Assignment 2
Neuroscience of Learning, Memory, Cognition
Dr. Karbalaei Aghajan

Mohammad Hossein Shafizadegan
99104781

March 30, 2023

Contents

1 Part I : FitzHugh-Nagumo Model

1.1 Explaining model and it's parameters

The FHN model consists of two variables: V, which represents the membrane potential; and w, which represents a recovery variable that accounts for the slow adaptation processes in neurons. Voltage dynamics can be approximated by a cubic function while dynamics of are linear in V and w.
The dynamics of these variables are governed by two coupled differential equations:

$$\dot{V} = V(a - V)(V - 1) - w + I_{ext}$$
$$\tau\dot{w} = V + c - bw$$

where τ is the time constant of w; a , b and c are constants that affect the shape of nullclines.

The V-nullcline is given by $w = V(a - V)(V - 1) + I_{ext}$, which is a cubic curve. The w-nullcline is given by $w = \frac{V+c}{b}$, which is a straight line with slope $\frac{1}{b}$.

The intersection points between these nullclines determine the fixed points or equilibrium states of the system. Depending on a, b, c and τ,the system can have one or three fixed points.

The FHN model exhibits various types of behaviors depending on its parameters and initial conditions. Some examples are:

- When slope of w-nullcline is greater than one ($b < 1$), there is only one fixed point. As input increases, the V-nullcline moves up therefor, fixed point moves to the right which makes the trajectories converge to a limit cycle.
- If the slope of w-nullcline is smaller than one ($b > 1$), there can be three fixed points based on different values of a and c. With positive input, the V-nullcline moves upward, so the left stable fixed point and the saddle merge and disappear while the right fixed point remains stable

In this question, the model parameters are as follows :

$$a = 1 \qquad \tau = 12.5 \qquad c = 0.7 \qquad b = 0.8$$

Since $b < 1$ then, there is only one fixed point in our case.

1.2 Simulation

Using python and odeint which is a builtin function for solving differential equations we will solve the coupled differential equation of the FitzHugh-Nagumo model.

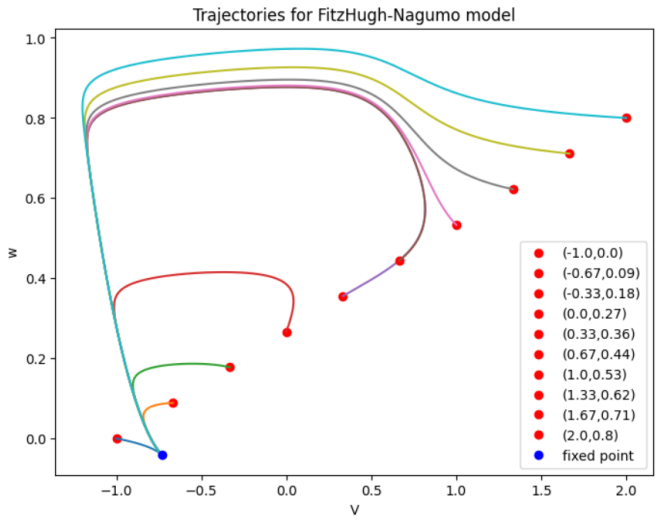
First, we will define a function called "FitzHugh_Nagumo" as follows :

```
def diff_equations(y, t, Iext):
    V,w = y # unpack state variables
    dVdt = V - V**3 - w + Iext
    dwdt = 0.08 * (V + 0.7 - 0.8*w)
    return [dVdt, dwdt]

def FitzHugh_Nagumo(init_v, init_w, Iext, T0):
    dt = 0.01
    T  = math.ceil(T0/dt) # [ms]
    t = np.arange(0,T)*dt
    y = odeint(diff_equations, [init_v, init_w], t, args=(Iext,))
    return y[:,0], y[:,1]
```

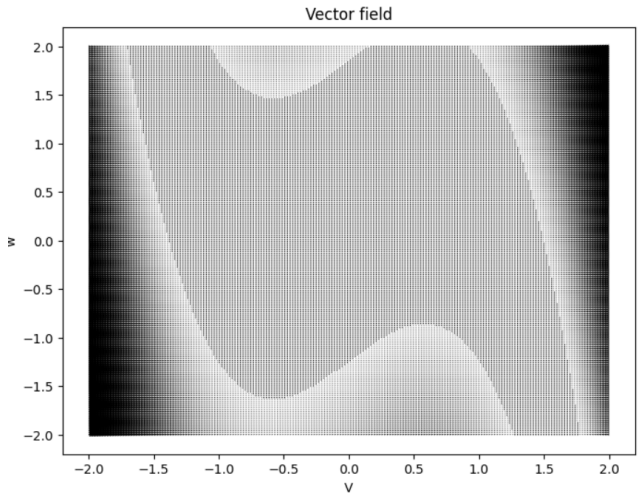
In the above code, we have defined the model differential equations in the "diff_equations" function and then it will be used as input argument needed to solve the differential equation using "odeint" function.

