In [11]:
```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.cm as cm
from scipy.integrate import solve_ivp
from scipy.integrate import ode
import time
from numba import njit
m = open('ERK_model.net','r')
```

In [12]:

```
mLines = m.readlines()

for i in range(len(mLines)):

    if ('begin' in mLines[i])&('parameters' in mLines[i]):
        parInit = i+1

    if ('end' in mLines[i])&('parameters' in mLines[i]):
        parEnd = i

    if ('begin' in mLines[i])&('reactions' in mLines[i]):
        reactInit = i+1

    if ('end' in mLines[i])&('reactions' in mLines[i]):
        reactEnd = i

    if ('begin' in mLines[i])&('species' in mLines[i]):
        speciesInit = i+1

    if ('end' in mLines[i])&('species' in mLines[i]):
        speciesEnd = i

parLines = mLines[parInit:parEnd]
lines = mLines[reactInit:reactEnd]
speciesLines = mLines[speciesInit:speciesEnd]
numPars = len(parLines)

for parID in range(len(parLines)):

    parLines[parID]=parLines[parID].replace("^","**")

    separated= parLines[parID].split(' ')

    for index in range(len(separated)-1,-1,-1):

        if len(separated[index])==0:
            del separated[index]

    exec(separated[1]+'='+separated[2])

numSpecies = len(speciesLines)
IC= np.zeros((numSpecies,))

for speciesID in range(numSpecies):

    separated= speciesLines[speciesID].split(' ')

    for index in range(len(separated)-1,-1,-1):

        if len(separated[index])==0:
            del separated[index]

    exec('IC[speciesID]='+separated[2])


reactants = np.zeros((len(lines),10))
products = np.zeros((len(lines),10))

rates = np.zeros((len(lines), ))
```

```
for reactionID in range(len(lines)):

    separated= lines[reactionID].split(' ')

    for index in range(len(separated)-1,-1,-1):

        if len(separated[index])==0:
            del separated[index]

    reactantSet = separated[1].split(',')
    reactants[reactionID][0]=len(reactantSet)

    for reactantID in range(len(reactantSet)):
        reactants[reactionID][reactantID+1] = int(reactantSet[reactantID])-1

    productSet = separated[2].split(',')
    products[reactionID][0]=len(productSet)

    for productID in range(len(productSet)):
        products[reactionID][productID+1] = int(productSet[productID])-1

    rates[reactionID]=eval(separated[3])


numReactions = len(rates)

minNum = 20
dimerizationIndex = np.zeros((numReactions,))
rdimerizationIndex = np.zeros((numReactions,))

for i in range(len(rates)):

    if minNum > reactants[i][0]:

        minNum = reactants[i][0]

    if (reactants[i][0]==2)&(reactants[i][1]==reactants[i][2]):
        print('reaction ID='+str(i)+' is dimerization!!')
        print(asarray(reactants[i])+1)
        print(asarray(products[i])+1)

        dimerizationIndex[i]=1.

reactants=reactants.astype('int')
products=products.astype('int')
```

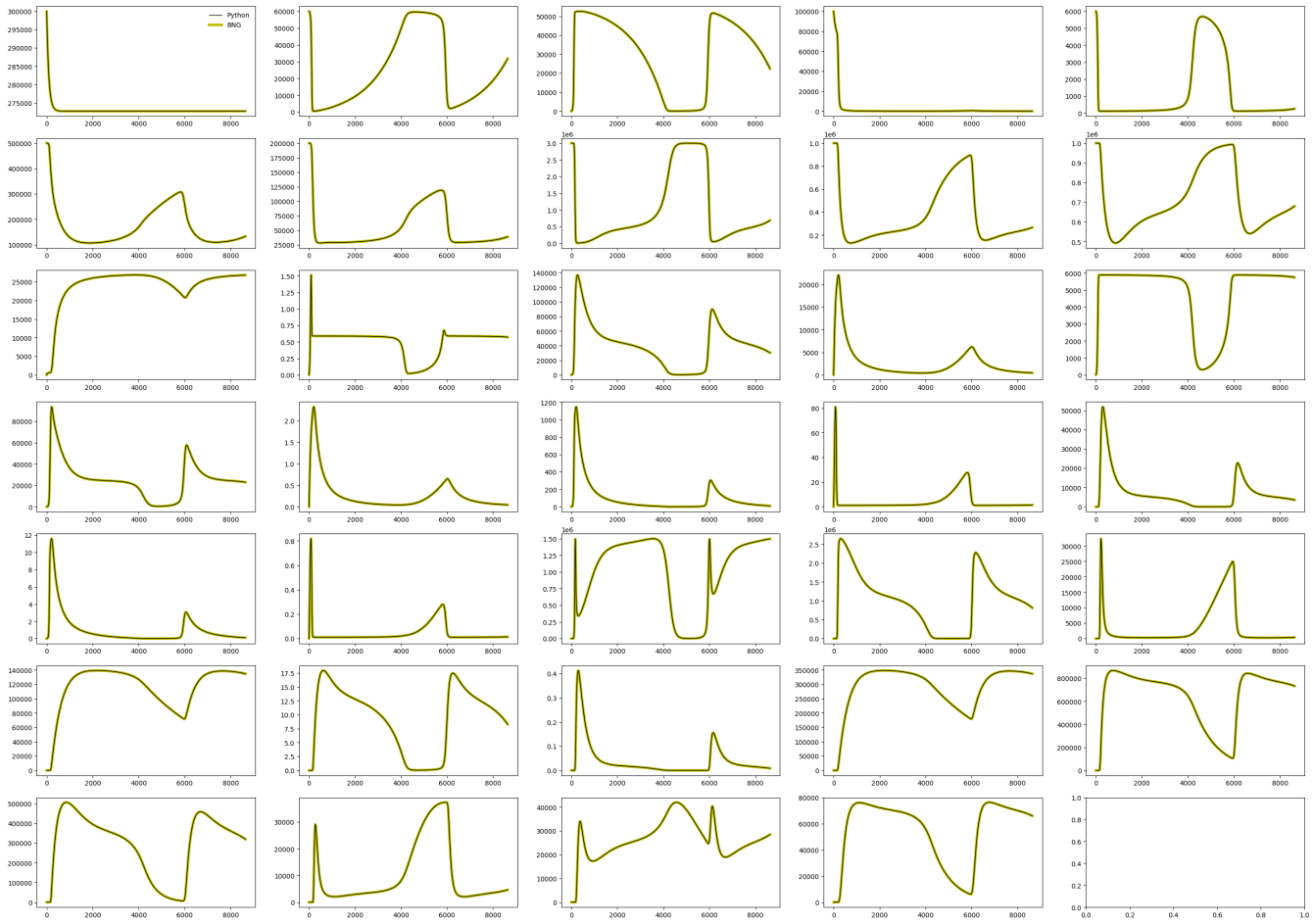
In [13]:
```
@njit
def dX(t,X):

    dx = np.zeros((numSpecies,))

    for i in range(numReactions):

        flux = rates[i]

        for j in range(1, reactants[i][0]+1):

            flux = flux*X[reactants[i][j]]

        for j in range(1, reactants[i][0]+1):

            dx[ reactants[i][j] ]  -= flux

        for j in range(1, products[i][0]+1):

            dx[ products[i][j] ]  += flux

    return dx
```

In [14]:
```
tStart = time.time()
method='LSODA'
tSpan = np.linspace(0,8640,1001)
sol = solve_ivp(dX,(0,tSpan[-1]),IC,t_eval=tSpan,method=method,rtol=1e-6,atol=1
e-6)
tEnd = time.time()

print(tEnd-tStart)
```

```
0.6869966983795166
```

In [15]:
```
bng = np.genfromtxt('ERK_model_ODE.cdat')
bng_t = bng[:,0]
bng_x = bng[:,1:].T
```

In [16]:
```
fig,ax = plt.subplots(7,5,figsize=(35,25))

for i in range(7):
    for j in range(5):
        index = i*5+j
        if index<34:
            ax[i][j].plot(sol.t, sol.y[index,:],zorder=2, lw=1, color='k', label='Python')
            ax[i][j].plot(bng_t, bng_x[index,:],zorder=1, lw=4, color='y', label='BNG')
            if index==0:
                ax[i][j].legend(loc=1, frameon=False)
```



In [ ]: