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 advance
#Across each line of the code, the respective unit along a short explanation is mention
#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\li
b\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\an
aconda3\lib\site-packages (from pandas->thermo) (2.8.1)
Requirement already satisfied: pytz>=2017.2 in c:\users\mamoun.al-rawashdeh\anaconda3\li
b\site-packages (from pandas->thermo) (2020.1)
Requirement already satisfied: six>=1.5 in c:\users\mamoun.al-rawashdeh\anaconda3\lib\si
te-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-pa
ckages (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 volumtric 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 oftotal 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/l #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*l #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
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/l #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*l #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 * l #cm2 #heat exchange surface area inside
osar          = np.pi * dtu * l #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
yH2Ooh    =  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
FH2Ooh    =  yH2Ooh * 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 simulatiom
z1       = np.linspace(0, l, 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):

    ###############################################################################

    #statments 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)**1.251)))) ** 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 reactio
r1h       = frac  * ((K3 * (pO2h**1.251))/4)*(((1 + ((8 * K2 * (CPCT) * pCH4h) / (K
r2h       = frac  * ((K3 * (pO2h**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
yHec      = 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
pHec      = F[15] * yHec #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) + (

#delCp
delCpr1c = (eCpc + (2*hCpc)) - (2*mCpc) #J/mol.K
delCpr2c = (bCpc + (3*hCpc)) - (3*eCpc) #J/mol.K
delCpr3c = (nCpc + (3*hCpc)) - (bCpc + (2*eCpc)) #J/mol.K

#delHr
delHr1c  =  delHro1c + (delCpr1c*(F[13]-Tr)) #J/mol ethylene
delHr2c  =  delHro2c + (delCpr2c*(F[13]-Tr)) #J/mol benzene
delHr3c  =  delHro3c + (delCpr3c*(F[13]-Tr)) #J/mol naphthalene

###############################################################################

#heat balance
Qgh      =  (-r1h * delHr1h) + (-r2h * delHr2h) #J/cm.s
Qex      =  (U*np.pi*dtu)*(F[12]-F[13]) #J/cm.s #(Uh*a*Ah)
Qgc      =  (-r1c * delHr1c) + (-r2c * delHr2c) + (-r3c * delHr3c) #J/cm.s

#net rates
#OCM
dFch4hdz  = -(r1h + (2*r2h)) #mol/cm.s
dFo2hdz   = -((2*r1h) + (0.5*r2h)) #mol/cm.s
dFco2hdz  = (r1h) #mol/cm.s
dFc2h6hdz = (r2h) #mol/cm.s
dFh2ohdz  = ((2*r1h) + (r2h)) #mol/cm.s
dFhehdz   = 0 #mol/cm.s

#MDA
dFch4cdz  = -2*r1c # mol/cm.s
dFc2h4cdz = r1c - (3*r2c) - (2*r3c) #mol/cm.s
dFc6h6cdz = r2c - r3c #mol/cm.s
dFc10h8cdz= r3c #mol/cm.s
dFh2cdz   = (2*r1c) + (3*r2c) + (3*r3c) #mol/cm.s
dFhecdz   = 0 #mol/cm.s
```

```
    #temperature profile
    dThdz      = (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, dFco2hdz, 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, FO2oh, 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*(FO2oh-F1[-1,1])/FO2oh #% #O2 conversion
oconvh    = 100*(FO2oh-F1[:,1]) / FO2oh #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 #C #temperature peak
Thlower   = Toh - min(F1[:,12]) #C #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 #C #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

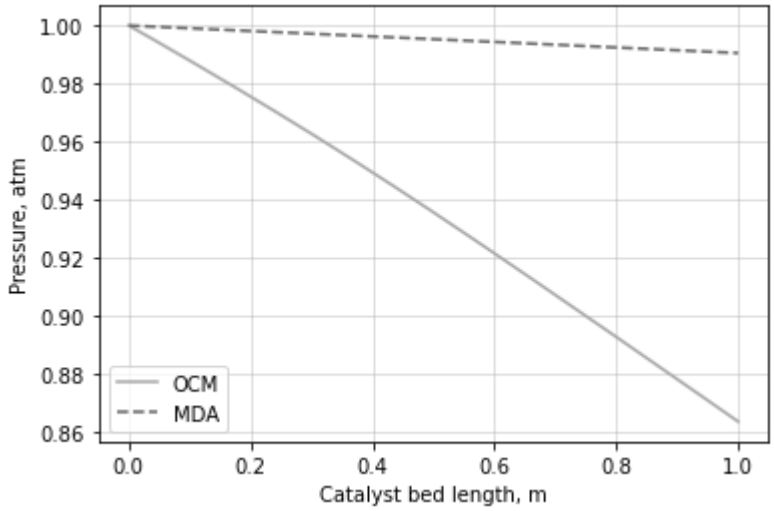
In [19]:
```
print('eps_cath', eps_cath, 'eps_catc', eps_catc, 'dph', '%.1f' %(dph*10), 'dpc', '%.1f
print('d_th','%.1f' %d_th, 'd_tc', '%.1f' %d_tc, 'd_tco', '%.1f' %d_tco, 'len',  '%.1f'
print('delp_h %','%.3f' %delPh, 'delp_c %', '%.3f' %delPc)
print('X_CH4 ', '%.1f' %x_ch4h, 'X_O2 ', '%.1f' %x_o2h,'S_C2 ', '%.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' %sdtc,
print('eps_diluenth', '%.3f' %eps_diluenth)
```

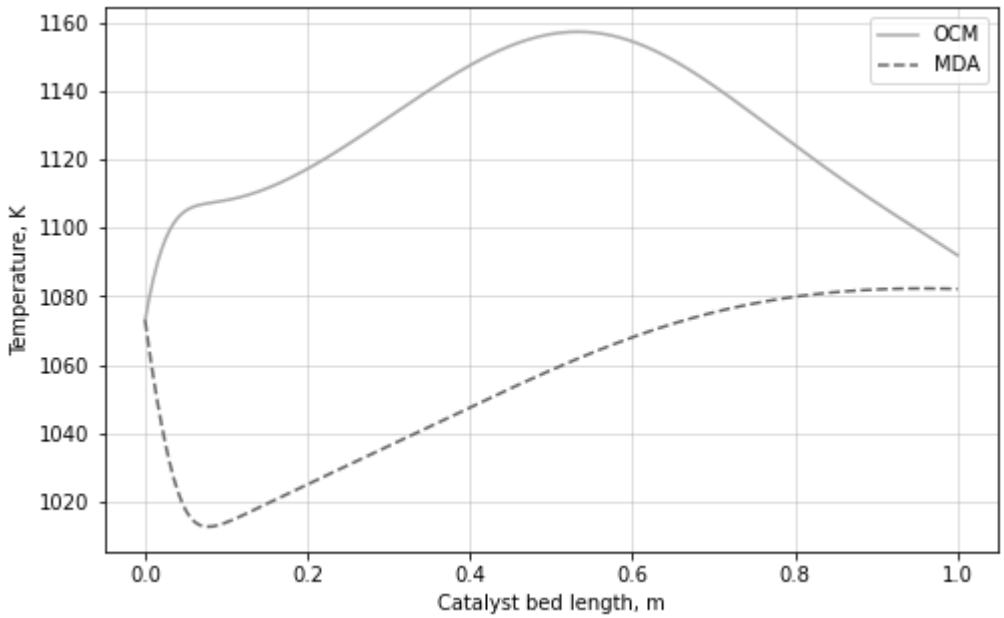
```
eps_cath 0.5 eps_catc 0.5 dph 3.0 dpc 3.0 d_rh 1.0 d_rc 5.8
d_th 3.0 d_tc 18.2 d_tco 21.8 len 100.0 tubes 1000
delp_h % 13.643 delp_c % 0.947
X_CH4  52.8 X_O2  100.0 S_C2  78.2 Y_C2  41.3 Uovl 1200.0
X_CH4  11.1 S_C6  18.4 Y_C6  2.0
Tpeak_H  84.3 Tlower_H  0.0 Tstd_H  20.7 ghsv_h 13987 sTh 0.3 uh 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
```

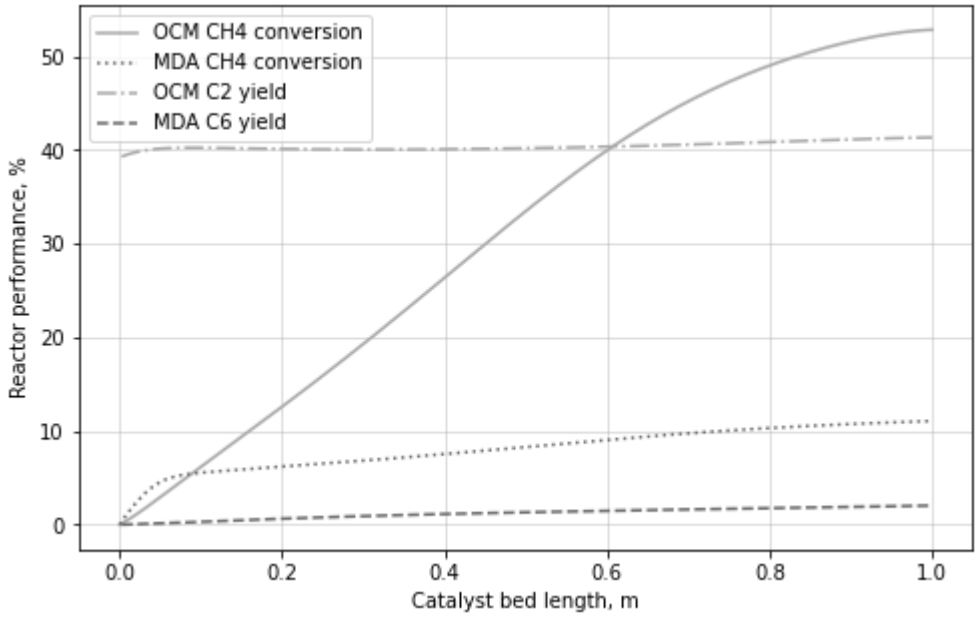
Cell 2.14 Plotting simulation results

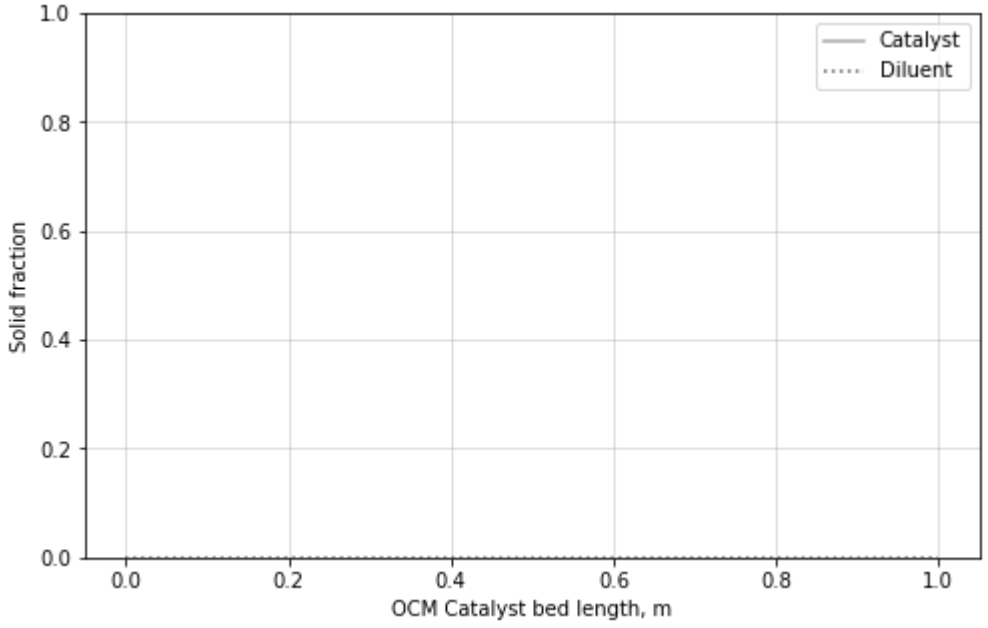
In [20]:
```
#pressure across the catalyst bed
plt.plot(z1/100,F1[:,14], '-', color='darkgray', label = r'OCM')
plt.plot(z1/100,F1[:,15], '--', color='dimgray', label = r'MDA')
plt.ylabel('Pressure, atm')
plt.xlabel('Catalyst bed length, m')
plt.grid(alpha=0.5)
plt.legend(loc='best')
plt.rcParams["figure.figsize"]=(8,5)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

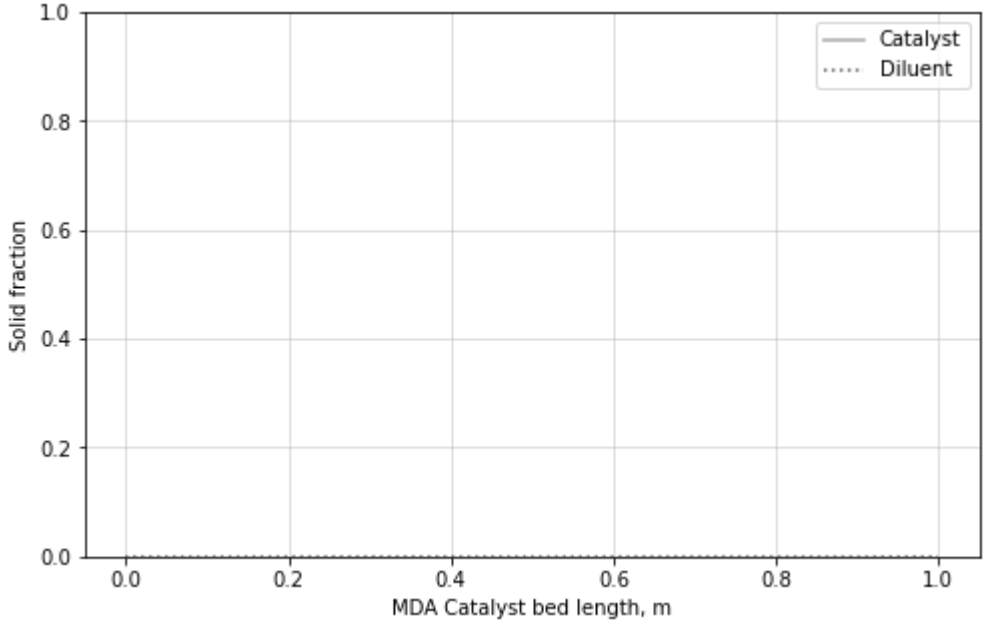
```
plt.plot(np.array(zed)/100, np.multiply(epch,2),  '-', color='darkgray', label = r'Cata
plt.plot(np.array(zed)/100, np.multiply(epdh,2),  ':', color='dimgray', label = r'Dilue
plt.ylim(0,1)
plt.ylabel('Solid fraction')
plt.xlabel('OCM Catalyst bed length, m')
plt.grid(alpha=0.5)
plt.legend(loc='best')
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'Catal
plt.plot(np.array(zed)/100, np.multiply(epdc,2), ':', color='dimgray', label = r'Diluen
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()
```






OCM CH4 conversion
MDA CH4 conversion
OCM C2 yield
MDA C6 yield
Reactor performance, %
Catalyst bed length, m


Catalyst
Diluent
Solid fraction
OCM Catalyst bed length, m


Catalyst
Diluent
Solid fraction
MDA Catalyst bed length, m