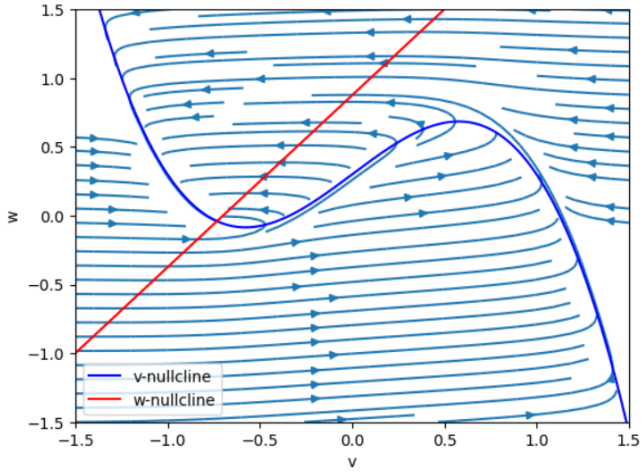
For $I_{ext} = 0.3$ we have plotted the trajectories for ten initial points.

```
    init_v = np.linspace(-1,2,10)
    init_w = np.linspace(0,0.8,10)
    Iext = 0.3
    T0 = 100

    for i in range(10):
        v, w = FitzHugh_Nagumo(init_v[i],init_w[i],Iext,T0)
        plt.plot(init_v[i],init_w[i],'ro',label="(" + str(round(init_v[i],2))+","+str(round(init_w[i],2))
            + ")")
        plt.plot(v,w)
```



Using the python "quiver" and "streamplot" functions we will also plot the corresponding vector filed , streamlines and nullclines.





1.3

$\dot{V} = V - V^3 - w + 0.3 \quad , \quad \dot{w} = 0.08(V + 0.7 - 0.8w)$

$V - \text{nullcline} : w = V - V^3 + 0.3 \quad , \quad w - \text{nullcline} : w = \dfrac{V + 0.7}{0.8}$

Finding nullclines intersections : $\Rightarrow V - V^3 + 0.3 = \dfrac{V + 0.7}{0.8} \Rightarrow P \cong (-0.732, -0.04)$

Forming the Jacubian matrix : $L = \begin{bmatrix} 1 - 3V^2\big|_{V=-0.732} & -1 \\ 0.08 & -0.064 \end{bmatrix} = \begin{bmatrix} -0.607 & -1 \\ 0.08 & -0.064 \end{bmatrix}$

$\tau = trace(L) = -0.671 \quad , \quad \Delta = det(L) \cong 0.12$

$\lambda_1 = \dfrac{\tau + \sqrt{\tau^2 - 4\Delta}}{2}$, $\lambda_2 = \dfrac{\tau - \sqrt{\tau^2 - 4\Delta}}{2} \Rightarrow \lambda_1 = -0.3355 + j0.08625$, $\lambda_2 = -0.3355 - j0.08625$

The eigenvalues are complex conjugate with negative real parts so the fixed point is a kind of a stable focus one.

The results completely match the trajectories plotted and shown in the previous part. As it can be clearly seen in the trajectories plot, all of the trajectories start from their initial point and after traveling a pseudo circular path, they have all reached the fixed point with coordinates of $(-0.732, -0.04)$

1.4

As we change the value of input current, the V-nullcline wil move upward or downward. so the value of the V for the fixed point will change. To see how this will affect the dynamics behavior, we first check the effects of V.

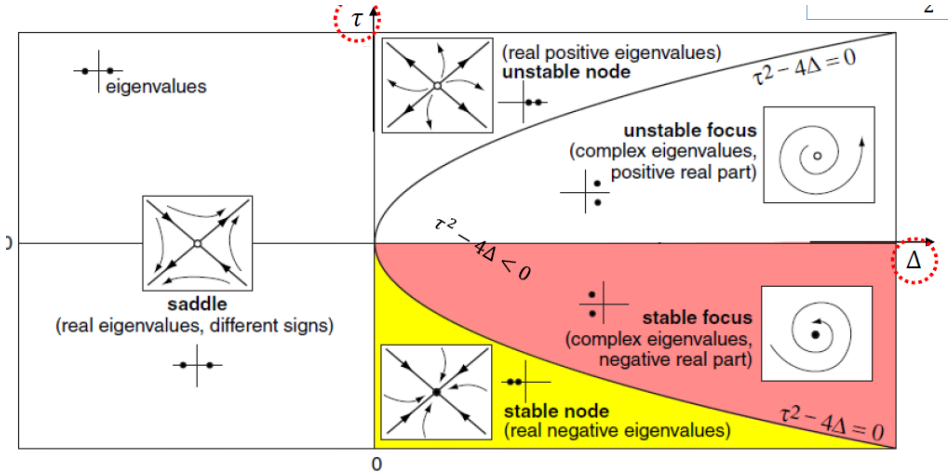
$L = \begin{bmatrix} 1 - 3V^2 & -1 \\ 0.08 & -0.064 \end{bmatrix} \Rightarrow \tau = 0.936 - 3V^2 \quad , \quad \Delta = 0.192V^2 + 0.016$

for $\forall V \ : \ \Delta > 0$ $\quad \begin{cases} \tau > 0 & V \in (-0.558, 0.558) \\ \tau \leq 0 & \text{O.W.} \end{cases}$

$\tau^2 - 4\Delta = 9v^4 - 6.384v^2 + 0.812096 \Rightarrow \begin{cases} \tau^2 - 4\Delta < 0 & V \in (-0.737,\ -0.407) \cup (0.407,\ 0.737) \\ \tau^2 - 4\Delta \geq 0 & \text{O.W.} \end{cases}$

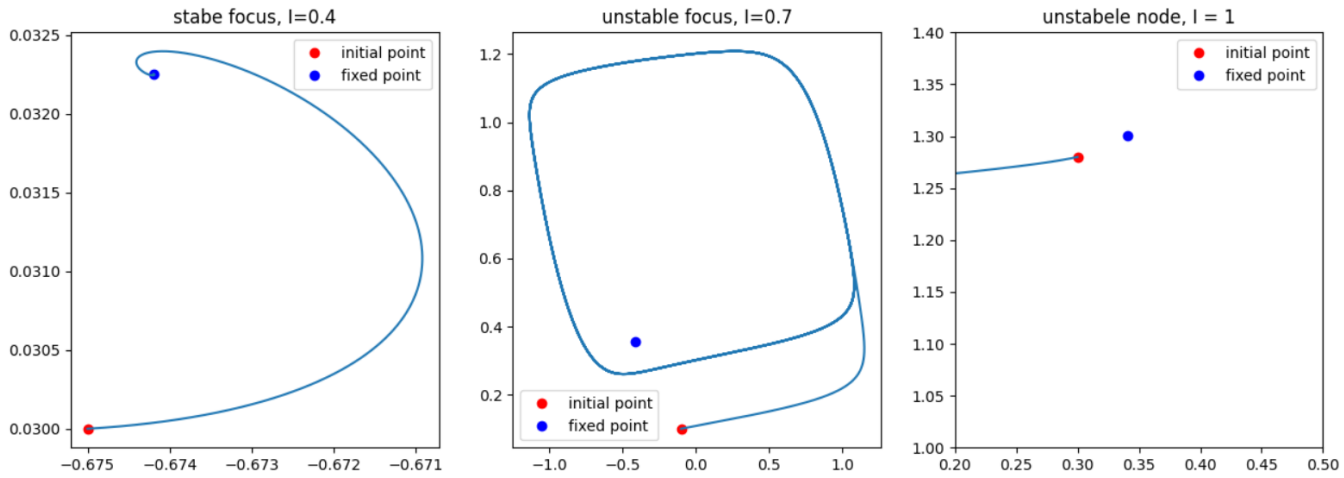
$\Rightarrow \begin{cases} \text{if } 0.558 < |V| < 0.737 \Rightarrow \tau < 0 \ , \ \tau^2 - 4\Delta < 0 \Rightarrow \text{stable focus} \\ \text{if } 0.407 < |V| < 0.558 \Rightarrow \tau > 0 \ , \ \tau^2 - 4\Delta < 0 \Rightarrow \text{unstable focus} \\ \text{if } |V| < 0.407 \Rightarrow \tau > 0 \ , \ \tau^2 - 4\Delta > 0 \Rightarrow \text{unstable node} \end{cases}$

$I = \dfrac{V + 0.7}{0.8} - V + V^3 \Rightarrow \begin{cases} \text{if } 0.29 < I < 0.56 \cup 1.19 < I < 1.46 \Rightarrow \tau < 0 \ , \ \tau^2 - 4\Delta < 0 \Rightarrow \text{stable focus} \\ \text{if } 0.56 < I < 0.71 \cup 1.04 < I < 1.19 \Rightarrow \tau > 0 \ , \ \tau^2 - 4\Delta < 0 \Rightarrow \text{unstable focus} \\ \text{if } 0.71 < I < 1.04 \Rightarrow \tau > 0 \ , \ \tau^2 - 4\Delta > 0 \Rightarrow \text{unstable node} \end{cases}$



Now based on the values of I calculated above, we will plot a trajectory for a initial value and check the dynamics behavior.

```
    v, w = FitzHugh_Nagumo(-0.675,0.03,0.4,T0)
    plt.plot(-0.674, 0.032, 'ro', label="fixed point")
    plt.plot(v,w)
    plt.legend()
    plt.title("stabe focus, I=0.4")
    plt.subplot(132)
    v, w = FitzHugh_Nagumo(-0.1,0.1,0.7,T0)
    plt.plot(-0.415, 0.357, 'ro', label="fixed point")
    plt.plot(v,w)
    plt.title("unstable focus, I=0.7")
    plt.legend()
    plt.subplot(133)
    v, w = FitzHugh_Nagumo(0.3,1.28,1,T0)
    plt.plot(0.341, 1.301, 'ro', label="fixed point")
    plt.plot(v,w)
    plt.title("unstabele node, I = 1")
```



1.5

In the previous part we observe that :

if $0.56 < I < 0.71 \cup 1.04 < I < 1.19 \Rightarrow 0.407 < |V| < 0.558 \Rightarrow \tau > 0$, $\tau^2 - 4\Delta < 0 \Rightarrow$ unstable focus

So under this condition there will be a unstable focus fixed point therefor, it a limit cycle can exist. We can also see that limit cycle in the figure shown in the previous part.

1.6

$$L = \begin{bmatrix} 1 - 3V^2 & -1 \\ a & -0.8a \end{bmatrix} \Rightarrow \tau = 1 - 3V^2 - 0.8a \quad , \quad \Delta = 2.4V^2a + 0.2a$$

$\tau^2 - 4\Delta = 0.64a^2 - 4.8V^2a - 2.4a + 9V^4 - 6V^2 + 1$

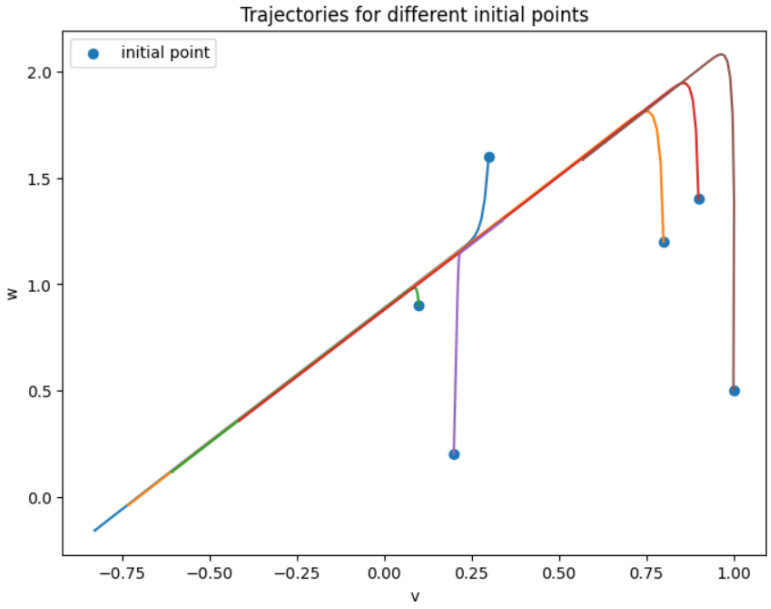
if $a \gg 1 \Rightarrow \Delta > 0 \quad , \quad \tau^2 - 4\Delta \cong 0.64a^2 - 4.8V^2a + 9V^4$

Limit cycle $\Rightarrow$ unstable focus $\Rightarrow \tau > 0$, $\tau^2 - 4\Delta < 0 \Rightarrow \begin{cases} a < \frac{1-3V^2}{0.8} \\ 0.64a^2 - 4.8V^2a + 9V^4 < 0 \end{cases}$

there is no real value of V that satisfies the inequality for $a \gg 1$ so there is no limit cycle available.

For different values of $Iext$ and initial values and for $a = 100$, we have simulated and plotted the trajectories which can be seen below.

```
Iext_2 = [0.1, 0.3, 0.5, 0.7, 1, 1.2]
init_v = [0.3, 0.8, 0.1, 0.9, 0.2, 1]
init_w = [1.6, 1.2, 0.9, 1.4, 0.2, 0.5]
```



It can be seen that for different values of $Iext$ there is no limit cycle.

2 Winner takes all network

2.1

Using python builtin function "odeint" we are going to solve the system of two coupled differential equations.

Fist we will define the $S(x)$ function, then using "E_diff_equations" function we will describe the differential equations and after that we will define the "winner_takes_all()" function as follows:

```
def S_x(x):
   sigma = 120
   N=2
   M = 100
   if(x>=0):
      return M*x**N/(sigma**N+x**N)
   else:
      return 0

def E_diff_equations(y, t, tau, k1, k2):
   E1,E2 = y # unpack state variables
   dE1dt = 1/tau * (-E1 + S_x(k1-3*E2))
   dE2dt = 1/tau * (-E2 + S_x(k2-3*E1))
   return [dE1dt, dE2dt]

def winner_takes_all(init_E1, init_E2, k1, k2, tau, T0):
   dt = 0.01
   T  = math.ceil(T0/dt) # [ms]
   t = np.arange(0,T)*dt
   y = odeint(E_diff_equations, [init_E1, init_E2], t, args=(tau, k1, k2,))
   return y[:,0], y[:,1]
```

Now by using the following code we can observe the vector field of E_1 and E_2 and also their nullcilnes.

For creating and simulating the vector field, we should first create a mesh grid using vectors of E_1 and E_2. Then we will create are differential equations using the our grid variables and finally using "streamplot()" we can plot the vector field.

```
# plotting vector field
e1, e2= np.meshgrid(E1,E2)
i1 = k2-3*e1
i2 = k1-3*e2
o1 = M*i1**N/(sigma**N+i1**N)
o2 = M*i2**N/(sigma**N+i2**N)
o1[i1 < 0] = 0
o2[i2 < 0] = 0
de1 = 1/tau * (-e1 + o2)
de2 = 1/tau * (-e2 + o1)
plt.streamplot(e1,e2,de1,de2)
```

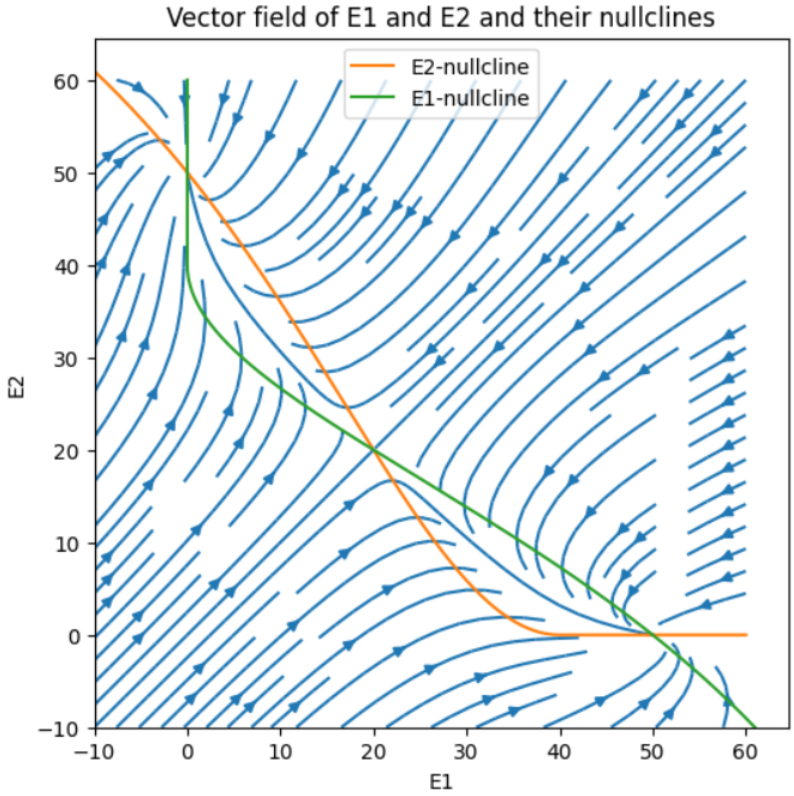
For plotting the nullclines we have to plot the following equations :

$$\frac{dE_1}{dt} = 0 \Rightarrow E_1 = S(k_1 - 3E_2)$$

$$\frac{dE_2}{dt} = 0 \Rightarrow E_2 = S(k_2 - 3E_1)$$

the code used for plotting the nullclines is as follows:

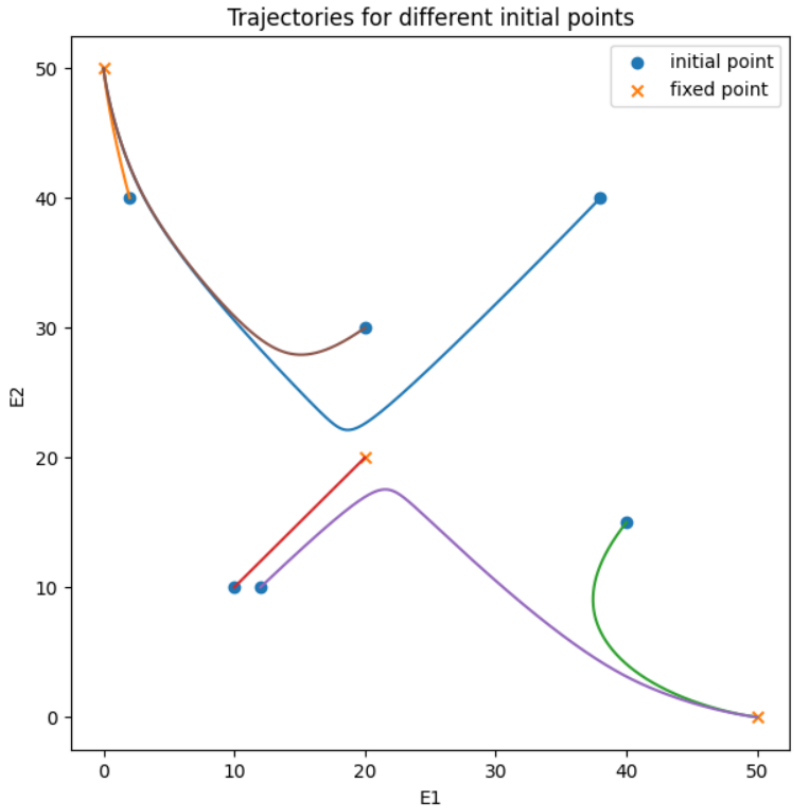
```
    indx1 = k2-3*E1
    indx2 = k1-3*E2
    out1 = M*indx1**N/(sigma**N+indx1**N)
    out2 = M*indx2**N/(sigma**N+indx2**N)
    out1[indx1 < 0] = 0
    out2[indx2 < 0] = 0
    plt.plot(E2, out1, label="E2-nullcline")
    plt.plot(out2, E1, label="E1-nullcline")
```



It can be clearly seen that the nullclines will intersect in three points so there are three fixed points. Also the behavior and the kind of the fixed points can be seen too.

Now for the following initial points we will plot the trajectories.

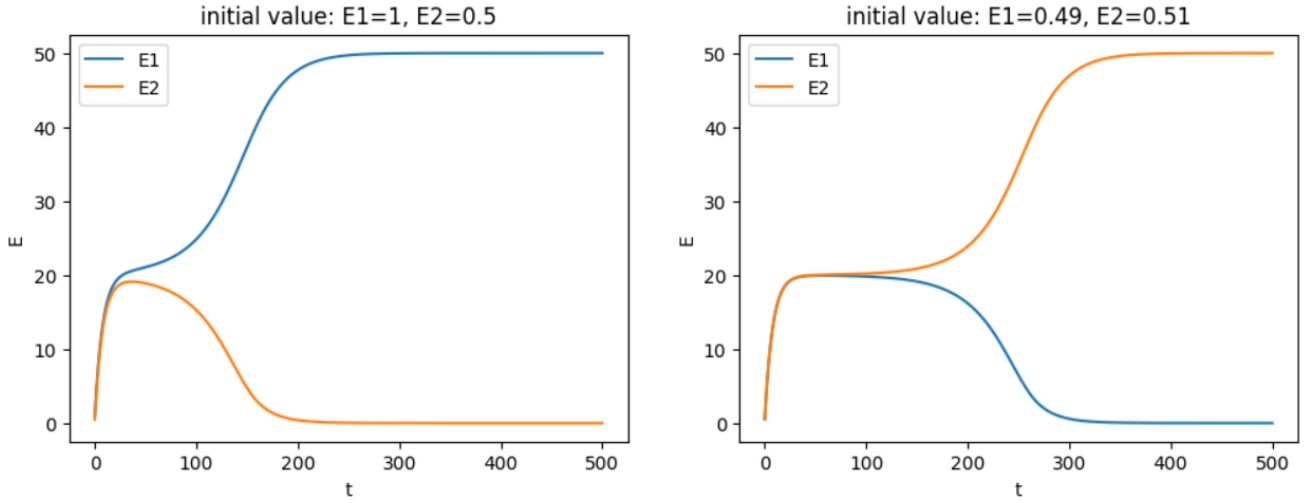
```
    tau = 20
    k1, k2 = 120, 120
    T = 500
    init_E1, init_E2 = [38, 2, 40, 10, 12, 20], [40, 40, 15, 10, 10, 30]
```



The results are completely vivid.

2.2

Using the previous defined functions we will easily plot the number of spikes versus time for the two different initial values given.



In the graph located at the left side, since the initial value of E_1 is greater than E_2, in competition between these neurons, E_1 has won so as we see the values of E_1 and E_2 will move toward the fixed point with coordinates of $(50, 0)$

and the second neuron which has lost the competition will become completely deactivated.

Based on the graph located at the right side, the initial value of E_2 is a bit greater than E_1 so in this case, E_2 will win the competition. As it can be seen the final value of E_2 is 50 and the finial value of E_1 is 0. The first neuron has became completely inactive. In this case based on the location of the initial point in the vector field and the fact that the initial values are approximately equal, at first the point have moved toward the point $(20, 20)$ which is saddle point and but the initial value of E_2 is a bit greater so finally E_2 has won. As it can be seen in this case the variables have reached their final values at a greater time.

2.3

Fist we calculate the coordinates of the fixed points.

$$\frac{dE_1}{dt} = 0 \Rightarrow E_1 = S(k_1 - 3E_2) \quad , \quad \frac{dE_2}{dt} = 0 \Rightarrow E_2 = S(k_2 - 3E_1) \ , \ k_1 = k_2 = 120$$

if $E_1 > 40$, $E_2 < 40 \Rightarrow 120 - 3E_1 < 0 \Rightarrow S(120 - 3E_1) = 0 \Rightarrow E_2 = 0 \Rightarrow S(120 - 3E_2) = 50 = E_1 \Rightarrow (50, 0)$
if $E_2 > 40$, $E_1 < 40 \Rightarrow 120 - 3E_2 < 0 \Rightarrow S(120 - 3E_2) = 0 \Rightarrow E_1 = 0 \Rightarrow S(120 - 3E_1) = 50 = E_2 \Rightarrow (0, 50)$
if $E_1 < 40$, $E_2 < 40 \Rightarrow \frac{100(120 - 3E_1)^2}{120^2 + (120 - 3E_1)^2} = \frac{100(120 - 3E_2)^2}{120^2 + (120 - 3E_2)^2} \Rightarrow (20, 20)$

Now for each fixed point we check the stability.

Forming the Jacubian matrix : $L = \frac{1}{\tau} \begin{bmatrix} -1 & -3\frac{dS}{dE_2}\big|_{k_1 - 3E_2} \\ -3\frac{dS}{dE_1}\big|_{k_2 - 3E_1} & -1 \end{bmatrix}$

$(E_1, E_2) = (0, 50)$:

$L = \frac{1}{\tau} \begin{bmatrix} -1 & 0 \\ x & -1 \end{bmatrix} \Rightarrow T = trace(L) = -\frac{2}{\tau} = -0.1 < 0 \quad , \quad \Delta = det(L) = \frac{1}{\tau^2} = \frac{1}{400} > 0$

$\Rightarrow T^2 - 4\Delta = 0 \Rightarrow$ stable node

$(E_1, E_2) = (50, 0)$:

$L = \frac{1}{\tau} \begin{bmatrix} -1 & x \\ 0 & -1 \end{bmatrix} \Rightarrow T = trace(L) = -\frac{2}{\tau} = -0.1 < 0 \quad , \quad \Delta = det(L) = \frac{1}{\tau^2} = \frac{1}{400} > 0$

$\Rightarrow T^2 - 4\Delta = 0 \Rightarrow$ stable node

$(E_1, E_2) = (20, 20)$:

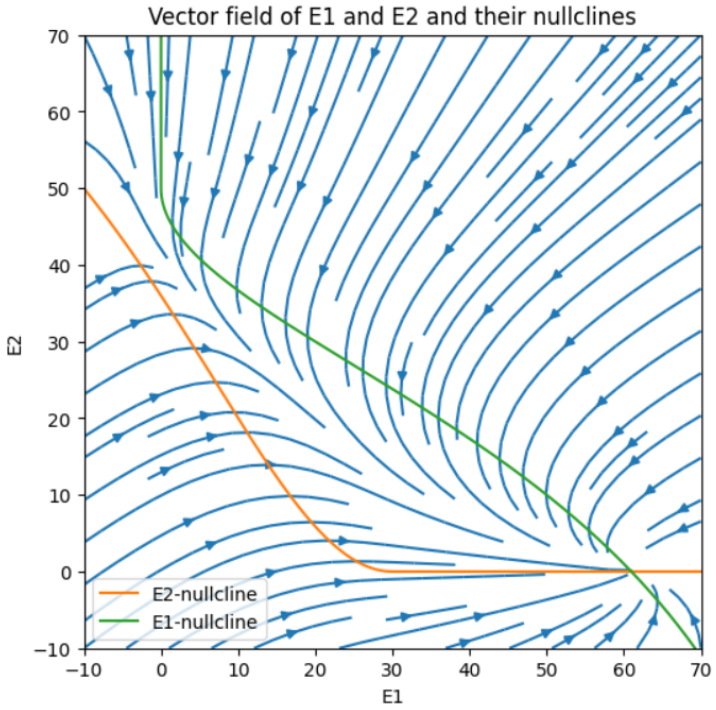
$L = \frac{1}{\tau} \begin{bmatrix} -1 & -1.6 \\ -1.6 & -1 \end{bmatrix} \Rightarrow T = trace(L) = -\frac{2}{\tau} = -0.1 < 0 \quad , \quad \Delta = det(L) = -\frac{1.56}{\tau^2} = -0.0039 < 0$

$\Rightarrow$ saddle node

As it can be seen based on the figures, the results completely match the simulations.

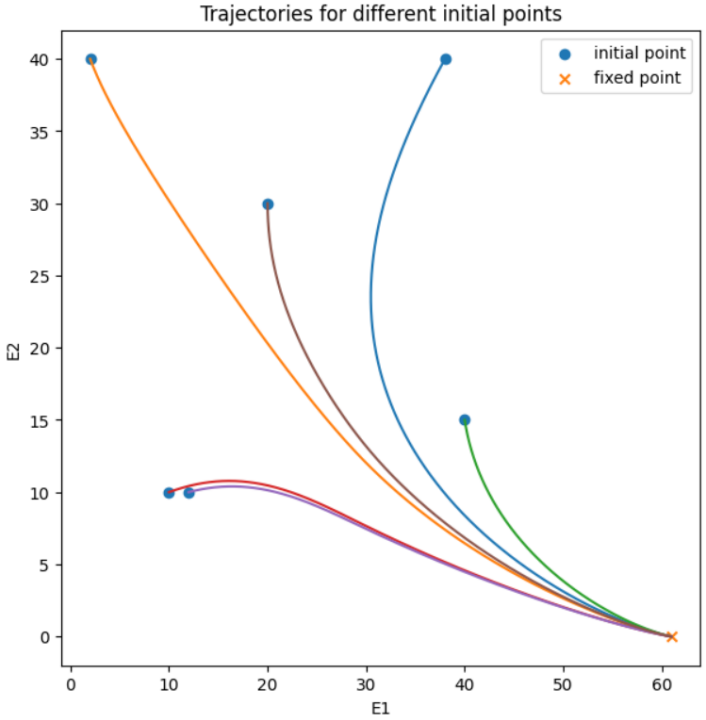
2.4

First we plot the vector field and the nullclines for new values of k_1 and k_2.

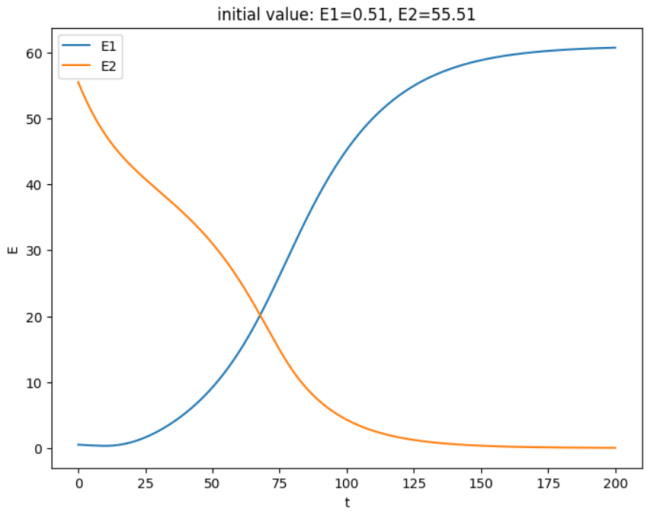


In this case, the nullclines will intersect in only one point.

Now we plot the trajectories.



The behavior of E_1 and E_2 over time is as follows:



For this new conditions and based on the previous figures, we saw that there is only one fixed point and that point is a stable node with coordinates of $(60, 0)$. In fact for this values of k_1 and k_2, the first neuron will always win the competition and at the end the second neuron will be deactivated. This can be also seen based on the above figure that although the initial value of E_2 is much greater than E_1, it's final value is 0.

Here we check the stability of the fixed points.

$$\frac{dE_1}{dt} = 0 \Rightarrow E_1 = S(k_1 - 3E_2) \quad , \quad \frac{dE_2}{dt} = 0 \Rightarrow E_2 = S(k_2 - 3E_1) \ , \ k_1 = 150 \, , \, k_2 = 90$$

if $E_1 > 30$, $E_2 < 50 \Rightarrow 90 - 3E_1 < 0 \Rightarrow S(90 - 3E_1) = 0 \Rightarrow E_2 = 0 \Rightarrow S(150 - 3E_2) \cong 60 = E_1 \Rightarrow (60, 0)$
if $E_2 > 50$, $E_1 < 30 \Rightarrow 150 - 3E_2 < 0 \Rightarrow S(150 - 3E_2) = 0 \Rightarrow E_1 = 0 \Rightarrow S(90 - 3E_1) = 36 = E_2 \Rightarrow$ not valid ✗
if $E_1 < 30$, $E_2 < 50 \Rightarrow \dfrac{100(150 - 3E_1)^2}{120^2 + (150 - 3E_1)^2} = \dfrac{100(90 - 3E_2)^2}{120^2 + (90 - 3E_2)^2} \Rightarrow$ no intersection

---

$(E_1, E_2) = (60, 0)$:

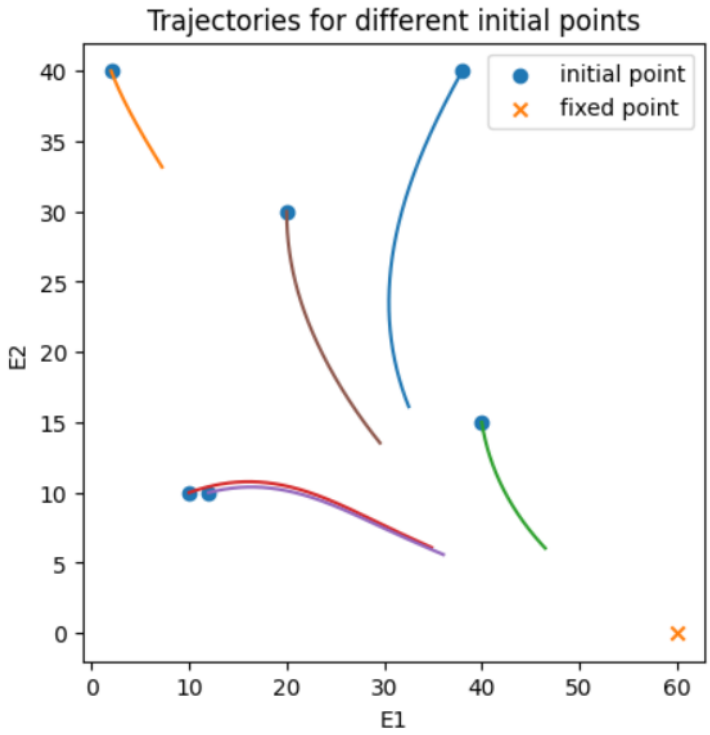
$$L = \frac{1}{\tau}\begin{bmatrix} -1 & x \\ 0 & -1 \end{bmatrix} \Rightarrow T = trace(L) = -\frac{2}{\tau} = -0.1 < 0 \quad , \quad \Delta = det(L) = \frac{1}{\tau^2} = \frac{1}{400} > 0$$

$\Rightarrow T^2 - 4\Delta = 0 \Rightarrow$ stable node

2.5

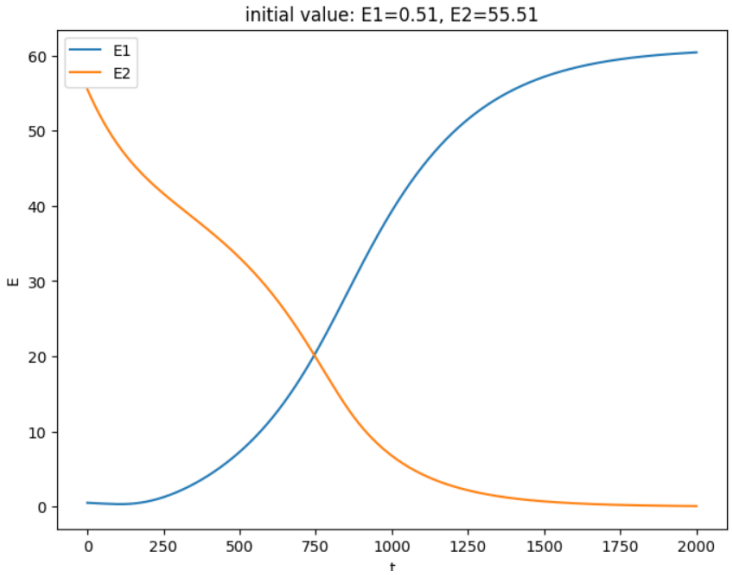
Based on the equations used for forming the nullclines, the time constant τ has no effect on the form of the nullclines and the vector field. As we expected the number and coordinate of the fixed point is the same as before.

By Increasing the value of τ, the time needed for the system variables to reach their final values will be increased and the system agility will decrease. In the previous parts we set the value of the simulation time to 200 ms for plotting the trajectories. Now for the same value of simulation time we will plot the trajectories for the new value of τ.



It can be seen that since the system is not as agile as it was before, after this time of simulation the trajectories and the variables have not reached their final values and we should increase the value of simulation time if we want the variables to reach their final values.

It is also true for the timing diagram of E_1 and E_2.



In the previous plot, after about 200 ms, E_1 and E_2 will reach their final values, 60 and 0. But as you can see here they will reach their final values after about 2000 ms.

2.6

Based on the previous parts it can be concluded that the result of decision making or dominance of a neuron over another is highly dependent to the inputs.

If the input of both neurons are the same there are three possible chances. Both neurons have a chance to win the competition and even a draw condition is possible. The initial values will determine the result.

But if the input of one of the neurons is greater than the others, there will be only one winner and it will be the neuron with greater input, no matter what are the initial conditions.

The time constant parameter determines the time needed for neurons to win the competition and the time needed for them to reach their final values. The greater the value of time constant, the more time needed for neuron spikes number to reach their final value.