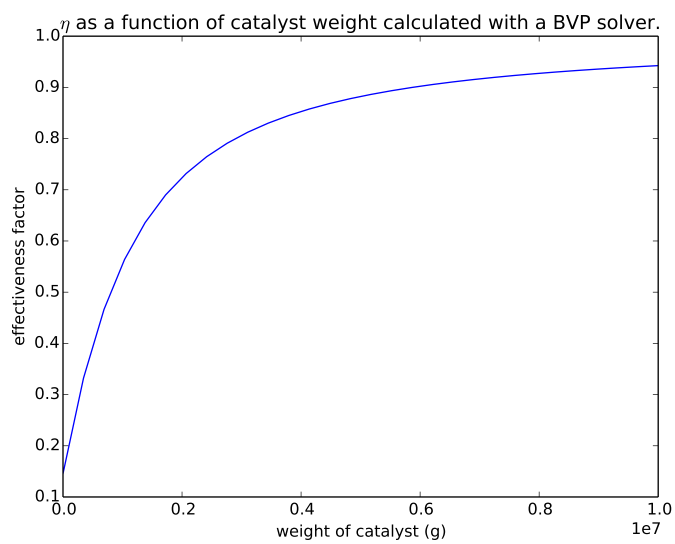
```python

#+NAME: Yang Song
#+ANDREWID: yangsong
#+COURSE: 06-625
#+ASSIGNMENT: thiele-2
'''The reaction A \rightarrow B occurs in a packed bed reactor. Given
this additional information:

r = k C_A**2     k = 2.25e5 cm**3/mol/s   (note this is a volumetric not mass based rate constant)
A is fed to the reactor at 10 mol/s
An inert gas is fed to the reactor at 10 mol/s
The total pressure is 4 atm, at 550 K.
d
The catalyst is a spherical pellet with a radius of 0.45 cm. the bed
density is 0.60 gm/cm**3, and the catalyst occupies 88% of the
volume. The effective diffusuivity of A is 0.008 cm**2/s.

The effective rate is given by:
reff = vol_fraction * eta * r

a) Estimate the mass of catalyst required to achieve 75\% conversion of A
b) Plot the effectiveness factor as a function of catalyst weight
'''

import numpy as np
from pycse import BVP_nl
from scipy.integrate import odeint
k = 2.25e5 # cm^3/mol/s

FA0 = 10.0 # mol/s

FI = 10.0 # mol/s

P = 4.0 # atm

T = 550.0 # K

R = 0.45 # cm

PA0 = P * FA0 / (FA0 + FI)
gas_const = 82.057 # atm*cm^3/mol/K

CA0 = PA0 / gas_const / T # mol/cm^3

v = (FI + FA0) * gas_const * T / P # cm^3/s, is a constant throughout the reactor.

phi = 1 - 0.88 # porosity

De = 0.008 # cm^2/s

a = R / 3.0 # reduced radius of catalyst pellet.
```

```python

roub = 0.6 # g/cm^3


# Method 1: Using BVP solver in pycse to solve /eta as a function of $C_{AS}$.

# Problem (a)

def eta_calc(CAS):
    R = 3.0
    r1 = 0.0
    r2 = R
    N = 100
    RBAR = np.linspace(r1, r2, N)
    thiele = np.sqrt(1.5* k * CAS * a ** 2 / De)
    p = 4
    c0 = 0.9
    init = c0 + (1 - c0) / R**p * RBAR**p
    def F(rbar, cbar, dcbardrbar):
        return -2.0 / rbar * dcbardrbar + 2.0 / (2.0 + 1.0) * thiele**2 * cbar
            **2.0

    def BCS(rbar, cbar):
        return [(cbar[1]-cbar[0])/(rbar[1]-rbar[0]), cbar[-1]-1]

    CBAR = BVP_nl(F, RBAR, BCS, init)
    eta = 1./9. * np.trapz(CBAR**2 * RBAR**2, RBAR)
    return eta

def dVdFA(V, FA):
    CAS = FA / v
    r = k * CAS ** 2.0
    eta = eta_calc(CAS)
    reff = (1-phi) * eta * r
    dVdFA = - 1.0 / reff
    return dVdFA

FAexit = (1 - 0.75) * FA0
FAspan = np.linspace(FA0, FAexit)
sol = odeint(dVdFA, 0, FAspan)
W = roub * sol[-1, 0]
print 'Mass of the catalyst required to achieve 75% conversion of A is {0} g.'
    .format(W)
```

```
Mass of the catalyst required to achieve 75% conversion of A is
    219646.687638 g.
```

```python

# Problem (b)

def dFAdV(FA, V):
    CAS = FA / v
    r = k * CAS ** 2.0
    eta = eta_calc(CAS)
    reff = (1-phi) * eta * r
```

```
    dFAdV = -reff
    return dFAdV

Wspan = np.linspace(0, 1e7, 30)
ETA = np.empty(Wspan.shape)
Vspan = Wspan / roub
FA = odeint(dFAdV, FA0, Vspan)
CAS = FA / v
for i, CASi in enumerate(CAS):
    ETA[i] = eta_calc(CASi)

import matplotlib.pyplot as plt
plt.figure()
plt.plot(Wspan, ETA)
plt.xlabel('weight of catalyst (g)')
plt.ylabel('effectiveness factor')
plt.title('η as a function of catalyst weight calculated with a BVP solver.')
plt.show()
```

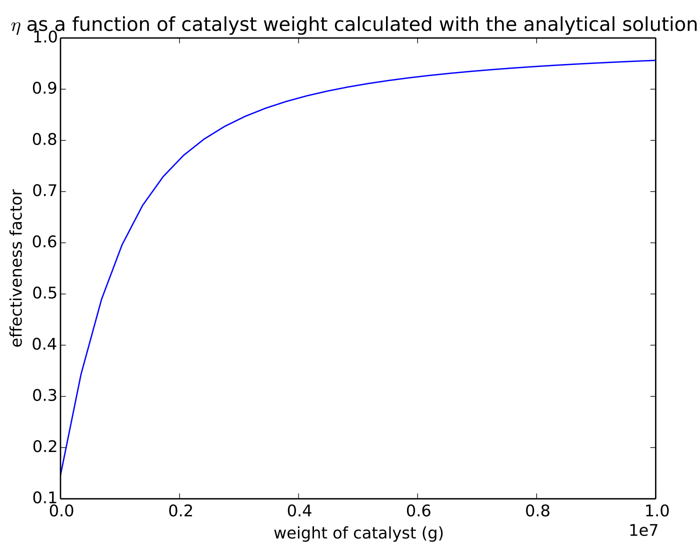



```

# Method 2: using the analytical solution of η for 1st-order, spherical particles.
# The error should be tolerable.

# Problem (a)
def dVdFA(V, FA):
    CAS = FA / v
    r = k * CAS ** 2.0
    thiele = np.sqrt(1.5 * k * CAS * a**2 / De)
    eta = 1.0 / thiele * (1.0 / np.tanh(3*thiele) - 1.0 / (3.0*thiele))
    reff = (1-phi) * eta * r
    dVdFA = - 1.0 / reff
    return dVdFA

FAexit = (1 - 0.75) * FA0
FAspan = np.linspace(FA0, FAexit)
```

```
sol = odeint(dVdFA, 0, FAspan)
W = roub * sol[-1, 0]
print 'Mass of the catalyst required to achieve 75% conversion of A is {0} g.'
    .format(W)
```

```
Mass of the catalyst required to achieve 75% conversion of A is
    216933.061881 g.
```

```

# Problem (b)
def dFAdV(FA, V):
    CAS = FA / v
    r = k * CAS ** 2.0
    thiele = np.sqrt(1.5 * k * CAS * a**2 / De)
    eta = 1.0 / thiele * (1.0 / np.tanh(3*thiele) - 1.0 / (3.0*thiele))
    reff = (1-phi) * eta * r
    dFAdV = -reff
    return dFAdV

Wspan = np.linspace(0, 1e7, 30)
ETA = np.empty(Wspan.shape)
Vspan = Wspan / roub
FA = odeint(dFAdV, FA0, Vspan)
CAS = FA / v
for i, CASi in enumerate(CAS):
    thiele = np.sqrt(1.5 * k * CASi * a**2 / De)
    ETA[i] = 1.0 / thiele * (1.0 / np.tanh(3*thiele) - 1.0 / (3.0*thiele))

import matplotlib.pyplot as plt
plt.figure()
plt.plot(Wspan, ETA)
plt.xlabel('weight of catalyst (g)')
plt.ylabel('effectiveness factor')
plt.title('$\eta$ as a function of catalyst weight calculated with the analytical
    solution.')
plt.show()
```



```python

# We can see the two methods yield very close results, indicating the BVP
    solver works well with this problem.
```

```
hostname: yangsong-PC.WV.CC.CMU.EDU
ipaddr:   128.237.170.64
mac:      145169596412
```

Press to select grade Grade: