HW3_neuro

October 14, 2020

1 Homework 3

1.0.1 Yanis Tazi

[33]:

```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.ticker as tck
import math
plt.rcParams["figure.figsize"] = (15,10)
```

1.0.2 To implement this model, the idea is to be in the steady state : r( )=F(I( )).

1.0.3 However, as discussed it is hard to calculate it directly . One reason is that this differential equation is a partial differential equation becauseit depends on t but also on . Therefore , we need to approximate numerically the steady state, the trick is to iterating over a long time period so that it converges to the steady state.

1.0.4 We also need to create different classesto include : connectivity ring with a function that will create the recurrent connectivities J_{ij} and a class dynamics to iterate over time . We also need an initial vector of r taking into consideration the initial. conditions r(i, t = 0) = a cos(2(i − 0)) for all i values, here we will chose 50 ranging from $-\frac{\pi}{2}$ to $\frac{\pi}{2}$. This ring created is considered as the coding space , not the physical space. and the angles ' from the connectivity are considered to be the angle of activity of presynaptic current.

2 Question 1 : Use J0 = −0.5, J2 = 1, A = 40 Hz and ϵ = 0.1.

[34]:

```
def h(theta,A,c,eps,theta_cue=0):
    return (A*c*(1-eps+eps*math.cos(2*(theta-theta_cue))))
```

[35]:

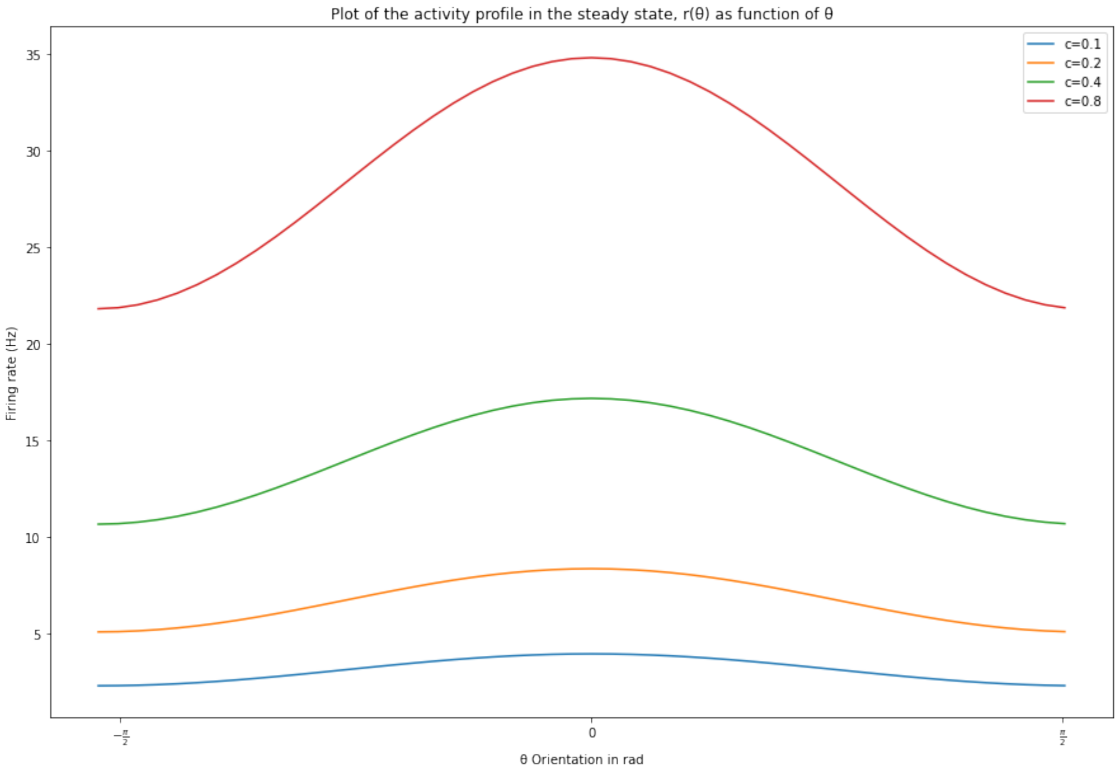
```
def numerical_integrations(N=50,a=2,theta_0=0,A=40,c=0.1,eps=0.1,J_0=-0.
↪5,J_2=1,d_t=0.0005,T=100,tau=10,theta_cue=0):

    thetas = [(math.pi/N)*i -  math.pi/2 for i in range(0,N)]
    r = [a*math.cos(2*(theta-theta_0)) for theta in thetas]
    for t in np.arange(0,T, d_t):
        #print(r)
        H = [h(theta=theta,A=A,c=c,eps=eps,theta_cue=theta_cue) for theta in␣
↪thetas]
        sum_I = [(1/(N-1)) * np.sum([(J_0+J_2*math.
↪cos(2*(thetas[i]-thetas[j]))*r[j])
                                    for j in range(N)]) for i in range(N)]
        I = [a + b for a, b in zip(H, sum_I)]
        I = [i if i>0 else 0 for i in I]
        r = r + (d_t/tau)*(np.negative(r)+I)
    return (r)
```

3 a) $\theta_{cue} = 0$

a=2, $\theta_0 = 0$

[36]:

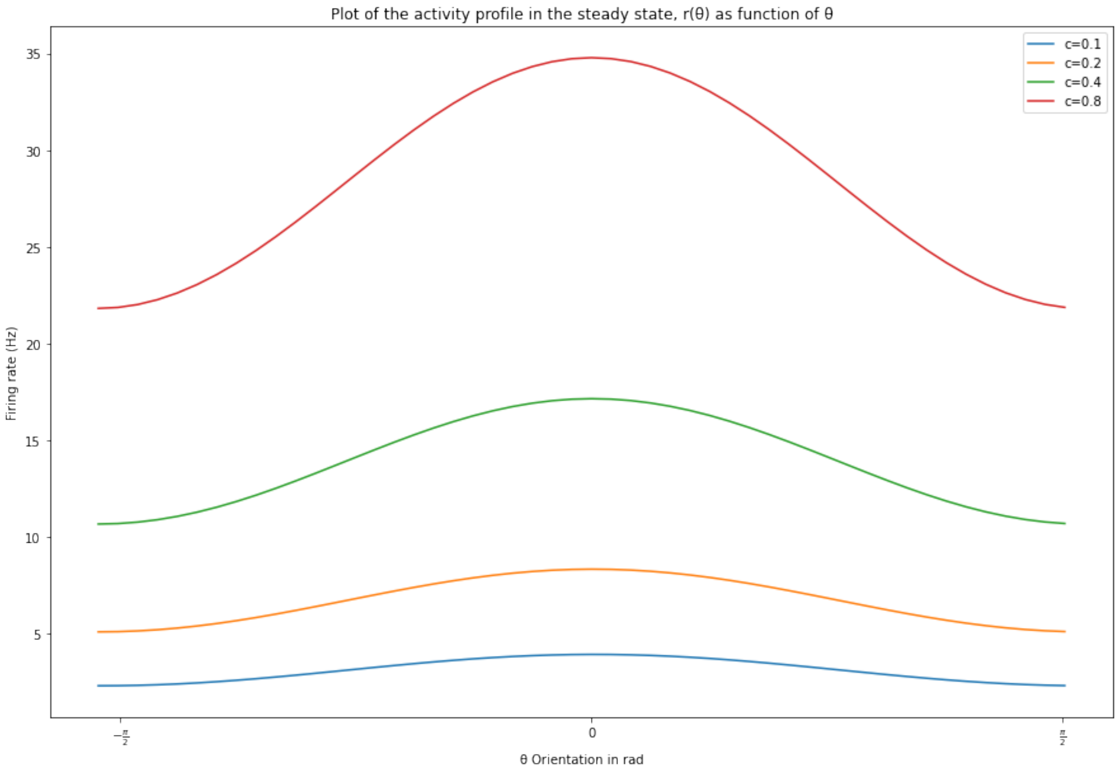
```
N=50
plt.plot([(math.pi/N)*i -  math.pi/2 for i in␣
↪range(0,N)],numerical_integrations(N=N,a=2,theta_0=0,c=0.1,d_t=0.
↪05,T=100),label='c=0.1')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in␣
↪range(0,N)],numerical_integrations(N=N,a=2,theta_0=0,c=0.2,d_t=0.
↪05,T=100),label='c=0.2')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in␣
↪range(0,N)],numerical_integrations(N=N,a=2,theta_0=0,c=0.4,d_t=0.
↪05,T=100),label='c=0.4')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in␣
↪range(0,N)],numerical_integrations(N=N,a=2,theta_0=0,c=0.8,d_t=0.
↪05,T=100),label='c=0.8')

plt.xlabel('  Orientation in rad')
plt.ylabel('Firing rate (Hz)')
plt.legend()
plt.title("Plot of the activity profile in the steady state, r( ) as function␣
↪of  ")
plt.xticks([-1.5,0,1.5], [r'$-\frac{\pi}{2}$', '0', r'$\frac{\pi}{2}$'])
plt.show()
```



a=2, $\theta_0 = \frac{\pi}{4}$

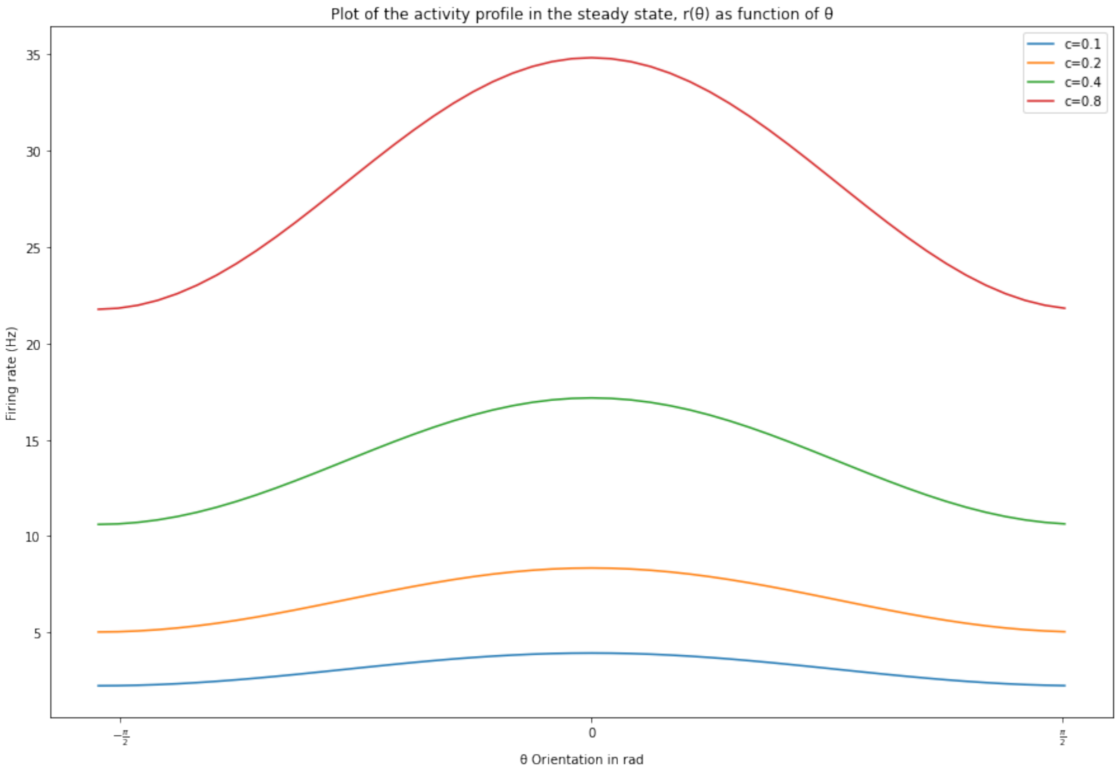
[37]:

```
N=50
plt.plot([(math.pi/N)*i -  math.pi/2 for i in␣
 ↪range(0,N)],numerical_integrations(N=N,a=2,theta_0=math.pi/4,c=0.1,d_t=0.
 ↪05,T=100),label='c=0.1')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in␣
 ↪range(0,N)],numerical_integrations(N=N,a=2,theta_0=math.pi/4,c=0.2,d_t=0.
 ↪05,T=100),label='c=0.2')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in␣
 ↪range(0,N)],numerical_integrations(N=N,a=2,theta_0=math.pi/4,c=0.4,d_t=0.
 ↪05,T=100),label='c=0.4')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in␣
 ↪range(0,N)],numerical_integrations(N=N,a=2,theta_0=math.pi/4,c=0.8,d_t=0.
 ↪05,T=100),label='c=0.8')
plt.xlabel('  Orientation in rad')
plt.ylabel('Firing rate (Hz)')
plt.legend()
plt.title("Plot of the activity profile in the steady state, r( ) as function␣
 ↪of  ")
plt.xticks([-1.5,0,1.5], [r'$-\frac{\pi}{2}$', '0', r'$\frac{\pi}{2}$'])
plt.show()
```



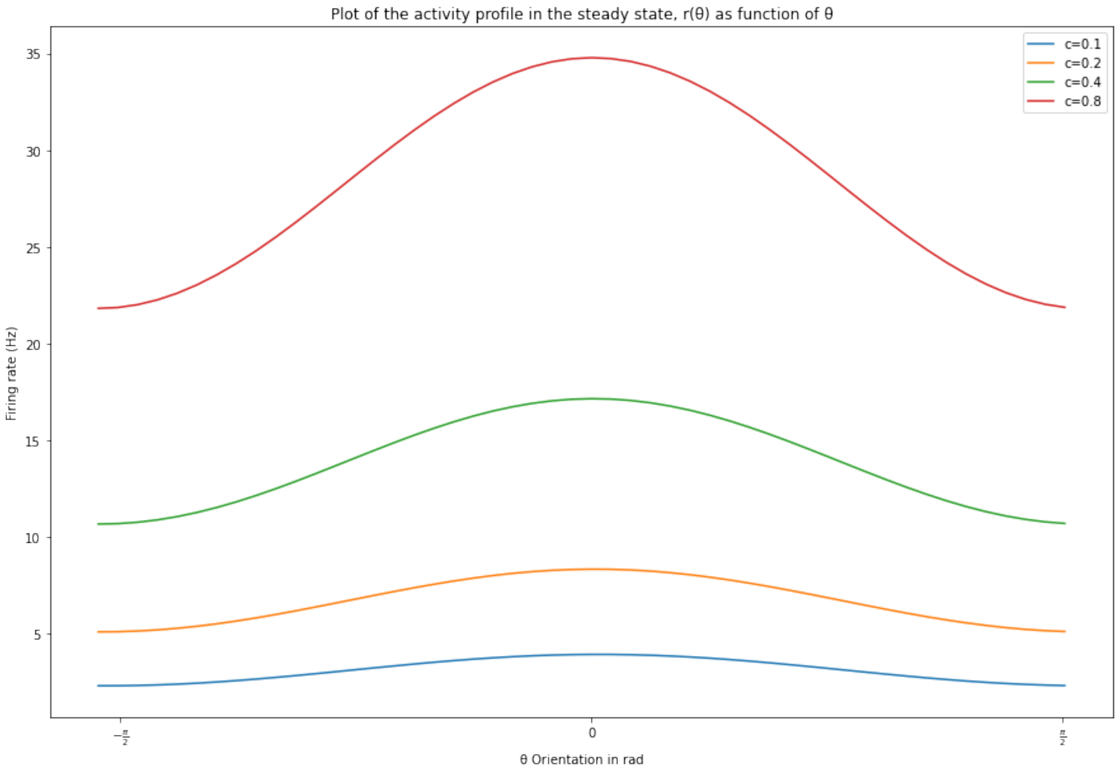
a=5, $\theta_0 = 0$

```
[38]: N=50
      plt.plot([(math.pi/N)*i -  math.pi/2 for i in␣
       ↪range(0,N)],numerical_integrations(N=N,a=5,theta_0=0,c=0.1,d_t=0.
       ↪05,T=100),label='c=0.1')
      plt.plot([(math.pi/N)*i -  math.pi/2 for i in␣
       ↪range(0,N)],numerical_integrations(N=N,a=5,theta_0=0,c=0.2,d_t=0.
       ↪05,T=100),label='c=0.2')
      plt.plot([(math.pi/N)*i -  math.pi/2 for i in␣
       ↪range(0,N)],numerical_integrations(N=N,a=5,theta_0=0,c=0.4,d_t=0.
       ↪05,T=100),label='c=0.4')
      plt.plot([(math.pi/N)*i -  math.pi/2 for i in␣
       ↪range(0,N)],numerical_integrations(N=N,a=5,theta_0=0,c=0.8,d_t=0.
       ↪05,T=100),label='c=0.8')
      plt.xlabel('  Orientation in rad')
      plt.ylabel('Firing rate (Hz)')
      plt.legend()
      plt.title("Plot of the activity profile in the steady state, r( ) as function␣
       ↪of  ")
      plt.xticks([-1.5,0,1.5], [r'$-\frac{\pi}{2}$', '0', r'$\frac{\pi}{2}$'])
      plt.show()
```



a=5, $\theta_0 = \frac{\pi}{4}$

[39]:
```
N=50
plt.plot([(math.pi/N)*i -  math.pi/2 for i in␣
 ↪range(0,N)],numerical_integrations(N=N,a=5,theta_0=math.pi/4,c=0.1,d_t=0.
 ↪05,T=100),label='c=0.1')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in␣
 ↪range(0,N)],numerical_integrations(N=N,a=5,theta_0=math.pi/4,c=0.2,d_t=0.
 ↪05,T=100),label='c=0.2')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in␣
 ↪range(0,N)],numerical_integrations(N=N,a=5,theta_0=math.pi/4,c=0.4,d_t=0.
 ↪05,T=100),label='c=0.4')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in␣
 ↪range(0,N)],numerical_integrations(N=N,a=5,theta_0=math.pi/4,c=0.8,d_t=0.
 ↪05,T=100),label='c=0.8')
plt.xlabel('  Orientation in rad')
plt.ylabel('Firing rate (Hz)')
plt.legend()
plt.title("Plot of the activity profile in the steady state, r( ) as function␣
 ↪of  ")
plt.xticks([-1.5,0,1.5], [r'$-\frac{\pi}{2}$', '0', r'$\frac{\pi}{2}$'])
plt.show()
```

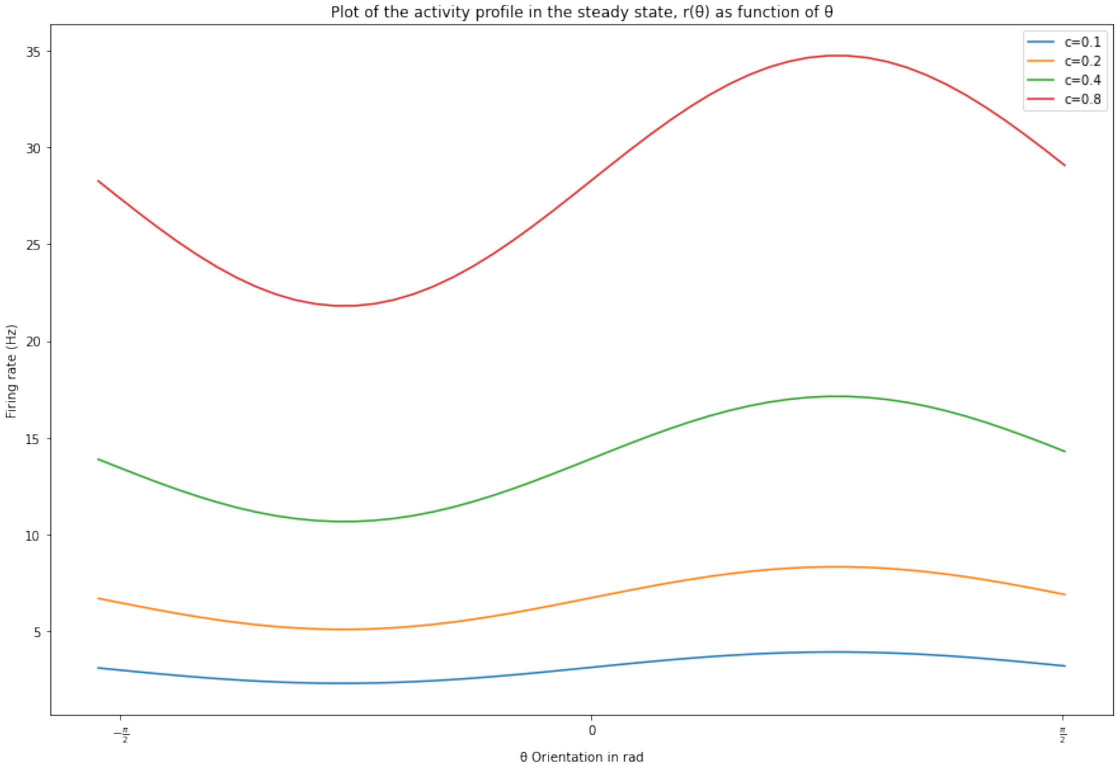


4 b) $\theta_{cue} = \frac{\pi}{4}$

a=2, $\theta_0 = 0$

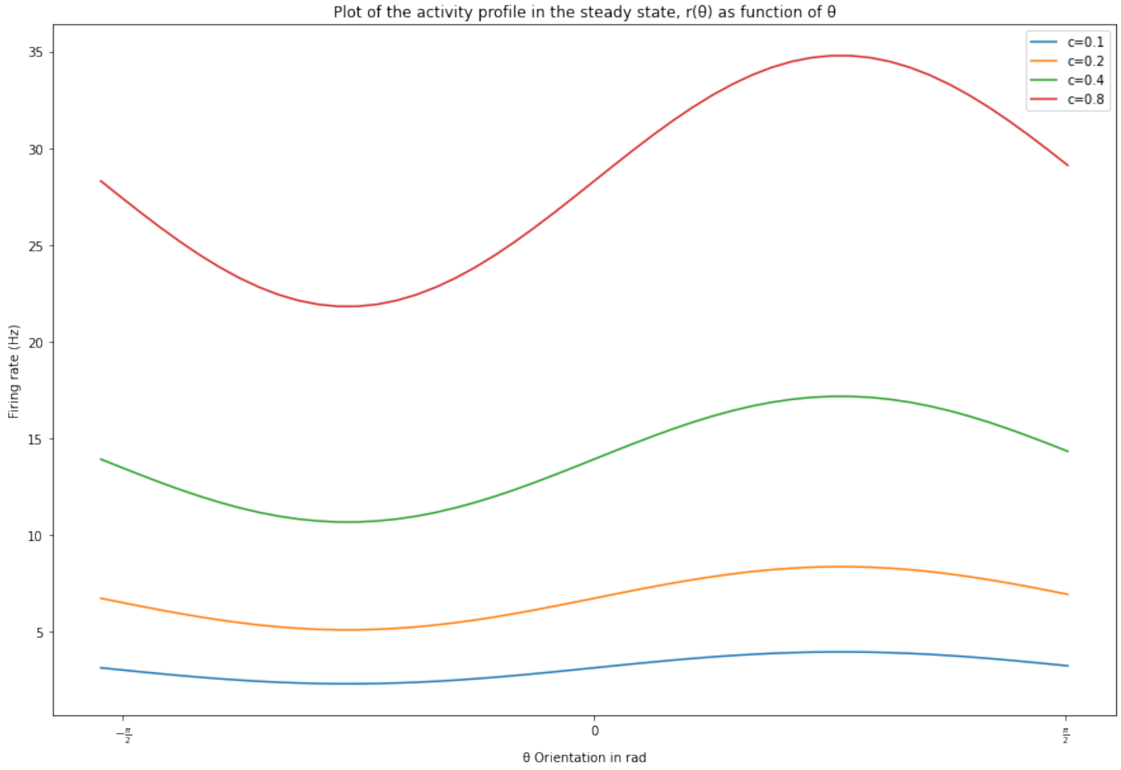
[40]:

```
N=50
plt.plot([(math.pi/N)*i -  math.pi/2 for i in range(0,N)],numerical_integrations(theta_cue=math.pi/4,N=N,a=2,theta_0=0,c=0.1,d_t=0.05,T=100),label='c=0.1')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in range(0,N)],numerical_integrations(theta_cue=math.pi/4,N=N,a=2,theta_0=0,c=0.2,d_t=0.05,T=100),label='c=0.2')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in range(0,N)],numerical_integrations(theta_cue=math.pi/4,N=N,a=2,theta_0=0,c=0.4,d_t=0.05,T=100),label='c=0.4')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in range(0,N)],numerical_integrations(theta_cue=math.pi/4,N=N,a=2,theta_0=0,c=0.8,d_t=0.05,T=100),label='c=0.8')

plt.xlabel('  Orientation in rad')
plt.ylabel('Firing rate (Hz)')
plt.legend()
plt.title("Plot of the activity profile in the steady state, r( ) as function␣
 ↪of  ")
plt.xticks([-1.5,0,1.5], [r'$-\frac{\pi}{2}$', '0', r'$\frac{\pi}{2}$'])
plt.show()
```



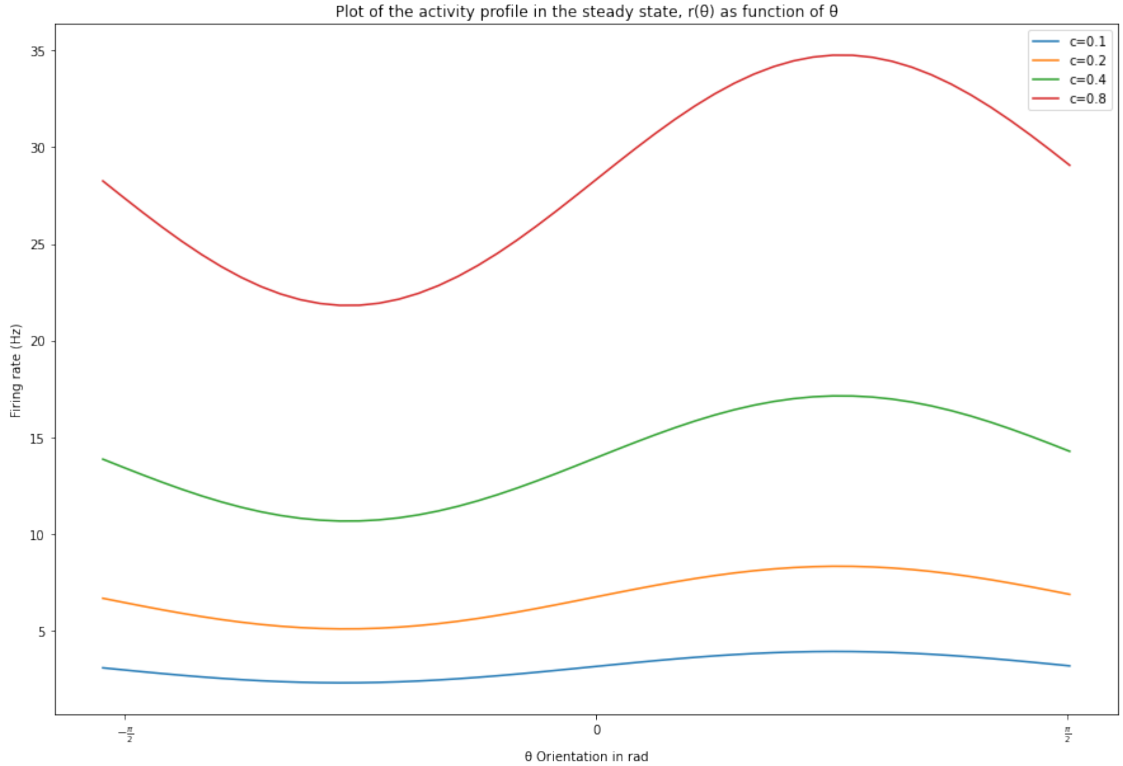
a=2, $\theta_0 = \frac{\pi}{4}$

```
[41]: N=50
plt.plot([(math.pi/N)*i -  math.pi/2 for i in␣
 ↪range(0,N)],numerical_integrations(theta_cue=math.pi/4,N=N,a=2,theta_0=math.
 ↪pi/4,c=0.1,d_t=0.05,T=100),label='c=0.1')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in␣
 ↪range(0,N)],numerical_integrations(theta_cue=math.pi/4,N=N,a=2,theta_0=math.
 ↪pi/4,c=0.2,d_t=0.05,T=100),label='c=0.2')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in␣
 ↪range(0,N)],numerical_integrations(theta_cue=math.pi/4,N=N,a=2,theta_0=math.
 ↪pi/4,c=0.4,d_t=0.05,T=100),label='c=0.4')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in␣
 ↪range(0,N)],numerical_integrations(theta_cue=math.pi/4,N=N,a=2,theta_0=math.
 ↪pi/4,c=0.8,d_t=0.05,T=100),label='c=0.8')
plt.xlabel('  Orientation in rad')
plt.ylabel('Firing rate (Hz)')
plt.legend()
plt.title("Plot of the activity profile in the steady state, r( ) as function␣
 ↪of  ")
plt.xticks([-1.5,0,1.5], [r'$-\frac{\pi}{2}$', '0', r'$\frac{\pi}{2}$'])
plt.show()
```



a=5, $\theta_0 = 0$

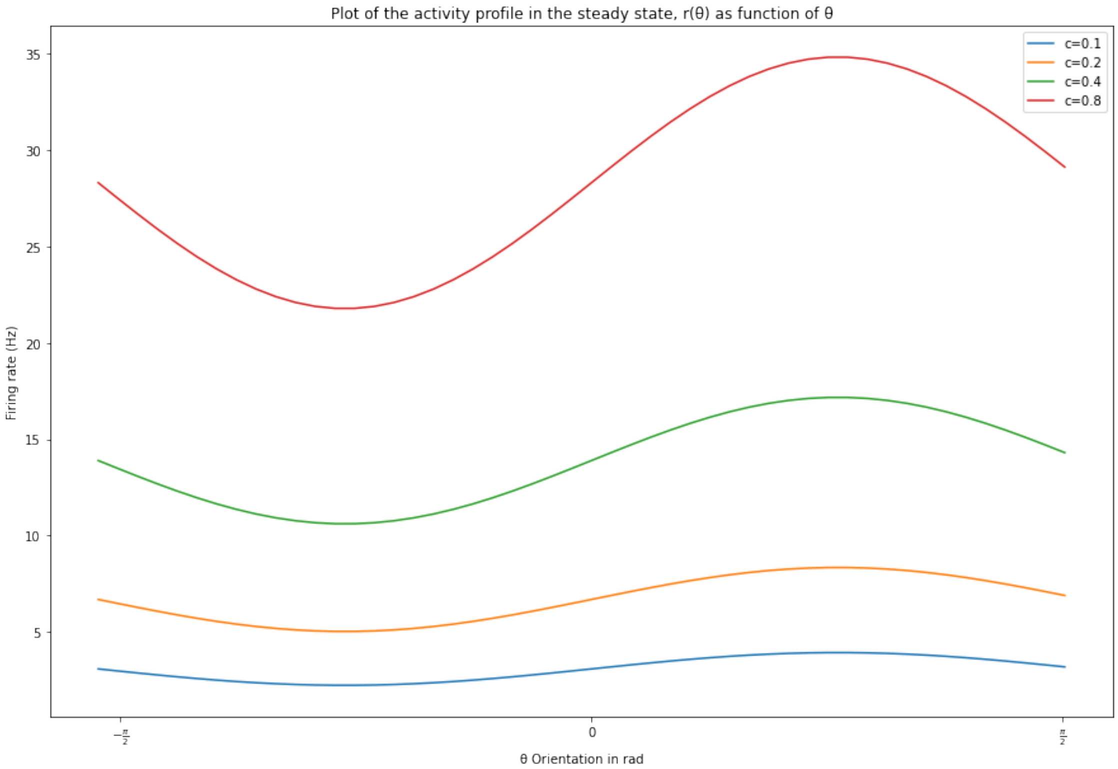
```
[42]: N=50
plt.plot([(math.pi/N)*i -  math.pi/2 for i in␣
 ↪range(0,N)],numerical_integrations(theta_cue=math.pi/4,N=N,a=5,theta_0=0,c=0.
 ↪1,d_t=0.05,T=100),label='c=0.1')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in␣
 ↪range(0,N)],numerical_integrations(theta_cue=math.pi/4,N=N,a=5,theta_0=0,c=0.
 ↪2,d_t=0.05,T=100),label='c=0.2')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in␣
 ↪range(0,N)],numerical_integrations(theta_cue=math.pi/4,N=N,a=5,theta_0=0,c=0.
 ↪4,d_t=0.05,T=100),label='c=0.4')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in␣
 ↪range(0,N)],numerical_integrations(theta_cue=math.pi/4,N=N,a=5,theta_0=0,c=0.
 ↪8,d_t=0.05,T=100),label='c=0.8')
plt.xlabel('  Orientation in rad')
plt.ylabel('Firing rate (Hz)')
plt.legend()
plt.title("Plot of the activity profile in the steady state, r( ) as function␣
 ↪of  ")
plt.xticks([-1.5,0,1.5], [r'$-\frac{\pi}{2}$', '0', r'$\frac{\pi}{2}$'])
plt.show()
```



a=5, $\theta_0 = \frac{\pi}{4}$

[43]:
```
N=50
plt.plot([(math.pi/N)*i -  math.pi/2 for i in␣
 ↪range(0,N)],numerical_integrations(theta_cue=math.pi/4,N=N,a=5,theta_0=math.
 ↪pi/4,c=0.1,d_t=0.05,T=100),label='c=0.1')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in␣
 ↪range(0,N)],numerical_integrations(theta_cue=math.pi/4,N=N,a=5,theta_0=math.
 ↪pi/4,c=0.2,d_t=0.05,T=100),label='c=0.2')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in␣
 ↪range(0,N)],numerical_integrations(theta_cue=math.pi/4,N=N,a=5,theta_0=math.
 ↪pi/4,c=0.4,d_t=0.05,T=100),label='c=0.4')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in␣
 ↪range(0,N)],numerical_integrations(theta_cue=math.pi/4,N=N,a=5,theta_0=math.
 ↪pi/4,c=0.8,d_t=0.05,T=100),label='c=0.8')
```

```
plt.xlabel('  Orientation in rad')
plt.ylabel('Firing rate (Hz)')
plt.legend()
plt.title("Plot of the activity profile in the steady state, r( ) as function␣
 ↪of  ")
plt.xticks([-1.5,0,1.5], [r'$-\frac{\pi}{2}$', '0', r'$\frac{\pi}{2}$'])
plt.show()
```



4.0.1 The parameter c also plays an important role here as it expresses the contrast and increasing c will increase linearly the thalamic input ,highlights the relative luminescence and increase the firing rate .

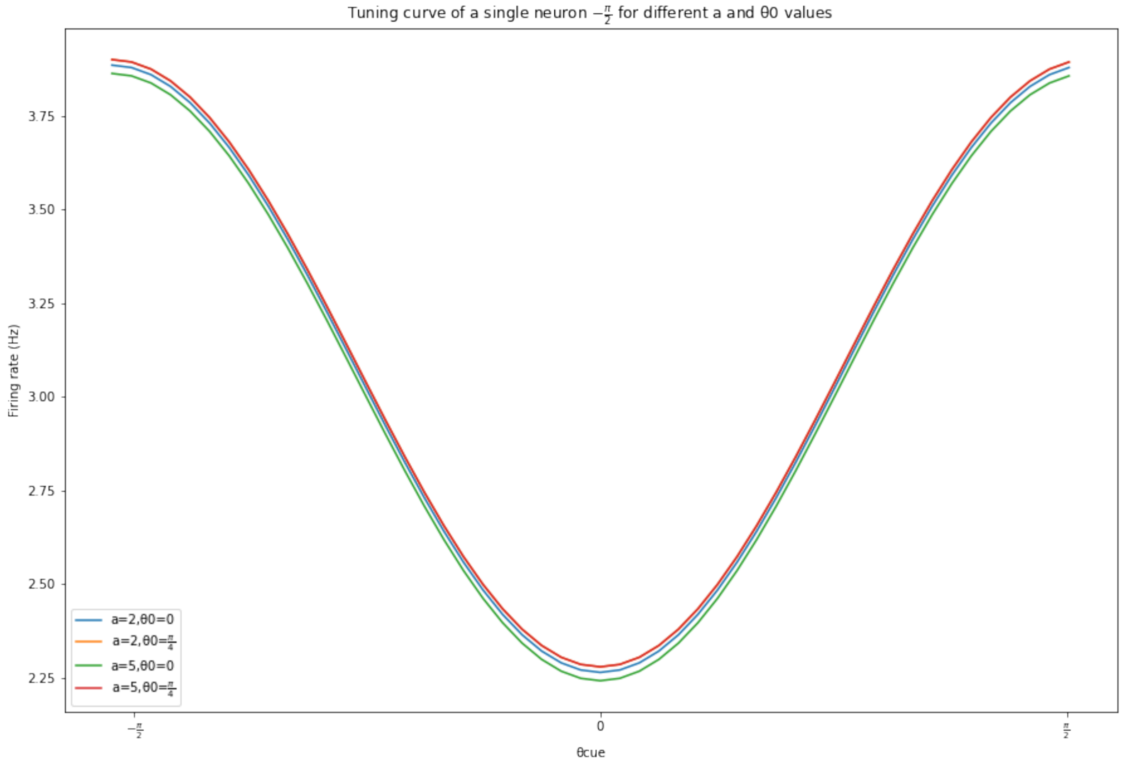
5 Single neuron : $-\frac{\pi}{2}$

[45]:

```
selected_neuron = 0
r_cues = []
r_cues1 = []
r_cues2 = []
r_cues3 = []
N=50
for cue in [(math.pi/N)*i -  math.pi/2 for i in range(0,N)]:
    r_cues.append(numerical_integrations(N=N,a=2,theta_0=0,c=0.1,d_t=0.
↪05,T=100,theta_cue=cue))
    r_cues1.append(numerical_integrations(N=N,a=2,theta_0=math.pi/4,c=0.1,d_t=0.
↪05,T=100,theta_cue=cue))
    r_cues2.append(numerical_integrations(N=N,a=5,theta_0=0,c=0.1,d_t=0.
↪05,T=100,theta_cue=cue))
    r_cues3.append(numerical_integrations(N=N,a=5,theta_0=math.pi/4,c=0.1,d_t=0.
↪05,T=100,theta_cue=cue))
```

[46]:

```
plt.plot([(math.pi/N)*i -  math.pi/2 for i in range(0,N)],[r_cues[i][0] for i␣
↪in range(50)],label='a=2, 0=0')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in range(0,N)],[r_cues1[i][0] for i␣
↪in range(50)],
         label='a=2, 0='+r'$\frac{\pi}{4}$')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in range(0,N)],[r_cues2[i][0] for i␣
↪in range(50)],label='a=5, 0=0')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in range(0,N)],[r_cues3[i][0] for i␣
↪in range(50)],
         label='a=5, 0='+r'$\frac{\pi}{4}$')
plt.legend()
plt.title("Tuning curve of a single neuron " + r'$-\frac{\pi}{2}$' +" for␣
↪different a and  0 values ")
plt.xlabel(" cue")
plt.ylabel("Firing rate (Hz)")
plt.xticks([-1.5,0,1.5], [r'$-\frac{\pi}{2}$', '0', r'$\frac{\pi}{2}$'])
plt.show()
```

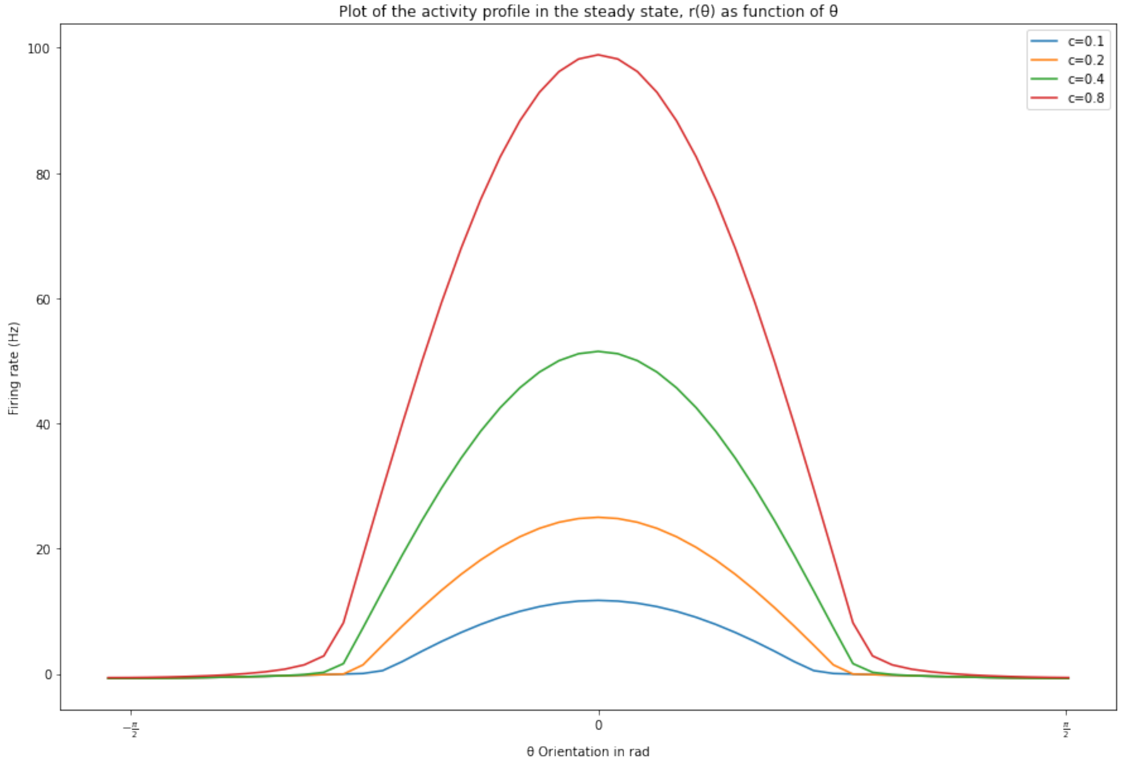


5.0.1 No differences for the tuning curves of a single neuron for different a and 0 values.

6 Question 2 : Use J0 $= -7.3$, J2 $= 11$, A $= 40$ Hz and $\epsilon = 0.1$.

a=2, $\theta_0 = 0$

```
[47]: N=50
plt.plot([(math.pi/N)*i -  math.pi/2 for i in range(0,N)],
         numerical_integrations(N=N,J_0=-7.3,J_2=11,a=2,theta_0=0,c=0.1,d_t=0.
 ↪005,T=10),label='c=0.1')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in range(0,N)],
         numerical_integrations(N=N,J_0=-7.3,J_2=11,a=2,theta_0=0,c=0.2,d_t=0.
 ↪005,T=10),label='c=0.2')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in range(0,N)],
         numerical_integrations(N=N,J_0=-7.3,J_2=11,a=2,theta_0=0,c=0.4,d_t=0.
 ↪005,T=10),label='c=0.4')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in range(0,N)],
         numerical_integrations(N=N,J_0=-7.3,J_2=11,a=2,theta_0=0,c=0.8,d_t=0.
 ↪005,T=10),label='c=0.8')

plt.xlabel('  Orientation in rad')
plt.ylabel('Firing rate (Hz)')
plt.legend()
plt.title("Plot of the activity profile in the steady state, r( ) as function␣
 ↪of  ")
plt.xticks([-1.5,0,1.5], [r'$-\frac{\pi}{2}$', '0', r'$\frac{\pi}{2}$'])
plt.show()
```

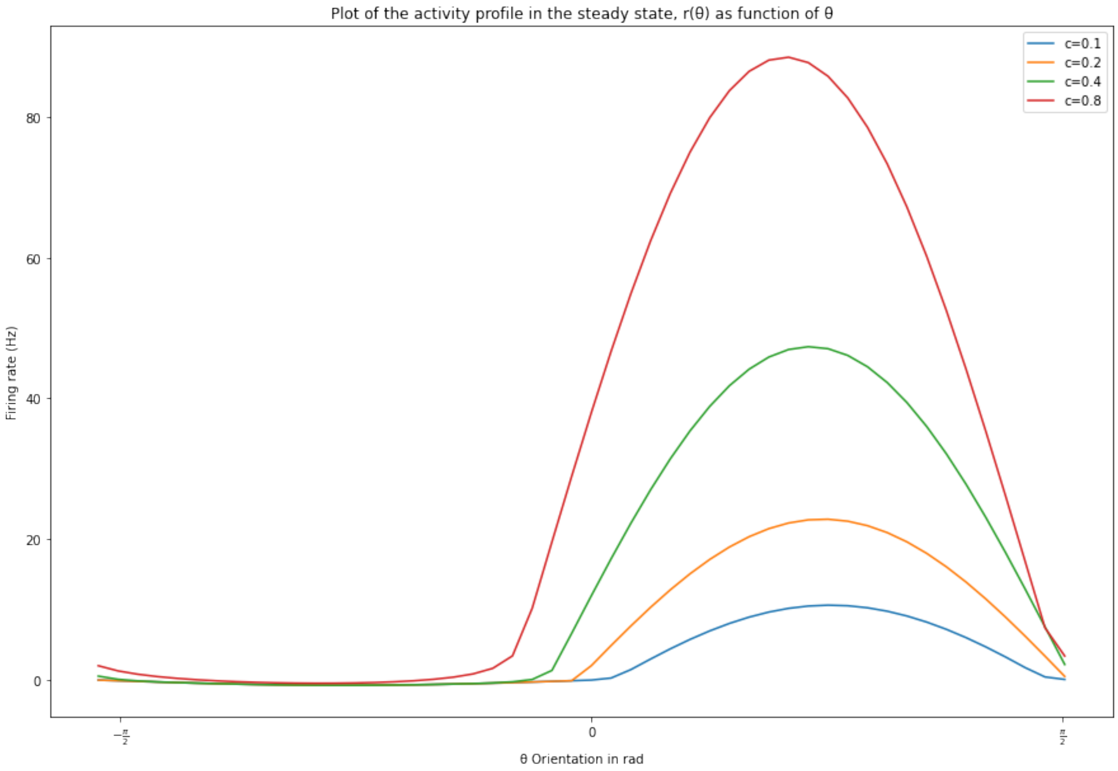


a=2, $\theta_0 = \frac{\pi}{4}$

[48]:

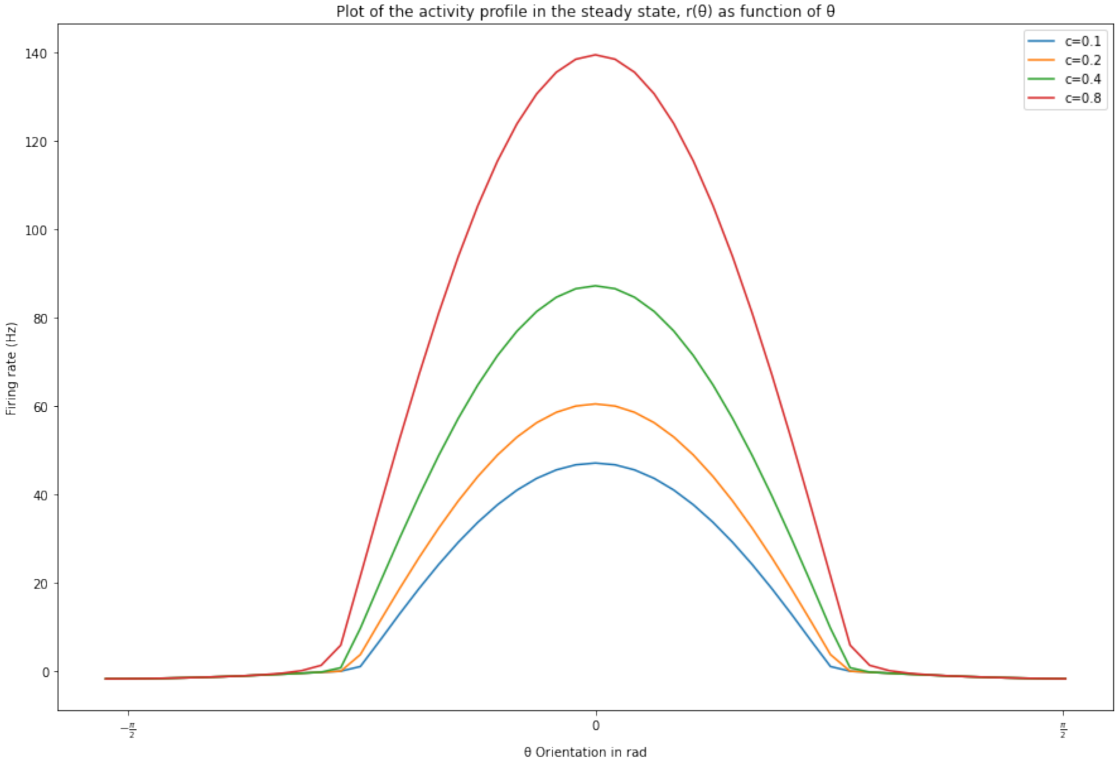
```
N=50
plt.plot([(math.pi/N)*i -  math.pi/2 for i in range(0,N)],
         numerical_integrations(N=N,J_0=-7.3,J_2=11,a=2,theta_0=math.pi/4,c=0.
 ↪1,d_t=0.005,T=10),label='c=0.1')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in range(0,N)],
         numerical_integrations(N=N,J_0=-7.3,J_2=11,a=2,theta_0=math.pi/4,c=0.
 ↪2,d_t=0.005,T=10),label='c=0.2')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in range(0,N)],
         numerical_integrations(N=N,J_0=-7.3,J_2=11,a=2,theta_0=math.pi/4,c=0.
 ↪4,d_t=0.005,T=10),label='c=0.4')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in range(0,N)],
         numerical_integrations(N=N,J_0=-7.3,J_2=11,a=2,theta_0=math.pi/4,c=0.
 ↪8,d_t=0.005,T=10),label='c=0.8')
```

```
plt.xlabel('  Orientation in rad')
plt.ylabel('Firing rate (Hz)')
plt.legend()
plt.title("Plot of the activity profile in the steady state, r( ) as function␣
 ↪of  ")
plt.xticks([-1.5,0,1.5], [r'$-\frac{\pi}{2}$', '0', r'$\frac{\pi}{2}$'])
plt.show()
```



a=5, $\theta_0 = 0$

```
[49]: N=50
plt.plot([(math.pi/N)*i -  math.pi/2 for i in range(0,N)],
         numerical_integrations(N=N,J_0=-7.3,J_2=11,a=5,theta_0=0,c=0.1,d_t=0.
 ↪005,T=10),label='c=0.1')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in range(0,N)],
         numerical_integrations(N=N,J_0=-7.3,J_2=11,a=5,theta_0=0,c=0.2,d_t=0.
 ↪005,T=10),label='c=0.2')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in range(0,N)],
         numerical_integrations(N=N,J_0=-7.3,J_2=11,a=5,theta_0=0,c=0.4,d_t=0.
 ↪005,T=10),label='c=0.4')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in range(0,N)],
         numerical_integrations(N=N,J_0=-7.3,J_2=11,a=5,theta_0=0,c=0.8,d_t=0.
 ↪005,T=10),label='c=0.8')
plt.xlabel('  Orientation in rad')
plt.ylabel('Firing rate (Hz)')
plt.legend()
plt.title("Plot of the activity profile in the steady state, r( ) as function␣
 ↪of  ")
plt.xticks([-1.5,0,1.5], [r'$-\frac{\pi}{2}$', '0', r'$\frac{\pi}{2}$'])
plt.show()
```

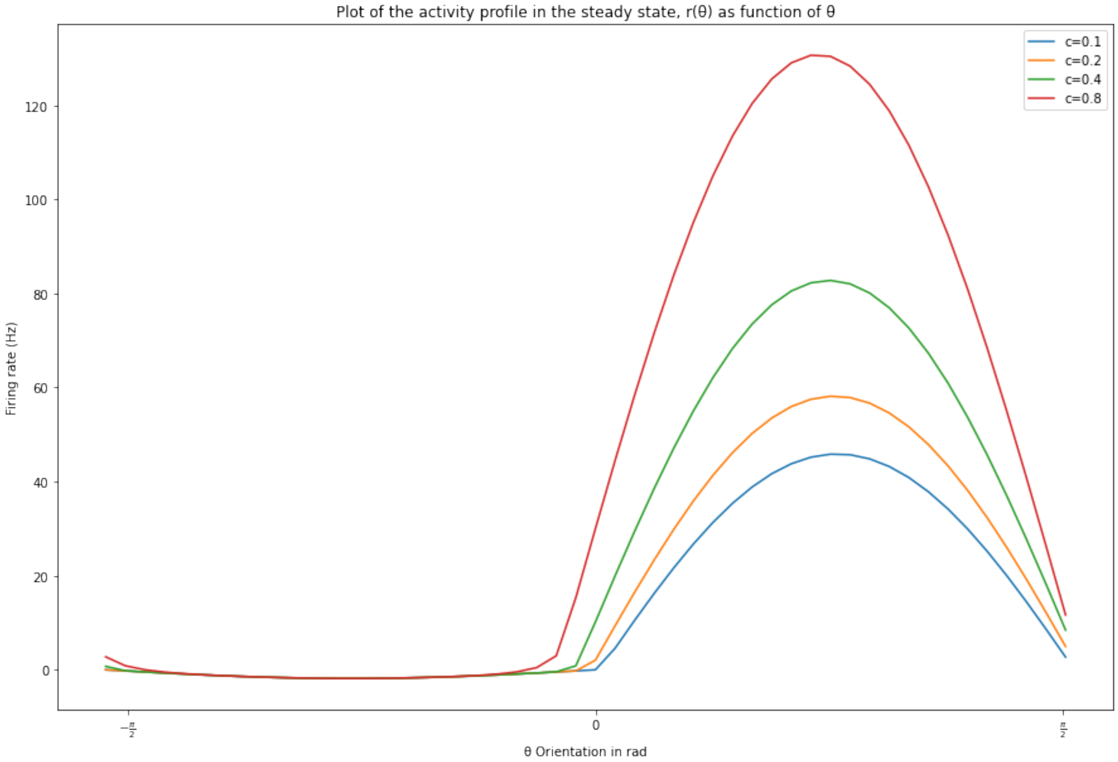


a=5, $\theta_0 = \frac{\pi}{4}$

[50]:

```
N=50
plt.plot([(math.pi/N)*i -  math.pi/2 for i in range(0,N)],
         numerical_integrations(N=N,J_0=-7.3,J_2=11,a=5,theta_0=math.pi/4,c=0.
 ↪1,d_t=0.005,T=10),label='c=0.1')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in range(0,N)],
         numerical_integrations(N=N,J_0=-7.3,J_2=11,a=5,theta_0=math.pi/4,c=0.
 ↪2,d_t=0.005,T=10),label='c=0.2')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in range(0,N)],
         numerical_integrations(N=N,J_0=-7.3,J_2=11,a=5,theta_0=math.pi/4,c=0.
 ↪4,d_t=0.005,T=10),label='c=0.4')
```

```
plt.plot([(math.pi/N)*i -  math.pi/2 for i in range(0,N)],
         numerical_integrations(N=N,J_0=-7.3,J_2=11,a=5,theta_0=math.pi/4,c=0.
 ↪8,d_t=0.005,T=10),label='c=0.8')
plt.xlabel('  Orientation in rad')
plt.ylabel('Firing rate (Hz)')
plt.legend()
plt.title("Plot of the activity profile in the steady state, r( ) as function␣
 ↪of  ")
plt.xticks([-1.5,0,1.5], [r'$-\frac{\pi}{2}$', '0', r'$\frac{\pi}{2}$'])
plt.show()
```



6.0.1 J_2 controls the strength of excitation while J_0 accounts for the recurrent inhibition.

6.0.2 We see that width of the model remains pretty much the same while c agains control the. firing rate amplitude . Those amplitudes are much stronger than earlier because the parameter space has changed and therefore the current phase too.

7 Question 3 : Use J0 $= -7.3$, J2 $= 11$, A $= 40$ Hz and $\epsilon = 0.1$.

8 $\theta_{cue} = 0$

[51]:

```
def h_noise(theta,A,c,eps,sigma=3,theta_cue=0):
    np.random.seed(17)
    return (A*c*(1-eps+eps*math.cos(2*(theta-theta_cue)))+sigma*np.random.
 ↪normal(0,1))
```

[52]:

```
def numerical_integrations_noise(N=50,a=2,theta_0=0,A=40,c=0.1,eps=0.1,J_0=-0.
 ↪5,J_2=1,d_t=0.01,T=300,tau=10,sigma=3,theta_cue=0):

    thetas = [(math.pi/N)*i -  math.pi/2 for i in range(0,N)]
    r = [a*math.cos(2*(theta-theta_0)) for theta in thetas]
    for t in np.arange(0,T, d_t):
        #print(r)
        H = [h_noise(theta=theta,A=A,c=c,eps=eps,theta_cue=theta_cue) for theta␣
 ↪in thetas]
        sum_I = [(1/(N-1)) * np.sum([(J_0+J_2*math.
 ↪cos(2*(thetas[i]-thetas[j]))*r[j])
                                     for j in range(N)]) for i in range(N)]
        I = [a + b for a, b in zip(H, sum_I)]
        I = [i if i>0 else 0 for i in I]
        r = r + (d_t/tau)*(np.negative(r)+I)
    return (r)
```

9 Like question 1 : Use J0 $= -0.5$, J2 $= 1$, A $= 40$ Hz and $\epsilon = 0.1$.

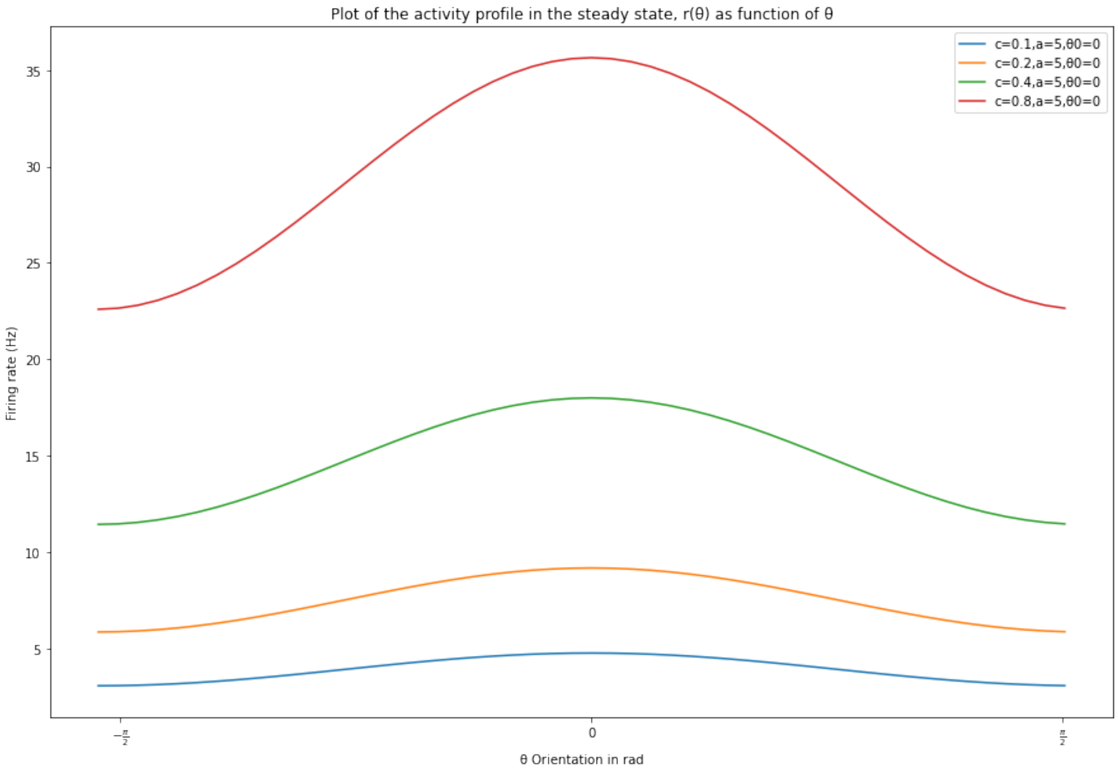
10 r( ) as function of

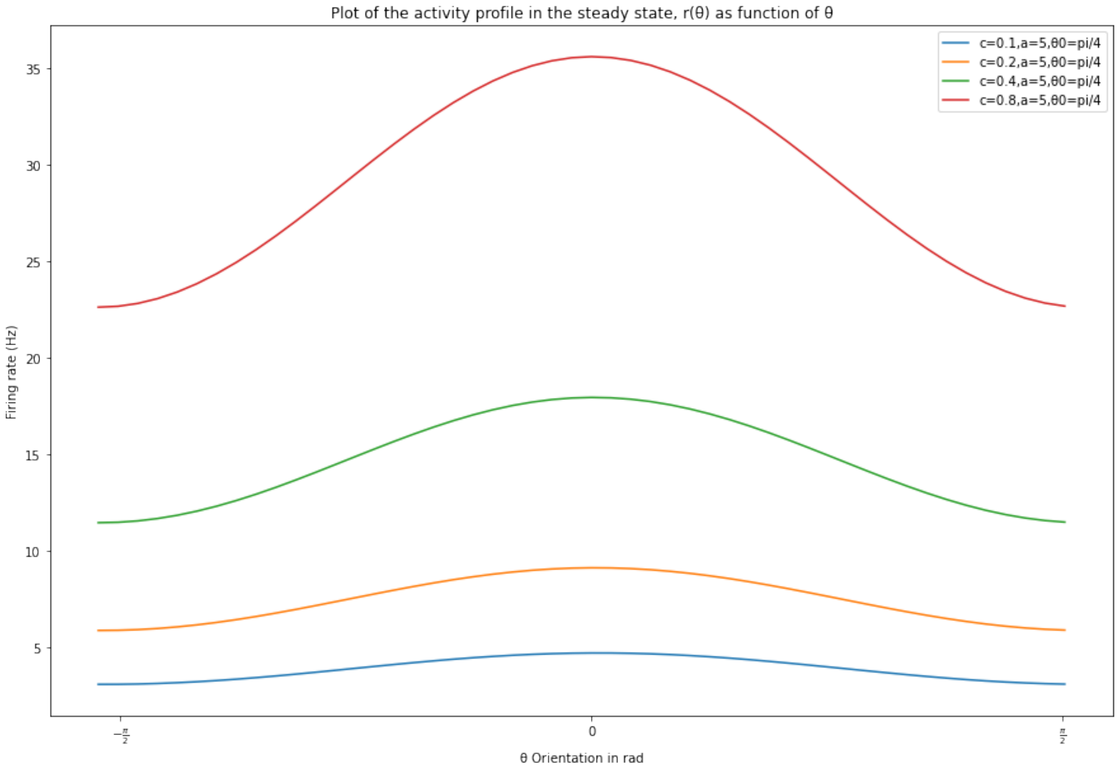
[53]:

```
N=50
plt.plot([(math.pi/N)*i -  math.pi/2 for i in␣
 ↪range(0,N)],numerical_integrations_noise(N=N,a=2,theta_0=0,c=0.1,d_t=0.
 ↪05,T=100),label='c=0.1,a=2, 0=0')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in␣
 ↪range(0,N)],numerical_integrations_noise(N=N,a=2,theta_0=0,c=0.2,d_t=0.
 ↪05,T=100),label='c=0.2,a=2, 0=0')
```

```
plt.plot([(math.pi/N)*i -  math.pi/2 for i in␣
 ↪range(0,N)],numerical_integrations_noise(N=N,a=2,theta_0=0,c=0.4,d_t=0.
 ↪05,T=100),label='c=0.4,a=2, 0=0')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in␣
 ↪range(0,N)],numerical_integrations_noise(N=N,a=2,theta_0=0,c=0.8,d_t=0.
 ↪05,T=100),label='c=0.8,a=2, 0=0')
plt.xlabel('  Orientation in rad')
plt.ylabel('Firing rate (Hz)')
plt.legend()
plt.title("Plot of the activity profile in the steady state, r( ) as function␣
 ↪of  ")
plt.xticks([-1.5,0,1.5], [r'$-\frac{\pi}{2}$', '0', r'$\frac{\pi}{2}$'])
plt.show()

N=50
plt.plot([(math.pi/N)*i -  math.pi/2 for i in␣
 ↪range(0,N)],numerical_integrations_noise(N=N,a=2,theta_0=math.pi/4,c=0.
 ↪1,d_t=0.05,T=100),label='c=0.1,a=2, 0=pi/4')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in␣
 ↪range(0,N)],numerical_integrations_noise(N=N,a=2,theta_0=math.pi/4,c=0.
 ↪2,d_t=0.05,T=100),label='c=0.2,a=2, 0=pi/4')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in␣
 ↪range(0,N)],numerical_integrations_noise(N=N,a=2,theta_0=math.pi/4,c=0.
 ↪4,d_t=0.05,T=100),label='c=0.4,a=2, 0=pi/4')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in␣
 ↪range(0,N)],numerical_integrations_noise(N=N,a=2,theta_0=math.pi/4,c=0.
 ↪8,d_t=0.05,T=100),label='c=0.8,a=2, 0=pi/4')
plt.xlabel('  Orientation in rad')
plt.ylabel('Firing rate (Hz)')
plt.legend()
plt.title("Plot of the activity profile in the steady state, r( ) as function␣
 ↪of  ")
plt.xticks([-1.5,0,1.5], [r'$-\frac{\pi}{2}$', '0', r'$\frac{\pi}{2}$'])
plt.show()

N=50
plt.plot([(math.pi/N)*i -  math.pi/2 for i in␣
 ↪range(0,N)],numerical_integrations_noise(N=N,a=5,theta_0=0,c=0.1,d_t=0.
 ↪05,T=100),label='c=0.1,a=5, 0=0')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in␣
 ↪range(0,N)],numerical_integrations_noise(N=N,a=5,theta_0=0,c=0.2,d_t=0.
 ↪05,T=100),label='c=0.2,a=5, 0=0')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in␣
 ↪range(0,N)],numerical_integrations_noise(N=N,a=5,theta_0=0,c=0.4,d_t=0.
 ↪05,T=100),label='c=0.4,a=5, 0=0')
```

```
plt.plot([(math.pi/N)*i -  math.pi/2 for i in␣
 ↪range(0,N)],numerical_integrations_noise(N=N,a=5,theta_0=0,c=0.8,d_t=0.
 ↪05,T=100),label='c=0.8,a=5, 0=0')
plt.xlabel('  Orientation in rad')
plt.ylabel('Firing rate (Hz)')
plt.legend()
plt.title("Plot of the activity profile in the steady state, r( ) as function␣
 ↪of  ")
plt.xticks([-1.5,0,1.5], [r'$-\frac{\pi}{2}$', '0', r'$\frac{\pi}{2}$'])
plt.show()

N=50
plt.plot([(math.pi/N)*i -  math.pi/2 for i in␣
 ↪range(0,N)],numerical_integrations_noise(N=N,a=5,theta_0=math.pi/4,c=0.
 ↪1,d_t=0.05,T=100),label='c=0.1,a=5, 0=pi/4')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in␣
 ↪range(0,N)],numerical_integrations_noise(N=N,a=5,theta_0=math.pi/4,c=0.
 ↪2,d_t=0.05,T=100),label='c=0.2,a=5, 0=pi/4')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in␣
 ↪range(0,N)],numerical_integrations_noise(N=N,a=5,theta_0=math.pi/4,c=0.
 ↪4,d_t=0.05,T=100),label='c=0.4,a=5, 0=pi/4')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in␣
 ↪range(0,N)],numerical_integrations_noise(N=N,a=5,theta_0=math.pi/4,c=0.
 ↪8,d_t=0.05,T=100),label='c=0.8,a=5, 0=pi/4')
plt.xlabel('  Orientation in rad')
plt.ylabel('Firing rate (Hz)')
plt.legend()
plt.title("Plot of the activity profile in the steady state, r( ) as function␣
 ↪of  ")
plt.xticks([-1.5,0,1.5], [r'$-\frac{\pi}{2}$', '0', r'$\frac{\pi}{2}$'])
plt.show()
```


Plot of the activity profile in the steady state, r(θ) as function of θ
c=0.1,a=5,θ0=0
c=0.2,a=5,θ0=0
c=0.4,a=5,θ0=0
c=0.8,a=5,θ0=0
Firing rate (Hz)
35
30
25
20
15
10
5
$-\frac{\pi}{2}$
0
$\frac{\pi}{2}$
θ Orientation in rad

