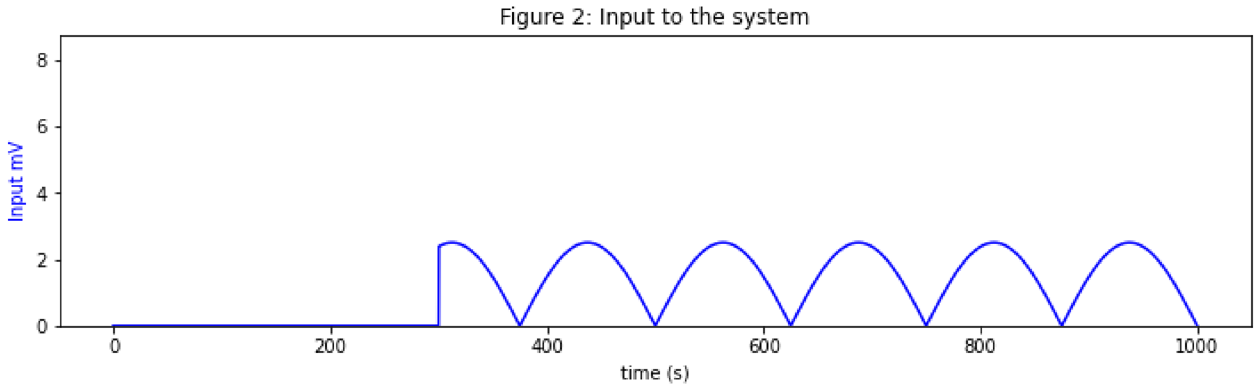
```
import numpy as np
import sys
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings("ignore")
```

```
h=0.3
input_onset=300
input_amp=2.5
f=[4,28]
time=np.arange(0,1000.1,h)
```

```
def sinus(f):
  return np.abs(input_amp*np.sin(2*np.pi*f*time/1000))
```

```
def Input(input_onset,sinus):
  I=np.zeros((len(sinus)))
  for k in range (0,len(sinus)):
    if time[k] >input_onset:
      I[k]=sinus[k]
  return I
```

```
fig, ax1 = plt.subplots(figsize=(12,3))
ax1.plot(time, Input(input_onset,sinus(f[0])), 'b-')
ax1.set_xlabel('time (s)')

ax1.set_ylabel('Input mV', color='b')
ax1.set_ylim(0,input_amp*3.5)
plt.title('Figure 2: Input to the system')
plt.show()
```



```
fig, ax1 = plt.subplots(figsize=(12,3))
ax1.plot(time, Input(input_onset,sinus(f[1])), 'b-')
ax1.set_xlabel('time (s)')

ax1.set_ylabel('Input mV', color='b')
ax1.set_ylim(0,input_amp*3.5)
plt.title('Figure 2: Input to the system')
plt.show()
```

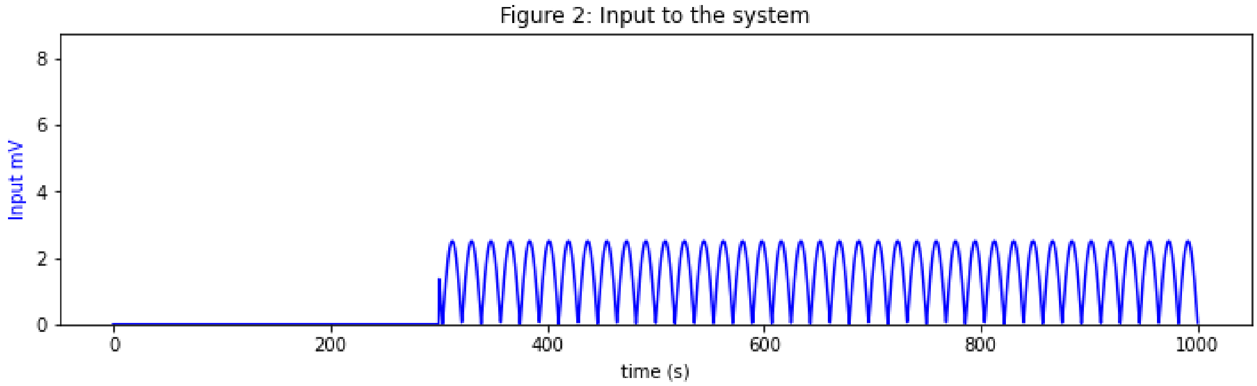


Figure 2: Input to the system

```
fig, ax1 = plt.subplots(figsize=(12,3))
ax1.plot(time, Input(input_onset,sinus(f[0])+sinus(f[1])), 'b-')
ax1.set_xlabel('time (s)')

ax1.set_ylabel('Input mV', color='b')
ax1.set_ylim(0,input_amp*3.5)
plt.title('Figure 2: Input to the system')
plt.show()
```

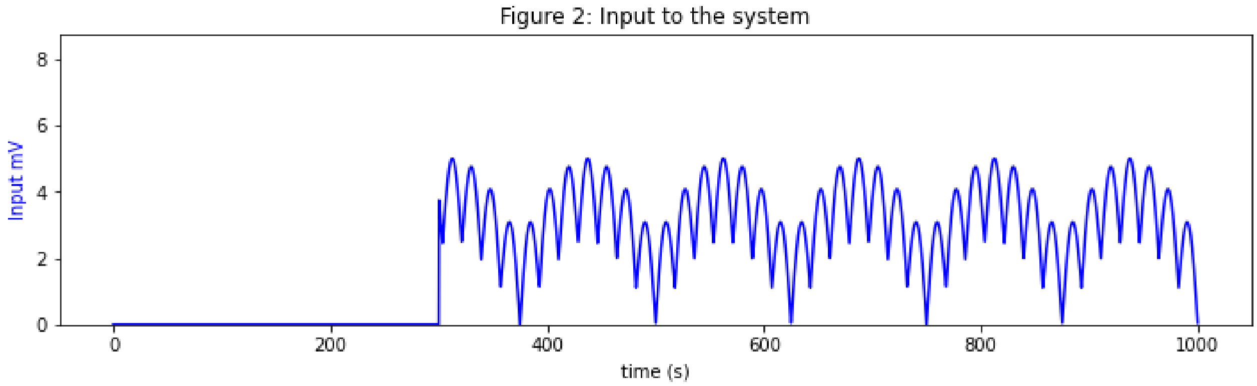


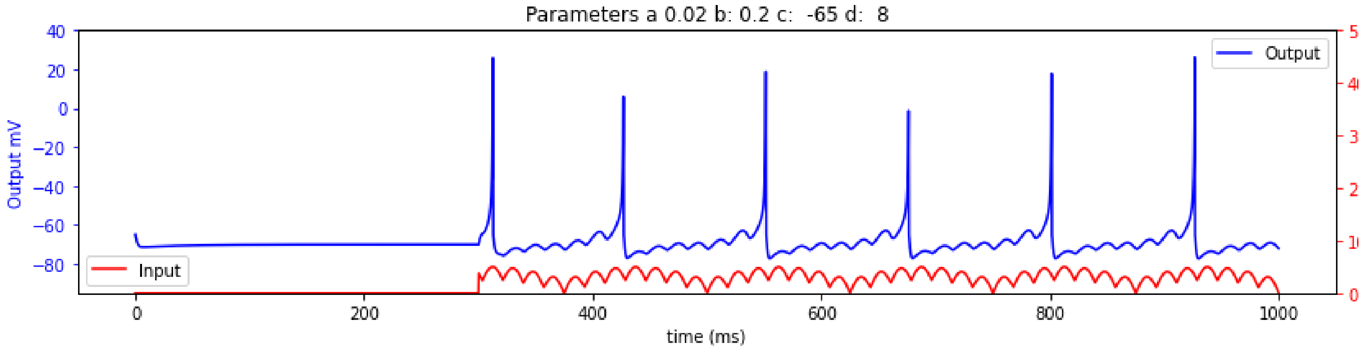
Figure 2: Input to the system

```
def Discrete_Model(a,b,u,v,I):
    v = v + h *(0.04*v*v+5*v+140-u+I)
    u = u + h *(a*(b*v-u))
    return u,v
```

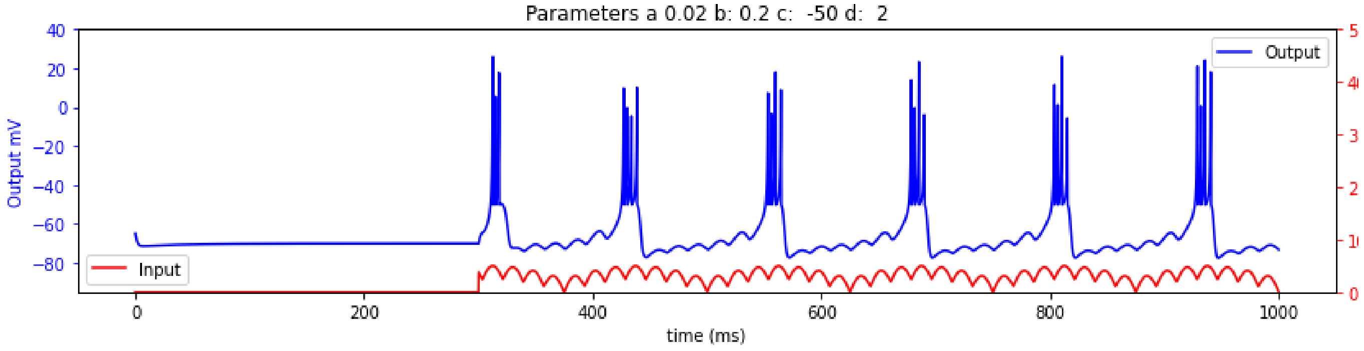
```
def Izhikevich(a,b,c,d):
    v=-65*np.ones((len(time)))
    u=0*np.ones((len(time)))
    u[0]=b*v[0]

    spiketime=[]
    fired=[]
    I=Input(input_onset,sinus(f[0])+sinus(f[1]))
    # EULER METHOD
    for k in range (0,len(time)-1):
        u[k+1],v[k+1]=Discrete_Model(a,b,u[k],v[k],I[k])

        if v[k+1]>30:
            v[k+1]=c
            u[k+1]=u[k+1]+d
    plot_input_output(time,v,I,a,b,c,d)
```

```
def plot_input_output(time,v,I,a,b,c,d):
    fig, ax1 = plt.subplots(figsize=(12,3))
    ax1.plot(time, v, 'b-', label = 'Output')
    ax1.set_xlabel('time (ms)')
    # Make the y-axis label, ticks and tick labels match the line color.
    # Plotting out put
    ax1.set_ylabel('Output mV', color='b')
    ax1.tick_params('y', colors='b')
    ax1.set_ylim(-95,40)
    ax2 = ax1.twinx()
    # Plotting input on a different axis
    ax2.plot(time, I, 'r', label = 'Input')
    ax2.set_ylim(0,input_amp*20)
    ax2.set_ylabel('Input (mV)', color='r')
    ax2.tick_params('y', colors='r')

    fig.tight_layout()
    ax1.legend(loc=1)
    ax2.legend(loc=3)
    ax1.set_title('Parameters a %s b: %s c:  %s d:  %s' %(a,b,c,d))
    plt.show()
```

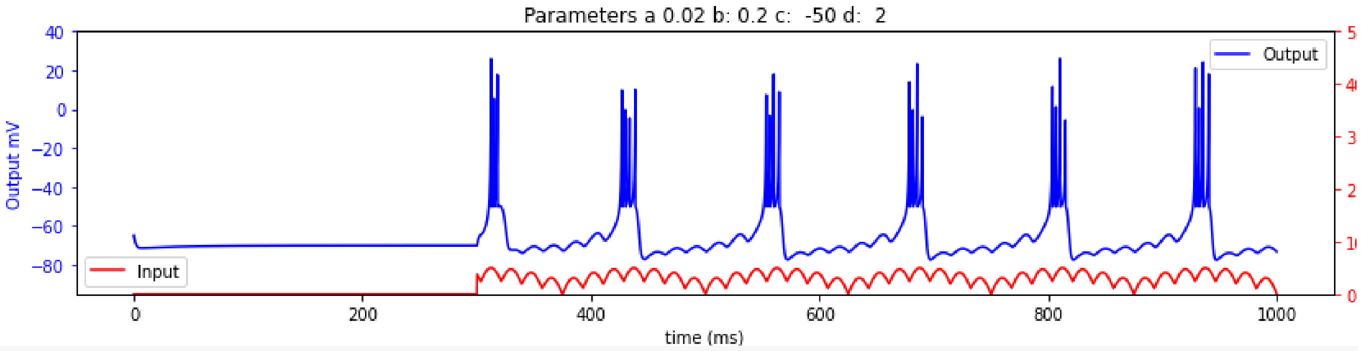
```
Izhikevich(0.02,0.2,-65,8) #RS
```



```
Izhikevich(0.02,0.2,-50,2) #IB
```

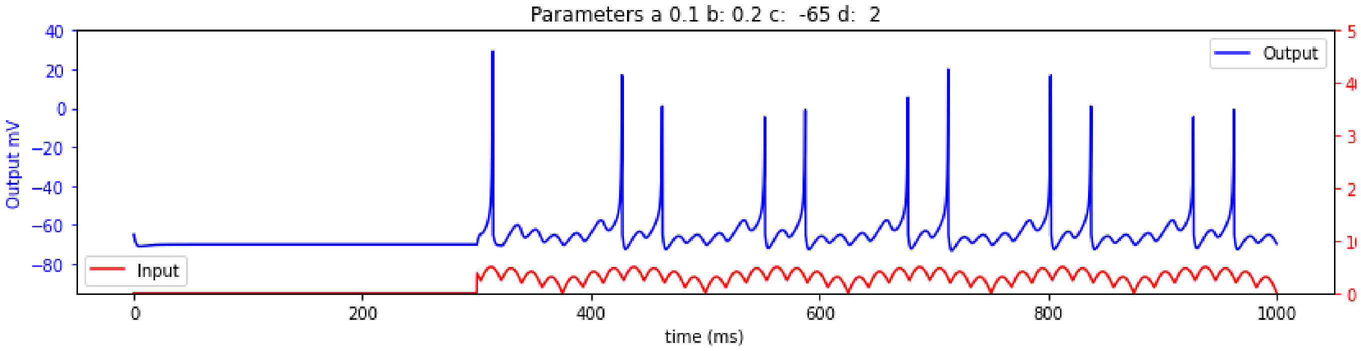


```
Izhikevich(0.02,0.2,-50,2) #CH
```



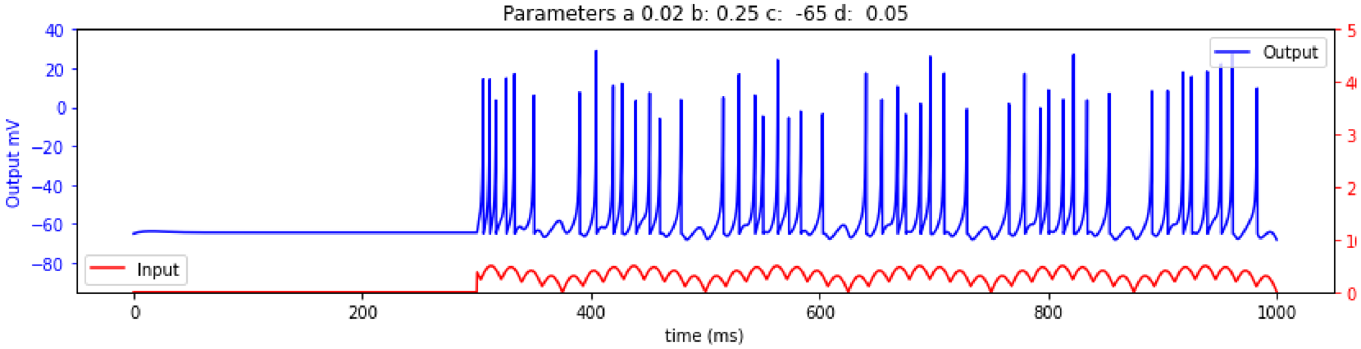
Parameters a 0.02 b: 0.2 c: -50 d: 2

```
Izhikevich(0.1,0.2,-65,2) #FS
```



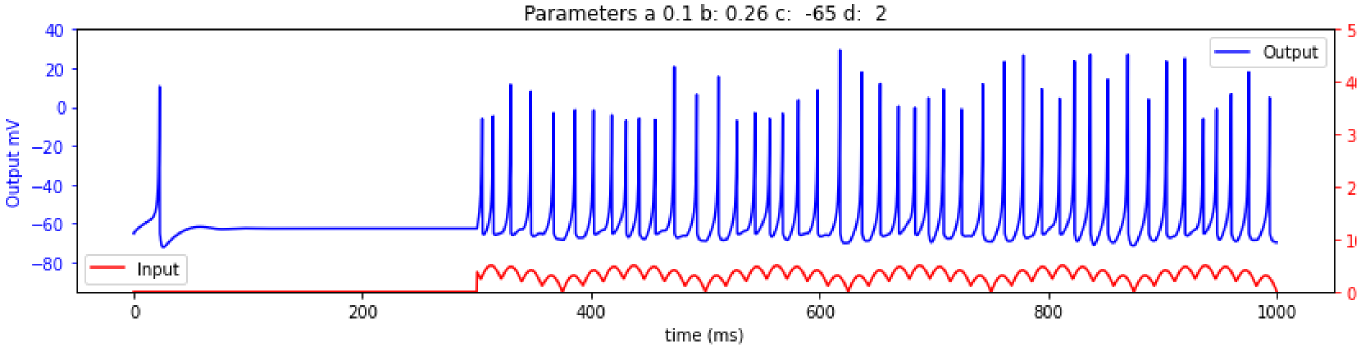
Parameters a 0.1 b: 0.2 c: -65 d: 2

```
Izhikevich(0.02,0.25,-65,0.05) #TC
```



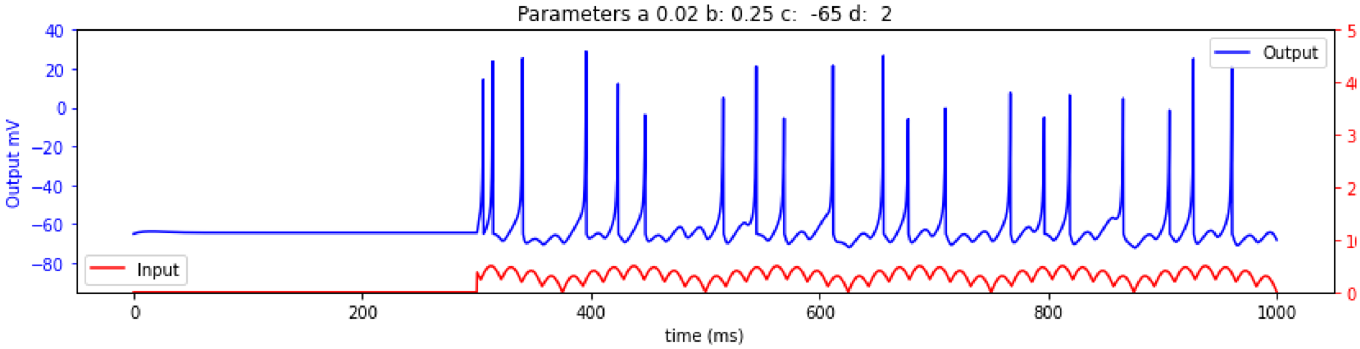
Parameters a 0.02 b: 0.25 c: -65 d: 0.05

```
Izhikevich(0.1,0.26,-65,2) #RZ
```

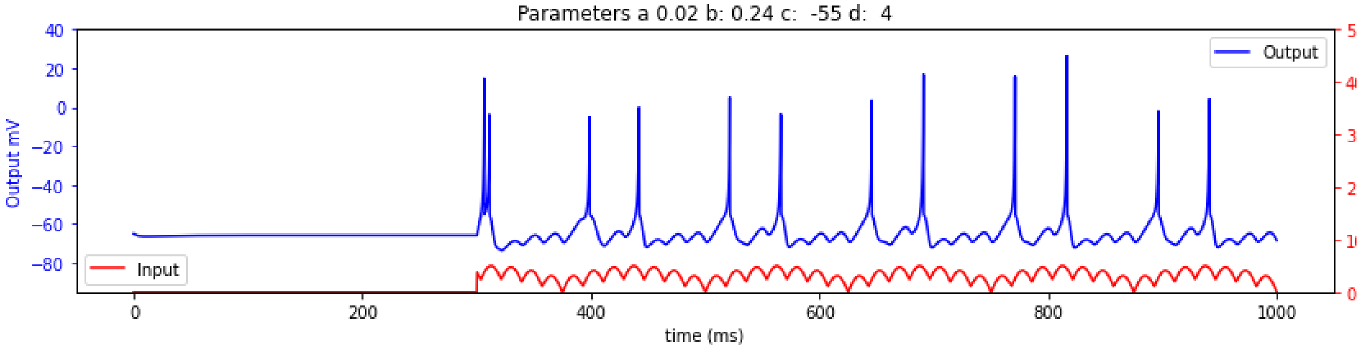


Parameters a 0.1 b: 0.26 c: -65 d: 2

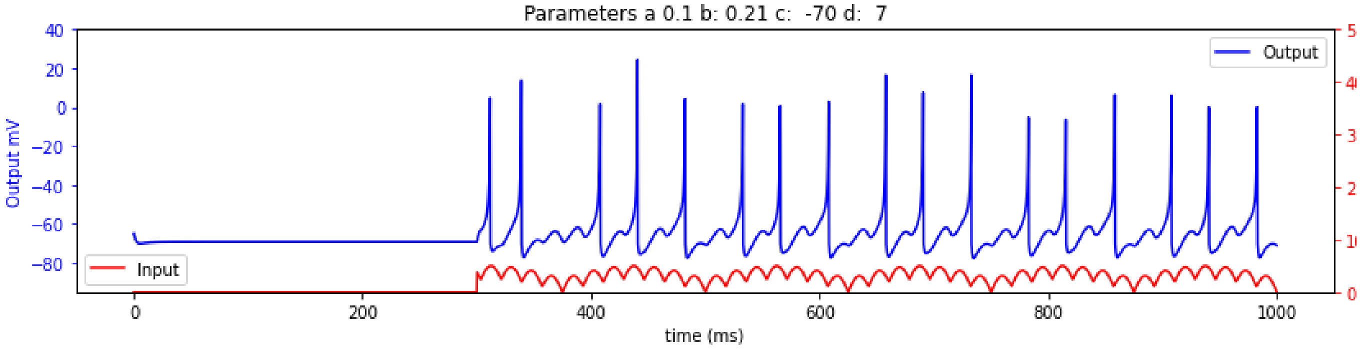
```
Izhikevich(0.02,0.25,-65,2) #LTS
```



```
Izhikevich(0.02,0.24,-55,4)
```



```
Izhikevich(0.1,0.21,-70,7)
```


Parameters a 0.1 b: 0.21 c: -70 d: 7
Output mV
Output
Input
time (ms)