CO2kgFuelkgEditField.Position = [205 200 100 22];

        % Create FlameTempPressureKEditFieldLabel
        app.FlameTempPressureKEditFieldLabel =
uilabel(app.Ryan_James_Powertrain_app);
        app.FlameTempPressureKEditFieldLabel.HorizontalAlignment =
'right';
        app.FlameTempPressureKEditFieldLabel.Position = [38 95 152 22];
        app.FlameTempPressureKEditFieldLabel.Text = 'Flame Temp + Pressure
(K)';

        % Create FlameTempPressureKEditField
        app.FlameTempPressureKEditField =
uieditfield(app.Ryan_James_Powertrain_app, 'numeric');
        app.FlameTempPressureKEditField.Editable = 'off';
        app.FlameTempPressureKEditField.Position = [205 95 100 22];

        % Create NOxPPMmcAllisterEditFieldLabel
        app.NOxPPMmcAllisterEditFieldLabel =
uilabel(app.Ryan_James_Powertrain_app);
        app.NOxPPMmcAllisterEditFieldLabel.HorizontalAlignment = 'right';
        app.NOxPPMmcAllisterEditFieldLabel.Position = [76 53 114 22];
        app.NOxPPMmcAllisterEditFieldLabel.Text = 'NOx PPM McAllister';

        % Create NOxPPMmcAllisterEditField
        app.NOxPPMmcAllisterEditField =
uieditfield(app.Ryan_James_Powertrain_app, 'numeric');
        app.NOxPPMmcAllisterEditField.Editable = 'off';
        app.NOxPPMmcAllisterEditField.Position = [205 53 100 22];

        % Create NOxPPMHeywoodEditFieldLabel
        app.NOxPPMHeywoodEditFieldLabel =
uilabel(app.Ryan_James_Powertrain_app);
        app.NOxPPMHeywoodEditFieldLabel.HorizontalAlignment = 'right';
        app.NOxPPMHeywoodEditFieldLabel.Position = [78 32 112 22];
        app.NOxPPMHeywoodEditFieldLabel.Text = 'NOx PPM Heywood';

        % Create NOxPPMHeywoodEditField
        app.NOxPPMHeywoodEditField =
uieditfield(app.Ryan_James_Powertrain_app, 'numeric');
        app.NOxPPMHeywoodEditField.Editable = 'off';
        app.NOxPPMHeywoodEditField.Position = [205 32 100 22];

        % Create N2KMolsm3EditFieldLabel

```

```

        app.N2KMolsm3EditFieldLabel =
uicontrol(app.Ryan_James_Powertrain_app);
        app.N2KMolsm3EditFieldLabel.HorizontalAlignment = 'right';
        app.N2KMolsm3EditFieldLabel.Position = [100 179 90 22];
        app.N2KMolsm3EditFieldLabel.Text = '[N2] KMols/m^3';

        % Create N2KMolsm3EditField
        app.N2KMolsm3EditField =
uicontrol(app.Ryan_James_Powertrain_app, 'numeric');
        app.N2KMolsm3EditField.Eitable = 'off';
        app.N2KMolsm3EditField.Position = [205 179 100 22];

        % Create O2KMolsm3EditFieldLabel
        app.O2KMolsm3EditFieldLabel =
uicontrol(app.Ryan_James_Powertrain_app);
        app.O2KMolsm3EditFieldLabel.HorizontalAlignment = 'right';
        app.O2KMolsm3EditFieldLabel.Position = [99 158 91 22];
        app.O2KMolsm3EditFieldLabel.Text = '[O2] KMols/m^3';

        % Create O2KMolsm3EditField
        app.O2KMolsm3EditField =
uicontrol(app.Ryan_James_Powertrain_app, 'numeric');
        app.O2KMolsm3EditField.Eitable = 'off';
        app.O2KMolsm3EditField.Position = [205 158 100 22];

        % Create VolumetricEfficiencyEditFieldLabel
        app.VolumetricEfficiencyEditFieldLabel =
uicontrol(app.Ryan_James_Powertrain_app);
        app.VolumetricEfficiencyEditFieldLabel.HorizontalAlignment =
'right';
        app.VolumetricEfficiencyEditFieldLabel.Position = [53 365 135 22];
        app.VolumetricEfficiencyEditFieldLabel.Text = 'Volumetric Efficiency
(%)';

        % Create VolumetricEfficiencyEditField
        app.VolumetricEfficiencyEditField =
uicontrol(app.Ryan_James_Powertrain_app, 'numeric');
        app.VolumetricEfficiencyEditField.BackgroundColor = [0.9294 0.6941
0.1255];
        app.VolumetricEfficiencyEditField.Position = [203 365 100 22];
        app.VolumetricEfficiencyEditField.Value = 40;

        % Create FlameTempKEditFieldLabel
        app.FlameTempKEditFieldLabel =
uicontrol(app.Ryan_James_Powertrain_app);
        app.FlameTempKEditFieldLabel.HorizontalAlignment = 'right';
        app.FlameTempKEditFieldLabel.Position = [99 116 91 22];
        app.FlameTempKEditFieldLabel.Text = 'Flame Temp (K)';

```

```
% Create FlameTempKEditField
app.FlameTempKEditField =
uieditfield(app.Ryan_James_Powertrain_app, 'numeric');
app.FlameTempKEditField.Editable = 'off';
app.FlameTempKEditField.Position = [205 116 100 22];

% Create COEditFieldLabel
app.COEditFieldLabel = uilabel(app.Ryan_James_Powertrain_app);
app.COEditFieldLabel.HorizontalAlignment = 'right';
app.COEditFieldLabel.Position = [156 137 34 22];
app.COEditFieldLabel.Text = 'CO%';

% Create COEditField
app.COEditField = uieditfield(app.Ryan_James_Powertrain_app,
'numeric');
app.COEditField.Editable = 'off';
app.COEditField.Position = [205 137 100 22];

% Create NOxPPMMerkerEditFieldLabel
app.NOxPPMMerkerEditFieldLabel =
uilabel(app.Ryan_James_Powertrain_app);
app.NOxPPMMerkerEditFieldLabel.HorizontalAlignment = 'right';
app.NOxPPMMerkerEditFieldLabel.Position = [91 74 99 22];
app.NOxPPMMerkerEditFieldLabel.Text = 'NOx PPM Merker';

% Create NOxPPMMerkerEditField
app.NOxPPMMerkerEditField =
uieditfield(app.Ryan_James_Powertrain_app, 'numeric');
app.NOxPPMMerkerEditField.Editable = 'off';
app.NOxPPMMerkerEditField.Position = [205 74 100 22];

% Create KeyEditFieldLabel
app.KeyEditFieldLabel = uilabel(app.Ryan_James_Powertrain_app);
app.KeyEditFieldLabel.HorizontalAlignment = 'right';
app.KeyEditFieldLabel.Position = [319 42 36 22];
app.KeyEditFieldLabel.Text = 'Key =';

% Create KeyEditField
app.KeyEditField = uieditfield(app.Ryan_James_Powertrain_app,
'text');
app.KeyEditField.Editable = 'off';
app.KeyEditField.BackgroundColor = [0.9294 0.6941 0.1255];
app.KeyEditField.Position = [365 42 65 22];
app.KeyEditField.Value = 'Adustable';
```

```

% Create EditField_6
app.EditField_6 = uieditfield(app.Ryan_James_Powertrain_app,
'text');
app.EditField_6.Editable = 'off';
app.EditField_6.BackgroundColor = [1 0 0];
app.EditField_6.Position = [633 42 80 22];
app.EditField_6.Value = 'Rich no EGR';

% Create EditField_7
app.EditField_7 = uieditfield(app.Ryan_James_Powertrain_app,
'text');
app.EditField_7.Editable = 'off';
app.EditField_7.FontColor = [1 1 1];
app.EditField_7.BackgroundColor = [0.7412 0.0784 0.1804];
app.EditField_7.Position = [722 42 73 22];
app.EditField_7.Value = 'Rich + EGR';

% Create EditField2
app.EditField2 = uieditfield(app.Ryan_James_Powertrain_app,
'text');
app.EditField2.Editable = 'off';
app.EditField2.BackgroundColor = [0 1 1];
app.EditField2.Position = [444 42 91 22];
app.EditField2.Value = 'Stoich no EGR ';

% Create EditField2_2
app.EditField2_2 = uieditfield(app.Ryan_James_Powertrain_app,
'text');
app.EditField2_2.Editable = 'off';
app.EditField2_2.FontColor = [1 1 1];
app.EditField2_2.BackgroundColor = [0 0 1];
app.EditField2_2.Position = [543 42 82 22];
app.EditField2_2.Value = 'Stoich + EGR';

% Create EditField_8
app.EditField_8 = uieditfield(app.Ryan_James_Powertrain_app,
'text');
app.EditField_8.Editable = 'off';
app.EditField_8.BackgroundColor = [0 1 0];
app.EditField_8.Position = [806 42 86 22];
app.EditField_8.Value = 'Lean no EGR';

% Create EditField2_3
app.EditField2_3 = uieditfield(app.Ryan_James_Powertrain_app,
'text');
app.EditField2_3.Editable = 'off';
app.EditField2_3.BackgroundColor = [0.4667 0.6745 0.1882];
app.EditField2_3.Position = [904 42 83 22];
app.EditField2_3.Value = 'Lean + EGR';

```

```
% Create EditField_9
app.EditField_9 = uicontrolfield(app.Ryan_James_Powertrain_app,
'text');
app.EditField_9.Editable = 'off';
app.EditField_9.BackgroundColor = [0.851 0.3294 0.102];
app.EditField_9.Position = [999 42 126 22];
app.EditField_9.Value = '% Fuel Mix Adjusted';

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

% App creation and deletion
methods (Access = public)

% Construct app
function app = PowertrainApp_Ryan

% Create UIFigure and components
createComponents(app)

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

if nargin == 0
    clear app
end
end

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

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

