Homework 5

Simon Bolding
NUEN 629

December 4, 2015

NUEN 629, Homework 5

Due Date Dec. 3

Solve the following problem and submit a detailed report, including a justification of why a reader should believe your results. 🙀

1 Clean Fusion Energy ¯_(ツ)_/¯

(100 points) Consider a thermonuclear fusion reactor producing neutrons of energy 14.1 and 2.45 MeV. The reactor is surrounded by FLiBe (a 2:1 mixture of LiF and BeF_2, `https://en.wikipedia.org/wiki/FLiBe`) to convert the neutron energy into heat. All the constituents in the FLiBe have their natural abundances. Using data from Janis (`https://www.oecd-nea.org/janis/`). Assume the total neutron flux is 10^{14} n/cm^2·s. Perform the following analyses.

1. (25 points) Write out the depletion (or in this case activation) chains that will occur in the system.
2. (50 points) Over a two-year cycle compute the inventory of nuclides in the system using two methods discussed in class. What is the maximum concentration of tritium?
3. (25 points) After discharging the FLiBe blanket, how long will it take until the material is less radioactive than Brazil nuts (444 Bq/kg, `http://www.orau.org/PTP/collection/consumer%20products/brazilnuts.htm`).

Solution 1-1:

I tracked all of the nuclides suggested in the document *FLiBE use in Fusion Reactors: an Initial Safety Assessment* by L.C Cadwallader and G.R. Longhurst. The decay paths from this document are given below.

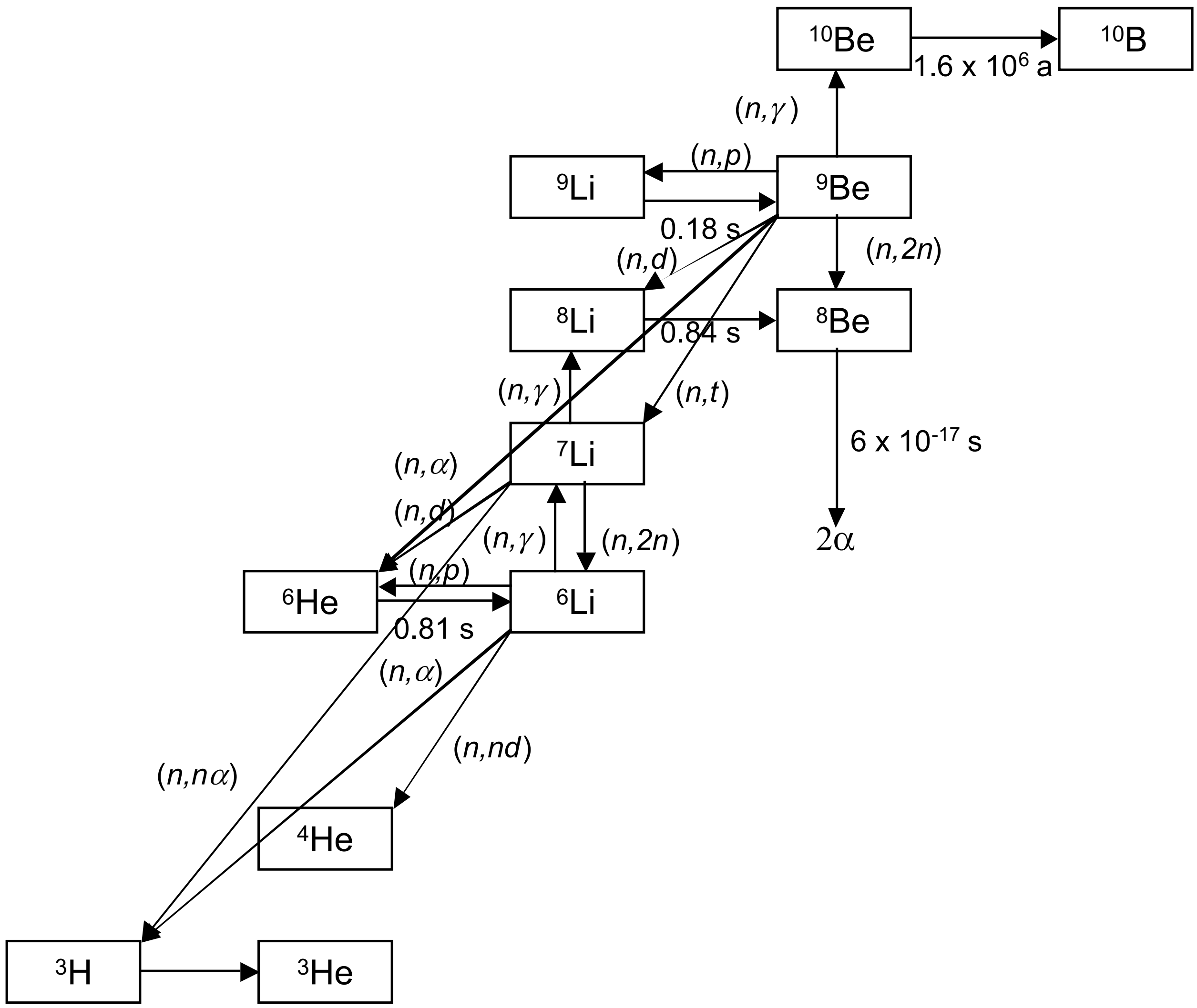


Figure 1. Activation and decay paths for lithium and beryllium in Flibe.[40]

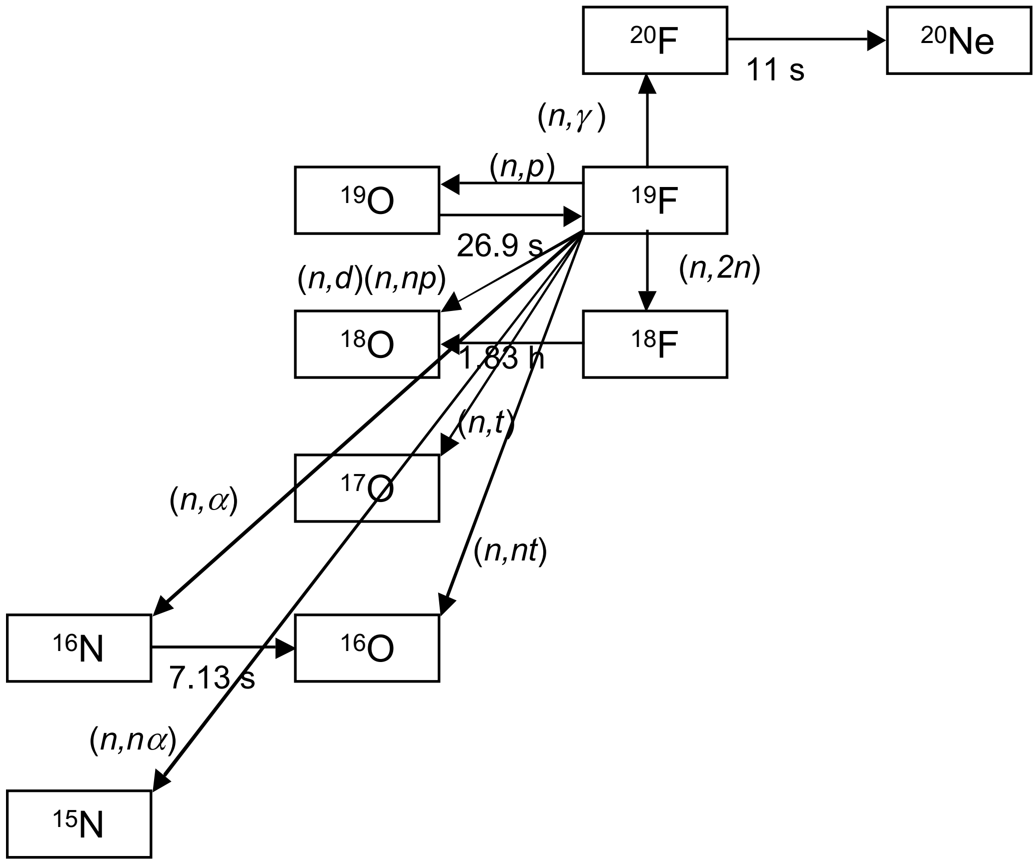


Figure 2. Activation and decay processes for fluorine in Flibe and oxygen impurity.[40]

Considering that stainless steel 316 with a Mn component (rather than nickel) might be present, erodants could be generated and activated [38], giving the following radioisotopes:

^{54}Mn	^{55}Fe	^{58}Co
^{56}Mn	^{59}Fe	^{60}Co

Some of these radioisotopes may oxidize in air, but concentrations should be low since the impurity concentrations were in the ppm range. The actual curie inventory of these radioisotopes depends on fluence and the level of impurity or erosion. For HYLIFE-II, Tobin [38] showed site boundary doses between 1 and 26 rem for each of the impurity and erodant radioisotopes listed above. The ^{18}F from pure Flibe gave a site boundary dose of 340 rem for HYLIFE-II. Considering that workers would be closer to the more concentrated source of these radioisotopes, their doses would be higher than the values stated for the site boundary unless appropriate mitigative actions are taken. Specific calculations need to be performed for the magnetic fusion designs under consideration to determine the worker and the site boundary doses. While these calculations are important, it is also noted that the MSRE operated successfully from 1965 to 1969 without any extreme personnel safety events (i.e., no radiation exposure over 15 rem, and

Solution 1-2:

The relative abundances of each isotope is computed assuming natural abundances and the chemical makeup of Li_2BeF_4. The initial atom fractions are

$$\vec{n_0} = \begin{pmatrix} ^6\text{Li} \\ ^7\text{Li} \\ ^9\text{Be} \\ ^{19}\text{F} \end{pmatrix} = \begin{pmatrix} 0.0214 \\ 0.2643 \\ 0.1429 \\ 0.5714 \end{pmatrix} \tag{1}$$

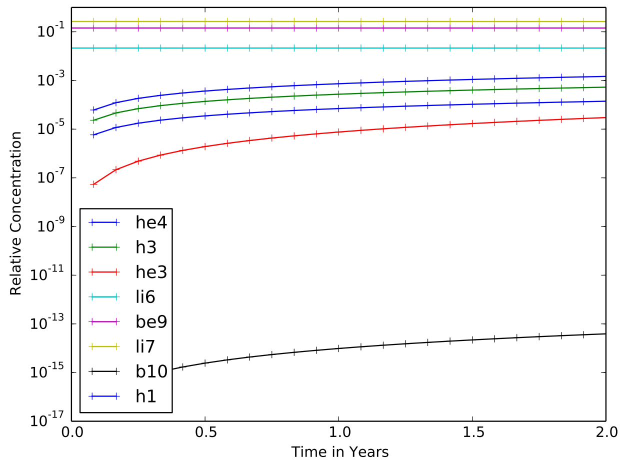
with the rest of the tracked isotopes from part 1 set to zero. The neutron flux was assumed to be 90% at 14.1 MeV and 10% at 2.45 MeV. The blanket is assumed thin enough that scattering is negligible, so neutrons remain at these two energies, which is fairly inaccurate as Be is a strong moderator. The invetory of nuclides in the system was computed using a parabolic contour integral approximation to the exponential matrix and with backward Euler. The backward Euler was implemented by taking 100 time steps between each data point, using the previous data point as the initial conditon. With this approach the results were found to agree to with at least two digits of accuracy. All tritium (h1) and α (he4) production are tracked as well. The plots of isotopic atom fractions are given below. The mass of tritium per initial kg of FLiBe after two years of irradiation is computed by dividing the atom fraction by the ratio of atomic masses; this ratio was found to be $\boxed{\text{0.017 kg of tritium per initial kg of FLiBe}}$.

Solution 1-3:

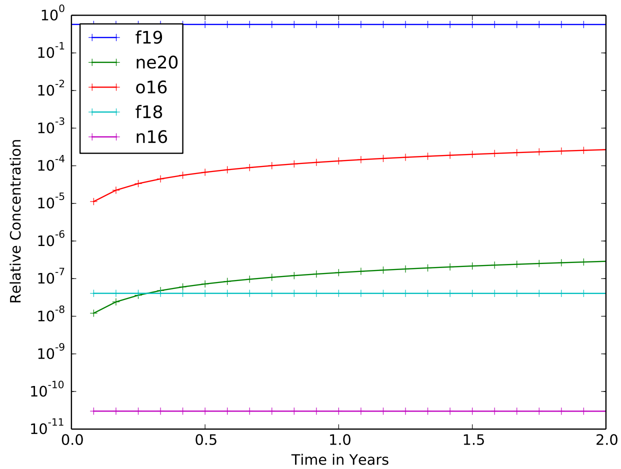
The activity, per unit mass of FLiBe, uted as

$$A = \sum_{j=1}^{N} \frac{\gamma_j(t) N_A}{\mathcal{A_{FLiBe}}} \lambda_j \tag{2}$$

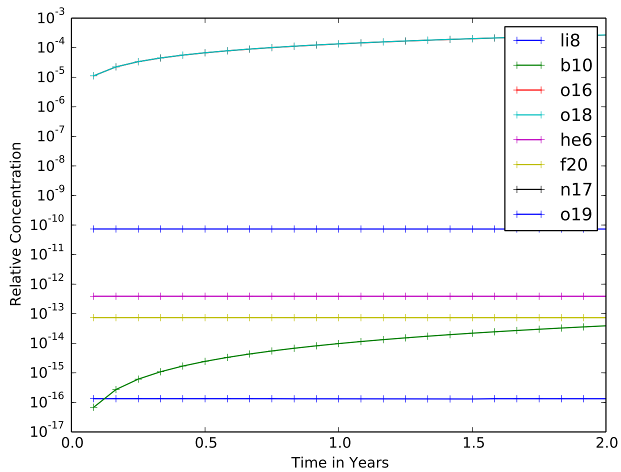
where γ_j is the atom fraction for the j-th of N isotopes. For simplicity, the flux is set to zero and the matrix exponential method is used to track the activity until it is below the threshold of 444 Bq.



Relative Concentration
Time in Years
he4
h3
he3
li6
be9
li7
b10
h1



Relative Concentration
Time in Years
f19
ne20
o16
f18
n16



Relative Concentration
Time in Years
li8
b10
o16
o18
he6
f20
n17
o19

Code

```
import numpy as np
import scipy as sp
import scipy.sparse as sparse
import scipy.sparse.linalg as splinalg
import matplotlib.pyplot as plt
import re

# In[107]:

N = 14
z = [5.623151880088747 + 1.194068309420004j,
       5.089353593691644 + 3.588821962583661j,
       3.993376923428209 + 6.004828584136945j,
       2.269789514323265 + 8.461734043748510j,
      -0.208754946413353 +10.991254996068200j,
      -3.703276086329081 +13.656363257468552j,
      -8.897786521056833 +16.630973240336562j]
c = [ -0.278754565727894 - 1.021482174078080j,
       0.469337296545605 + 0.456439548888464j,
      -0.234984158551045 - 0.058083433458861j,
       0.048071353323537 - 0.013210030313639j,
      -0.003763599179114 + 0.003351864962866j,
       0.000094388997996 - 0.000171848578807j,
      -0.000000715408253 + 0.000001436094999j,]


## Depletion Example

# In[84]:

#cross-sections in barns
ids = ['h1','h3','he3','he4','he6','li6','li7','li8','li9',
             'be9','be10','b10','f19','f20','ne20','f19','f18',
             'o16','n16','n17','o19','o18']
pos = {}
for i in range(len(ids)):
    pos[ids[i]]=i


stoa = 1./31557600

siga = {'h1':[0.0,0.0],
         'h3':[0.0,0.98287-0.93317],
         'he4':[0.0,0.0],
         'he3':[3.090702-2.364785,1.17-0.97],
         'he6':[0.0,0.0],
         'li6':[1.5421-1.27645,1.448249-0.861308],
         'li7':[1.920451-1.733316,1.44098-1.01017],
         'li8':[0.0,0.0],
         'li9':[0.0,0.0],
         'be9':[2.510747-2.401078,1.52753-1.10746],
         'be10':[0.0,0.0],
         'b10':[2.05-1.6767,1.4676-0.907458],
         'f19':[3.040797-2.134332,1.76351-0.9528]
         }

#Missing ones to zero
for i in ids:
    if i not in siga:
        siga[i] = [0.0,0.0]

cap = {'li6':['li7',1.106851e-5,1.017352e-5],
        'li7':['li8',4.670126e-6,4.1075e-6],
        'be9':['be10',1.e-4,1.e-4],
        'f19':['f20', 8.641807e-5,3.495e-5]          }
```

```
n2n = {'li7':['li6',0.0,0.03174],
        'be9':['he4-he4',0.0255,0.486034],
        'f19':['f18',0,0.04162]         }

nalpha = {'li7':['h3-he4',0.0,0.30215],
           'li6':['h3-he4',0.206155,0.025975],
           'be9':['he6-he4',0.090833,0.0105],
           'f19':['he4-n16',2.551667e-5,0.028393]      }

ntrit  = {'be9':['li7-h3',0.0,0.02025],
           'f19':['o16-h3',0.0,0.01303],}

nprot = {'li6':['he6-h1',0.0,0.0359],
          'be9':['li9-h1',0.0,0.02025],
          'f19':['o19-h1',0.0,0.018438]}


decays = {'h3':['he3',12.5],
           'he6':['li6',0.81*stoa],
           'li8':['he4-he4',0.84*stoa],
           'li9':['be9',0.5*0.18*stoa,'he4-he4',0.5*0.18*stoa],
           'be10':['b10',1.6E6],
           'f18':['o18',6586*stoa],
           'f20':['ne20',11.16*stoa],
           'o19':['f19',26.9*stoa],
           'n16':['o16',7.13*stoa]}

#Convert halflifes to decays
for i in decays:
    for j in range(0,len(decays[i]),2):
        decays[i][j+1] = np.log(2)/decays[i][j+1]

A = np.zeros((len(ids),len(ids)))

phi = 1.0e14 * 60 * 60 * 24 * 365  #10^14 1/cm^2/s in 1/cm^2/year
phi = phi*1.0e-24 # neutrons/barns-year
phi1 = 0.1*phi
phi2 = 0.9*phi
for i in ids:
    row = pos[i]
    A[row,row] =- phi1*siga[i][0] - phi2*siga[i][1]
    if i in decays:
        #sum over branching ratios
        A[row,row] -= sum(decays[i][j+1] for j in range(0,len(decays[i]),2))

    #Loop over all reaction types
    for r in [cap,n2n,nalpha,ntrit,nprot]:
        if i in r:
            target = r[i][0].split("-")
            for t in target: #from i to target
                A[pos[t],row] += phi1*r[i][1] + phi2*r[i][2]

    #Loop over decays
    if i in decays:
        for j in range(0,len(decays[i]),2): #in sets of 2
            #the first member is what it decays to, second is decay constant
            target = decays[i][0+j].split("-") #if goes to two things, hyphen
            for t in target:
                A[pos[t],row] += decays[i][1+j] #A[target,src] = decay




#Initial condition
n0 = np.zeros(len(ids))
abund = {'li6':(0.075*2),
          'li7':(0.925*2),
```

```
          'be9':1.,
          'f19':4.}
for i in ids:
    if i in abund:
        n0[pos[i]] = abund[i]

n0 /= sum(n0)


from scipy.linalg import expm

Npoints = (12,) #(2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32)
times = np.linspace(0,2,num=25) #in years
conc_exp = np.zeros((times.shape[0],n0.shape[0]))
conc_be  = np.zeros((times.shape[0],n0.shape[0]))

for ti in range(times.shape[0]):
    t = times[ti]
    n = n0.copy()
    for N in Npoints:
        pos1 = 0
        theta = np.pi*np.arange(1,N,2)/N
        z = N*(.1309 - 0.1194*theta**2 + .2500j*theta)
        w = N*(- 2*0.1194*theta + .2500j)
        c = 1.0j/N*np.exp(z)*w
        #plt.plot(np.real(z),np.imag(z),'o-')
        #plt.show()
        u = np.zeros(len(n))
        for k in range(int(N/2)):
            n,code = splinalg.gmres(z[k]*sparse.identity(len(n))  - A*t,n0, tol=1e-12,
                    maxiter=20000)
            if (code):
                print(code)
            u -= c[k]*n
        u = 2*np.real(u)
    conc_exp[ti,:] = u

    #Backward euler
    T = 100
    if ti == 0 :
        dt = 0.0
        n = n0.copy()
    else:
        dt = (t - times[ti-1])/T
        n = conc_be[ti-1,:].copy()

    I = sparse.identity(len(n0))
    for i in range(T):
#     print("Iteration", i)
        n = splinalg.gmres(I - A*dt, n,tol=1e-12)[0]
    conc_be[ti,:] = n

plot1 = ['he4','h3','he3','li6','be9','li7','b10','h1']
plot2 = ['f19','ne20','o16','f18','n16']
plot3 = ['li8','b10','o16','o18','he6','f20','n17','o19']

A_trit = 3.0160492
A_flibe = 18.99*4 + 6.94 * 2 + 9.01
print("Mass ratio of tritium", conc_exp[-1,pos['h3']]*A_flibe/A_trit)

#Plot concentrations

plt.figure()
for i in plot1:
    plt.semilogy(times,conc_be[:,pos[i]],"-+",label=i)
plt.legend(loc='best')
plt.xlabel("Time in Years")
plt.ylabel("Relative Concentration")
```

```
plt.savefig('p1.pdf',bbox_inches='tight')

plt.figure()
for i in plot2:
    plt.semilogy(times,conc_be[:,pos[i]],"-+",label=i)
plt.legend(loc='best')
plt.xlabel("Time in Years")
plt.ylabel("Relative Concentration")
plt.savefig('p2.pdf',bbox_inches='tight')

plt.figure()
for i in plot3:
    plt.semilogy(times,conc_be[:,pos[i]],"-+",label=i)
plt.legend(loc='best')
plt.xlabel("Time in Years")
plt.ylabel("Relative Concentration")
plt.savefig('p3.pdf',bbox_inches='tight')


#Compute activities
n = conc_be[-1,:]

#Rebuild A with no flux

while true:

    t = times[ti]
    n = n0.copy()
    for N in Npoints:
        pos1 = 0
        theta = np.pi*np.arange(1,N,2)/N
        z = N*(.1309 - 0.1194*theta**2 + .2500j*theta)
        w = N*(- 2*0.1194*theta + .2500j)
        c = 1.0j/N*np.exp(z)*w
        #plt.plot(np.real(z),np.imag(z),'o-')
        #plt.show()
        u = np.zeros(len(n))
        for k in range(int(N/2)):
            n,code = splinalg.gmres(z[k]*sparse.identity(len(n))  - A*t,n0, tol=1e-12,
                    maxiter=20000)
            if (code):
                print(code)
            u -= c[k]*n
        u = 2*np.real(u)
    conc_exp[ti,:] = u

    #Backward euler
    T = 100
    if ti == 0 :
        dt = 0.0
        n = n0.copy()
    else:
        dt = (t - times[ti-1])/T
        n = conc_be[ti-1,:].copy()

    I = sparse.identity(len(n0))
    for i in range(T):
#     print("Iteration", i)
        n = splinalg.gmres(I - A*dt, n,tol=1e-12)[0]
    conc_be[ti,:] = n
```