IN THE NAME OF GOD

HW2

Neuroscience, Learning, Memory, Cognition Course

Sharif University of Technology

Dr.Karbalaei

SaraRezanezhad99101643

April2023

1.1)

The **FitzHugh-Nagumo model**

$$\begin{aligned}\dot{V} &= V - V^3/3 - W + I \\ \dot{W} &= 0.08(V + 0.7 - 0.8W)\end{aligned}$$

is a two-dimensional simplification of the Hodgkin-Huxley model of spike generation in squid giant axons. Here,

- V is the membrane potential,
- W is a recovery variable,
- I is the magnitude of stimulus current.

another shape of this model:

dv/dt = v - v^3/3 - w + I

dw/dt = (v + a - bw)/tau

The parameter a determines the shape of the nullcline for w. Nullcline is a contour along which some variable has no change. In other words, it is a curve along which one of the state variables is stationary. The value of a sets the position and slope of this nullcline. Here the magnitude of parameter a is 0.7.

The parameter b determines the shape of the nullcline for v. The value of b sets the position and slope of this nullcline. In this equation parameter b = 0.8.

Finally, tau is the time constant for the recovery variable. It determines how quickly w returns to its resting state after being activated (in this equation tau can be 12.5)

1.2)

The written code is to define two above equation and plot a phase diagram to show the relation between 2 parameters, V & W.

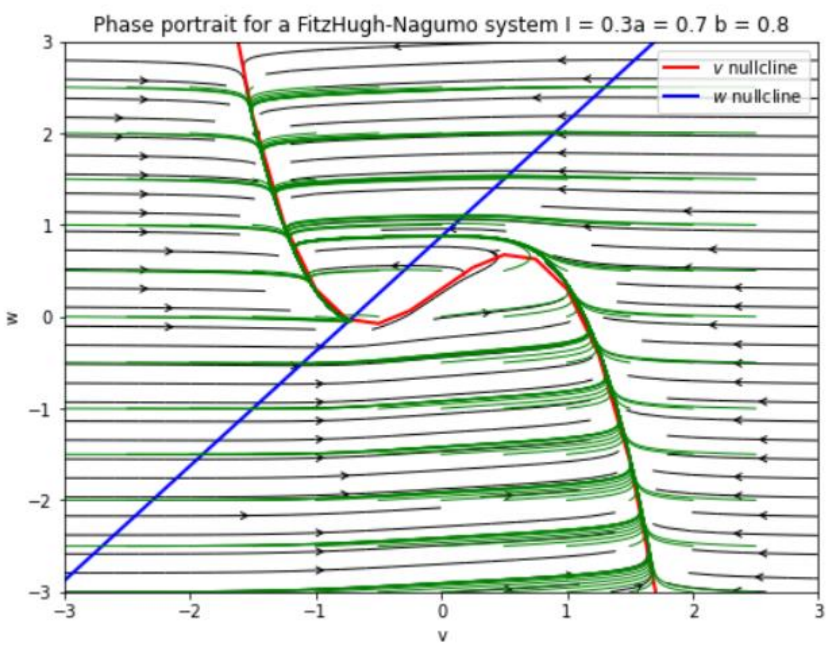
The explanation about each line is written in the code as comments; but In summary, first we initialized some of the parameters then we wrote a function to use the differential equations. We initialized values for main parameters in phase page.

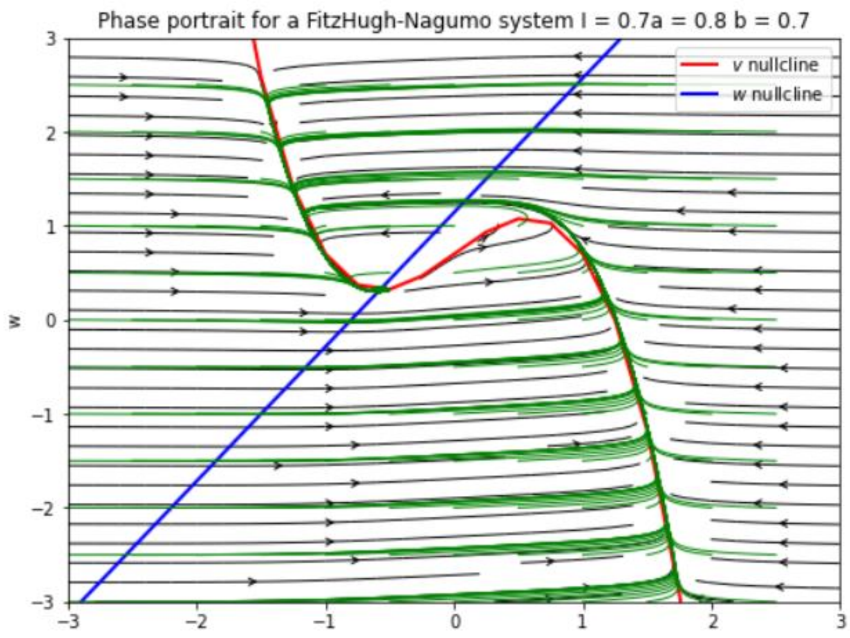
We solved the interaction between mode and time with a numerical method in which we used for loops and *odeint* function (As it's been recommended) to solve the ordinary differential equations. In each cycle lines were drawn to show the final shape of trajectory on the w diagram in terms of v.

