Time-Dependent Probability Distribution for the A<->B Reaction

We will use $p(t) = e^{R*t}p(0)$, to compute the time evolution of the copy numbers of A

In [1]:
```
from scipy import *
from scipy import linalg
```

In [2]:
```
N=10 # The number of states
t=0.48 # time elapsed in seconds
kf=2.5 # forward rate in s^-1
kb=2.5 # backward rate in s^-1
```

In [3]:
```
R=zeros([N+1,N+1])
```

In [4]:
```
for n in r_[1:N]:
    R[n,n]=-(kf*n+kb*(N-n))
    R[n,n+1]=kf*(n+1)
    R[n,n-1]=kb*(N-(n-1))
R[0,0]=-kb*N
R[10,10]=-kf*N
R[0,1]=kf*1
R[N-1,N]=kf*N
R[N,N-1]=kb*1
print R
print sum(R,axis=0)
```

```
[[-25.    2.5   0.    0.    0.    0.    0.    0.    0.    0.    0. ]
 [ 25.  -25.    5.    0.    0.    0.    0.    0.    0.    0.    0. ]
 [  0.   22.5 -25.    7.5   0.    0.    0.    0.    0.    0.    0. ]
 [  0.    0.   20.  -25.   10.    0.    0.    0.    0.    0.    0. ]
 [  0.    0.    0.   17.5 -25.   12.5   0.    0.    0.    0.    0. ]
 [  0.    0.    0.    0.   15.  -25.   15.    0.    0.    0.    0. ]
 [  0.    0.    0.    0.    0.   12.5 -25.   17.5   0.    0.    0. ]
 [  0.    0.    0.    0.    0.    0.   10.  -25.   20.    0.    0. ]
 [  0.    0.    0.    0.    0.    0.    0.    7.5 -25.   22.5   0. ]
 [  0.    0.    0.    0.    0.    0.    0.    0.    5.  -25.   25. ]
 [  0.    0.    0.    0.    0.    0.    0.    0.    0.    2.5 -25. ]]
[ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.]
```

In [5]:
```
ONES=ones(N+1)
print ONES
```

```
[ 1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.]
```

In [6]:
```
dot(ONES,R)
```

Out[6]:
```
array([ 0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.])
```

In [7]:
```
T=linalg.expm2(R*t)
```

In [8]:
```
dot(ONES,T)
```

Out[8]:
```
array([ 1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.])
```

In [9]:
```
p0=zeros(N+1)
p0[0]=1
print p0
```

```
[ 1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.]
```

In [10]:
```
pt=dot(T,p0)
sum(pt)
```

Out[10]:
```
0.99999999999999711
```

In [11]:
```
print pt
```

```
[ 0.00232715  0.01940041  0.07277957  0.16179473  0.23604161  0.23613261
  0.1640442   0.0781464   0.02443017  0.00452585  0.0003773 ]
```

Analytical answer from the file: A_reversible_B_explicit_solved.mw

In [14]:
```
p_analyt=array([0.002327151517,0.1940040583e-1, 0.7277956968e-1, .1617947
print p_analyt
```

```
[ 0.00232715  0.01940041  0.07277957  0.16179473  0.23604161  0.23613261
  0.1640442   0.0781464   0.02443016  0.00452585  0.00037729]
```

In [13]:
```
allclose(p_analyt,pt)
```

Out[13]:
```
True
```

In [17]:
```
for i in r_[0:N+1]: print "P[", i, "] = ", pt[N-i]
```

```
P[ 0 ] =  0.000377299479391
P[ 1 ] =  0.00452584890933
P[ 2 ] =  0.0244301655874
P[ 3 ] =  0.0781464022211
P[ 4 ] =  0.164044200964
P[ 5 ] =  0.236132613555
P[ 6 ] =  0.236041612807
P[ 7 ] =  0.161794729459
```

```
P[ 8 ] =  0.0727795696645
P[ 9 ] =  0.0194004058362
P[ 10 ] =  0.00232715151 73
```