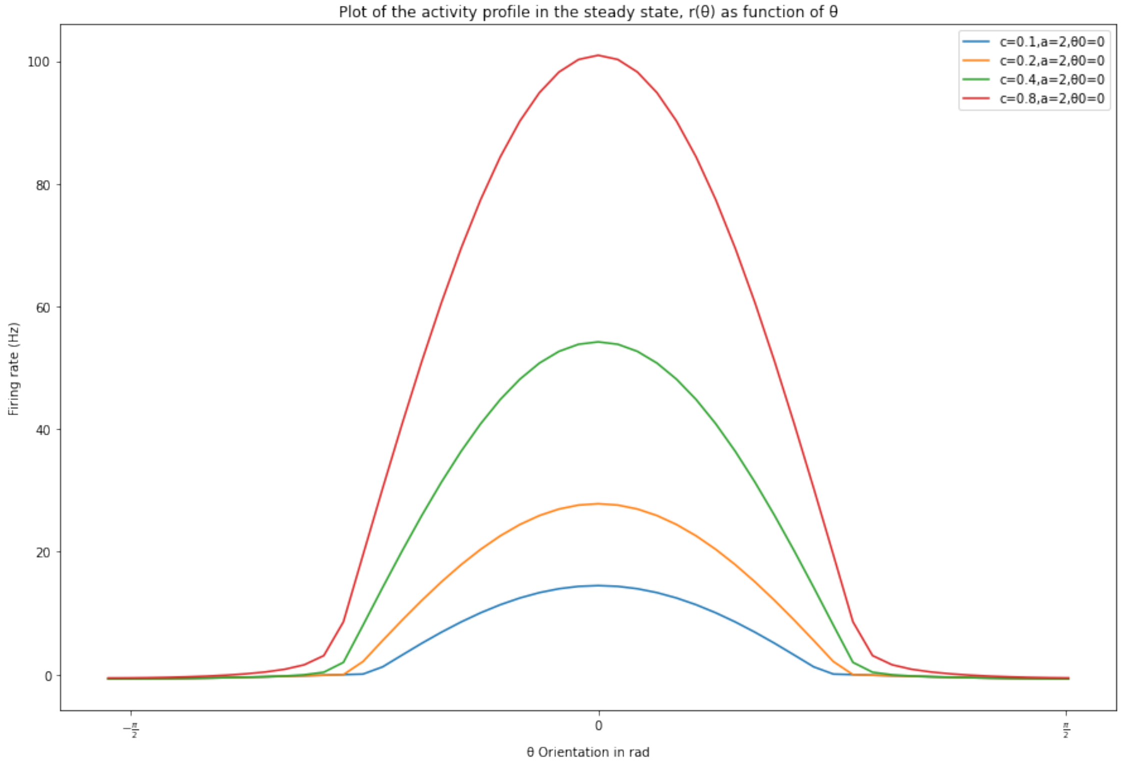

Plot of the activity profile in the steady state, r(θ) as function of θ
c=0.1,a=5,θ0=pi/4
c=0.2,a=5,θ0=pi/4
c=0.4,a=5,θ0=pi/4
c=0.8,a=5,θ0=pi/4
Firing rate (Hz)
35
30
25
20
15
10
5
$-\frac{\pi}{2}$
0
$\frac{\pi}{2}$
θ Orientation in rad

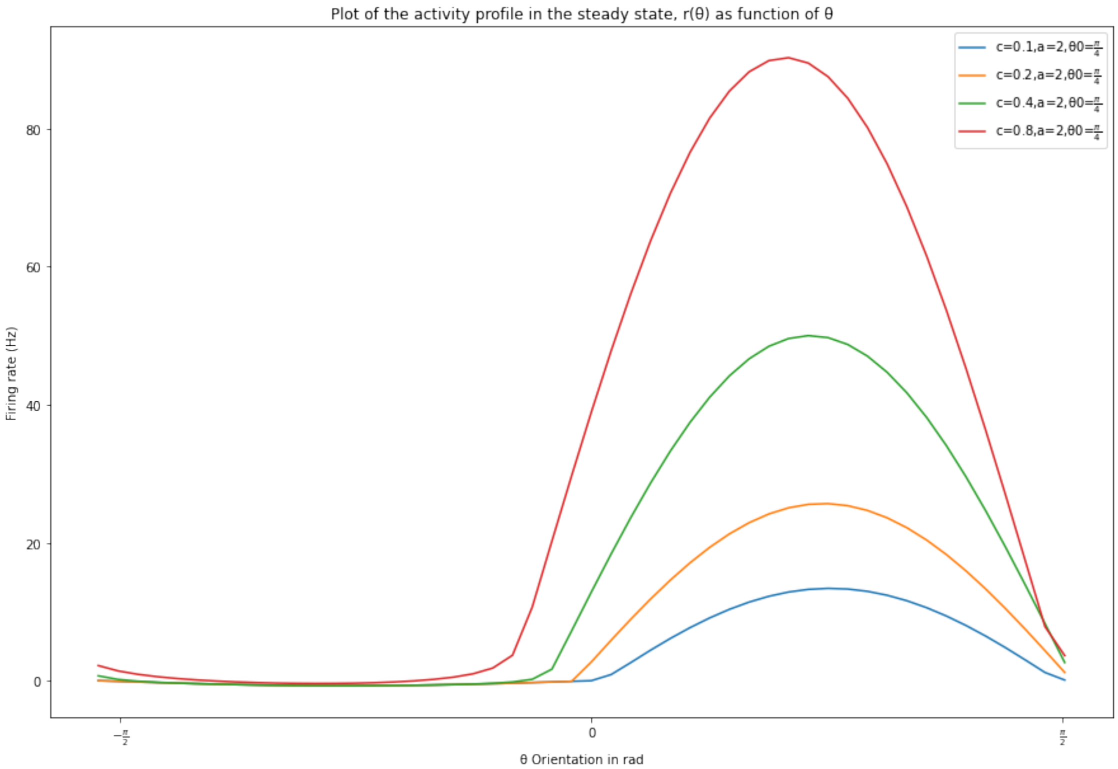
11 Like question 2 : Use J0 = −7.3, J2 = 11, A = 40 Hz and ϵ = 0.1.

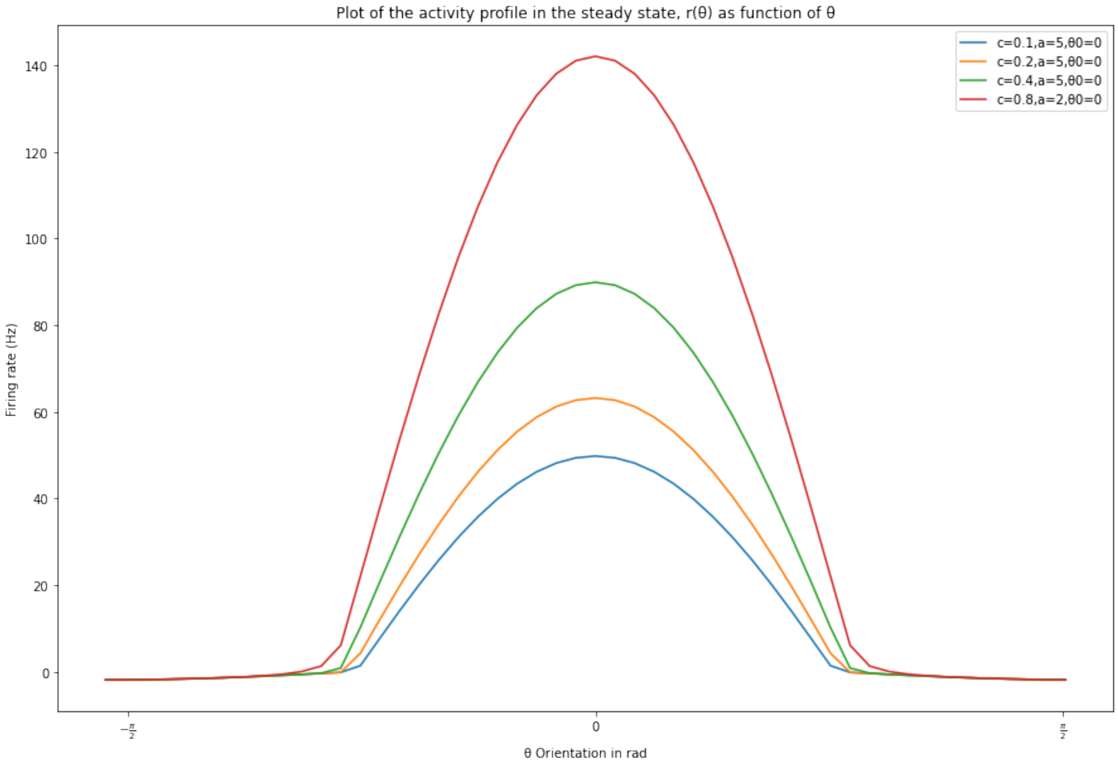
12 r( ) as function of

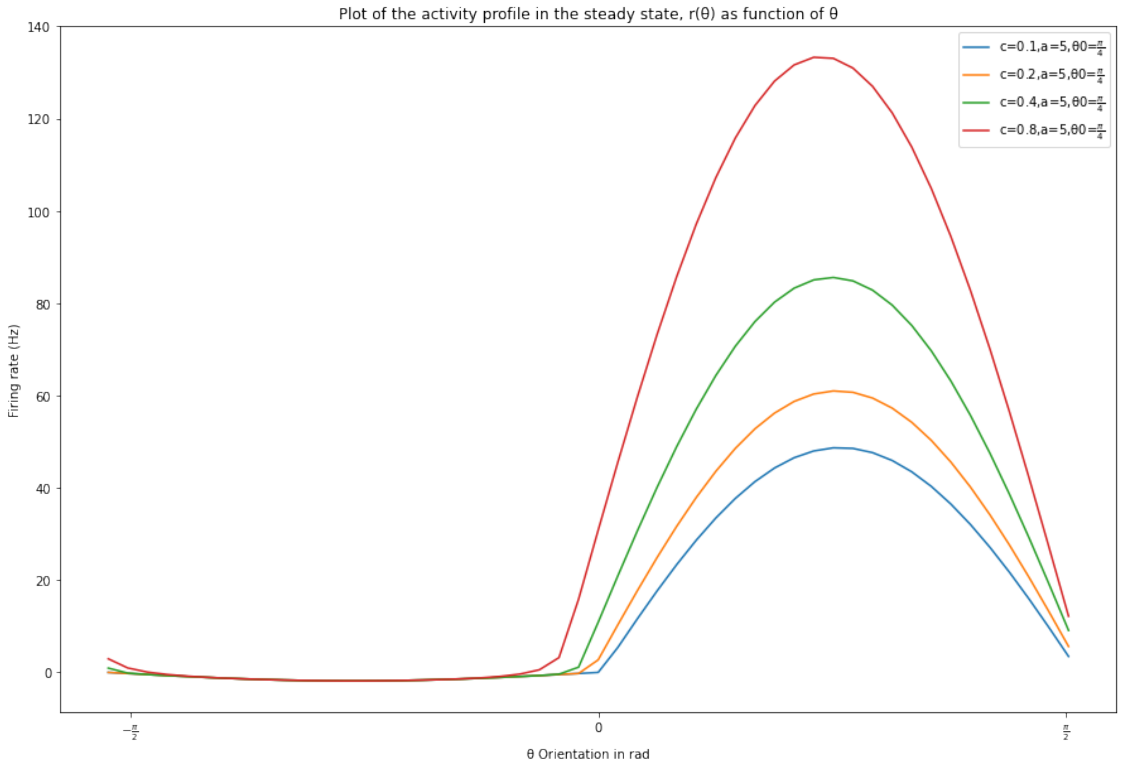
[54]:
```
N=50

plt.plot([(math.pi/N)*i -  math.pi/2 for i in range(0,N)],
         numerical_integrations_noise(N=N,J_0=-7.3,J_2=11,a=2,theta_0=0,c=0.
 ↪1,d_t=0.005,T=10),label='c=0.1,a=2, 0=0')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in range(0,N)],
         numerical_integrations_noise(N=N,J_0=-7.3,J_2=11,a=2,theta_0=0,c=0.
 ↪2,d_t=0.005,T=10),label='c=0.2,a=2, 0=0')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in range(0,N)],
         numerical_integrations_noise(N=N,J_0=-7.3,J_2=11,a=2,theta_0=0,c=0.
 ↪4,d_t=0.005,T=10),label='c=0.4,a=2, 0=0')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in range(0,N)],
         numerical_integrations_noise(N=N,J_0=-7.3,J_2=11,a=2,theta_0=0,c=0.
 ↪8,d_t=0.005,T=10),label='c=0.8,a=2, 0=0')
plt.xlabel('  Orientation in rad')
plt.ylabel('Firing rate (Hz)')
plt.legend()
plt.title("Plot of the activity profile in the steady state, r( ) as function␣
 ↪of  ")
plt.xticks([-1.5,0,1.5], [r'$-\frac{\pi}{2}$', '0', r'$\frac{\pi}{2}$'])
plt.show()

plt.plot([(math.pi/N)*i -  math.pi/2 for i in range(0,N)],
         numerical_integrations_noise(N=N,J_0=-7.3,J_2=11,a=2,theta_0=math.pi/
 ↪4,c=0.1,d_t=0.005,T=10),
         label='c=0.1,a=2, 0=' + r'$\frac{\pi}{4}$')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in range(0,N)],
         numerical_integrations_noise(N=N,J_0=-7.3,J_2=11,a=2,theta_0=math.pi/
 ↪4,c=0.2,d_t=0.005,T=10),
         label='c=0.2,a=2, 0=' + r'$\frac{\pi}{4}$')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in range(0,N)],
         numerical_integrations_noise(N=N,J_0=-7.3,J_2=11,a=2,theta_0=math.pi/
 ↪4,c=0.4,d_t=0.005,T=10),
         label='c=0.4,a=2, 0=' + r'$\frac{\pi}{4}$')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in range(0,N)],
         numerical_integrations_noise(N=N,J_0=-7.3,J_2=11,a=2,theta_0=math.pi/
 ↪4,c=0.8,d_t=0.005,T=10),
         label='c=0.8,a=2, 0=' + r'$\frac{\pi}{4}$')
```

```
plt.xlabel('  Orientation in rad')
plt.ylabel('Firing rate (Hz)')
plt.legend()
plt.title("Plot of the activity profile in the steady state, r( ) as function␣
 ↪of  ")
plt.xticks([-1.5,0,1.5], [r'$-\frac{\pi}{2}$', '0', r'$\frac{\pi}{2}$'])
plt.show()

plt.plot([(math.pi/N)*i -  math.pi/2 for i in range(0,N)],
         numerical_integrations_noise(N=N,J_0=-7.3,J_2=11,a=5,theta_0=0,c=0.
 ↪1,d_t=0.005,T=10),label='c=0.1,a=5, 0=0')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in range(0,N)],
         numerical_integrations_noise(N=N,J_0=-7.3,J_2=11,a=5,theta_0=0,c=0.
 ↪2,d_t=0.005,T=10),label='c=0.2,a=5, 0=0')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in range(0,N)],
         numerical_integrations_noise(N=N,J_0=-7.3,J_2=11,a=5,theta_0=0,c=0.
 ↪4,d_t=0.005,T=10),label='c=0.4,a=5, 0=0')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in range(0,N)],
         numerical_integrations_noise(N=N,J_0=-7.3,J_2=11,a=5,theta_0=0,c=0.
 ↪8,d_t=0.005,T=10),label='c=0.8,a=2, 0=0')
plt.xlabel('  Orientation in rad')
plt.ylabel('Firing rate (Hz)')
plt.legend()
plt.title("Plot of the activity profile in the steady state, r( ) as function␣
 ↪of  ")
plt.xticks([-1.5,0,1.5], [r'$-\frac{\pi}{2}$', '0', r'$\frac{\pi}{2}$'])
plt.show()

plt.plot([(math.pi/N)*i -  math.pi/2 for i in range(0,N)],
         numerical_integrations_noise(N=N,J_0=-7.3,J_2=11,a=5,theta_0=math.pi/
 ↪4,c=0.1,d_t=0.005,T=10),
         label='c=0.1,a=5, 0=' + r'$\frac{\pi}{4}$')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in range(0,N)],
         numerical_integrations_noise(N=N,J_0=-7.3,J_2=11,a=5,theta_0=math.pi/
 ↪4,c=0.2,d_t=0.005,T=10)
         ,label='c=0.2,a=5, 0=' + r'$\frac{\pi}{4}$')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in range(0,N)],
         numerical_integrations_noise(N=N,J_0=-7.3,J_2=11,a=5,theta_0=math.pi/
 ↪4,c=0.4,d_t=0.005,T=10),
         label='c=0.4,a=5, 0=' + r'$\frac{\pi}{4}$')
plt.plot([(math.pi/N)*i -  math.pi/2 for i in range(0,N)],
         numerical_integrations_noise(N=N,J_0=-7.3,J_2=11,a=5,theta_0=math.pi/
 ↪4,c=0.8,d_t=0.005,T=10),
         label='c=0.8,a=5, 0=' + r'$\frac{\pi}{4}$')
plt.xlabel('  Orientation in rad')
plt.ylabel('Firing rate (Hz)')
plt.legend()
plt.title("Plot of the activity profile in the steady state, r( ) as function␣
 ↪of  ")
plt.xticks([-1.5,0,1.5], [r'$-\frac{\pi}{2}$', '0', r'$\frac{\pi}{2}$'])
plt.show()
```




Plot of the activity profile in the steady state, r(θ) as function of θ
c=0.1,a=2,θ0=$\frac{\pi}{4}$
c=0.2,a=2,θ0=$\frac{\pi}{4}$
c=0.4,a=2,θ0=$\frac{\pi}{4}$
c=0.8,a=2,θ0=$\frac{\pi}{4}$
Firing rate (Hz)
80
60
40
20
0
$-\frac{\pi}{2}$
0
$\frac{\pi}{2}$
θ Orientation in rad


Plot of the activity profile in the steady state, r(θ) as function of θ
c=0.1,a=5,θ0=0
c=0.2,a=5,θ0=0
c=0.4,a=5,θ0=0
c=0.8,a=2,θ0=0
Firing rate (Hz)
140
120
100
80
60
40
20
0
$-\frac{\pi}{2}$
0
$\frac{\pi}{2}$
θ Orientation in rad

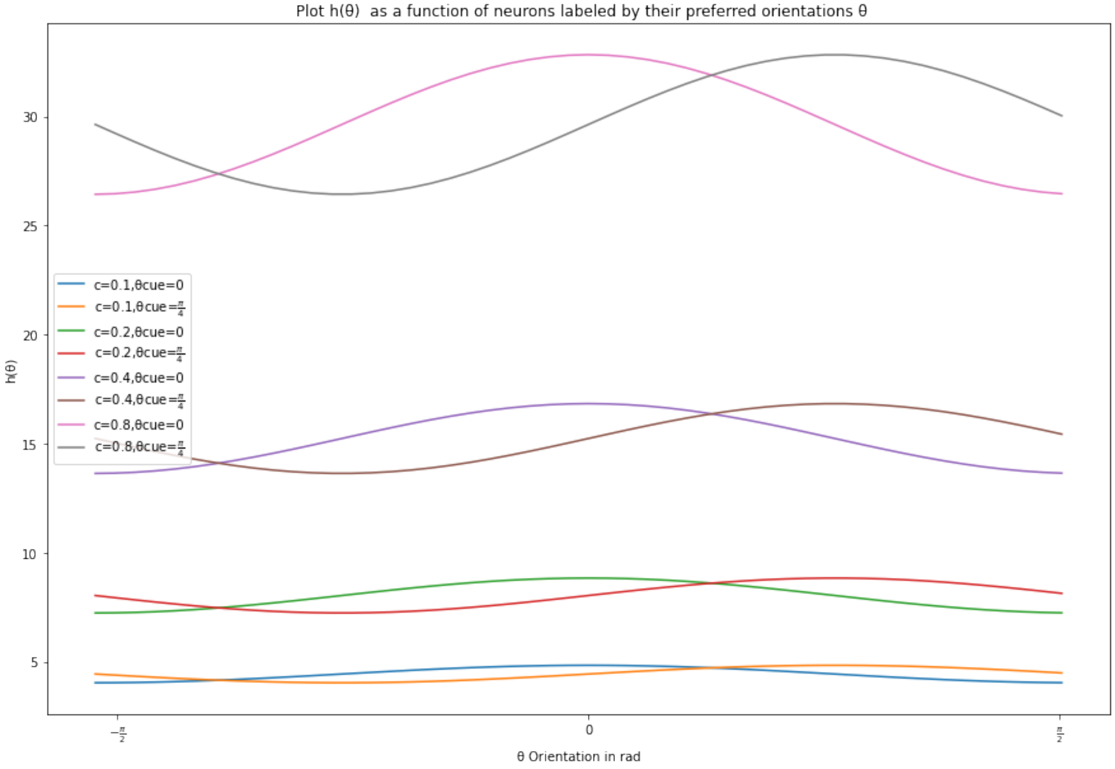


13 h( ) as function of

14 a) h with noise

```
[55]: thetas = [(math.pi/N)*i -  math.pi/2 for i in range(0,N)]
      plt.plot(thetas,[h_noise(theta=theta,c=0.1,A=40,eps=0.1,theta_cue=0) for theta␣
       ↪in thetas],label='c=0.1, cue=0')
      plt.plot(thetas,[h_noise(theta=theta,c=0.1,A=40,eps=0.1,theta_cue=math.pi/4)␣
       ↪for theta in thetas],
               label='c=0.1, cue=' + r'$\frac{\pi}{4}$')
      plt.plot(thetas,[h_noise(theta=theta,c=0.2,A=40,eps=0.1,theta_cue=0) for theta␣
       ↪in thetas],label='c=0.2, cue=0')
      plt.plot(thetas,[h_noise(theta=theta,c=0.2,A=40,eps=0.1,theta_cue=math.pi/4)␣
       ↪for theta in thetas],
               label='c=0.2, cue=' + r'$\frac{\pi}{4}$')
      plt.plot(thetas,[h_noise(theta=theta,c=0.4,A=40,eps=0.1,theta_cue=0) for theta␣
       ↪in thetas],label='c=0.4, cue=0')
      plt.plot(thetas,[h_noise(theta=theta,c=0.4,A=40,eps=0.1,theta_cue=math.pi/4)␣
       ↪for theta in thetas],
               label='c=0.4, cue=' + r'$\frac{\pi}{4}$')
```

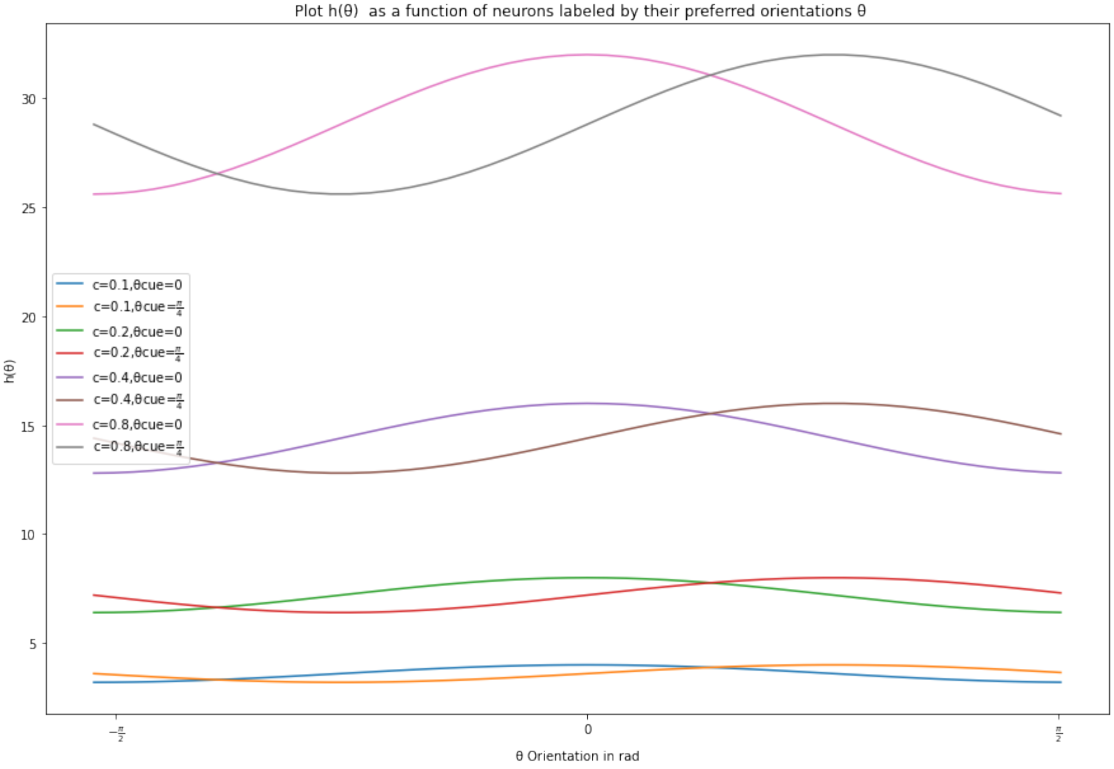
```
plt.plot(thetas,[h_noise(theta=theta,c=0.8,A=40,eps=0.1,theta_cue=0) for theta␣
 ↪in thetas],label='c=0.8, cue=0')
plt.plot(thetas,[h_noise(theta=theta,c=0.8,A=40,eps=0.1,theta_cue=math.pi/4)␣
 ↪for theta in thetas],
         label='c=0.8, cue=' + r'$\frac{\pi}{4}$')
plt.legend()
plt.xlabel('  Orientation in rad')
plt.ylabel('h( )')
plt.title(" Plot h( )  as a function of neurons labeled by their preferred␣
 ↪orientations  ")
plt.xticks([-1.5,0,1.5], [r'$-\frac{\pi}{2}$', '0', r'$\frac{\pi}{2}$'])
plt.show()
```



15 b) h without noise

[60]:
```
thetas = [(math.pi/N)*i -  math.pi/2 for i in range(0,N)]
plt.plot(thetas,[h(theta=theta,c=0.1,A=40,eps=0.1,theta_cue=0) for theta in␣
 ↪thetas],label='c=0.1, cue=0')
plt.plot(thetas,[h(theta=theta,c=0.1,A=40,eps=0.1,theta_cue=math.pi/4) for␣
 ↪theta in thetas],
         label='c=0.1, cue=' + r'$\frac{\pi}{4}$')
```

```
plt.plot(thetas,[h(theta=theta,c=0.2,A=40,eps=0.1,theta_cue=0) for theta in␣
 ↪thetas],label='c=0.2, cue=0')
plt.plot(thetas,[h(theta=theta,c=0.2,A=40,eps=0.1,theta_cue=math.pi/4) for␣
 ↪theta in thetas],
         label='c=0.2, cue=' + r'$\frac{\pi}{4}$')
plt.plot(thetas,[h(theta=theta,c=0.4,A=40,eps=0.1,theta_cue=0) for theta in␣
 ↪thetas],label='c=0.4, cue=0')
plt.plot(thetas,[h(theta=theta,c=0.4,A=40,eps=0.1,theta_cue=math.pi/4) for␣
 ↪theta in thetas],
         label='c=0.4, cue=' + r'$\frac{\pi}{4}$')
plt.plot(thetas,[h(theta=theta,c=0.8,A=40,eps=0.1,theta_cue=0) for theta in␣
 ↪thetas],label='c=0.8, cue=0')
plt.plot(thetas,[h(theta=theta,c=0.8,A=40,eps=0.1,theta_cue=math.pi/4) for␣
 ↪theta in thetas],
         label='c=0.8, cue=' + r'$\frac{\pi}{4}$')
plt.legend()
plt.xlabel('  Orientation in rad')
plt.ylabel('h( )')
plt.title(" Plot h( )  as a function of neurons labeled by their preferred␣
 ↪orientations  ")
plt.xticks([-1.5,0,1.5], [r'$-\frac{\pi}{2}$', '0', r'$\frac{\pi}{2}$'])
plt.show()
```



16 Question 4 : Use ϵ =0 , $J_0 = -1$ and $J_2 = 1, 1.5, 2.5, 4.5$

16.0.1 This parameter ϵ explains how well h is tuned . The greater ϵ the better the thalamic input is tuned (feedforward model).

16.0.2 Indeed for $\epsilon = 0$, the input is flat and tuning is non existant.

16.0.3 When $\epsilon = 0$, there is no stimulus orientation since we have a constant input h.

16.0.4 The parameter space is divided into three domains.

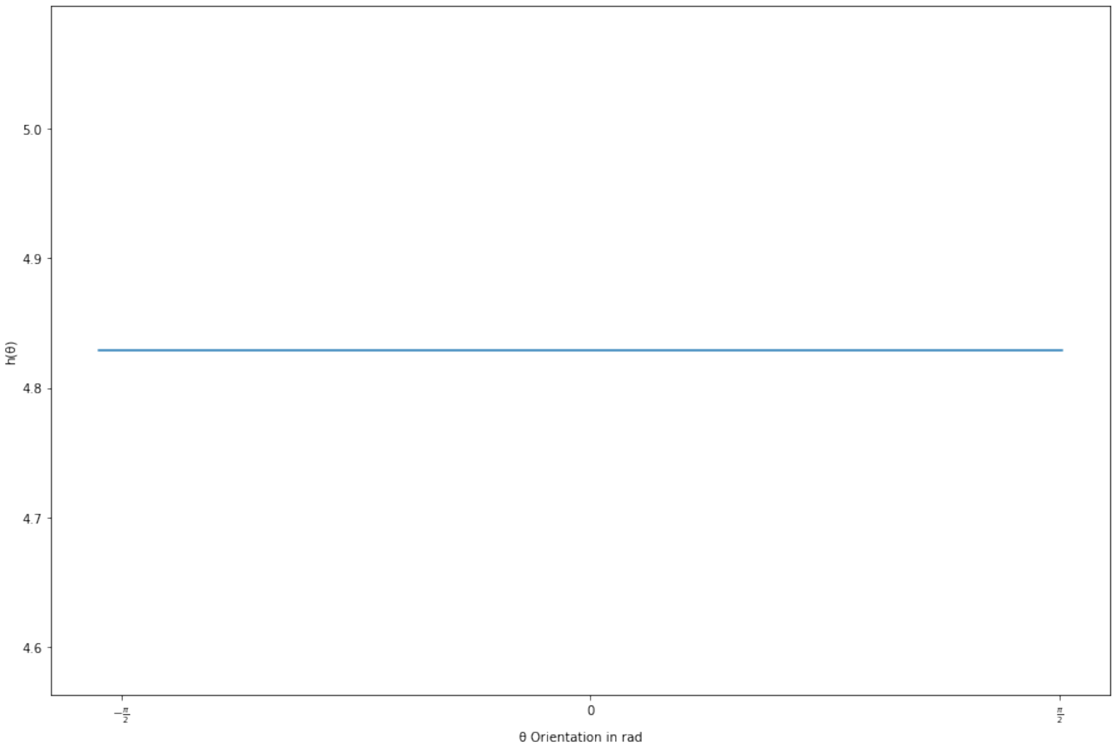
16.0.5 When $\epsilon = 0$, for fixed $J_o = -1$ and $J_2 \leq 2$, we have lots of inhibition and a little bit of excitation and therefrore the firing rate will be almost constant across the network model : homogeneous state. However, when we change ϵ , this does not hold anymore because the input is tuned.

16.0.6 When $\epsilon = 0$, for fixed $J_o = -1$ and $J_2 > 1$, we have a more balanced regime where we have strong recurrent inhibition as well as strong recurrent excitation : marginal .

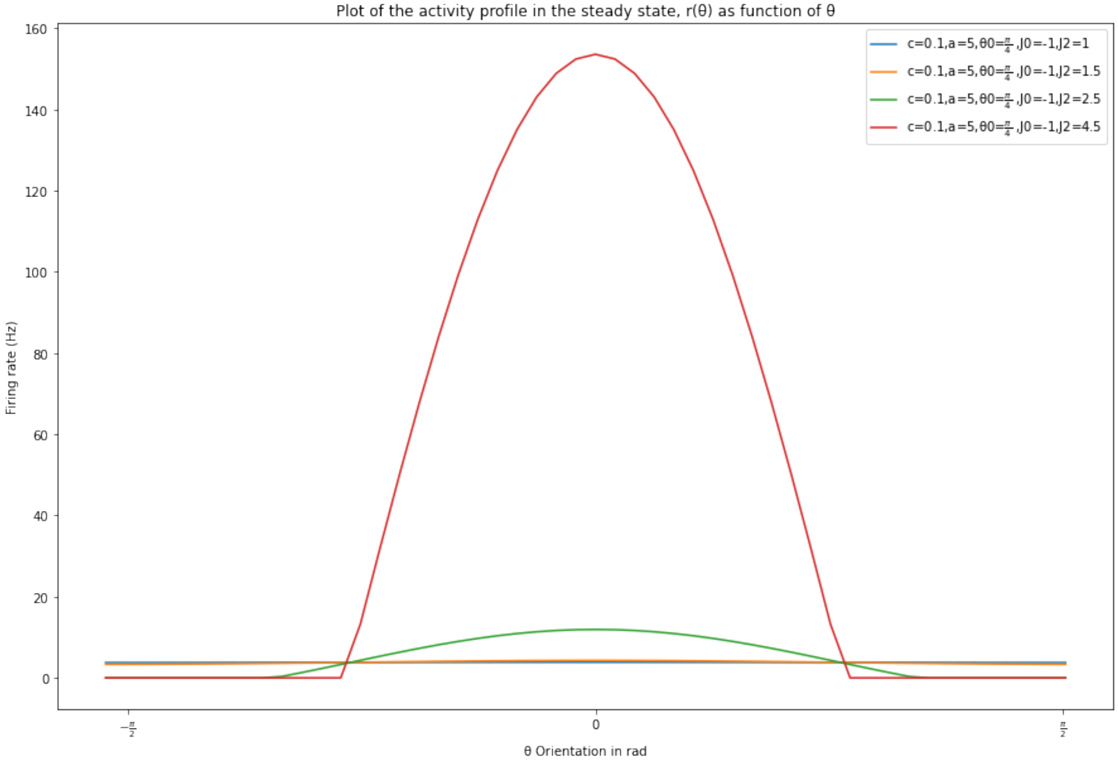
16.0.7 First let's show that h is fixed when $\epsilon = 0$

[56]:

```
N=50
thetas = [(math.pi/N)*i -  math.pi/2 for i in range(0,N)]
plt.plot(thetas,[h_noise(theta=theta,c=0.1,A=40,eps=0,theta_cue=0) for theta in thetas],label='c=0.1, cue=0')
plt.xlabel('  Orientation in rad')
plt.ylabel('h( )')
plt.xticks([-1.5,0,1.5], [r'$-\frac{\pi}{2}$', '0', r'$\frac{\pi}{2}$'])
plt.show()
```



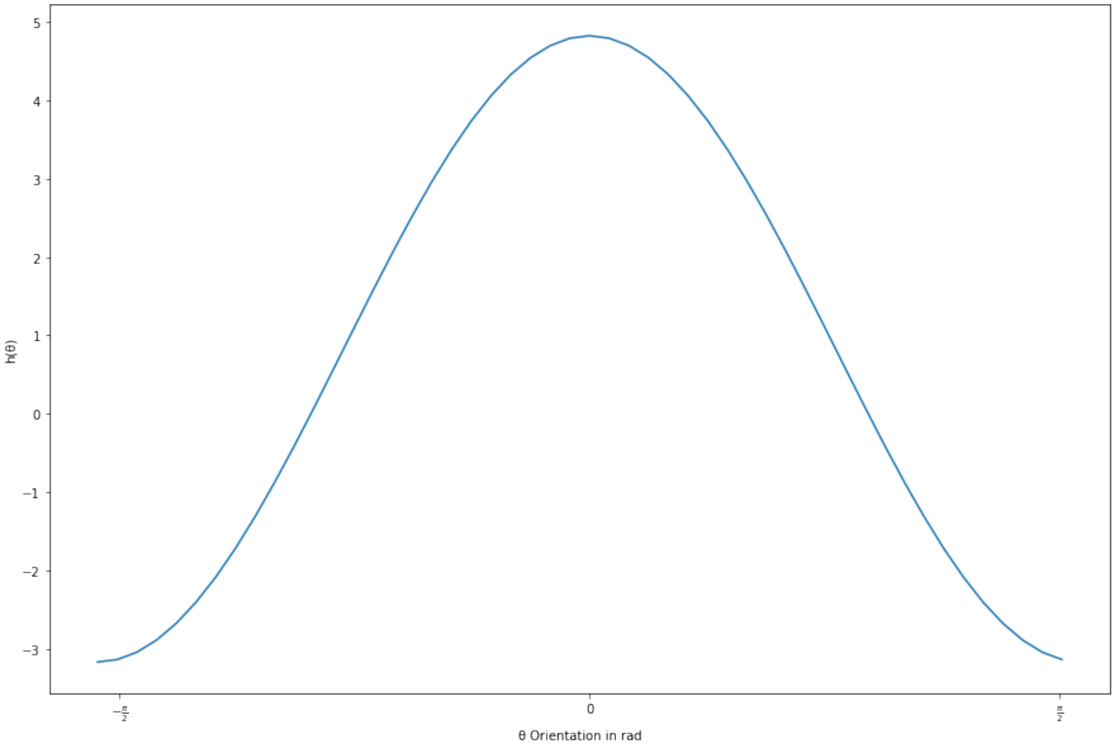
```
[57]: for j2 in [1,1.5,2.5,4.5]:
          plt.plot([(math.pi/N)*i -  math.pi/2 for i in range(0,N)],
               numerical_integrations_noise(N=N,J_0=-1,J_2=j2,a=5,theta_0=0,c=0.
       ↪1,d_t=0.05,T=100,eps=0),
                   label='c=0.1,a=5, 0='+ r'$\frac{\pi}{4}$'+' ,J0=-1,J2='+str(j2))
      plt.xlabel('  Orientation in rad')
      plt.ylabel('Firing rate (Hz)')
      plt.legend()
      plt.title("Plot of the activity profile in the steady state, r( ) as function␣
       ↪of  ")
      plt.xticks([-1.5,0,1.5], [r'$-\frac{\pi}{2}$', '0', r'$\frac{\pi}{2}$'])
      plt.show()
```



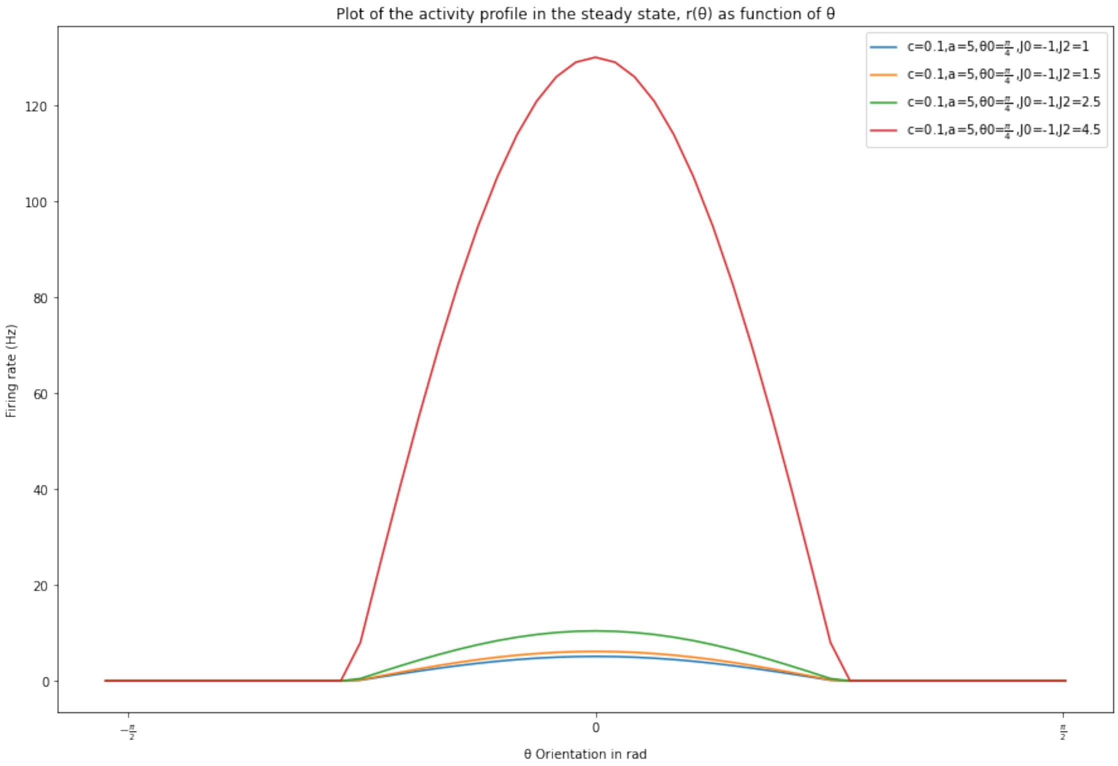
16.0.8 Flat firing for J2 <2 as expected.

16.0.9 Let's try for $\epsilon = 1$ when the thalamic input is tuned.

```
[58]: N=50
      thetas = [(math.pi/N)*i -  math.pi/2 for i in range(0,N)]
      plt.plot(thetas,[h_noise(theta=theta,c=0.1,A=40,eps=1,theta_cue=0) for theta in␣
       ↪thetas],label='c=0.1, cue=0')
      plt.xlabel('  Orientation in rad')
      plt.ylabel('h( )')
      plt.xticks([-1.5,0,1.5], [r'$-\frac{\pi}{2}$', '0', r'$\frac{\pi}{2}$'])
      plt.show()
```



```
[59]: for j2 in [1,1.5,2.5,4.5]:
          plt.plot([(math.pi/N)*i -  math.pi/2 for i in range(0,N)],
               numerical_integrations_noise(N=N,J_0=-1,J_2=j2,a=5,theta_0=0,c=0.
       ↪1,d_t=0.05,T=100,eps=1),
                   label='c=0.1,a=5, 0='+ r'$\frac{\pi}{4}$'+' ,J0=-1,J2='+str(j2))
      plt.xlabel('  Orientation in rad')
      plt.ylabel('Firing rate (Hz)')
      plt.legend()
      plt.title("Plot of the activity profile in the steady state, r( ) as function␣
       ↪of  ")
      plt.xticks([-1.5,0,1.5], [r'$-\frac{\pi}{2}$', '0', r'$\frac{\pi}{2}$'])
      plt.show()
```



16.0.10 We can see that the regime have changed when the input is tuned with $\epsilon = 1$. Now, we do not have this flat firing for J2<2

[ ]: