

# General introduction

```
In [1]: #The code is developed for OCM and MDA thermally coupled reactors  
# The values used are for case 1* in Table 8 of the manuscript  
#The code is publicly disclosed and is free to use for educational purposes for advanced  
#Across each line of the code, the respective unit along a short explanation is mentioned  
#Authors: Muhammad Umar Jamil and Ma'moun Al-Rawashdeh (mamoun.al-rawashdeh@qatar.tamu.)
```

## Table of cells

```
In [2]: # Cell 1.0 Installing and importing required Libraries  
## Cell 1.1 Importing required Libraries  
## Cell 1.2 Installing thermo library  
## Cell 1.3 Upgrading thermo library  
## Cell 1.4 Importing relevant thermo Libraries  
  
# Cell 2.0 Reactor model  
## Cell 2.1 Constants  
## Cell 2.2 General tube design parameters  
## Cell 2.3 Tube design parameters  
## Cell 2.4 Flow rates and composition  
## Cell 2.5 Defining chemical names for thermodynamic properties  
## Cell 2.6 Heat of formation  
## Cell 2.7 Specific heat  
## Cell 2.8 Gas mixture physical properties  
## Cell 2.9 Model simulation parameters  
## Cell 2.10 Reactor model  
## Cell 2.11 Solve ODE  
## Cell 2.12 Reactor outputs  
## Cell 2.13 Printing reactor outputs  
## Cell 2.14 Plotting reactor outputs and simulation results
```

## Cell 1.0 Installing and importing required libraries

### Cell 1.1 Importing required libraries

```
In [3]: import numpy as np  
from sympy import symbols, solve  
from scipy.integrate import solve_bvp, odeint  
import matplotlib.pyplot as plt  
from numpy import trapz  
import mpmath as mpm  
import statistics as st  
import pandas as pd
```

### Cell 1.2 Installing thermo library

```
In [4]: pip install thermo
```

```
Requirement already satisfied: thermo in c:\users\mamoun.al-rawashdeh\anaconda3\lib\site-packages (0.1.40)
Note: you may need to restart the kernel to use updated packages.
Requirement already satisfied: fluids>=0.1.77 in c:\users\mamoun.al-rawashdeh\anaconda3\lib\site-packages (from thermo) (1.0.2)
Requirement already satisfied: scipy in c:\users\mamoun.al-rawashdeh\anaconda3\lib\site-packages (from thermo) (1.5.2)
Requirement already satisfied: pandas in c:\users\mamoun.al-rawashdeh\anaconda3\lib\site-packages (from thermo) (1.1.3)
Requirement already satisfied: numpy>=1.5.0 in c:\users\mamoun.al-rawashdeh\anaconda3\lib\site-packages (from fluids>=0.1.77->thermo) (1.19.2)
Requirement already satisfied: python-dateutil>=2.7.3 in c:\users\mamoun.al-rawashdeh\anaconda3\lib\site-packages (from pandas->thermo) (2.8.1)
Requirement already satisfied: pytz>=2017.2 in c:\users\mamoun.al-rawashdeh\anaconda3\lib\site-packages (from pandas->thermo) (2020.1)
Requirement already satisfied: six>=1.5 in c:\users\mamoun.al-rawashdeh\anaconda3\lib\site-packages (from python-dateutil>=2.7.3->pandas->thermo) (1.15.0)
```

## Cell 1.3 Upgrading thermo library

```
In [5]: pip install --upgrade pip
```

```
Requirement already satisfied: pip in c:\users\mamoun.al-rawashdeh\anaconda3\lib\site-packages (23.1.2)
Note: you may need to restart the kernel to use updated packages.
```

## Cell 1.4 Importing relevant thermo libraries

```
In [6]: from thermo.chemical import Chemical
from thermo.mixture import Mixture
```

## Cell 2.0 Reactor model

### Cell 2.1 Constants

```
In [7]: #OCM
Rh      = 1.9858775 * (10**-3) #kcal K-1 mol-1 #gas constant
Poh     = 1 #atm #initial pressure
Prefh   = Poh * 101325 #Pa #reference pressure
Toh     = 800 + 273 #K #initial temperature
Tr      = 298.15 #K #reference temperature
rhobh   = 0.345 #g/cm3 #catalyst bulk density

#MDA
Rc      = 101325*0.022414/273 #J/gmol.K #Gas constant
Poc     = 1 #atm #initial pressure
Prefc   = Poc * 101325 #Pa #reference pressure
Toc     = 800 + 273 #K #initial temperature
rhobc   = 0.510 #g/cm3 #catalyst bulk density
```

## Cell 2.2 General tube design parameters

```
In [8]: l       = 100 #cm #reactor length to design
ntubes  = 1000 #number of reactor tubes
```

```

lembdaw = 80/100 #J/cm.s.K #thermal conductivity of the wall for steel material
U       = 1200/10000 #W/m2.K #overall coefficient of heat transfer (division by 10000)

```

## Cell 2.3 Tube design parameters

In [9]:

```

#OCM
voht      = 10000 * (10**6)/3600 #cm3(STP)/s #inlet total volumetric flowrate
voh       = voht/ntubes #cm3(STP)/s #inlet volumetric flowrate per channel

w_h       = 0.12333*10**6 #g #catalyst weight
eps_gash  = 0.5 #gas void fraction
eps_sh    = 1 - eps_gash #solid fraction
percent_dilh = 0 #percentage of diluent to catalyst within the solid fraction (this is
eps_cath  = eps_sh*(1-percent_dilh) #= solid fraction *((solid fraction - diluent f
eps_diluenth = 1 - eps_gash - eps_cath # diluent percentage in total reactor (this is

vol_cat_particles = w_h/rhobh # volume of catalyst particles only
vol_Reac          = vol_cat_particles/eps_cath # volume of total reactor
vol_c              = eps_cath*vol_Reac # volume of catalyst only
vol_g              = eps_gash*vol_Reac # volume of gas only
vol_d              = eps_diluenth*vol_Reac # volume of diluent only
Vtot               = vol_c + vol_g + vol_d # total reactor volume by addition of gas+d

a_rh      = vol_Reac/1 #cm2 reactor cross sectional area
d_rh      = 2*((a_rh/np.pi)**0.5) #cm #total reactor diameter
a_th      = a_rh/ntubes #cm2 #cross sectional area of one tube
d_th      = (2*((a_th/np.pi)**0.5)) #cm #tube diameter used in reaction kinetics
vol_th   = a_th*1 #volume of one tube

#MDA
voct      = 72320* (10**6)/3600 #cm3(STP)/s total inlet flow rate
voc       = voct/ntubes #cm3(STP)/s inlet flow rte per channel

w_c       = 6.66667*10**6 #g #catalyst weight
eps_gasc  = 0.5 #gas void fraction
eps_sc    = 1 - eps_gasc #solid fraction
percent_dilc = 0 #percentage of diluent to catalyst within the solid fraction
eps_catc  = eps_sc*(1-percent_dilc) #= solid fraction *((solid fraction - diluent f
eps_diluentc = 1 - eps_gasc - eps_catc #diluent percentage in total reactor

vol_catc  = w_c/rhobc/eps_gasc #cm3 catalyst volume
vol_rc    = vol_catc/(1-eps_diluentc) #cm3 reactor volume based on new calculations

a_rc      = vol_rc/1 #cm2 reactor cross sectional area
d_rc      = 2*((a_rc/np.pi)**0.5) #cm #total reactor diameter
a_tc      = a_rc/ntubes #cm2 #cross sectional area of one tube
d_tc      = (2*((a_tc/np.pi)**0.5)) #cm #tube diameter used in reaction kinetics
vol_tc   = a_tc*1 #volume of one tube

# design parameters
dw        = 0.5 #cm #thickness of tube wall
dtu       = dw + d_th #cm #used in heat transfer
d_tco    = dtu + d_tc #cm #total diameter of inner tube
isar     = np.pi * d_th * 1 #cm2 #heat exchange surface area inside
osar     = np.pi * dtu * 1 #cm2 #heat exchange surface area outside
lsar     = (osar - isar)/(np.log(osar/isar)) #cm2 #log mean heat exchange surface

```

```

# OCM side
dph      = 0.3 #cm #spherical particle diameter bed side
rph      = dph/2 #cm #spherical particle radius bed side
volch    = (4/3)*np.pi*(rph**3) #cm3 #volume of catalyst particle
uh       = voh/a_th #cm3/s.cm2 reactor #superficial velocity
ghsv_h   = voh/vol_th #Flow Rate of gas passing in the reactor/volume of catalyst

# MDA side
dpc      = 0.3 #cm #spherical particle diameter heat exchanger side
rpc      = dpc/2 #cm #spherical particle radius heat exchanger side
volcc    = (4/3)*np.pi*(rpc**3) #cm3 #volume of catalyst particle
uc       = voc/a_tc #cm3/s.cm2 reactor #superficial velocity
ghsv_c   = voc/vol_tc #flow Rate of gas passing in the reactor/volume of catalyst

```

## Cell 2.4 Inlet composition and molar flow rates

In [10]:

```

#OCM
yCH4oh   = (1-0.178297381)*3/4 #initial CH4 composition
yO2oh    = (1-0.178297381)*1/4 #initial O2 composition
yH2oh    = 0 #initial H2O composition
yC2H6oh  = 0 #initial C2H6 composition
yCO2oh   = 0 #initial CO2 composition
yHeoh    = 0.178297381 #initial He composition

Ftotoh   = voh/22414 #mol/s #initial molar flowrate per channel
FCH4oh   = yCH4oh * Ftotoh #initial CH4 molar flowrate
FO2oh    = yO2oh * Ftotoh #initial O2 flowrate
FH20oh   = yH2oh * Ftotoh #initial H2O flowrate
FC2H6oh  = yC2H6oh * Ftotoh #initial C2H6 composition
FCO2oh   = yCO2oh * Ftotoh #initial CO2 composition
FHeoh    = yHeoh * Ftotoh #initial He composition

#MDA
yCH4oc   = 0.95 #initial CH4 composition
yHeoc    = 1 - yCH4oc #initial He composition
yC2H4oc  = 0 #initial C2H4 composition
yC6H6oc  = 0 #initial C6H6 composition
yC10H8oc = 0 #initial C10H8 composition
yH2oc    = 0 #initial H2 composition

Ftotoc   = voc/22414 #mol/s #initial molar flowrate per channel
FCH4oc   = yCH4oc * Ftotoc #initial CH4 molar flowrate
FHeoc    = yHeoc * Ftotoc #initial He flowrate

```

## Cell 2.5 Defining chemical names for thermodynamic properties

In [11]:

```

#OCM
methh   = Chemical('methane', Toh, Prefh)
oxyh    = Chemical ('oxygen', Toh, Prefh)
carbdh  = Chemical ('carbon dioxide', Toh, Prefh)
ethah   = Chemical ('ethane', Toh, Prefh)
wath    = Chemical ('water', Toh, Prefh)
helh    = Chemical ('helium', Toh, Prefh)
mh     = Mixture(['methane', 'oxygen', 'carbon dioxide', 'ethane','water', 'helium'],

```

```

#MDA
methc = Chemical('methane', Toc, Prefc)
ethc = Chemical ('ethylene', Toc, Prefc)
benzc = Chemical ('benzene', Toc, Prefc)
naphc = Chemical ('naphthalene', Toc, Prefc)
hydc = Chemical ('hydrogen', Toc, Prefc)
helc = Chemical ('helium', Toc, Prefc)
mc = Mixture(['methane', 'ethylene', 'benzene', 'naphthalene', 'hydrogen', 'helium']

```

## Cell 2.6 Heat of formation

In [12]:

```

#OCM
mHfh = methh.Hf #J/mol
oHfh = 0 #J/mol #zero since pure component
cdHfh = carbdh.Hf #J/mol
eaHfh = ethah.Hf #J/mol
wHfh = wath.Hf #J/mol
heHfh = 0 #J/mol

#MDA
mHfc = methc.Hf #J/mol
eHfc = ethc.Hf #J/mol
bHfc = benzc.Hf #J/mol
nHfc = naphc.Hf #J/mol
hHfc = 0 # #J/mol
heHfc = 0 #J/mol

```

## Cell 2.7 Specific heat

In [13]:

```

#OCM
mCph = methh.Cpm #J/mol.K
oCph = oxyh.Cpm #J/mol.K
cdCph = carbdh.Cpm #J/mol.K
eaCph = ethah.Cpm #J/mol.K
wCph = wath.Cpm #J/mol.K
heCph = helh.Cpm #J/mol.K

#MDA
mCpc = methc.Cpm #J/mol.K
eCpc = ethc.Cpm #J/mol.K
bCpc = benzc.Cpm #J/mol.K
nCpc = naphc.Cpm #J/mol.K
hCpc = hydc.Cpm #J/mol.K
heCpc = helc.Cpm #J/mol.K

```

## Cell 2.8 Gas mixture physical properties

In [14]:

```

#OCM
meuh = mh.mu * 10 #Pa.s to g/cm.s #viscosity
rhoh = mh.rho * 0.001 #kg/m3 to g/cm3 #density
Cph = mh.Cpm #J/mol.K #specific heat
Reh = dph*rhoh*uh/meuh #reynold's number
Cphn = mh.Cp/1000 #J/g.K #specific heat
lembdah = (mh.k)/100 #J/cm.s.K #thermal conductivity of the bed fluid
Prh = Cphn * meuh/lembdah #prandtl number

```

```
#MDA
meuc    = mc.mu * 10 #Pa.s to g/cm.s #viscosity
rhoc    = mc.rho * 0.001 #kg/m3 to g/cm3 #density
Cpc     = mc.Cpm #J/mol.K #specific heat
Rec     = dpc*rhoc*uc/meuc #reynold's number
Cpcn    = mc.Cp/1000 #J/g.K #specific heat
lembdac = (mc.k)/100 #J/cm.s.K #thermal conductivity of the bed fluid
Prc     = Cpcn * meuc/lembdac #prandtl number
```

## Cell 2.9 Model simulation parameters

```
In [15]: n      = 400 #number of iterations for the model simulation
z1     = np.linspace(0, 1, n) #cm #catalyst bed length points
gheath, rheatex, gheatc, zed, epch, epdh, epgh, epcc, epdc, epgc =[], [], [], [], [], [
```

## Cell 2.10 Reactor model

```
In [16]: #define reactor model
def model (F,z):

#####
#statements to make sure molar flow rates can not be smaller than zero
if F[0] < 0:
    F[0] = 0
if F[1] < 0:
    F[1] = 0
if F[2] < 0:
    F[2] = 0
if F[3] < 0:
    F[3] = 0
if F[4] < 0:
    F[4] = 0
if F[5] < 0:
    F[5] = 0
if F[6] < 0:
    F[6] = 0
if F[7] < 0:
    F[7] = 0
if F[8] < 0:
    F[8] = 0
if F[9] < 0:
    F[9] = 0
if F[10] < 0:
    F[10] = 0
if F[11] < 0:
    F[11] = 0
if F[12] < 0:
    F[12] = 0
if F[13] < 0:
    F[13] = 0
if F[14] < 0:
    F[14] = 0
if F[15] < 0:
    F[15] = 0
```

```

#statements to make sure no nan values in the solution
F[1]      = np.where(np.isnan(F[1]), 0, F[1])
F[2]      = np.where(np.isnan(F[2]), 0, F[2])
F[3]      = np.where(np.isnan(F[3]), 0, F[3])
F[4]      = np.where(np.isnan(F[4]), 0, F[4])
F[5]      = np.where(np.isnan(F[5]), 0, F[5])
F[6]      = np.where(np.isnan(F[6]), 0, F[6])
F[7]      = np.where(np.isnan(F[7]), 0, F[7])
F[8]      = np.where(np.isnan(F[8]), 0, F[8])
F[9]      = np.where(np.isnan(F[9]), 0, F[9])
F[10]     = np.where(np.isnan(F[10]), 0, F[10])
F[11]     = np.where(np.isnan(F[11]), 0, F[11])
F[12]     = np.where(np.isnan(F[12]), 0, F[12])
F[13]     = np.where(np.isnan(F[13]), 0, F[13])
F[14]     = np.where(np.isnan(F[14]), 0, F[14])
F[15]     = np.where(np.isnan(F[15]), 0, F[15])

#####
#OCM
#constants for rate equation
K1o      = 2.472 * 10**7 * np.exp(-49.64/(Rh*Toh)) #Ea units (kcal/mol)
K2o      = 10.10 * np.exp(-23.15/(Rh*Toh)) #Ea units (kcal/mol)
K3o      = 1.103 * 10**-3 * np.exp(-4.548/(Rh*Toh)) #Ea units (kcal/mol)
K4o      = 2.093 * 10**-4 * np.exp(27.94/(Rh*Toh)) #Ea units (kcal/mol)

K1      = K1o * np.exp((-49.64/Rh)*((1/F[12])-(1/Toh))) #Ea units (kcal/mol)
K2      = K2o * np.exp((-23.15/Rh)*((1/F[12])-(1/Toh))) #Ea units (kcal/mol)
K3      = K3o * np.exp((-4.548/Rh)*((1/F[12])-(1/Toh))) #Ea units (kcal/mol)
K4      = K4o * np.exp((27.94/Rh)*((1/F[12])-(1/Toh))) #Ea units (kcal/mol)

#total flowrate
FTh      = F[0]+F[1]+F[2]+F[3]+F[4]+F[5] #mol/s
vh       = FTh * 22414 #cm/s #total velocity

#compositions
yCH4h    = F[0]/FTh #CH4
yO2h     = F[1]/FTh #O2
yCO2h    = F[2]/FTh #CO2
yC2H6h   = F[3]/FTh #C2H6
yH2Oh   = F[4]/FTh #H2O
yHeh    = F[5]/FTh #He

#partial pressures
pCH4h    = F[14] * yCH4h #atm #CH4
pO2h     = F[14] * yO2h #atm #O2
pCO2h    = F[14] * yCO2h #atm #CO2
pC2h     = F[14] * yC2H6h #atm #C2H6
pH2Oh   = F[14] * yH2Oh #atm #O2
pHeh    = F[14] * yHeh #atm #He

#reactions #derived from stoichiometry of the rate equations
#step 1: CH4 + 2O2      -> CO2 + 2H2O
#step 2: 2CH4 + 0.5O2    -> C2H6 + H2O

#net rate Equations
#rate parameters
Z        = (K1 * ((pO2h)**0.5)) / ((K1 * ((pO2h)**0.5)) + (K1 * K2 * K4) + (K2 * p
So      = 2 / (((1 + (8 * Z * ((K2 * pCH4h)) / (K3 * ((pO2h)**1.25)))) ** 0.5) +
CPCT   = (K1 * (pO2h ** 0.5)) / ((K1 * (pO2h ** 0.5)) + (K1 * K2 * K4) + (K2 * (

```

```

#rate equations
frac = vol_c/l/ntubes           #FRACTION needed for calculating OCM rate of reaction
r1h      = frac * ((K3 * (p02h**1.251))/4)*(((1 + ((8 * K2 * (CPCT) * pCH4h) / (K
r2h      = frac * ((K3 * (p02h**1.251))/16)*(((1 + ((8 * K2 * (CPCT) * pCH4h) /
                                          
#delHro
delHro1h = (cdHfh + (2*wHfh)) - (mHfh + (2*oHfh)) #J/mol carbon dioxide
delHro2h = (eaHfh + (wHfh)) - ((2*mHfh) + (0.5*oHfh)) #J/mol ethane

SFCph     = (F[0]*mCph) + (F[1]*oCph) + (F[2]*cdCph) + (F[3]*eaCph) + (F[4]*wCph) +
                                          
#delCp
delCpr1h = (cdCph + (2*wCph)) - (mCph + (2*oCph)) #J/mol carbon dioxide
delCpr2h = (eaCph + (wCph)) - ((2*mCph) + (0.5*oCph)) #J/mol ethane

#delHr
delHr1h  = delHro1h + (delCpr1h*(F[12]-Tr)) #J/mol carbon dioxide
delHr2h  = delHro2h + (delCpr2h*(F[12]-Tr)) #J/mol ethane

#####
#MDA
#constants for rate equation
A1      = 500.3 #mole, h, g cat, atm
A2      = 7.837 #mole, h, g cat, atm
A3      = 0.420 #mole, h, g cat, atm

Ea1     = 65.58 * 1000 #J/mol
Ea2     = 30.84 * 1000 #J/mol
Ea3     = -19.74 * 1000 #J/mol

Kp1     = 3.219*10**-5 * np.exp(4.4184 * (49.92*10**3/ Rc) * ((1/973) - (1/F[13]))
Kp2     = 1.478*10**5 * np.exp(4.4184 * (-13.978*10**3/ Rc) * ((1/973) - (1/F[13])
Kp3     = 9.985*10**3 * np.exp(4.4184 * (-4.485*10**3/ Rc) * ((1/973) - (1/F[13]))

kf1o    = A1 * np.exp(-Ea1/(Rc*Toc)) #mole, h, g cat, atm
kf2o    = A2 * np.exp(-Ea2/(Rc*Toc)) #mole, h, g cat, atm
kf3o    = A3 * np.exp(-Ea3/(Rc*Toc)) #mole, h, g cat, atm

kf1     = kf1o * np.exp((-Ea1/Rc)*((1/F[13])-(1/Toc))) #mole, h, g cat, atm
kf2     = kf2o * np.exp((-Ea2/Rc)*((1/F[13])-(1/Toc))) #mole, h, g cat, atm
kf3     = kf3o * np.exp((-Ea3/Rc)*((1/F[13])-(1/Toc))) #mole, h, g cat, atm

#total flowrate
FTc     = F[6]+F[7]+F[8]+F[9]+F[10]+F[11] #mol/s
vc      = FTc * 22414 #cm/s #velocity

#compositions
yCH4c   = F[6]/FTc #CH4
yC2H4c  = F[7]/FTc #C2H4
yC6H6c  = F[8]/FTc #C6H6
yC10H8c = F[9]/FTc #C10H8
yH2c    = F[10]/FTc #H2
yHe    = F[11]/FTc #He

#partial pressures
pCH4c   = F[15] * yCH4c #atm #CH4

```

```

pC2H4c      = F[15] * yC2H4c #atm #C2H4
pC6H6c      = F[15] * yC6H6c #atm #C6H6
pC10H8c     = F[15] * yC10H8c #atm #C10H8
pH2c        = F[15] * yH2c #atm #H2
pHe          = F[15] * yHe #atm #He

#reactions
#Step 1: 2CH4 <-> C2H4 + 2H2
#Step 2: 3C2H4 <-> C6H6 + 3H2
#Step 3: C6H6 + 2C2H4 <-> C10H8 + 3H2

#Rate Equations
n1      = (pC2H4c*pH2c**2) / (10**-6+((pCH4c**2)*Kp1))
n2      = (pC6H6c*pH2c**3) / (10**-6+((pC2H4c**3)*Kp2))
n3      = (pC10H8c*pH2c**3) / (10**-6+(pC6H6c*(pC2H4c**2)*Kp3))

r1c     = (eps_catc) * rhobc * a_tc * kf1 * pCH4c * (1-n1)/3600 #mol/cm.s
r2c     = (eps_catc) * rhobc * a_tc * kf2 * pC2H4c * (1-n2)/3600 #mol/cm.s
r3c     = (eps_catc) * rhobc * a_tc * kf3 * pC2H4c * pC6H6c * (1-n3)/3600 #mol/cm.

#delHro
delHro1c = (eHfc + (2*hHfc)) - (2*mHfc) #J/mol ethylene
delHro2c = (bHfc + (3*hHfc)) - (3*eHfc) #J/mol benzene
delHro3c = (nHfc + (3*hHfc)) - (bHfc + (2*eHfc)) #J/mol naphthalene

SFCpc   = (F[6]*mCpc) + (F[7]*eCpc) + (F[8]*bCpc) + (F[9]*nCpc) + (F[10]*hCpc) + (
           (F[11]*oCpc) + (F[12]*dCpc) + (F[13]*lCpc) + (F[14]*tCpc) + (F[15]*cCpc) + (F[16]*vCpc) + (F[17]*sCpc) + (F[18]*pCpc) + (F[19]*uCpc) + (F[20]*yCpc) + (F[21]*gCpc) + (F[22]*rCpc) + (F[23]*wCpc) + (F[24]*xCpc) + (F[25]*zCpc) + (F[26]*nCpc) + (F[27]*hCpc) + (F[28]*eCpc) + (F[29]*bCpc) + (F[30]*tCpc) + (F[31]*cCpc) + (F[32]*vCpc) + (F[33]*yCpc) + (F[34]*gCpc) + (F[35]*rCpc) + (F[36]*wCpc) + (F[37]*xCpc) + (F[38]*zCpc) + (F[39]*nCpc) + (F[40]*hCpc) + (F[41]*eCpc) + (F[42]*bCpc) + (F[43]*tCpc) + (F[44]*cCpc) + (F[45]*vCpc) + (F[46]*yCpc) + (F[47]*gCpc) + (F[48]*rCpc) + (F[49]*wCpc) + (F[50]*xCpc) + (F[51]*zCpc) + (F[52]*nCpc) + (F[53]*hCpc) + (F[54]*eCpc) + (F[55]*bCpc) + (F[56]*tCpc) + (F[57]*cCpc) + (F[58]*vCpc) + (F[59]*yCpc) + (F[60]*gCpc) + (F[61]*rCpc) + (F[62]*wCpc) + (F[63]*xCpc) + (F[64]*zCpc) + (F[65]*nCpc) + (F[66]*hCpc) + (F[67]*eCpc) + (F[68]*bCpc) + (F[69]*tCpc) + (F[70]*cCpc) + (F[71]*vCpc) + (F[72]*yCpc) + (F[73]*gCpc) + (F[74]*rCpc) + (F[75]*wCpc) + (F[76]*xCpc) + (F[77]*zCpc) + (F[78]*nCpc) + (F[79]*hCpc) + (F[80]*eCpc) + (F[81]*bCpc) + (F[82]*tCpc) + (F[83]*cCpc) + (F[84]*vCpc) + (F[85]*yCpc) + (F[86]*gCpc) + (F[87]*rCpc) + (F[88]*wCpc) + (F[89]*xCpc) + (F[90]*zCpc) + (F[91]*nCpc) + (F[92]*hCpc) + (F[93]*eCpc) + (F[94]*bCpc) + (F[95]*tCpc) + (F[96]*cCpc) + (F[97]*vCpc) + (F[98]*yCpc) + (F[99]*gCpc) + (F[100]*rCpc) + (F[101]*wCpc) + (F[102]*xCpc) + (F[103]*zCpc) + (F[104]*nCpc) + (F[105]*hCpc) + (F[106]*eCpc) + (F[107]*bCpc) + (F[108]*tCpc) + (F[109]*cCpc) + (F[110]*vCpc) + (F[111]*yCpc) + (F[112]*gCpc) + (F[113]*rCpc) + (F[114]*wCpc) + (F[115]*xCpc) + (F[116]*zCpc) + (F[117]*nCpc) + (F[118]*hCpc) + (F[119]*eCpc) + (F[120]*bCpc) + (F[121]*tCpc) + (F[122]*cCpc) + (F[123]*vCpc) + (F[124]*yCpc) + (F[125]*gCpc) + (F[126]*rCpc) + (F[127]*wCpc) + (F[128]*xCpc) + (F[129]*zCpc) + (F[130]*nCpc) + (F[131]*hCpc) + (F[132]*eCpc) + (F[133]*bCpc) + (F[134]*tCpc) + (F[135]*cCpc) + (F[136]*vCpc) + (F[137]*yCpc) + (F[138]*gCpc) + (F[139]*rCpc) + (F[140]*wCpc) + (F[141]*xCpc) + (F[142]*zCpc) + (F[143]*nCpc) + (F[144]*hCpc) + (F[145]*eCpc) + (F[146]*bCpc) + (F[147]*tCpc) + (F[148]*cCpc) + (F[149]*vCpc) + (F[150]*yCpc) + (F[151]*gCpc) + (F[152]*rCpc) + (F[153]*wCpc) + (F[154]*xCpc) + (F[155]*zCpc) + (F[156]*nCpc) + (F[157]*hCpc) + (F[158]*eCpc) + (F[159]*bCpc) + (F[160]*tCpc) + (F[161]*cCpc) + (F[162]*vCpc) + (F[163]*yCpc) + (F[164]*gCpc) + (F[165]*rCpc) + (F[166]*wCpc) + (F[167]*xCpc) + (F[168]*zCpc) + (F[169]*nCpc) + (F[170]*hCpc) + (F[171]*eCpc) + (F[172]*bCpc) + (F[173]*tCpc) + (F[174]*cCpc) + (F[175]*vCpc) + (F[176]*yCpc) + (F[177]*gCpc) + (F[178]*rCpc) + (F[179]*wCpc) + (F[180]*xCpc) + (F[181]*zCpc) + (F[182]*nCpc) + (F[183]*hCpc) + (F[184]*eCpc) + (F[185]*bCpc) + (F[186]*tCpc) + (F[187]*cCpc) + (F[188]*vCpc) + (F[189]*yCpc) + (F[190]*gCpc) + (F[191]*rCpc) + (F[192]*wCpc) + (F[193]*xCpc) + (F[194]*zCpc) + (F[195]*nCpc) + (F[196]*hCpc) + (F[197]*eCpc) + (F[198]*bCpc) + (F[199]*tCpc) + (F[200]*cCpc) + (F[201]*vCpc) + (F[202]*yCpc) + (F[203]*gCpc) + (F[204]*rCpc) + (F[205]*wCpc) + (F[206]*xCpc) + (F[207]*zCpc) + (F[208]*nCpc) + (F[209]*hCpc) + (F[210]*eCpc) + (F[211]*bCpc) + (F[212]*tCpc) + (F[213]*cCpc) + (F[214]*vCpc) + (F[215]*yCpc) + (F[216]*gCpc) + (F[217]*rCpc) + (F[218]*wCpc) + (F[219]*xCpc) + (F[220]*zCpc) + (F[221]*nCpc) + (F[222]*hCpc) + (F[223]*eCpc) + (F[224]*bCpc) + (F[225]*tCpc) + (F[226]*cCpc) + (F[227]*vCpc) + (F[228]*yCpc) + (F[229]*gCpc) + (F[230]*rCpc) + (F[231]*wCpc) + (F[232]*xCpc) + (F[233]*zCpc) + (F[234]*nCpc) + (F[235]*hCpc) + (F[236]*eCpc) + (F[237]*bCpc) + (F[238]*tCpc) + (F[239]*cCpc) + (F[240]*vCpc) + (F[241]*yCpc) + (F[242]*gCpc) + (F[243]*rCpc) + (F[244]*wCpc) + (F[245]*xCpc) + (F[246]*zCpc) + (F[247]*nCpc) + (F[248]*hCpc) + (F[249]*eCpc) + (F[250]*bCpc) + (F[251]*tCpc) + (F[252]*cCpc) + (F[253]*vCpc) + (F[254]*yCpc) + (F[255]*gCpc) + (F[256]*rCpc) + (F[257]*wCpc) + (F[258]*xCpc) + (F[259]*zCpc) + (F[260]*nCpc) + (F[261]*hCpc) + (F[262]*eCpc) + (F[263]*bCpc) + (F[264]*tCpc) + (F[265]*cCpc) + (F[266]*vCpc) + (F[267]*yCpc) + (F[268]*gCpc) + (F[269]*rCpc) + (F[270]*wCpc) + (F[271]*xCpc) + (F[272]*zCpc) + (F[273]*nCpc) + (F[274]*hCpc) + (F[275]*eCpc) + (F[276]*bCpc) + (F[277]*tCpc) + (F[278]*cCpc) + (F[279]*vCpc) + (F[280]*yCpc) + (F[281]*gCpc) + (F[282]*rCpc) + (F[283]*wCpc) + (F[284]*xCpc) + (F[285]*zCpc) + (F[286]*nCpc) + (F[287]*hCpc) + (F[288]*eCpc) + (F[289]*bCpc) + (F[290]*tCpc) + (F[291]*cCpc) + (F[292]*vCpc) + (F[293]*yCpc) + (F[294]*gCpc) + (F[295]*rCpc) + (F[296]*wCpc) + (F[297]*xCpc) + (F[298]*zCpc) + (F[299]*nCpc) + (F[300]*hCpc) + (F[301]*eCpc) + (F[302]*bCpc) + (F[303]*tCpc) + (F[304]*cCpc) + (F[305]*vCpc) + (F[306]*yCpc) + (F[307]*gCpc) + (F[308]*rCpc) + (F[309]*wCpc) + (F[310]*xCpc) + (F[311]*zCpc) + (F[312]*nCpc) + (F[313]*hCpc) + (F[314]*eCpc) + (F[315]*bCpc) + (F[316]*tCpc) + (F[317]*cCpc) + (F[318]*vCpc) + (F[319]*yCpc) + (F[320]*gCpc) + (F[321]*rCpc) + (F[322]*wCpc) + (F[323]*xCpc) + (F[324]*zCpc) + (F[325]*nCpc) + (F[326]*hCpc) + (F[327]*eCpc) + (F[328]*bCpc) + (F[329]*tCpc) + (F[330]*cCpc) + (F[331]*vCpc) + (F[332]*yCpc) + (F[333]*gCpc) + (F[334]*rCpc) + (F[335]*wCpc) + (F[336]*xCpc) + (F[337]*zCpc) + (F[338]*nCpc) + (F[339]*hCpc) + (F[340]*eCpc) + (F[341]*bCpc) + (F[342]*tCpc) + (F[343]*cCpc) + (F[344]*vCpc) + (F[345]*yCpc) + (F[346]*gCpc) + (F[347]*rCpc) + (F[348]*wCpc) + (F[349]*xCpc) + (F[350]*zCpc) + (F[351]*nCpc) + (F[352]*hCpc) + (F[353]*eCpc) + (F[354]*bCpc) + (F[355]*tCpc) + (F[356]*cCpc) + (F[357]*vCpc) + (F[358]*yCpc) + (F[359]*gCpc) + (F[360]*rCpc) + (F[361]*wCpc) + (F[362]*xCpc) + (F[363]*zCpc) + (F[364]*nCpc) + (F[365]*hCpc) + (F[366]*eCpc) + (F[367]*bCpc) + (F[368]*tCpc) + (F[369]*cCpc) + (F[370]*vCpc) + (F[371]*yCpc) + (F[372]*gCpc) + (F[373]*rCpc) + (F[374]*wCpc) + (F[375]*xCpc) + (F[376]*zCpc) + (F[377]*nCpc) + (F[378]*hCpc) + (F[379]*eCpc) + (F[380]*bCpc) + (F[381]*tCpc) + (F[382]*cCpc) + (F[383]*vCpc) + (F[384]*yCpc) + (F[385]*gCpc) + (F[386]*rCpc) + (F[387]*wCpc) + (F[388]*xCpc) + (F[389]*zCpc) + (F[390]*nCpc) + (F[391]*hCpc) + (F[392]*eCpc) + (F[393]*bCpc) + (F[394]*tCpc) + (F[395]*cCpc) + (F[396]*vCpc) + (F[397]*yCpc) + (F[398]*gCpc) + (F[399]*rCpc) + (F[400]*wCpc) + (F[401]*xCpc) + (F[402]*zCpc) + (F[403]*nCpc) + (F[404]*hCpc) + (F[405]*eCpc) + (F[406]*bCpc) + (F[407]*tCpc) + (F[408]*cCpc) + (F[409]*vCpc) + (F[410]*yCpc) + (F[411]*gCpc) + (F[412]*rCpc) + (F[413]*wCpc) + (F[414]*xCpc) + (F[415]*zCpc) + (F[416]*nCpc) + (F[417]*hCpc) + (F[418]*eCpc) + (F[419]*bCpc) + (F[420]*tCpc) + (F[421]*cCpc) + (F[422]*vCpc) + (F[423]*yCpc) + (F[424]*gCpc) + (F[425]*rCpc) + (F[426]*wCpc) + (F[427]*xCpc) + (F[428]*zCpc) + (F[429]*nCpc) + (F[430]*hCpc) + (F[431]*eCpc) + (F[432]*bCpc) + (F[433]*tCpc) + (F[434]*cCpc) + (F[435]*vCpc) + (F[436]*yCpc) + (F[437]*gCpc) + (F[438]*rCpc) + (F[439]*wCpc) + (F[440]*xCpc) + (F[441]*zCpc) + (F[442]*nCpc) + (F[443]*hCpc) + (F[444]*eCpc) + (F[445]*bCpc) + (F[446]*tCpc) + (F[447]*cCpc) + (F[448]*vCpc) + (F[449]*yCpc) + (F[450]*gCpc) + (F[451]*rCpc) + (F[452]*wCpc) + (F[453]*xCpc) + (F[454]*zCpc) + (F[455]*nCpc) + (F[456]*hCpc) + (F[457]*eCpc) + (F[458]*bCpc) + (F[459]*tCpc) + (F[460]*cCpc) + (F[461]*vCpc) + (F[462]*yCpc) + (F[463]*gCpc) + (F[464]*rCpc) + (F[465]*wCpc) + (F[466]*xCpc) + (F[467]*zCpc) + (F[468]*nCpc) + (F[469]*hCpc) + (F[470]*eCpc) + (F[471]*bCpc) + (F[472]*tCpc) + (F[473]*cCpc) + (F[474]*vCpc) + (F[475]*yCpc) + (F[476]*gCpc) + (F[477]*rCpc) + (F[478]*wCpc) + (F[479]*xCpc) + (F[480]*zCpc) + (F[481]*nCpc) + (F[482]*hCpc) + (F[483]*eCpc) + (F[484]*bCpc) + (F[485]*tCpc) + (F[486]*cCpc) + (F[487]*vCpc) + (F[488]*yCpc) + (F[489]*gCpc) + (F[490]*rCpc) + (F[491]*wCpc) + (F[492]*xCpc) + (F[493]*zCpc) + (F[494]*nCpc) + (F[495]*hCpc) + (F[496]*eCpc) + (F[497]*bCpc) + (F[498]*tCpc) + (F[499]*cCpc) + (F[500]*vCpc) + (F[501]*yCpc) + (F[502]*gCpc) + (F[503]*rCpc) + (F[504]*wCpc) + (F[505]*xCpc) + (F[506]*zCpc) + (F[507]*nCpc) + (F[508]*hCpc) + (F[509]*eCpc) + (F[510]*bCpc) + (F[511]*tCpc) + (F[512]*cCpc) + (F[513]*vCpc) + (F[514]*yCpc) + (F[515]*gCpc) + (F[516]*rCpc) + (F[517]*wCpc) + (F[518]*xCpc) + (F[519]*zCpc) + (F[520]*nCpc) + (F[521]*hCpc) + (F[522]*eCpc) + (F[523]*bCpc) + (F[524]*tCpc) + (F[525]*cCpc) + (F[526]*vCpc) + (F[527]*yCpc) + (F[528]*gCpc) + (F[529]*rCpc) + (F[530]*wCpc) + (F[531]*xCpc) + (F[532]*zCpc) + (F[533]*nCpc) + (F[534]*hCpc) + (F[535]*eCpc) + (F[536]*bCpc) + (F[537]*tCpc) + (F[538]*cCpc) + (F[539]*vCpc) + (F[540]*yCpc) + (F[541]*gCpc) + (F[542]*rCpc) + (F[543]*wCpc) + (F[544]*xCpc) + (F[545]*zCpc) + (F[546]*nCpc) + (F[547]*hCpc) + (F[548]*eCpc) + (F[549]*bCpc) + (F[550]*tCpc) + (F[551]*cCpc) + (F[552]*vCpc) + (F[553]*yCpc) + (F[554]*gCpc) + (F[555]*rCpc) + (F[556]*wCpc) + (F[557]*xCpc) + (F[558]*zCpc) + (F[559]*nCpc) + (F[560]*hCpc) + (F[561]*eCpc) + (F[562]*bCpc) + (F[563]*tCpc) + (F[564]*cCpc) + (F[565]*vCpc) + (F[566]*yCpc) + (F[567]*gCpc) + (F[568]*rCpc) + (F[569]*wCpc) + (F[570]*xCpc) + (F[571]*zCpc) + (F[572]*nCpc) + (F[573]*hCpc) + (F[574]*eCpc) + (F[575]*bCpc) + (F[576]*tCpc) + (F[577]*cCpc) + (F[578]*vCpc) + (F[579]*yCpc) + (F[580]*gCpc) + (F[581]*rCpc) + (F[582]*wCpc) + (F[583]*xCpc) + (F[584]*zCpc) + (F[585]*nCpc) + (F[586]*hCpc) + (F[587]*eCpc) + (F[588]*bCpc) + (F[589]*tCpc) + (F[590]*cCpc) + (F[591]*vCpc) + (F[592]*yCpc) + (F[593]*gCpc) + (F[594]*rCpc) + (F[595]*wCpc) + (F[596]*xCpc) + (F[597]*zCpc) + (F[598]*nCpc) + (F[599]*hCpc) + (F[600]*eCpc) + (F[601]*bCpc) + (F[602]*tCpc) + (F[603]*cCpc) + (F[604]*vCpc) + (F[605]*yCpc) + (F[606]*gCpc) + (F[607]*rCpc) + (F[608]*wCpc) + (F[609]*xCpc) + (F[610]*zCpc) + (F[611]*nCpc) + (F[612]*hCpc) + (F[613]*eCpc) + (F[614]*bCpc) + (F[615]*tCpc) + (F[616]*cCpc) + (F[617]*vCpc) + (F[618]*yCpc) + (F[619]*gCpc) + (F[620]*rCpc) + (F[621]*wCpc) + (F[622]*xCpc) + (F[623]*zCpc) + (F[624]*nCpc) + (F[625]*hCpc) + (F[626]*eCpc) + (F[627]*bCpc) + (F[628]*tCpc) + (F[629]*cCpc) + (F[630]*vCpc) + (F[631]*yCpc) + (F[632]*gCpc) + (F[633]*rCpc) + (F[634]*wCpc) + (F[635]*xCpc) + (F[636]*zCpc) + (F[637]*nCpc) + (F[638]*hCpc) + (F[639]*eCpc) + (F[640]*bCpc) + (F[641]*tCpc) + (F[642]*cCpc) + (F[643]*vCpc) + (F[644]*yCpc) + (F[645]*gCpc) + (F[646]*rCpc) + (F[647]*wCpc) + (F[648]*xCpc) + (F[649]*zCpc) + (F[650]*nCpc) + (F[651]*hCpc) + (F[652]*eCpc) + (F[653]*bCpc) + (F[654]*tCpc) + (F[655]*cCpc) + (F[656]*vCpc) + (F[657]*yCpc) + (F[658]*gCpc) + (F[659]*rCpc) + (F[660]*wCpc) + (F[661]*xCpc) + (F[662]*zCpc) + (F[663]*nCpc) + (F[664]*hCpc) + (F[665]*eCpc) + (F[666]*bCpc) + (F[667]*tCpc) + (F[668]*cCpc) + (F[669]*vCpc) + (F[670]*yCpc) + (F[671]*gCpc) + (F[672]*rCpc) + (F[673]*wCpc) + (F[674]*xCpc) + (F[675]*zCpc) + (F[676]*nCpc) + (F[677]*hCpc) + (F[678]*eCpc) + (F[679]*bCpc) + (F[680]*tCpc) + (F[681]*cCpc) + (F[682]*vCpc) + (F[683]*yCpc) + (F[684]*gCpc) + (F[685]*rCpc) + (F[686]*wCpc) + (F[687]*xCpc) + (F[688]*zCpc) + (F[689]*nCpc) + (F[690]*hCpc) + (F[691]*eCpc) + (F[692]*bCpc) + (F[693]*tCpc) + (F[694]*cCpc) + (F[695]*vCpc) + (F[696]*yCpc) + (F[697]*gCpc) + (F[698]*rCpc) + (F[699]*wCpc) + (F[700]*xCpc) + (F[701]*zCpc) + (F[702]*nCpc) + (F[703]*hCpc) + (F[704]*eCpc) + (F[705]*bCpc) + (F[706]*tCpc) + (F[707]*cCpc) + (F[708]*vCpc) + (F[709]*yCpc) + (F[710]*gCpc) + (F[711]*rCpc) + (F[712]*wCpc) + (F[713]*xCpc) + (F[714]*zCpc) + (F[715]*nCpc) + (F[716]*hCpc) + (F[717]*eCpc) + (F[718]*bCpc) + (F[719]*tCpc) + (F[720]*cCpc) + (F[721]*vCpc) + (F[722]*yCpc) + (F[723]*gCpc) + (F[724]*rCpc) + (F[725]*wCpc) + (F[726]*xCpc) + (F[727]*zCpc) + (F[728]*nCpc) + (F[729]*hCpc) + (F[730]*eCpc) + (F[731]*bCpc) + (F[732]*tCpc) + (F[733]*cCpc) + (F[734]*vCpc) + (F[735]*yCpc) + (F[736]*gCpc) + (F[737]*rCpc) + (F[738]*wCpc) + (F[739]*xCpc) + (F[740]*zCpc) + (F[741]*nCpc) + (F[742]*hCpc) + (F[743]*eCpc) + (F[744]*bCpc) + (F[745]*tCpc) + (F[746]*cCpc) + (F[747]*vCpc) + (F[748]*yCpc) + (F[749]*gCpc) + (F[750]*rCpc) + (F[751]*wCpc) + (F[752]*xCpc) + (F[753]*zCpc) + (F[754]*nCpc) + (F[755]*hCpc) + (F[756]*eCpc) + (F[757]*bCpc) + (F[758]*tCpc) + (F[759]*cCpc) + (F[760]*vCpc) + (F[761]*yCpc) + (F[762]*gCpc) + (F[763]*rCpc) + (F[764]*wCpc) + (F[765]*xCpc) + (F[766]*zCpc) + (F[767]*nCpc) + (F[768]*hCpc) + (F[769]*eCpc) + (F[770]*bCpc) + (F[771]*tCpc) + (F[772]*cCpc) + (F[773]*vCpc) + (F[774]*yCpc) + (F[775]*gCpc) + (F[776]*rCpc) + (F[777]*wCpc) + (F[778]*xCpc) + (F[779]*zCpc) + (F[780]*nCpc) + (F[781]*hCpc) + (F[782]*eCpc) + (F[783]*bCpc) + (F[784]*tCpc) + (F[785]*cCpc) + (F[786]*vCpc) + (F[787]*yCpc) + (F[788]*gCpc) + (F[789]*rCpc) + (F[790]*wCpc) + (F[791]*xCpc) + (F[792]*zCpc) + (F[793]*nCpc) + (F[794]*hCpc) + (F[795]*eCpc) + (F[796]*bCpc) + (F[797]*tCpc) + (F[798]*cCpc) + (F[799]*vCpc) + (F[800]*yCpc) + (F[801]*gCpc) + (F[802]*rCpc) + (F[803]*wCpc) + (F[804]*xCpc) + (F[805]*zCpc) + (F[806]*nCpc) + (F[807]*hCpc) + (F[808]*eCpc) + (F[809]*bCpc) + (F[810]*tCpc) + (F[811]*cCpc) + (F[812]*vCpc) + (F[813]*yCpc) + (F[814]*gCpc) + (F[815]*rCpc) + (F[816]*wCpc) + (F[817]*xCpc) + (F[818]*zCpc) + (F[819]*nCpc) + (F[820]*hCpc) + (F[821]*eCpc) + (F[822]*bCpc) + (F[823]*tCpc) + (F[824]*cCpc) + (F[825]*vCpc) + (F[826]*yCpc) + (F[827]*gCpc) + (F[828]*rCpc) + (F[829]*wCpc) + (F[830]*xCpc) + (F[831]*zCpc) + (F[832]*nCpc) + (F[833]*hCpc) + (F[834]*eCpc) + (F[835]*bCpc) + (F[836]*tCpc) + (F[837]*cCpc) + (F[838]*vCpc) + (F[839]*yCpc) + (F[840]*gCpc) + (F[841]*rCpc) + (F[842]*wCpc) + (F[843]*xCpc) + (F[844]*zCpc) + (F[845]*nCpc) + (F[846]*hCpc) + (F[847]*eCpc) + (F[848]*bCpc) + (F[849]*tCpc) + (F[850]*cCpc) + (F[851]*vCpc) + (F[852]*yCpc) + (F[853]*gCpc) + (F[854]*rCpc) + (F[855]*wCpc) + (F[856]*xCpc) + (F[857]*zCpc) + (F[858]*nCpc) + (F[859]*hCpc) + (F[860]*eCpc) + (F[861]*bCpc) + (F[862]*tCpc) + (F[863]*cCpc) + (F[864]*vCpc) + (F[865]*yCpc) + (F[866]*gCpc) + (F[867]*rCpc) + (F[868]*wCpc) + (F[869]*xCpc) + (F[870]*zCpc) + (F[871]*nCpc) + (F[872]*hCpc) + (F[873]*eCpc) + (F[874]*bCpc) + (F[875]*tCpc) + (F[876]*cCpc) + (F[877]*vCpc) + (F[878]*yCpc) + (F[879]*gCpc) + (F[880]*rCpc) + (F[881]*wCpc) + (F[882]*xCpc) + (F[883]*zCpc) + (F[884]*nCpc) + (F[885]*hCpc) + (F[886]*eCpc) + (F[887]*bCpc) + (F[888]*tCpc) + (F[889]*cCpc) + (F[890]*vCpc) + (F[891]*yCpc) + (F[892]*gCpc) + (F[893]*rCpc) + (F[894]*wCpc) + (F[895]*xCpc) + (F[896]*zCpc) + (F[897]*nCpc) + (F[898]*hCpc) + (F[899]*eCpc) + (F[900]*bCpc) + (F[901]*tCpc) + (F[902]*cCpc) + (F[903]*vCpc) + (F[904]*yCpc) + (F[905]*gCpc) + (F[906]*rCpc) + (F[907]*wCpc) + (F[908]*xCpc) + (F[909]*zCpc) + (F[910]*nCpc) + (F[911]*hCpc) + (F[912]*eCpc) + (F[913]*bCpc) + (F[914]*tCpc) + (F[915]*cCpc) + (F[916]*vCpc)
```

```

#temperature profile
dTdz      = (Qgh - Qex) / SFCph #negative for exothermic
dTcdz     = (Qgc + Qex) / SFCpc #positive for endothermic

#pressure profile
dphdz     = -((1-eps_gash)*uh/(dph*eps_gash**3))*((150*meuh*(1-eps_gash)/dph)+(1.75
dpcdz     = -((1-eps_gasc)*uc/(dpc*eps_gasc**3))*((150*meuc*(1-eps_gasc)/dpc)+(1.75

#flowrate profile
dFdz      = [dFch4hdz, dFo2hdz, dFc2h6hdz, dFh2ohdz, dFhehdz, dFch4cdz, d

#appending values for result outputs
gheath.append(Qgh)
gheatc.append(Qgc/3600)
rheatex.append(Qex/3600)
zed.append(z)
epch.append(eps_cath)
epdh.append(eps_diluenth)
epgh.append(eps_gash)
epcc.append(eps_catc)
epdc.append(eps_diluentc)
epgc.append(eps_gasc)

#return model values
return dFdz

```

## Cell 2.11 Solve reactor model

```
In [17]: F0      = [FCH4oh, F02oh, 0, 0, 0, FHeoh, FCH4oc, 0, 0, 0, 0, FHeoc, Toh, Toc, Poh, Po
F1      = odeint(model,F0,z1) #reactor model
```

## Cell 2.12 Reactor outputs

```

In [18]: #OCM
x_ch4h = 100*(FCH4oh-F1[-1,0])/FCH4oh %% #CH4 conversion
mconvh = 100*(FCH4oh-F1[:,0]) / FCH4oh %% #CH4 conversion
x_o2h = 100*(F02oh-F1[-1,1])/F02oh %% #O2 conversion
oconvh = 100*(F02oh-F1[:,1]) / F02oh #O2 conversion %
s_c2h = 100 * (2*F1[-1,3])/(F1[0,0]-F1[-1,0]) %% #C2 selectivity
c2sh = 100 * (2*F1[1:,3])/(F1[0,0]-F1[1:,0]) %% #C2 selectivity
y_c2h = x_ch4h * s_c2h/100 %% #C2 yield
c2yh = x_ch4h * c2sh/100 %% #C2 yield
delPh = 100*(Poh-F1[-1,14])/Poh %% #pressure drop
Thpeak = max(F1[:,12]) - Toh %% #temperature peak
Thlower = Toh - min(F1[:,12]) %% #temperature dip
sdth = st.pstdev(F1[:,12]) #measures the amount of variation or dispersion of a set

#MDA
x_ch4c = 100*(FCH4oc-F1[-1,6])/FCH4oc %% #CH4 conversion
mconvc = 100*(FCH4oc-F1[:,6]) / FCH4oc %% #CH4 conversion
s_c6c = 100 * (2*F1[-1,8])/(F1[0,6]-F1[-1,6]) %% #C6 selectivity
c6sc = 100 * (2*F1[1:,8])/(F1[0,6]-F1[1:,6]) %% #C6 selectivity
y_c6c = x_ch4c * s_c6c/100 %% #C6 yield
c6yc = x_ch4c * c6sc/100 %% #C6 yield
delPc = 100*(Poc-F1[-1,15])/Poc %% #pressure drop
Tcpeak = max(F1[:,13]) - Toc %% #temperature peak

```

```

Tclower = Toc - min(F1[:,13]) #C #temperature dip
sdtc    = st.pstdev(F1[:,13]) #measures the amount of variation or dispersion of a set

```

## Cell 2.13 Printing reactor outputs

```
In [19]: print('eps_cath', eps_cath, 'eps_catc', eps_catc, 'dph', '%.1f' %(dph*10), 'dpc', '%.1f'
print('d_th','%.1f' %d_th, 'd_tc', '%.1f' %d_tc, 'd_tco', '%.1f' %d_tco, 'len', '%.1f'
print('delp_h %','%.3f' %delPh, 'delp_c %', '%.3f' %delPc)
print('X_CH4 ', '%.1f' %x_ch4h, 'X_O2 ', '%.1f' %x_o2h,'S_C2 ', '%.1f' %s_c2h, 'Y_C2 ',
print('X_CH4 ', '%.1f' %x_ch4c, 'S_C6 ', '%.1f' %s_c6c, 'Y_C6 ', '%.1f' %y_c6c)
print('Tpeak_H ', '%.1f' %Thpeak,'Tlower_H ', '%.1f' %Thlower, 'Tstd_H ', '%.1f' %sdth,
print('Tpeak_C ', '%.1f' %Tcpeak,'Tlower_C ', '%.1f' %Tclower, 'Tstd_C ', '%.1f' %sdth,
print('eps_diluenth', '%.3f' %eps_diluenth)
```

```
eps_cath 0.5 eps_catc 0.5 dph 3.0 dpc 3.0 d_rh 1.0 d_rc 5.8
d_th 3.0 d_tc 18.2 d_tco 21.8 len 100.0 tubes 1000
delp_h % 13.643 delp_c % 0.947
X_CH4 52.8 X_O2 100.0 S_C2 78.2 Y_C2 41.3 Uovl 1200.0
X_CH4 11.1 S_C6 18.4 Y_C6 2.0
Tpeak_H 84.3 Tlower_H 0.0 Tstd_H 20.7 ghsv_h 13987 sTh 0.3 uh 388.5
Tpeak_C 9.3 Tlower_C 60.3 Tstd_C 23.4 ghsv_c 2766 sTc 1.3 uc 76.8
eps_diluenth 0.000
```

## Cell 2.14 Plotting simulation results

```
In [20]: #pressure across the catalyst bed
plt.plot(z1/100,F1[:,14], '-', color='darkgray', label = r'OCM')
plt.plot(z1/100,F1[:,15], '--', color='dimgray', label = r'MDA')
plt.ylabel('Pressure, atm')
plt.xlabel('Catalyst bed length, m')
plt.grid(alpha=0.5)
plt.legend(loc='best')
plt.rcParams["figure.figsize"]=(8,5)
plt.show()

#temperature across the catalyst bed
plt.plot(z1/100,F1[:,12], '-', color='darkgray', label = r'OCM')
plt.plot(z1/100,F1[:,13], '--', color='dimgray', label = r'MDA')
plt.ylabel('Temperature, K')
plt.xlabel('Catalyst bed length, m')
plt.grid(alpha=0.5)
plt.legend(loc='upper right')
plt.rcParams["figure.figsize"]=(8,5)
plt.show()

#reactor performance % across the catalyst bed
plt.plot(z1/100,mconvh, '-', color='darkgray', label = r'OCM CH4 conversion')
plt.plot(z1/100,mconvc, ':', color='dimgray', label = r'MDA CH4 conversion')
plt.plot(z1[1:]/100,c2yh, '-.', color='darkgray', label = r'OCM C2 yield')
plt.plot(z1[1:]/100,c6yc, '--', color='dimgray', label = r'MDA C6 yield')
plt.ylabel('Reactor performance, %')
plt.xlabel('Catalyst bed length, m')
plt.grid(alpha=0.5)
plt.legend(loc='upper left')
plt.rcParams["figure.figsize"]=(8,5)
plt.show()

#solid fraction across OCM bed
```

```

plt.plot(np.array(zed)/100, np.multiply(epch,2), '--', color='darkgray', label = r'Catalyst bed length, m')
plt.plot(np.array(zed)/100, np.multiply(epdh,2), ':', color='dimgray', label = r'Diluent bed length, m')
plt.ylim(0,1)
plt.ylabel('Solid fraction')
plt.xlabel('OCM Catalyst bed length, m')
plt.grid(alpha=0.5)
plt.legend(loc='best')
plt.rcParams["figure.figsize"]=(8,5)
plt.show()

#solid fraction across MDA bed
plt.plot(np.array(zed)/100, np.multiply(epcc,2), '--', color='darkgray', label = r'Catalyst bed length, m')
plt.plot(np.array(zed)/100, np.multiply(epdc,2), ':', color='dimgray', label = r'Diluent bed length, m')
plt.ylim(0,1)
plt.ylabel('Solid fraction')
plt.xlabel('MDA Catalyst bed length, m')
plt.grid(alpha=0.5)
plt.legend(loc='best')
plt.rcParams["figure.figsize"]=(8,5)
plt.show()

```





