```
1
2 % Clear the console
3 clc;
4 % Close all the windows
5 close all;
6 % Clear Workspace Variables
7 clear;
8
9 global S1_MIN S1_MAX S1_POINTS;
10 global S2_MIN S2_MAX S2_POINTS;
11 global INVALID_FLOWRATE;
12 global Fethyl_S1S2_plotOpt;
13 global MT_PER_KT G_PER_KT GJ_PER_KJ;
14 global VALUE_ETHANE VALUE_ETHYLENE VALUE_HYDROGEN_CHEM;
15 global COST_RATES_STEAM;
16 global VALUE_HYDROGEN_FUEL VALUE_METHANE_FUEL VALUE_PROPANE_FUEL VALUE_BUTANE_FUEL;
17 global VALUE_NATGAS_FUEL VALUE_NUM2OIL_FUEL;
18 global ENTHALPY_PROPANE ENTHALPY_BUTANE;
19 global MOLMASS_PROPANE MOLMASS_BUTANE;
20 global PROFIT_S1S2_OPT;
21 global HEAT_CAPACITY_ETHANE;
22 global HEAT_FORMATION_ETHANE;
23 global STEAM_30PSIA STEAM_50PSIA STEAM_100PSIA STEAM_200PSIA STEAM_500PSIA STEAM_750PSIA;
24 global HYDROGEN METHANE ETHYLENE PROPANE BUTANE;
25 global ENTHALPY_METHANE ENTHALPY_PROPANE ENTHALPY_BUTANE HEAT_CAPACITY_ETHANE;
26 global KT_PER_G KG_PER_KT KJ_PER_GJ MT_PER_G ENTHALPY_NAT_GAS MOLMASS_ETHANE...
27     MOLMASS_ETHYLENE MOLMASS_NATGAS;
28 global MT_CO2_PER_KT_METHANE MT_CO2_PER_KT_PROPANE MT_CO2_PER_KT_BUTANE ...
29     MT_CO2_PER_KT_NATURALGAS;
30 global TAX_CO2_PER_MT;
31 global STEAM_PRESSURE_COL STEAM_TEMP_COL;
32 global MOLMASS_METHANE MOLMASS_WATER BAR_PER_PSIA;
33 global C_TO_K HEAT_CAPACITY_WATER;
34 global R k1_f k1_r k2 k3 R_2 C_TO_K YR_PER_SEC SEC_PER_YR MOLMASS_HYDROGEN
35 global PSA_TOGGLE ENTHALPY_HYDROGEN T_SEPARATION P_SEPARATION M3_PER_L DENSITY_LIQ_WATER
36 global MAX_CAPEX MAX_OPEX MAX_TFCI PRESS_RXTR
37
38 % USER NOTES_____________________________________________________________
39
40 % Note: The primary (high level) units of this script are ...
41 % Mass         kta
42 % Energy       GJ
43 % Pressure     Bar
44 % Temperature  Celcius
45 % Moles        Moles
46 % Value        Dollars
47
48 % [ __ ] THIS MEANS DIMENSIONLESS UNITS
49
50 % USER INPUTS | DESIGN PARAMETERS_____________________________________
51
52 % Product
53 P_ETHYLENE_DES = 200;          % [ kta ]
```

```
54    % Note! This design parameter's units are changed prior to the matrix def
55
56 YEARS_IN_OPERATION = 10 ;
57
58 % USER INPUTS | GLOBAL CONSTANTS_____________________________________
59
60 % USER INPUTS | 3D PLOT, CONTOUR, LVL 2 & 3 CALCS_____________________
61
62 % Reactor Conditions | 3D PLOT & CONTOUR PLOT (S1 S2) && THE LVL3 CALCS
63 STEAM_TO_FEED_RATIO_MOLS = 0.6;    % [ __ ] 0.6 to 1.0
64 TEMP_RXTR = 825;              % [ C ]
65 PRESS_RXTR = 2;               % [ Bar ] 2 to 5 bar
66 TEMP_ETHANE_FEED = 25;        % [ C ]
67 CONVERSION =  0.17053;        % [ __ ] % Level 2 & 3 Calculations
68 USERINPUT_S1 = 0.96971 ;      % [ __ ] % Level 2 & 3 Calculations
69 USERINPUT_S2 = 0.00011843;    % [ __ ] % Level 2 & 3 Calculations
70 STEAM_CHOICE = 1;
71 %   STEAM_30PSIA = 1;
72 %   STEAM_50PSIA = 2;
73 %   STEAM_100PSIA = 3;
74 %   STEAM_200PSIA = 4;
75 %   STEAM_500PSIA = 5;
76 %   STEAM_750PSIA = 6;
77       % % Steam
78       % % [ psia Temp[C] $/MT  kJ/kg ]
79       % COST_RATES_STEAM = [
80       %    30  121      2.38  2213;
81       %    50  138      3.17  2159;
82       %    100 165      4.25  2067;
83       %    200 194      5.32  1960;
84       %    500 242      6.74  1755;
85       %    750 266      7.37  1634
86       % ];
87
88 % Plotting | 3D PLOT & CONTOUR PLOT (S1 S2)
89 NUM_POINTS = 10^4;
90
91 % USER INPUTS | RXTR TABLE PARAMETERS_____________________________________
92
93 % Reactor Script Parameters | RXTR TABLE OUTPUT
94 V_MIN = 0.1;                % [ L ]
95 V_MAX = 10^4;               % [ L ]
96 NUM_V_POINTS = 20;          % [ __ ]
97
98 P_MIN = 2;                  % [ Bar ]
99 P_MAX = 5;                  % [ Bar ]
100 NUM_P_POINTS = 2;          % [ __ ]
101
102 T_MIN = 775;               % [ Celcius ]
103 T_MAX = 825;               % [ Celcius ]
104 NUM_T_POINTS = 2;          % [ __ ]
105
106 STEAM_MIN = 0.6;           % [ __ ]
```

```
STEAM_MAX = 1.0;              % [ __ ]
NUM_STEAM_POINTS = 2;         % [ __ ]

% Table Overrides | RXTR TABLE OUTPUT
T_P_OVERRIDE = true;
   T_OVERRIDE = 825;          %[C]
   P_OVERRIDE = 2;            %[Bar]
   STEAM_MR_OVERRIDE = 0.6;%   [__]

% Output fuel costs
CONSOLE_OUTPUT_EFFECTIVE_VALUE_FUELS = true;

% Output the level 2 and 3 calculations
OUTPUT_LVL3_FLOWRATES_TO_CONSOLE = true;
   SANITY_CHECK_CALCULATIONS = true;

% Plot the 3D and Contour plot's
CALCULATE_ALL_SELECTIVITIES = true;
    PLOT_ECON_3D = true;
    PLOT_ECON_COUNTOUR = true;

% Output the Reactor Design tables
CALCULATE_REACTOR_FLOWS = true;

% PSA Toggle switch
PSA_TOGGLE = true;

% Do you want to add the work of the compressor to the heat flux of heating
% the steam from the temp it's avilable at, to the temp of the reactor?
ADD_COMPRESSOR_WORK_TO_STEAM_HEATFLUX = true;

% Separation System Thermodynamics
T_SEPARATION = 173.15;        % [ K ]
P_SEPARATION = PRESS_RXTR;    % [ bar ]
MAX_OPEX = false;     % [ __ ]
MAX_TFCI = false;
MAX_CAPEX = false;

% Zeolite and waste stream
% zeo 1.2 - 2.2 wt% absobtion = max of zeolite (g/g)

% NOTE SEARCH FOR "??" TO SEE MY ASSUMPTIONS AND OTHER NOTES IN THE CODE

% WORK OF THE COMPRESSOR HAS NOT BEEN IMPLEMENTED
% THE STEAM TO FEED RATIO LIKELY HAS UNIT ISSUES OF (g/g) vs (mol/mol)
%      I think I implemented both

%_____________________________________________________________________
% DON'T TOUCH ANYTHING BELOW THIS LINE
%_____________________________________________________________________


% CONSTANTS | PLOTTING_____________________________________________
```

```
160
161 CONSOLE_SECTION_DIVIDER = ...
162     "_____________________________________________________";
163 S1_MIN = 0.01;
164 S1_MAX = 1.00;
165 S1_POINTS = NUM_POINTS ^ (1/2) ;
166 S2_MIN = 0.01;
167 S2_MAX = 1.00;
168 S2_POINTS = NUM_POINTS ^ (1/2);
169 INVALID_FLOWRATE = 0;
170 Fethyl_S1S2_plotOpt = { ...
171     'S_1 Selectivity', ...
172     'S_2 Selectivity', ...
173     'Ethylene Flowrate [kta]',...
174     'P_ethylene_VS_S1_S2.jpg'};
175 PROFIT_S1S2_OPT = { ...
176     'S_1 Selectivity', ...
177     'S_2 Selectivity', ...
178     'Annual Profit [$ MM USD]',...
179     'P_ethylene_VS_S1_S2.jpg'};
180
181 % CONSTANTS | UNITS_____________________________________________
182
183 % Mass
184 MT_PER_KT = 10^3;       % [ MT / kt ]
185
186 G_PER_KT = 10^9;        % [ g / kt ]
187 KT_PER_G = 10^-9;       % [ kt / g ]
188
189 KG_PER_KT = 10^6;       % [ kg / MT ]
190
191 MT_PER_G = 10^-6;       % [ MT / g ]
192
193 % Energy
194 GJ_PER_KJ = 10^-6;      % [ GJ / kJ ]
195 KJ_PER_GJ = 10^6;       % [ kJ / GJ ]
196
197 % Temperature
198 C_TO_K = 273.15;        % [ C -> K ]
199 % Value
200 MMDOLLA_PER_DOLLA = 10^-6;  % [ $ MM / $]
201 DOLLA_PER_MMDOLLA = 10^6;   % [ $ / $ MM ]
202
203 % Pressure
204 BAR_PER_PSIA = 0.0689476;   % [ Bar / Psia ]
205
206 % Time
207 YR_PER_SEC = 1 / (3.154 * 10^7);    % [ yr / s ]
208 SEC_PER_YR = 3.154 * 10^7;          % [ s / yr ]
209
210 % Volumes
211 M3_PER_L = 0.001;
212
```

```
213 % CONSTANTS | PHYSICAL_____________________________________________
214
215 DENSITY_LIQ_WATER = 10^3;          % [ kg / m^3 ]
216
217 % CONSTANTS | CHEMICAL_____________________________________________
218
219 % Chemical | Molar Mass
220 MOLMASS_HYDROGEN = 2.01588;           % [ g / mol ]
221     % Source :  https://webbook.nist.gov/cgi/cbook.cgi?ID=1333-74-0
222 MOLMASS_METHANE = 16.0425;            % [ g / mol ]
223     % Source :  https://webbook.nist.gov/cgi/cbook.cgi?ID=74-82-8
224 MOLMASS_WATER = 18.015;               % [ g / mol ]
225     % source : https://pubchem.ncbi.nlm.nih.gov/compound/Water
226 MOLMASS_CO2 = 44.01;                  % [ g / mol ]
227     % Source : https://pubchem.ncbi.nlm.nih.gov/compound/Carbon-dioxide-water
228 MOLMASS_PROPANE = 44.0956;            % [ g / mol ]
229     % Source : https://webbook.nist.gov/cgi/cbook.cgi?ID=C74986&Mask=1
230 MOLMASS_BUTANE = 58.1222;             % [ g / mol ]
231     % Source : https://webbook.nist.gov/cgi/cbook.cgi?ID=C106978&Mask=1
232 MOLMASS_ETHANE = 30.0690;             % [ g / mol ]
233     % Source : https://webbook.nist.gov/cgi/cbook.cgi?ID=C74840
234 MOLMASS_ETHYLENE = 28.0532;           % [ g / mol ]
235     % Source = https://webbook.nist.gov/cgi/cbook.cgi?ID=74-85-1&Type=IR-SPEC&Index=QUANT-IR,20
236 MOLMASS_NATGAS = 16.04;               % [ g / mol ]
237     % ASSUMING NATURAL GAS IS ALL METHANE
238
239 % Chemical | Combustion Stochiometery
240 CO2_TO_METHANE_COMBUSTION_STOICH = 1;
241 CO2_TO_PROPANE_COMBUSTION_STOICH = 3;
242 CO2_TO_BUTANE_COMBUSTION_STOICH = 4;
243 C02_TO_NATGAS_COMBUSTION_STOICH = CO2_TO_METHANE_COMBUSTION_STOICH;
244     % Natural gas is asuumed to be entirely methane
245
246
247 % CONSTANTS | THERMODYNAMICS_____________________________________
248
249 % Gas Constant
250 R = 8.314;                          % [ J / mol K ]
251 R_2 = 0.0831446261815324;           % [ L bar / K mol ]
252
253 % Heat capacities
254 HEAT_CAPACITY_WATER = 33.79 * 10^-3;    % [ kJ / mol K ] Ref Temp = 298K
255     % Source : https://webbook.nist.gov/cgi/cbook.cgi?ID=C14940637&Mask=1&Type=JANAFG&Table=on
256 HEAT_CAPACITY_ETHANE = 52.71 * 10^-3;   % [ kJ / mol K ] Reference Temp = 300K
257     % Source : https://webbook.nist.gov/cgi/cbook.cgi?ID=C74840&Units=SI&Mask=1EFF
258
259 % Heats of Formation (at 25C)
260 HEAT_FORMATION_ETHANE = -83.8;        % [ kJ / mol ] reference Temp = std
261     % Source : https://webbook.nist.gov/cgi/cbook.cgi?ID=C74840&Units=SI&Mask=1EFF
262 HEAT_FORMATION_METHANE = -74.87;      % [ kJ / mol ] reference Temp = std
263     % Source : https://webbook.nist.gov/cgi/cbook.cgi?ID=C74828&Mask=1
264 HEAT_FORMATION_ETHYLENE = 52.47;      % [ kJ / mol ] reference Temp = std
265     % Source : https://webbook.nist.gov/cgi/cbook.cgi?ID=C74851&Mask=1
```

```
266 HEAT_FORMATION_HYDROGEN = 0;          % [ kJ / mol ] reference Temp = std
267 HEAT_FORMATION_PROPANE = -104.7;      % [ kJ / mol ] reference Temp = std
268     % Source : https://webbook.nist.gov/cgi/cbook.cgi?ID=C74986&Mask=1
269 HEAT_FORMATION_BUTANE = -125.6;       % [ kJ / mol ] reference Temp = std
270     % Source : https://webbook.nist.gov/cgi/cbook.cgi?ID=C106978&Mask=1
271
272 % Enthalpy of combustion (std conditions)
273 ENTHALPY_HYDROGEN = 286;
274     % Source : https://chem.libretexts.org/Courses/University_of_Kentucky/UK%3A_General_Chemistry/05%3A_Thermochemistry/5.3%3A_Enthalpy
275 ENTHALPY_METHANE = 890;               % [ kJ / mol ]
276     % Source : https://webbook.nist.gov/cgi/cbook.cgi?ID=C74828&Mask=1
277 ENTHALPY_PROPANE = 2219.2;            % [ kJ / mol ]
278     % Source : https://webbook.nist.gov/cgi/cbook.cgi?ID=C74986&Mask=1
279 ENTHALPY_BUTANE = 2877.5;             % [ kJ / mol ]
280     % Source : https://webbook.nist.gov/cgi/cbook.cgi?ID=C106978&Mask=1
281 ENTHALPY_NAT_GAS = ENTHALPY_METHANE;
282     % Source : https://afdc.energy.gov/fuels/natural_gas_basics.html#:~:text=Natural%20gas%20is%20an%20odorless,used%20in%20the%20United%20States.
283     % Natural gas is mostly methane, so assumed to be 100% methane in the calcs
284
285 % Enthalpy of Reactions [ kJ / extent rxn]
286 ENTHALPY_RXN_1 = HEAT_FORMATION_HYDROGEN + HEAT_FORMATION_ETHYLENE ...
287                                 - HEAT_FORMATION_ETHANE;
288 ENTHALPY_RXN_2 = HEAT_FORMATION_METHANE + HEAT_FORMATION_PROPANE ...
289                                 - 2 * HEAT_FORMATION_ETHANE;
290 ENTHALPY_RXN_3 = HEAT_FORMATION_ETHANE - HEAT_FORMATION_ETHANE ...
291                                 - HEAT_FORMATION_ETHYLENE;
292 % CONSTANTS | ECONOMICS_____________________________________________
293
294 %  Chemicals
295 VALUE_ETHANE = 200;             % [ $ / MT ]
296 VALUE_ETHYLENE = 900;           % [ $ / MT ]
297 VALUE_HYDROGEN_CHEM = 1400;     % [ $ / MT ]
298
299 % Steam
300 % [ psia Temp[C] $/MT  kJ/kg ]
301 COST_RATES_STEAM = [
302     30  121     2.38  2213;
303     50  138     3.17  2159;
304     100 165     4.25  2067;
305     200 194     5.32  1960;
306     500 242     6.74  1755;
307     750 266     7.37  1634
308 ];
309
310 % Accessing the Steam P,T Data
311     STEAM_PRESSURE_COL = 2;
312     STEAM_TEMP_COL = 1;
313     STEAM_COST_COL = 3;
314     STEAM_30PSIA = 1;
315     STEAM_50PSIA = 2;
316     STEAM_100PSIA = 3;
317     STEAM_200PSIA = 4;
318     STEAM_500PSIA = 5;
```

```
319    STEAM_750PSIA = 6;
320
321 % Economic | Fuel
322 VALUE_HYDROGEN_FUEL = 3;          % [ $ / GJ ]
323 VALUE_METHANE_FUEL = 3;          % [ $ / GJ ]
324 VALUE_PROPANE_FUEL = 3;          % [ $ / GJ ]
325 VALUE_BUTANE_FUEL = 3;           % [ $ / GJ ]
326 VALUE_NATGAS_FUEL = 3;           % [ $ / GJ ]
327 VALUE_NUM2OIL_FUEL = 4.5;        % [ $ / US Gallon ]
328
329 % Economics | Enviormental
330 TAX_CO2_PER_MT = 125;            % [ $ / MT ]
331
332 % [$ / GJ]  = 1GJ(basis) * (KJ / GJ)   * (mol gas / KJ) *         (mol CO2 / mol gas)        *  (g / mol C02)*(MT / g) * ($ / MT)
333 TAX_CO2_PER_GJ_METHANE = KJ_PER_GJ * (1 / ENTHALPY_METHANE) * CO2_TO_METHANE_COMBUSTION_STOICH * MOLMASS_CO2 * MT_PER_G * TAX_CO2_PER_MT;
334 TAX_CO2_PER_GJ_PROPANE = KJ_PER_GJ * (1 / ENTHALPY_PROPANE) * CO2_TO_PROPANE_COMBUSTION_STOICH * MOLMASS_CO2 * MT_PER_G * TAX_CO2_PER_MT;
335 TAX_CO2_PER_GJ_BUTANE = KJ_PER_GJ * (1 / ENTHALPY_BUTANE) * CO2_TO_BUTANE_COMBUSTION_STOICH * MOLMASS_CO2 * MT_PER_G * TAX_CO2_PER_MT;
336 TAX_CO2_PER_GJ_NATGAS = TAX_CO2_PER_GJ_METHANE; %
337
338 % Chemistry | MT of C02 per KT of Fuel used
339 % (MT CO2) = 1KT(basis) * (g / KT) * (mol gas/ g gas) *
340 MT_CO2_PER_KT_METHANE = G_PER_KT * (1/MOLMASS_METHANE) *...
341    ... % (mol CO2 / mol gas) *        (g CO2 / mol CO2) * (MT / g)
342    CO2_TO_METHANE_COMBUSTION_STOICH * MOLMASS_CO2 * MT_PER_G;
343 MT_CO2_PER_KT_PROPANE = G_PER_KT * (1/MOLMASS_PROPANE) *...
344    CO2_TO_PROPANE_COMBUSTION_STOICH * MOLMASS_CO2 * MT_PER_G;
345 MT_CO2_PER_KT_BUTANE = G_PER_KT * (1/MOLMASS_BUTANE) *...
346    CO2_TO_BUTANE_COMBUSTION_STOICH * MOLMASS_CO2 * MT_PER_G;
347 MT_CO2_PER_KT_NATURALGAS = MT_CO2_PER_KT_METHANE;
348
349 % FUNCTIONS | FLOWRATE_____________________________________________
350
351 P_ETHYLENE = P_ETHYLENE_DES;
352 P_ETHYLENE_DES = P_ETHYLENE_DES * (1 / MOLMASS_ETHYLENE);
353 P_PROPANE = @(s1, s2)      (s2 / s1 *P_ETHYLENE_DES) * ...
354                            MOLMASS_PROPANE;
355 P_BUTANE = @(s1, s2)       (P_ETHYLENE_DES*(1/(2*s1) - s2/s1 - 1/2)) * ...
356                            MOLMASS_BUTANE;
357 F_ETHANE = @(s1, s2)       (P_ETHYLENE_DES / s1) * ...
358                            MOLMASS_ETHANE;
359 P_METHANE = @(s1, s2) (s2 / s1 * P_ETHYLENE_DES) * ...
360                            MOLMASS_METHANE;
361 P_HYDROGEN = @(s1, s2) (P_ETHYLENE_DES * ((1/(2*s1) - s2/s1 + 1/2))) * ...
362                            MOLMASS_HYDROGEN;
363
364 % FUNCTIONS | EXTENT OF REACTION_____________________________________
365
366 % Returns molar flowrates [ mol / yr ]
367 get_xi = @(flowrates) [ flowrates(HYDROGEN) * G_PER_KT / MOLMASS_HYDROGEN, ...
368                 flowrates(PROPANE) * G_PER_KT / MOLMASS_PROPANE, ...
369                 flowrates(BUTANE) * G_PER_KT / MOLMASS_BUTANE ];
370
371 % FUNCTIONS | VALIDATION_____________________________________________
```

```
372
373 flowrates_valid = @( flowrates ) all(flowrates >= 0);
374
375 % FUNCTIONS | ECONOMICS_____________________________________________
376
377 % ($ / yr) =              (kta) *   (MT / KT) * ($ / MT)
378 value_ethane = @(P_ethane) P_ethane * MT_PER_KT * VALUE_ETHANE;
379 value_ethylene = @(P_ethylene) P_ethylene * MT_PER_KT * VALUE_ETHYLENE;
380 value_h2_chem = @(P_h2_chem) P_h2_chem * MT_PER_KT * VALUE_HYDROGEN_CHEM;
381 value_methane = @(P_methane) P_methane * MT_PER_KT * VALUE_METHANE_FUEL;
382 value_propane = @(P_propane) P_propane * MT_PER_KT * VALUE_PROPANE_FUEL;
383 value_butane = @(P_butane) P_butane * MT_PER_KT * VALUE_BUTANE_FUEL;
384
385 % ($ / yr) =                   (kta) *  (MT / kt) * ($ / MT)
386 cost_steam = @(F_steam, steam_rate) F_steam * MT_PER_KT * steam_rate;
387
388 % FUNCTIONS | THEROMODYNAMICS_____________________________________
389 % (GJ / yr) =                   (kta) *  (g / KT)   * (mol gas/ g gas)    * (kJ / mol K)         * (GJ / KJ) * (K)
390 heat_ethane = @(F_ethane, T0, Tf) F_ethane * G_PER_KT * (1 / MOLMASS_ETHANE) * HEAT_CAPACITY_ETHANE * GJ_PER_KJ * (Tf - T0);
391
392 % (GJ / yr)  =     (mol / yr) * (kJ / mol) *   (GJ / kJ)
393 heat_rxn1 = @(xi_1) xi_1 * ENTHALPY_RXN_1 * GJ_PER_KJ;
394 heat_rxn2 = @(xi_2) xi_2 * ENTHALPY_RXN_2 * GJ_PER_KJ;
395 heat_rxn3 = @(xi_3) xi_3 * ENTHALPY_RXN_3 * GJ_PER_KJ;
396 heat_rxn = @(xi) heat_rxn1(xi(1)) + heat_rxn2(xi(2)) + heat_rxn3(xi(3));
397
398 % FUNCTIONS | RATE CONTANTS_____________________________________
399
400 % T is [ Kelvin ]  R is [ J / mol K ]
401 k1_f = @(T) (4.652 * 10^13) * exp( (-273000 / (R * (T ))));
402 k1_r = @(T) (9.91 * 10^8) * exp( (-137800 / (R * (T ))));
403 k2 = @(T) (4.652 * 10^11) * exp( (-273000 / (R * (T ))));
404 k3 = @(T) (7.083 * 10^13) * exp( (-252600 / (R * (T ))));
405
406
407 % DESIGN PARAMS_____________________________________________
408 STEAM_TO_FEED_RATIO_MASS = (MOLMASS_WATER / MOLMASS_ETHANE) * STEAM_TO_FEED_RATIO_MOLS;
409
410
411 % SCRIPT_____________________________________________________
412
413 % Economics | Post-Tax Value of different fuel sources
414 if (CONSOLE_OUTPUT_EFFECTIVE_VALUE_FUELS)
415     disp(" [ $ / GJ ] ")
416     EFFECTIVE_VALUE_HYDROGEN_FUEL = VALUE_HYDROGEN_FUEL
417     EFFECTIVE_VALUE_METHANE_FUEL = VALUE_METHANE_FUEL + TAX_CO2_PER_GJ_METHANE
418     EFFECTIVE_VALUE_PROPANE_FUEL = VALUE_PROPANE_FUEL + TAX_CO2_PER_GJ_PROPANE
419     EFFECTIVE_VALUE_BUTANE_FUEL = VALUE_BUTANE_FUEL + TAX_CO2_PER_GJ_BUTANE
420     EFFECTIVE_VALUE_NAT_GAS_FUEL = VALUE_NATGAS_FUEL + TAX_CO2_PER_GJ_NATGAS
421 %   EFFECTIVE_VALUE_NUM2_FUEL = VALUE_NATGAS_FUEL + TAX_CO2_PER_GJ_NUM2;
422
423 end
424
```

```
425 if (OUTPUT_LVL3_FLOWRATES_TO_CONSOLE)
426
427  % Calculate the flow rates of each species (kta)
428  P_hydrogen = P_HYDROGEN(USERINPUT_S1, USERINPUT_S2);
429  P_methane = P_METHANE(USERINPUT_S1, USERINPUT_S2);
430  P_ethylene = P_ETHYLENE;
431  P_propane = P_PROPANE(USERINPUT_S1, USERINPUT_S2);
432  P_butane = P_BUTANE(USERINPUT_S1, USERINPUT_S2);
433  F_ethane = F_ETHANE(USERINPUT_S1, USERINPUT_S2);
434  P_flowrates = [ P_hydrogen, P_methane, P_ethylene, P_propane, P_butane ];
435
436  disp(CONSOLE_SECTION_DIVIDER)
437  if (flowrates_valid(P_flowrates))
438
439    fprintf("Flowrates for the reactor given that s1 = %f, s2 = %f conv = %f\n\n", ...
440        USERINPUT_S1, USERINPUT_S2, CONVERSION)
441
442    disp(CONSOLE_SECTION_DIVIDER)
443    disp("Level 2 Flowrates  in / out of the entire plant [ kt / yr ]")
444    P_hydrogen
445    P_methane
446    P_ethylene
447    P_propane
448    P_butane
449
450    disp("Fresh Feed Flowrate")
451    F_ethane
452
453    disp(CONSOLE_SECTION_DIVIDER)
454    disp("Level 3 Flowrates [ kt / yr ] ")
455
456    disp("Recycle Stream Flowrate")
457
458    R_ethane = F_ethane * ((1-CONVERSION) / (CONVERSION))
459    % R_ethane = (P_ethylene/USERINPUT_S1) * ((1-CONVERSION)/CONVERSION)
460
461    disp("Reactor Flowrates")
462
463    F_ethane_into_reactor = R_ethane + F_ethane
464
465    if SANITY_CHECK_CALCULATIONS
466      disp(CONSOLE_SECTION_DIVIDER)
467      disp("Sanity Checking the Calculations")
468      Conservation_of_mass = F_ethane - sum(P_flowrates)
469      if Conservation_of_mass
470        fprintf("WARNING : YOU ARE NOT CONSERVING MASS\n\n")
471      end
472    end
473  else
474    disp("ERROR : Selectivities S1 S2 chosen are not physically possible")
475  end
476 end
477
```

```
478 % SCRIPT | PLOTTING_____________________________________________________
479
480 if (CALCULATE_ALL_SELECTIVITIES)
481     disp(CONSOLE_SECTION_DIVIDER)
482     disp("Calculating all selectivities... ")
483     % Iterates through each value of selectivities S1 and S2 to find the economic
484     % potential for different reaction conditions
485     s1_domain = linspace(S1_MIN, S1_MAX, S1_POINTS);
486     s2_domain = linspace(S2_MIN, S2_MAX, S2_POINTS);
487     [s1_mesh, s2_mesh] = meshgrid(s1_domain, s2_domain);
488     % All flowrates are initialized as matricies of zeros
489     ethylene_flowrates = (s1_mesh + s2_mesh) .* 0;
490     hydrogen_flowrates = (s1_mesh + s2_mesh) .* 0;
491     methane_flowrates = (s1_mesh + s2_mesh) .* 0;
492     ethylene_flowrates = (s1_mesh + s2_mesh) .* 0;
493     propane_flowrates = (s1_mesh + s2_mesh) .* 0;
494     butane_flowrates = (s1_mesh + s2_mesh) .* 0;
495     ethane_flowrates = (s1_mesh + s2_mesh) .* 0;
496
497     profit = (s1_mesh + s2_mesh) .* 0;
498
499     % Flow rate Indicies | For the flowrates(i) array
500     HYDROGEN = 1;
501     METHANE = 2;
502     ETHYLENE = 3;
503     PROPANE = 4;
504     BUTANE = 5;
505
506     i = 1;
507     for s1 = s1_domain
508         for s2 = s2_domain
509
510             P_hydrogen = P_HYDROGEN(s1, s2);
511             P_methane = P_METHANE(s1, s2);
512             P_ethylene = P_ETHYLENE;
513             P_propane = P_PROPANE(s1, s2);
514             P_butane = P_BUTANE(s1, s2);
515             F_ethane = F_ETHANE(s1, s2);
516
517             P_flowrates = [ P_hydrogen, P_methane, P_ethylene, P_propane, P_butane ];
518
519             if (flowrates_valid(P_flowrates))
520
521                 % Store for plotting (kta)
522                 hydrogen_flowrates(i) = P_HYDROGEN(s1, s2);
523                 methane_flowrates(i) = P_METHANE(s1, s2);
524                 ethylene_flowrates(i) = P_ETHYLENE;
525                 propane_flowrates(i) = P_PROPANE(s1, s2);
526                 butane_flowrates(i) = P_BUTANE(s1, s2);
527                 ethane_flowrates(i) = F_ETHANE(s1, s2);
528
529                 % F_ethane = F_ETHANE(select_1(i), select_2(i));
530                 % F_fresh_ethane = F_ethane;
```

```
531             % F_ethane_rxtr = F_ethane(i) * ( conversion(i) / (1 - conversion(i)) );
532
533             xi = [];
534             % Calculate the heat flux needed to keep reactor isothermal
535             heat_flux = 0;
536             xi = get_xi(P_flowrates);
537             F_steam = STEAM_TO_FEED_RATIO_MASS * F_ethane;
538             heat_flux = heat_flux + heat_ethane(F_ethane, TEMP_ETHANE_FEED, TEMP_RXTR);
539             % heat_flux = heat_flux + heat_ethane(F_ethane_into_reactor, TEMP_SEPARATION, TEMP_RXTR);
540             heat_flux = heat_flux + heat_steam(F_steam, STEAM_CHOICE, PRESS_RXTR, TEMP_RXTR) ;
541             heat_flux = heat_flux + heat_rxn(xi);
542
543     %         % Use the heat flux to calculate the fuel cost
544             [combusted_fuel_flow_rates, heat_flux_remaining] = fuel_combustion(heat_flux, P_flowrates);
545
546             % Calculate how much natural gas you needed to combust
547             F_natural_gas = natgas_combustion(heat_flux_remaining);
548
549             % Determine how much of the product streams were combusted to keep the reactor isothermal
550
551             combusted_hydrogen = combusted_fuel_flow_rates(HYDROGEN);
552             combusted_methane = combusted_fuel_flow_rates(METHANE);
553             combusted_propane = combusted_fuel_flow_rates(PROPANE);
554             combusted_butane = combusted_fuel_flow_rates(BUTANE);
555
556     %         % VALUE CREATED | Primary Products
557             profit(i) = profit(i) + value_ethylene(P_ethylene);
558             profit(i) = profit(i) + value_h2_chem(P_hydrogen - combusted_hydrogen);
559
560             % VALUE CREATED | Non-combusted fuels
561             % profit(i) = profit(i) + value_methane(P_methane - combusted_methane);
562                 % ?? I don't think you can sell methane. IH - need to
563                 % determine energy requirements for compressors +
564                 % separation + cooling (will likely need to purchase
565                 % Nat Gas)
566             profit(i) = profit(i) + value_propane(P_propane - combusted_propane);
567             profit(i) = profit(i) + value_butane(P_butane - combusted_butane);
568
569             % COSTS INCURRED
570             profit(i) = profit(i) - tax_C02(combusted_fuel_flow_rates, F_natural_gas);
571             profit(i) = profit(i) - cost_steam(F_steam, COST_RATES_STEAM(STEAM_CHOICE, STEAM_COST_COL));
572             profit(i) = profit(i) - value_ethane(F_ethane);
573             profit(i) = profit(i) - cost_natural_gas_fuel(F_natural_gas);
574             profit(i) = profit(i) - cost_waste_stream(F_steam);
575
576         else
577             profit(i) = INVALID_FLOWRATE;
578             ethylene_flowrates(i) = INVALID_FLOWRATE;
579         end
580         i = i + 1;
581     end
582 end
583
```

```
584    profit = profit ./ 10^6; % Convert to Millions of dollars
585    profit(profit < 0) = 0; % remove irrelvant data
586
587    if (PLOT_ECON_COUNTOUR)
588        disp("Plotting EP Contour Map")
589        plot_contour(s1_mesh, s2_mesh, profit, PROFIT_S1S2_OPT);
590    end
591    if (PLOT_ECON_3D)
592        disp("Plotting 3D EP Surface Function")
593        plot_3D(s1_mesh, s2_mesh, profit, PROFIT_S1S2_OPT);
594    end
595
596    % Prepare the array of flow rate matrices
597    % flowRatesArray = {hydrogen_flowrates, methane_flowrates, ethylene_flowrates, propane_flowrates, butane_flowrates, ethane_flowrates};
598
599    % Call the function with the desired row
600    % plotFlowRatesForRow(4, flowRatesArray); % To plot the first row across all matrices
601 end
602
603
604 % SCRIPT | REACTOR ____________________________________________________
605
606 T_RANGE = linspace(T_MIN, T_MAX, NUM_T_POINTS);
607 P_RANGE = linspace(P_MIN, P_MAX, NUM_P_POINTS);
608 STEAM_RANGE = linspace(STEAM_MIN, STEAM_MAX, NUM_STEAM_POINTS);
609 V_RANGE = [V_MIN, V_MAX]; % WARNING THESE ARE IN LITERS
610 % H2 Methane Ethane Propane Butane Ethylene
611 F_INTIAL_COND = [ 0; 0; 0; 0; 0; 10]; % These are in kta
612
613    % Product flow rate indicies
614    HYDROGEN = 1;
615    METHANE = 2;
616    ETHYLENE = 3;
617    PROPANE = 4;
618    BUTANE = 5;
619
620    % Feed flow rate index
621    ETHANE = 6;
622
623 if (CALCULATE_REACTOR_FLOWS)
624    disp("Reactor Script ")
625    for T_i = T_RANGE
626        for P_i = P_RANGE
627            for MR_S_i = STEAM_RANGE
628
629                % override the T_i and P_i with user input
630                if T_P_OVERRIDE
631                    disp("WARNING: OVERRIDE HAS BEEN ACTIVATED")
632                    T_i = T_OVERRIDE;
633                    P_i = P_OVERRIDE;
634                    MR_S_i = STEAM_MR_OVERRIDE;
635                end
636
```

```
637         fprintf("\n\nT = %f [C], P = %f [bar] MR = %f [__]\n", T_i, P_i, MR_S_i)
638
639         % Setup the PFR Design Equations
640
641         % BASIS CALCULATIONS_____________________________________________________
642
643         % CONVERT TO MOLES_______________________________________________________
644         % Convert all of the initial conditions to mol / s
645         % (mol / s) =         (kt / yr) *             (g / kt) *   ( mol / g )      * ( yr / s)
646         F_INTIAL_COND(METHANE) = F_INTIAL_COND(METHANE) * G_PER_KT * (1/MOLMASS_METHANE) * YR_PER_SEC;
647         F_INTIAL_COND(HYDROGEN) = F_INTIAL_COND(HYDROGEN) * G_PER_KT * (1/MOLMASS_HYDROGEN) * YR_PER_SEC;
648         F_INTIAL_COND(ETHANE) = F_INTIAL_COND(ETHANE) * G_PER_KT * (1/MOLMASS_ETHANE) * YR_PER_SEC;
649         F_INTIAL_COND(ETHYLENE) = F_INTIAL_COND(ETHYLENE) * G_PER_KT * (1/MOLMASS_ETHYLENE) * YR_PER_SEC;
650         F_INTIAL_COND(PROPANE) = F_INTIAL_COND(PROPANE) * G_PER_KT * (1/MOLMASS_PROPANE) * YR_PER_SEC;
651
652         % Calculate the molar flow rate of the steam
653         % mol/s = __    * mol / s
654         F_steam = MR_S_i * F_INTIAL_COND(ETHANE);
655
656         % Solve the system ODE's
657         %   (L, mol / s)         (L, mol/s, Celcius, Bar, mol/s)
658         odes = @(V, F) reactionODEs(V, F, T_i, P_i, F_steam);
659         [V_soln_ODE, F_soln_ODE] = ode45(odes, V_RANGE, F_INTIAL_COND);
660
661         % Calculate the conversion
662         conversion = (F_INTIAL_COND(ETHANE) - F_soln_ODE(:, ETHANE)) / F_INTIAL_COND(ETHANE);
663
664         % put handles length of the solution and the initial ethane flow
665         len = length(F_soln_ODE(:, 1));
666         F_ethane_initial = ones(len, 1) * F_INTIAL_COND(ETHANE);
667
668         % Calculate the Selectivities, for each row (aka V_rxtr)
669         select_1 = (F_soln_ODE(:, ETHYLENE) ) ./ (F_ethane_initial - F_soln_ODE(:, ETHANE));
670         select_2 = (F_soln_ODE(:, PROPANE) ) ./ (F_ethane_initial - F_soln_ODE(:, ETHANE));
671
672         % Calculate the inlet volumetric flow rate
673         % (L / s) ???????????????
674         P_sum = F_soln_ODE(:, HYDROGEN:BUTANE);
675         % Turn these constants into vectors to operation is valid
676         F_steam = ones(length(P_sum(:,1)), 1) .* F_steam;
677         % put handles on terms, to make the code readable
678         sum_flowrates_into_reactor = F_INTIAL_COND(ETHANE) + F_steam;
679         % Calculate the flow rate into the reactor
680         q0 = (R_2  * (T_i + C_TO_K) / P_i) .* sum_flowrates_into_reactor;
681           % This is F.30 in the 'Design PFR Algorithm Appendix'
682
683         % PLANT CALCULATIONS_____________________________________________________
684
685         % Calculate the the flowrates of the plant sized reactor given S1, S2 from ODE's
686         F_ethane = [];
687         P_ethylene = [];
688         for row = 1:length(select_1)
689           % mol / s        = (kt / yr)   * (g / kt)  * (mol / g)          * (yr / s)
```

```
690             P_ethylene(row, 1) = P_ETHYLENE .* G_PER_KT .* (1/MOLMASS_ETHYLENE) * YR_PER_SEC;
691         end
692
693         % Calculate the scaling factor of the plant, from the basis
694         % mol / mol = ...
695         scaling_factor = P_ethylene(:, 1) ./ F_soln_ODE(:, ETHYLENE);
696
697         % Calculate the volume of the plant sized reactor
698         % L / s = ( L / s )       * (  (mol / s )  ) / (   (mol / s)       )
699         %       BASIS           *    PLANT_FLOW   /     BASIS_FLOW
700         V_plant = V_soln_ODE(:, 1) .* scaling_factor;
701
702         % cost of the reactor
703         cost_rxt_vec = zeros(size(V_plant));
704         for row = 1:length(V_plant)
705             % ( $ )
706             cost_rxt_vec(row) = cost_reactor(V_plant(row,1) * M3_PER_L);
707             cost_rxt_vec(row) = cost_rxt_vec(row) / YEARS_IN_OPERATION;
708         end
709
710         % inlet flow of the plant scaled reactor
711         q0_plant = q0(:, 1) .* scaling_factor;
712             % Eqn F.35 in 'Design PFR Algorithm Appendix'
713
714         % Scaling all of the molar flowrates to the size of the plant
715         F_soln_ODE(:, METHANE) = F_soln_ODE(:, METHANE) .* scaling_factor;
716         F_soln_ODE(:, HYDROGEN) = F_soln_ODE(:, HYDROGEN) .* scaling_factor;
717         F_soln_ODE(:, ETHANE) = F_soln_ODE(:, ETHANE) .* scaling_factor;
718         F_soln_ODE(:, ETHYLENE) = F_soln_ODE(:, ETHYLENE) .* scaling_factor;
719         F_soln_ODE(:, BUTANE) = F_soln_ODE(:, BUTANE) .* scaling_factor;
720         F_soln_ODE(:, PROPANE) = F_soln_ODE(:, PROPANE) .* scaling_factor;
721
722         % CONVERT BACK TO MASS_____________________________________________
723
724         % convert back to kta
725         % kt / yr =  mol / s    * g / mol       * kt / g   * s / yr
726         F_soln_ODE(:, METHANE) = F_soln_ODE(: ,METHANE) * MOLMASS_METHANE * KT_PER_G * SEC_PER_YR;
727         F_soln_ODE(:, ETHANE) = F_soln_ODE(:, ETHANE) * MOLMASS_ETHANE * KT_PER_G * SEC_PER_YR;
728         F_soln_ODE(:, HYDROGEN) = F_soln_ODE(:, HYDROGEN) * MOLMASS_HYDROGEN * KT_PER_G * SEC_PER_YR;
729         F_soln_ODE(:, ETHYLENE) = F_soln_ODE(:, ETHYLENE) * MOLMASS_ETHYLENE * KT_PER_G * SEC_PER_YR;
730         F_soln_ODE(:, BUTANE) = F_soln_ODE(:, BUTANE) * MOLMASS_BUTANE * KT_PER_G * SEC_PER_YR;
731         F_soln_ODE(:, PROPANE) = F_soln_ODE(:, PROPANE) * MOLMASS_PROPANE * KT_PER_G * SEC_PER_YR;
732
733                 % Check if you're conserving mass
734         conserv_mass = zeros(length(F_soln_ODE(:,1)), 1);
735
736         % ECONOMIC CALCULATIONS_____________________________________________
737         profit = zeros(length(F_soln_ODE(:,1)), 1);
738         for i = 1:length(F_soln_ODE(:, 1))
739
740             % DEBUGGING
741             if i > 500
742                 disp("")
```

```
743             end
744             % P_flowrates = [ P_hydrogen, P_methane, P_ethylene, P_propane, P_butane ];
745             P_flowrates = F_soln_ODE(i , HYDROGEN:BUTANE);
746
747             P_hydrogen = P_flowrates(HYDROGEN);
748             P_methane = P_flowrates(METHANE);
749             P_ethylene = P_flowrates(ETHYLENE);
750             P_propane = P_flowrates(PROPANE);
751             P_butane = P_flowrates(BUTANE);
752
753             F_fresh_ethane = F_ETHANE(select_1(i), select_2(i));
754             R_ethane = F_fresh_ethane * ( ( 1 - conversion(i)) / conversion(i) );
755             R_ethane = F_soln_ODE(i, ETHANE);
756                 % ?? These two values R should be the same
757
758             if (~flowrates_valid(P_flowrates))
759                 disp("WARNING SOME FLOWATES MAY BE INVALID")
760             end
761
762             % Calculate the heat flux needed to keep reactor isothermal
763             heat_flux = 0;
764             xi = get_xi(P_flowrates);
765             F_steam = STEAM_TO_FEED_RATIO_MASS * (F_fresh_ethane + R_ethane);
766             heat_flux = heat_flux + heat_ethane(F_fresh_ethane, TEMP_ETHANE_FEED, TEMP_RXTR);
767             heat_flux = heat_flux + heat_ethane(R_ethane, T_SEPARATION - C_TO_K, TEMP_RXTR);
768             heat_flux = heat_flux + heat_steam(F_steam, STEAM_CHOICE, PRESS_RXTR, TEMP_RXTR) ;
769             heat_flux = heat_flux + heat_rxn(xi);
770
771     %         % Use the heat flux to calculate the fuel cost
772             [combusted_fuel_flow_rates, heat_flux_remaining] = fuel_combustion(heat_flux, P_flowrates);
773
774             % Calculate how much natural gas you needed to combust
775             F_natural_gas = natgas_combustion(heat_flux_remaining);
776
777             % Determine how much of the product streams were combusted to keep the reactor isothermal
778             combusted_hydrogen = combusted_fuel_flow_rates(HYDROGEN);
779             combusted_methane = combusted_fuel_flow_rates(METHANE);
780             combusted_propane = combusted_fuel_flow_rates(PROPANE);
781             combusted_butane = combusted_fuel_flow_rates(BUTANE);
782
783     %         % VALUE CREATED | Primary Products
784             profit(i, 1) = profit(i, 1) + value_ethylene(P_ethylene);
785             profit(i, 1) = profit(i, 1) + value_h2_chem(P_hydrogen - combusted_hydrogen);
786
787             % VALUE CREATED | Non-combusted fuels
788             % The commented line can be removed or modified as per the context.
789             % profit(i, 1) = profit(i, 1) + value_methane(P_methane - combusted_methane);
790             profit(i, 1) = profit(i, 1) + value_propane(P_propane - combusted_propane);
791             profit(i, 1) = profit(i, 1) + value_butane(P_butane - combusted_butane);
792
793             % COSTS INCURRED
794             profit(i, 1) = profit(i, 1) - tax_C02(combusted_fuel_flow_rates, F_natural_gas);
795             profit(i, 1) = profit(i, 1) - cost_steam(F_steam, COST_RATES_STEAM(STEAM_CHOICE, STEAM_COST_COL));
```

```
796                 profit(i, 1) = profit(i, 1) - value_ethane(F_fresh_ethane);
797                 profit(i, 1) = profit(i, 1) - cost_natural_gas_fuel(F_natural_gas);
798                 profit(i, 1) = profit(i, 1) - cost_waste_stream(F_steam);
799                 profit(i, 1) = profit(i, 1) - cost_separation_system(P_flowrates, F_steam, R_ethane);
800                  cost_sep = cost_separation_system(P_flowrates, F_steam, R_ethane) ;
801                 if cost_sep > 0
802
803                     fprintf("cost sep = %3.3e \n",cost_sep)
804                 end
805                 % fprintf("cost %s\n", cost_separation_system(P_flowrates, F_steam, R_ethane)')
806
807                 % Checking if I still have any sanity left after this, who knows...
808                 conserv_mass(i, 1) = F_fresh_ethane - sum(P_flowrates);
809             end
810
811             % PLOTTING_____________________________________________________________
812             col_names = {'V_rxtr [L] ', 'Hydrogen [kta]', 'Methane', ...
813                 'Ethylene', 'Propane', 'Butane','Ethane', 'conversion', ...
814                 'S1', 'S2', 'q0 [ L /s ]', 'Vol_plant [ L ]', 'q0 plant', 'cost reactor', 'profit', 'net profit', 'conserv mass'};
815             soln_table = table( V_soln_ODE, F_soln_ODE(:, HYDROGEN), ...
816                     F_soln_ODE(:, METHANE), F_soln_ODE(:, ETHYLENE), ...
817                     F_soln_ODE(:, PROPANE), F_soln_ODE(:, BUTANE), ...
818                     F_soln_ODE(:, ETHANE), conversion,select_1, ...
819                     select_2,q0,V_plant,q0_plant,cost_rxt_vec,profit, profit - cost_rxt_vec, conserv_mass,'VariableNames',col_names)
820 %           soln_table.Properties.VariableNames = col_names;
821
822             % Computer Selectivity vs conversion relationships
823
824             % Use Selectivity vs Conversion Relationships with lvl 2 & 3 balances
825             % to calculate the true feed flow rates into the reactor
826             npv.ethyleneValue = value_ethylene(P_ethylene);
827             npv.hydrogenValue = value_h2_chem(P_hydrogen - combusted_hydrogen);
828             npv.ethane = value_ethane(F_fresh_ethane);
829             npv.utilitiesCost = cost_steam(F_steam, COST_RATES_STEAM(STEAM_CHOICE, STEAM_COST_COL));
830             npv.CO2sustainabilityCharge = tax_C02(combusted_fuel_flow_rates, F_natural_gas);
831             npv.conversion = conversion(i);
832             npv.isbl = cost_rxt_vec + cost_separation_system(P_flowrates, F_steam, R_ethane);
833             % NPV CALCS
834             npv_graphs(npv)
835
836         end
837     end
838 end
839 end
840
841
842
843
844 disp("The Script is done running  )  "
845 % HELPER FUNCTIONS | PLOTTING_____________________________________________
846
847 function z = plot_contour(x, y, z, options)
848     global PSA_TOGGLE
```

```
849     % Unpack options
850     x_label = options{1};
851     y_label = options{2};
852     plt_title = options{3};
853     plt_saveName = options{4};
854
855     if PSA_TOGGLE
856         stringValue = 'true';
857     else
858         stringValue = 'false';
859     end
860     plt_title = plt_title + sprintf(" PSA %s ", stringValue);
861
862     hold on
863     figure
864     [C, h] = contourf(x, y, z); % Create filled contours
865     clabel(C, h,  'FontSize', 10, 'Color', 'k', 'LabelSpacing', 200); % Customize label properties
866     xlabel(x_label);
867     ylabel(y_label);
868     title(plt_title);
869     saveas(gcf, plt_saveName);
870     hold off
871 end
872
873 function plot_3D(x, y, z, options)
874     global PSA_TOGGLE
875
876     % Unpack options
877     x_label = options{1};
878     y_label = options{2};
879     plt_title = options{3};
880     plt_saveName = options{4};
881
882     if PSA_TOGGLE
883         stringValue = 'true';
884     else
885         stringValue = 'false';
886     end
887     plt_title = plt_title + sprintf(" PSA %s ", stringValue);
888
889      % Create a new figure
890      hold on; % Hold on to add multiple plot elements
891     figure
892     surf(x, y, z); % Create a 3D surface plot
893
894     % Customizing the plot
895     xlabel(x_label);
896     ylabel(y_label);
897     zlabel('Z Value'); % Add a label for the z-axis
898     title(plt_title);
899     colorbar; % Adds a color bar to indicate the scale of z values
900 %     shading interp; % Option for smoother color transition on the surface
901
```

```
902     hold off; % Release the figure
903     saveas(gcf, plt_saveName); % Save the figure to file
904 end
905
906
907 function plotFlowRatesForRow(row, flowRatesArray)
908     % flowRatesArray is expected to be an array of matrices, where each matrix corresponds to a species' flow rates
909
910     % Names of the gases for labeling purposes
911     gasNames = {'Hydrogen', 'Methane', 'Ethylene', 'Propane', 'Butane', 'Ethane'};
912
913     % Create a figure
914     figure;
915     hold on; % Hold on to plot all data on the same figure
916
917     % Loop through each flow rate matrix in the array
918     for i = 1:length(flowRatesArray)
919         % Extract the specified row from the current matrix
920         currentRow = flowRatesArray{i}(row, :);
921
922         % Plot the current row with a marker
923         plot(currentRow, '-o', 'DisplayName', gasNames{i});
924     end
925
926     % Adding plot features
927     title(sprintf('Flow Rates for Row %d', row));
928     xlabel('Selectivity 1 (S2 fixed)');
929     ylabel('Flow Rate');
930     legend('show');
931     hold off; % Release the figure for other plots
932 end
933
934 % HELPER FUNCTIONS | HEAT ____________________________________
935
936 function [combusted_fuel_flowrates, heatflux_left] = fuel_combustion(heat_flux, flowrates)
937     global HYDROGEN METHANE ETHYLENE PROPANE BUTANE;
938     global ENTHALPY_METHANE ENTHALPY_PROPANE ENTHALPY_BUTANE HEAT_CAPACITY_ETHANE;
939     global MT_PER_KT G_PER_KT GJ_PER_KJ KJ_PER_GJ MOLMASS_METHANE KT_PER_G MOLMASS_BUTANE ...
940         MOLMASS_PROPANE PSA_TOGGLE ENTHALPY_HYDROGEN MOLMASS_HYDROGEN
941
942     % Note! : Longest Chain Hydrocarbons are cheapest to combust
943
944     % initialize all values in the array to be zero
945     combusted_fuel_flowrates = flowrates * 0;
946
947     % LOGIC : Goes through each heat source in order, returns if the heat flux supplied is sufficient.
948     heatflux_left = heat_flux;
949
950     % (GJ / yr)         = (kt / yr)         * (g / kt) * (kJ / g)       * (GJ / kJ)
951     Q_combust_all_hydrogen = flowrates(HYDROGEN) * G_PER_KT * ENTHALPY_HYDROGEN * GJ_PER_KJ;
952
953     if (~PSA_TOGGLE)
954         % Hydrogen
```

```
955     if (heatflux_left > Q_combust_all_hydrogen)
956         combusted_fuel_flowrates(HYDROGEN) = flowrates(HYDROGEN);
957         heatflux_left = heatflux_left - Q_combust_all_hydrogen;
958     else
959         % (kt / yr)                  = ((GJ)              ) * (KJ / GJ) *
960         combusted_fuel_flowrates(HYDROGEN) = (heatflux_left) * KJ_PER_GJ * ...
961             ... % (mol / KJ)      * (g / mol)      * (kt / g)
962             ( 1 / ENTHALPY_HYDROGEN) * MOLMASS_HYDROGEN * KT_PER_G;
963         heatflux_left = 0;
964         return
965     end
966 end
967
968 % (GJ / yr)         = (kt / yr)        * (g / kt) * (kJ / g)       * (GJ / kJ)
969 Q_combust_all_methane = flowrates(METHANE) * G_PER_KT * ENTHALPY_METHANE * GJ_PER_KJ;
970
971 % Methane
972 if (heatflux_left > Q_combust_all_methane)
973     combusted_fuel_flowrates(METHANE) = flowrates(METHANE);
974     heatflux_left = heatflux_left - Q_combust_all_methane;
975 else
976     % (kt / yr)                  = ((GJ)              ) * (KJ / GJ) *
977     combusted_fuel_flowrates(METHANE) = (heatflux_left) * KJ_PER_GJ * ...
978         ... % (mol / KJ)      * (g / mol)      * (kt / g)
979         ( 1 / ENTHALPY_METHANE) * MOLMASS_METHANE * KT_PER_G;
980     heatflux_left = 0;
981     return
982 end
983
984 % (GJ / yr)         = (kt / yr)        * (g / kt) * (kJ / g)       * (GJ / kJ)
985 Q_combust_all_propane = flowrates(PROPANE) * G_PER_KT * ENTHALPY_PROPANE * GJ_PER_KJ;
986
987 % Propane
988 if (heatflux_left > Q_combust_all_propane)
989     combusted_fuel_flowrates(PROPANE) = flowrates(PROPANE);
990     heatflux_left = heatflux_left - Q_combust_all_propane;
991 else
992     % (kt / yr)                  = ((GJ)              ) * (KJ / GJ) *
993     combusted_fuel_flowrates(PROPANE) = (heatflux_left) * KJ_PER_GJ * ...
994         ... % (mol / KJ)      * (g / mol)      * (kt / g)
995         ( 1 / ENTHALPY_PROPANE) * MOLMASS_PROPANE * KT_PER_G;
996     heatflux_left = 0;
997     return
998 end
999
1000  % (GJ / yr)         = (kt / yr)        * (g / kt) * (kJ / g)       * (GJ / kJ)
1001  Q_combust_all_butane = flowrates(BUTANE) * G_PER_KT * ENTHALPY_BUTANE * GJ_PER_KJ;
1002
1003  % Butane
1004  if (heatflux_left > Q_combust_all_butane)
1005      combusted_fuel_flowrates(BUTANE) = flowrates(BUTANE);
1006      heatflux_left = heatflux_left - Q_combust_all_butane;
1007  else
```

```
1008        % (kt / yr)                    = ((GJ)              ) * (KJ / GJ) *
1009        combusted_fuel_flowrates(BUTANE) = (heatflux_left) * KJ_PER_GJ * ...
1010            ... % (mol / KJ)       * (g / mol)      * (kt / g)
1011            ( 1 / ENTHALPY_BUTANE) * MOLMASS_BUTANE * KT_PER_G;
1012        heatflux_left = 0;
1013        return
1014    end
1015 end
1016
1017 %       GJ   =          (kta   ,  __        , bar     , C )
1018 function heat = heat_steam(F_steam, STEAM_CHOICE, P_reactor, T_reactor)
1019    global COST_RATES_STEAM;
1020    global STEAM_PRESSURE_COL STEAM_TEMP_COL COST_RATES_STEAM G_PER_KT ...
1021           MOLMASS_WATER BAR_PER_PSIA C_TO_K HEAT_CAPACITY_WATER GJ_PER_KJ;
1022
1023    P_steam = COST_RATES_STEAM(STEAM_CHOICE, STEAM_PRESSURE_COL); % [ psia ]
1024    T_steam = COST_RATES_STEAM(STEAM_CHOICE, STEAM_TEMP_COL);     % [ C ]
1025    P_steam = P_steam * BAR_PER_PSIA;
1026    T_steam = T_steam + C_TO_K;
1027    T_reactor = T_reactor + C_TO_K;
1028
1029    if (P_steam > P_reactor) % Adiabatic Expansion
1030        T_adibatic = (T_steam) * (P_reactor / P_steam);
1031        T_steam = T_adibatic;
1032    elseif (P_steam < P_reactor) % Compression
1033        W = compressor_work(T_reactor, P_steam, P_reactor);
1034        if ADD_COMPRESSOR_WORK_TO_STEAM_HEATFLUX
1035            heat = heat + W;
1036        end
1037        % I should add this to the heat flux probably ??
1038    end
1039
1040    % KJ = kta     * (G / KT) * (mol / g)        * (KJ / MOL K)       * (K - K)
1041    heat = F_steam * G_PER_KT * (1/MOLMASS_WATER) * HEAT_CAPACITY_WATER * (T_reactor - T_steam);
1042    % GJ = KJ   * (KJ / GJ)
1043    heat = heat * GJ_PER_KJ;
1044
1045
1046    % Heat flux after temperture
1047
1048
1049
1050 end
1051
1052 function T_f = adiabatic_temp(T_0, P_0, P_f)
1053
1054    T_f = T_0 * ( P_0 / P_f);
1055 end
1056
1057 function W = compressor_work(T, P_0, P_f)
1058    R = 8.314;      % [ J / mol K]
1059
1060    W = - n * R * T * log(P_f / P_0);
```

```
1061
1062    % ?? THIS ALWAYS RETURNS 0 OR NULL, NOT IMPLEMENTED YET
1063
1064 end
1065
1066 % HELPER FUNCTIONS | TAXES_____________________________________
1067
1068 function cost = tax_C02(combusted_flowrates, F_natural_gas)
1069     global HYDROGEN METHANE ETHYLENE PROPANE BUTANE TAX_CO2_PER_MT;
1070     global MT_CO2_PER_KT_METHANE MT_CO2_PER_KT_PROPANE MT_CO2_PER_KT_BUTANE ...
1071    MT_CO2_PER_KT_NATURALGAS;
1072
1073     % Calculate the cost per kt (in tax) of each combusted fuel
1074     methane = combusted_flowrates(METHANE);
1075     propane = combusted_flowrates(PROPANE);
1076     butane = combusted_flowrates(BUTANE);
1077
1078     mt_c02 = 0;
1079     % kta  =  (MT)  + ( (kt fuel / yr) * (MT CO2 / KT FUEL) )
1080     mt_c02 = mt_c02 + methane * MT_CO2_PER_KT_METHANE;
1081    mt_c02 = mt_c02 + propane * MT_CO2_PER_KT_PROPANE;
1082     mt_c02 = mt_c02 + butane * MT_CO2_PER_KT_BUTANE;
1083     mt_c02 = mt_c02 + F_natural_gas * MT_CO2_PER_KT_NATURALGAS;
1084
1085     cost = mt_c02 * TAX_CO2_PER_MT;
1086 end
1087
1088 % HELPER FUNCTIONS | FUEL COSTS_____________________________________
1089
1090 function cost = cost_natural_gas_fuel(heat_flux_remaining)
1091     global VALUE_NATGAS_FUEL
1092     % $ / yr = (GJ)          * ($ / GJ)
1093     cost = heat_flux_remaining * VALUE_NATGAS_FUEL;
1094 end
1095
1096 % HELPER FUNCTIONS | FUEL FLOWRATES_____________________________________
1097
1098 function F_natural_gas = natgas_combustion(heat_flux_remaining)
1099     global KJ_PER_GJ ENTHALPY_NAT_GAS KT_PER_G MOLMASS_NATGAS;
1100     % output should be in kta, input is in GJ
1101
1102     %      kt         GJ            * (kJ / GJ) * (mol / kJ) *         (g / mol) *           (kt / g)
1103     F_natural_gas = heat_flux_remaining * KJ_PER_GJ * (1/ENTHALPY_NAT_GAS) * (MOLMASS_NATGAS) * KT_PER_G;
1104
1105 end
1106
1107 % FUNCTIONS | REACTOR ODE SYSTEM_____________________________________
1108
1109 function dFdV = reactionODEs(V, F, T, P, F_steam)
1110     global R_2 k1_f k1_r k2 k3 C_TO_K MOLMASS_METHANE MOLMASS_ETHANE MOLMASS_ETHYLENE ...
1111         MOLMASS_PROPANE MOLMASS_HYDROGEN MOLMASS_BUTANE YR_PER_SEC G_PER_KT SEC_PER_YR KT_PER_G
1112     % INPUT UNITS
1113     % V [ L ]
```

```
1114    % F [ kta ]
1115    % T [ Celcius ]
1116    % P [ bar ]
1117
1118    % Change the input units so that evrything is consistent
1119    % P = P * ATM_PER_BAR;
1120    T = T + C_TO_K;
1121
1122    % Product flow rate indicies
1123    HYDROGEN = 1;
1124    METHANE = 2;
1125    ETHYLENE = 3;
1126    PROPANE = 4;
1127    BUTANE = 5;
1128
1129    % Feed flow rate index
1130    ETHANE = 6;
1131
1132    F_tot = sum(F) + F_steam;
1133
1134
1135    % Hydrogen = A
1136    dFAdV = (k1_f(T) * ( (F(ETHANE) * P) / (F_tot * R_2 * T) )   ) - ...
1137          (k1_r(T) * ( F(ETHYLENE) * F(HYDROGEN) * P^2) ) / (F_tot * R_2 * T)^2;
1138
1139    % Methane = B
1140    dFBdV = (k2(T) * (F(ETHANE) * P)^2) / (F_tot * R_2 * T)^2;
1141
1142    % Ethylene = C
1143    dFCdV = (k1_f(T) * (F(ETHANE) * P / (F_tot * R_2 * T))) - ...
1144          (k1_r(T) * (F(ETHYLENE) * F(HYDROGEN) * P^2) / (F_tot * R_2 * T)^2) - ...
1145          (k3(T) * (F(ETHANE) * F(ETHYLENE) * P^2) / (F_tot * R_2 * T)^2);
1146
1147    % Propane = E
1148    dFEdV = k2(T) * (F(ETHANE) * P)^2 / (F_tot * R_2 * T)^2;
1149
1150    % Butane = F
1151    dFFdV = (k3(T) * (F(ETHANE) * F(ETHYLENE) * P^2)) / (F_tot * R_2 * T)^2;
1152
1153    % Ethane = D
1154    dFDdV = (-k1_f(T) * (F(ETHANE) * P / (F_tot * R_2 * T))) + ...
1155          (k1_r(T) * (F(ETHYLENE) * F(HYDROGEN) * P^2)/(F_tot * R_2 * T)^2) - ...
1156          (k2(T) * F(ETHANE)^2 * P^2 / (F_tot * R_2 * T)^2) - ...
1157          (k3(T) * F(ETHANE) * F(ETHYLENE) * P^2 / (F_tot * R_2 * T)^2);
1158
1159    T = T - C_TO_K;
1160
1161    dFdV = [dFAdV; dFBdV; dFCdV; dFEdV; dFFdV; dFDdV];
1162
1163 end
1164
1165 function cost = cost_reactor(V_plant_input)
1166    global FT_PER_METER STEAM_TO_FEED_RATIO
```

```
1167    FT_PER_METER = 3.28084;
1168    % ??? WHAT ARE THE UNITS OF TIME
1169    %
1170    pi = 3.14159;
1171    D = 0.05;                       % [m]
1172    V_plant_max = pi * (0.025)^2 * 20;     %[m^3]
1173
1174    % Reactors have a max length, so calculate the number of full size reactors
1175    % and add it to the cost of the one non-max length reactor
1176
1177    cost = 0;
1178
1179    % Find the Cost of the max-sized reactors
1180    num_of_additional_reactors = int64(V_plant_input / V_plant_max);
1181    num_of_additional_reactors = double(num_of_additional_reactors);
1182
1183    V_plant = V_plant_max;
1184    factor_1 = 4.18;
1185    factor_2 = (V_plant / (pi * (D/2)^2) * FT_PER_METER)^0.82;
1186    factor_3 = (101.9 * D * FT_PER_METER)^1.066;
1187    factor_4 = (1800 / 280);
1188    cost_max_reactor = factor_1 * factor_2 * factor_3 * factor_4;
1189    cost = cost + num_of_additional_reactors * cost_max_reactor;
1190
1191
1192    % Find the cost of the non-max size reactor
1193    V_plant = V_plant_input - V_plant_max * num_of_additional_reactors;
1194    if V_plant < 0
1195        V_plant = 0;
1196    end
1197    factor_1 = 4.18;
1198    factor_2 = (V_plant / (pi * (D/2)^2) * FT_PER_METER)^0.82;
1199    factor_3 = (101.9 * D * FT_PER_METER)^1.066;
1200    factor_4 = (1800 / 280);
1201    cost = cost + factor_1 * factor_2 * factor_3 * factor_4;
1202
1203
1204 end
1205
1206 %       [$]  =              ( kta   )
1207 function cost = cost_waste_stream(F_steam)
1208    global MOLMASS_WATER G_PER_KT YR_PER_SEC R_2 M3_PER_L T_SEPARATION ...
1209           P_SEPARATION SEC_PER_YR C_TO_K DENSITY_LIQ_WATER KG_PER_KT
1210
1211   % m^3 / s = (kt / yr) * (kg / kt)   * (m^3 / kg)  * (yr / s)
1212    q = F_steam * KG_PER_KT * (1 / DENSITY_LIQ_WATER) * YR_PER_SEC;
1213       % ?? Assume that all of the water out of the sep system is liquid
1214
1215    a = 0.001 + 2e-4*q^(-0.6);
1216       %Source: Uldrich and Vasudevan
1217    b=0.1;
1218       %Source: Uldrich and Vasudevan
1219    CEPCI = 820;
```

```
1220      %Source: Lecture slides
1221    C_f = 3.0;                  % [ $ / GJ ]
1222
1223    %$/m^3 waste water
1224    cost_waste_water = a*CEPCI + b*C_f;
1225
1226    % m^3 / s = (m^3 / s) * (s / yr)
1227    q = q * SEC_PER_YR;
1228    cost = cost_waste_water * q;
1229
1230  end
1231
1232  function cost = cost_separation_system(P_flowrates, F_steam, R_ethane)
1233    global MOLMASS_METHANE MOLMASS_HYDROGEN MOLMASS_ETHANE MOLMASS_ETHYLENE ...
1234        MOLMASS_PROPANE MOLMASS_BUTANE YR_PER_SEC
1235    global T_SEPARATION R PRESS_RXTR R ...
1236     MAX_OPEX MAX_TFCI MAX_CAPEX G_PER_KT MOLMASS_WATER
1237
1238    % Product flow rate indicies
1239    HYDROGEN = 1;
1240    METHANE = 2;
1241   ETHYLENE = 3;
1242    PROPANE = 4;
1243    BUTANE = 5;
1244
1245    % Feed flow rate index
1246    ETHANE = 6;
1247
1248    % SEPARATION_EFFICIENCY_FACTOR = 30;
1249    T = T_SEPARATION; % [ K ]
1250
1251   %Using compositions from ASPEN
1252   %Component mole flow rate out of rxtr over total mole flow rate out of reactor
1253    % Mol fractions out of the reactoor
1254
1255    % (mol / s) = (kt / yr) * (g / kt) * (mol / g) * (yr / s)
1256    P_flowrates(METHANE) = P_flowrates(METHANE) * G_PER_KT * (1/MOLMASS_METHANE) * YR_PER_SEC;
1257    P_flowrates(HYDROGEN) = P_flowrates(HYDROGEN) * G_PER_KT * (1/MOLMASS_HYDROGEN) * YR_PER_SEC;
1258    R_ethane = R_ethane * G_PER_KT * (1/MOLMASS_ETHANE) * YR_PER_SEC;
1259    P_flowrates(ETHYLENE) = P_flowrates(ETHYLENE) * G_PER_KT * (1/MOLMASS_ETHYLENE) * YR_PER_SEC;
1260    P_flowrates(PROPANE) = P_flowrates(PROPANE) * G_PER_KT * (1/MOLMASS_PROPANE) * YR_PER_SEC;
1261   P_flowrates(BUTANE) = P_flowrates(BUTANE) * G_PER_KT * (1/MOLMASS_BUTANE) * YR_PER_SEC; % Add this line for butane
1262    F_steam = F_steam * G_PER_KT * (1/MOLMASS_WATER) * YR_PER_SEC;
1263
1264    %CONVERT TO_MOLES
1265
1266    P_tot = sum(P_flowrates(HYDROGEN:BUTANE)) + F_steam + R_ethane;
1267
1268    z_methane = P_flowrates(METHANE) / P_tot;
1269    z_hydrogen = P_flowrates(HYDROGEN) / P_tot;
1270    z_ethane = R_ethane / P_tot;
1271   z_ethylene = P_flowrates(ETHYLENE) / P_tot;
1272    z_propane = P_flowrates(PROPANE) / P_tot;
```

```
z_butane = P_flowrates(BUTANE) / P_tot;
z_water = F_steam / P_tot;

%Mol fractions leaving each separation system (refer to Isa's drawing in GN)
% leaving sep 1
x_water = 1;

% leaving sep 4
x_ethane = 1;
x_ethylene = 1;

% leaving sep 2
x_butane = 0.0003;
x_propane = 1 - x_butane;

% leaving sep 5 (PSA)
x_methane = 4.03293090303065e-004;
x_hydrogen = 1 - x_methane;
   % ?? How should I implement the PSA toggle switch on this

%Pressures of PSA system [bar]
P_in = PRESS_RXTR;
P_H2 = 10;          % [ bar ]
P_ME = 1;           % [ bar ]
   % These outlet pressures are constant for PSA system. DONT change

%Using flow rates from ASPEN [NOTE: FOR MATLAB USE THE VALUES FROM THE
%SOLN_TABLE. WE USED THESE AS EXPECTED COSTS)

% Flowrates of each exiting stream from the sep system

F_water = F_steam;                      % mol/s
F_LPG = P_flowrates(BUTANE) + P_flowrates(PROPANE); % (mol / s)
F_ethylene = P_flowrates(ETHYLENE);             % (mol / s)
F_ethane = R_ethane;                    % (mol / s)
F_H2 = P_flowrates(HYDROGEN);                   % (mol / s)
F_ME = P_flowrates(METHANE);                    % (mol / s);

%(J/s) =    (mol/s) * (J/mol K) * (T)
W_min_Sep_System = F_water*R*T*log(x_water/z_water) + ...
            F_LPG*R*T*log(x_propane/z_propane + ...
                    x_butane/z_butane) + ...
            F_ethylene*R*T*log(x_ethylene/z_ethylene) + ...
            F_ethane*R*T*log(x_ethane/z_ethane) + ...
            R*T*( ...
               F_H2*log(P_H2/P_in)+ ...
               F_H2*log(x_hydrogen/z_hydrogen) +...
                F_ME*log(x_methane/z_methane) +...
               F_ME*log(P_ME/P_in)...
               );


lamdba_min = 20;
```

```
1326    lambda_max = 50;
1327    cost_energy = 3;       % ( $ / GJ )
1328
1329    if MAX_OPEX
1330    %($/yr)           =   (J/s)    * (GJ/J) * (Work Efficiency) *($/GJ)* (s/yr)
1331        opex =  W_min_Sep_System*1e-9 * lambda_max * cost_energy * 30.24e6;
1332    else
1333        opex = W_min_Sep_System*1e-9 * lamdba_min * cost_energy * 30.24e6;
1334    end
1335
1336    if MAX_CAPEX
1337    %($)              = ($/W)    (Efficiency) * (J/s)
1338        capex = 1 * lambda_max * W_min_Sep_System;
1339    else
1340        capex = 0.5 * lamdba_min * W_min_Sep_System;
1341    end
1342
1343    cost = 2.5 * capex ;
1344
1345 end
1346
1347
1348 function void = npv_graphs(npv)
1349
1350    % npv.ethyleneValue = value_ethylene(P_ethylene);
1351    % npv.hydrogenValue = value_h2_chem(P_hydrogen - combusted_hydrogen);
1352    % npv.ethane = value_ethane(F_fresh_ethane);
1353    % npv.utilitiesCost = cost_steam(F_steam, COST_RATES_STEAM(STEAM_CHOICE, STEAM_COST_COL));
1354    % npv.CO2sustainabilityCharge = tax_C02(combusted_fuel_flow_rates, F_natural_gas);
1355    % npv.conversion = conversion(i);
1356    % npv.isbl = cost_rxt_vec + cost_separation_system(P_flowrates, F_steam, R_ethane);
1357
1358    % NPV CALCS
1359
1360    filename = 'npv_code.xlsx';
1361    data = readtable(filename);
1362
1363    sheetName = 'cf'; % Specify the sheet name or number
1364    data = readtable(filename, 'Sheet', sheetName);
1365
1366    n = 5; % Number of times to duplicate the sheet
1367    for i = 1:n
1368        % Assuming 'data' is the table you want to duplicate
1369        writetable(data, filename, 'Sheet', ['DuplicatedSheet' num2str(i)]);
1370    end
1371
1372    cellRow = 10; % Row of the cell to modify
1373    cellCol = 'C'; % Column of the cell to modify, use column index if numerical indexing preferred
1374    newValue = 69; % New value to assign
1375
1376    for i = 1:n
1377        modifiedData = data; % Make a copy of the original data
1378        modifiedData{cellRow, cellCol} = newValue; % Modify the specific cell
```

```
1379
1380        % Export the modified data to a new sheet
1381        writetable(modifiedData, filename, 'Sheet', ['ModifiedSheet' num2str(i)]);
1382    end
1383
1384
1385
1386 end
1387
1388
1389
1390 function installedCost = calculate_installed_cost(Q, F_d, F_m, F_p)
1391
1392    Q = Q * 106BTU_PER_GJ * yr/hr;
1393    % Constants
1394    M_and_S = 1800; % Marshall and Swift index
1395    base_cost = 5.52 * 10^3;
1396
1397    % Purchased cost calculation
1398    % F_c = F_d + F_m + F_p;
1399    F_c = 1.1;
1400
1401    % Installed cost calculation
1402    installedCost = (M_and_S / 280) * (base_cost * Q^0.85 * (1.27 + F_c));
1403 end
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
```