The next part of the code, is for draw plots and in the final step, we draw a diagram to display each certain equation when the differential derivatives are zero.

For using *streamplot* and it's options for drawing trajectory with arrows, we should create a meshgrid

These are examples with different amount of initial values:





Phase portrait for a FitzHugh-Nagumo system I = 0.7a = 0.8 b = 0.7
v nullcline
w nullcline
w

1.3)

Original Formula

$$\frac{dv}{dt} = v(a-v)(v-1) - w + I \Rightarrow \dot{v} = v - v^3 - w + I$$

$$\frac{dw}{dt} = -cw + bv \Rightarrow \dot{w} = 0.08(v + 0.7 - 0.8w)$$

* V_nullcline: $\dot{v} = 0$

$$w = V(a-V)(V-1) + I \Rightarrow w = v - v^3 + I$$

$$\Rightarrow a = -1 \qquad f(V,w) = -v^3 + (a+1)v^2 - av - w + I = 0$$

$$\xrightarrow{\text{if } I = 0.3} -v^3 + v - w + 0.3 = 0$$

$$\boxed{w = -v^3 + v + 0.3} \ (\mathrm{I})$$

* w_nullcline: $\dot{w} = 0$

$$w = \frac{b}{c} V \Rightarrow \frac{b}{c} = \frac{5}{4} \Rightarrow \boxed{w = \frac{5v}{4} + \frac{7}{8}} \ (\mathrm{II})$$

$$g(V,w) = -cw + bV = 0$$

$$\Rightarrow v + 0.7 - 0.8w$$

$$(\mathrm{I}), (\mathrm{II}) \Rightarrow -v^3 + v + 0.3 = \frac{5}{4}v + \frac{7}{8}$$

$$\Rightarrow v^3 + \frac{v}{4} + \frac{23}{40} = 0 \Rightarrow v_1 \simeq -0.73187$$

$$v_2 = 0.3659 - 0.807j$$

$$v_3 = 0.3659 + 0.807j$$

Jacobian matrix at fixed point:

$$L = \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} \partial f/\partial v & \partial f/\partial w \\ \partial g/\partial v & \partial g/\partial w \end{bmatrix}$$

$$L = \begin{bmatrix} -3v^2 + 1 & -1 \\ 1 & -,8 \end{bmatrix}$$

$$\boxed{\begin{aligned} \tau &= trace(L) = -3v^2 + ,2 \\ \Delta &= det(L) = +2,4v^2 + ,2 \end{aligned}}$$

For fixed point v_1 : (= -,73187)

- trace(L) = -1,4.69
- det(L) = 1,4855

$$* L = \begin{bmatrix} -,6.69 & -1 \\ 1 & -,8 \end{bmatrix}$$

- eigenvalues: $\begin{vmatrix} -,6.69 - \lambda & -1 \\ 1 & -,8 - \lambda \end{vmatrix} = 0$

$$\lambda^2 + 1,4.69\lambda + ,4855 + 1 = \lambda^2 + 1,4.69\lambda + 1,4855$$

No real eigeValue => $\lambda_1 = -,7.345 + ,995318j$

$\lambda_2 = -,7.345 - ,995318j$

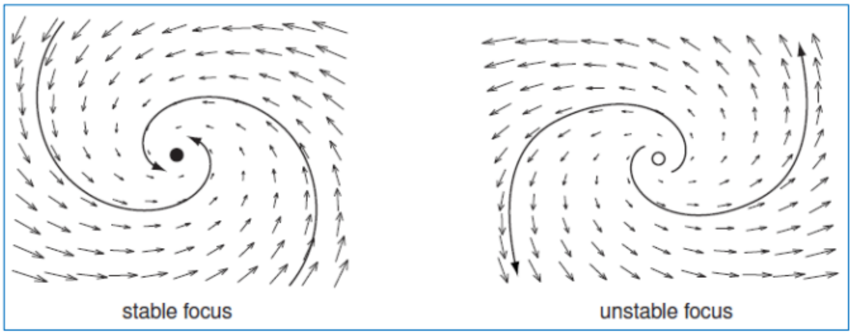
From what we concluded in the previous part, it can be understood that V_1(as a real fixed point) is a Focus kind of fixed points:

Classification of fixed points:

$$\begin{bmatrix} u(t) \\ w(t) \end{bmatrix} = c_1 e^{\lambda_1 t} v_1 + c_2 e^{\lambda_2 t} v_2$$

3. *Focus:*

- **Complex conjugate eigenvalues:** $\lambda_1 = \sigma + j\omega,\ \lambda_2 = \sigma - j\omega$
- Stable when real parts negative: $\sigma < 0$
- Unstable when real parts positive: $\sigma > 0$
- Imaginary part ω determines the frequency of rotation of trajectories around the focus



According to the real part of the eigenvalues (σ), this fixed point is a stable one.

We could also conclude this result with these specific conditions for the FitzHugh-Nagumo model:

- Fixed point is stable when:

$$\tau = \mathrm{trace}(L) = -a - c < 0$$

and:

$$\Delta = \det(L) = ac + b > 0$$

There are 2 parts in our code, for this part of the question, which are specifically for calculating the fixed points:

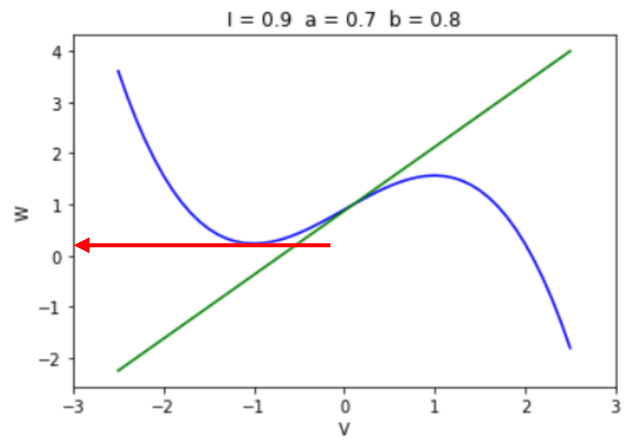
```
x = sym.symbols('x')
y = sym.symbols('y')

eq1 = sym.Eq(y, x -  x**3 +I)
eq2 = sym.Eq(y,  1.25*x+7/8)
sol = sym.solve((eq1,eq2), (x, y))
print(sol)
```

```python
from sympy import *

#initial value
I = 0.3
a = 0.7
b = 0.8
tau = 12.5

v = symbols('v')
A, B, C, D= symbols('A, B, C, D')
A=1
B=0
C =-1+1/b
D=-I+a/b

eq = Eq(A*v**3 +B*v**2 + C*v + D, 0)

soln = solve(eq, v)

print(soln)
```

For specific initial values that’s given in the question , we can see that the simulating is matching with differential equations’ result.

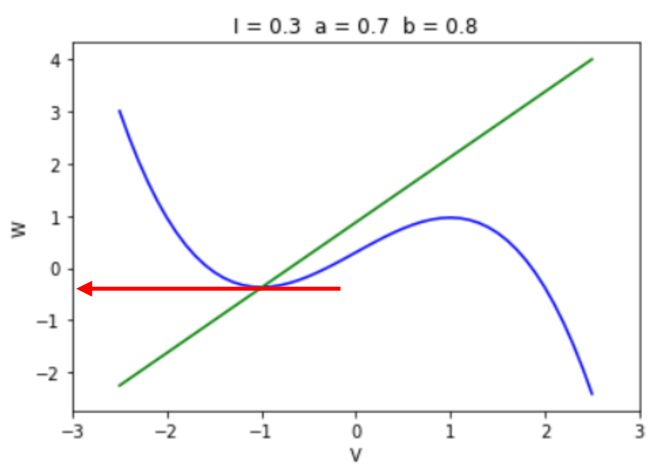
1.4)

For a higher value of I, more input current is applied to the neuron, which increases the neuronal activity. As a result, the acceleration of the neuron's action potential (spiking) increases and the V curve is deflected upwards (for negative voltage and less number of spikes).



On the other hand, for lower values of I, the least current is applied to the neuron, which reduces the neuronal activity. As a result, the acceleration of the action potential (spiking) of the neuron becomes less and the V curve deviates downwards (for negative voltage and less number of spikes).

(for more sensitivity to the changes of I, we used the model with Coefficient 1/3 for v^3)



1.5)

The Fitzhugh-Nagomo neuron model has a limit cycle in which information is transferred from the input layer to the output layer through neurons and synaptic connections between them. This limit cycle in the model has caused this model to be known as a model with short-term memory and has the ability to store and retrieve information ahead of time.

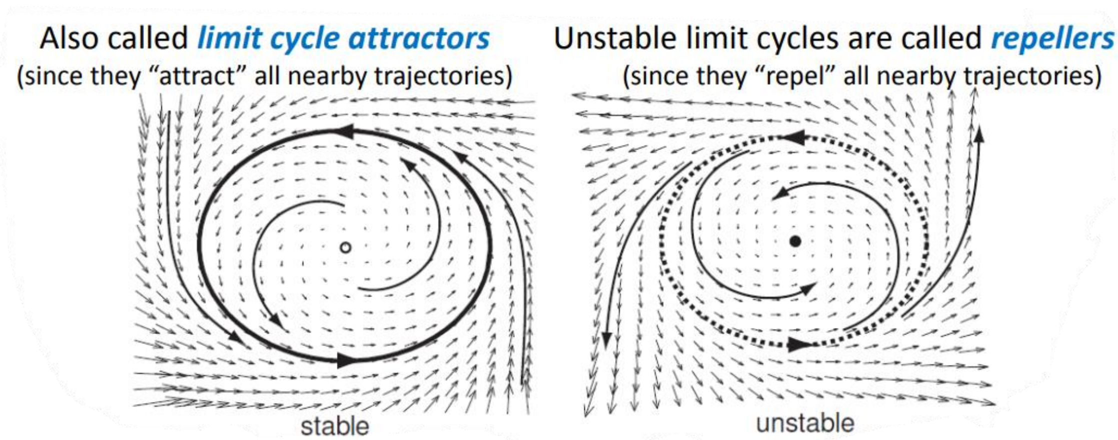
In fact, whenever input is provided to the model, information is sent to a specific internal state, and in mammals, this internal state is called a latent neural state.

Over time, the inputs are combined with the previous internal value and are sent to the network as new inputs, and as a result, the network will have a different internal state over time, which is known as the limit state or steady state

1.6)

By changing the value of a, there is no difference in the null lines. Therefore, there is no change in the coordinates of the fixed point or in the Jacobian matrix. So eventually the eigenvalues remain constant and if they have caused a stable or unstable limit cycle, the new situation will be the same.

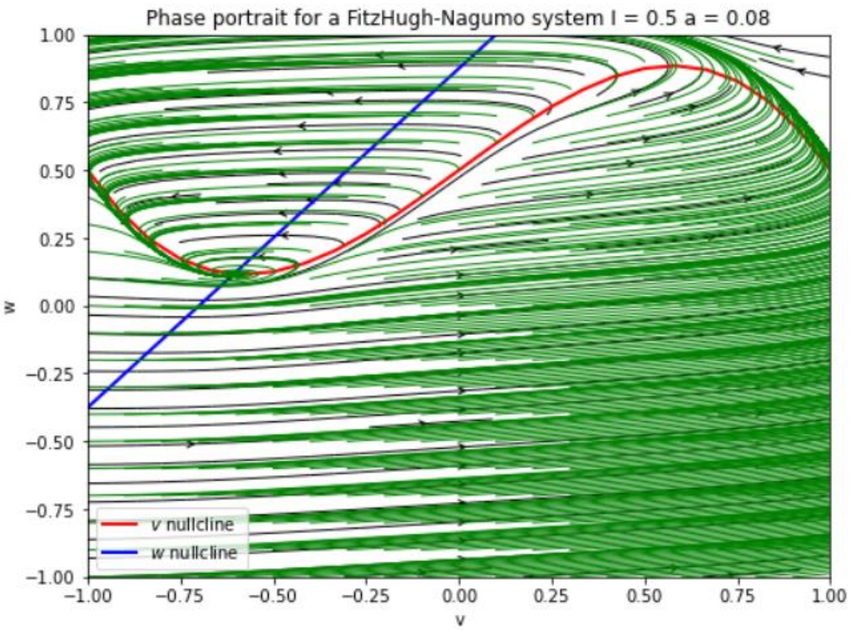
As we compare the limit cycle examples with our simulation, for normal value of I and a (as a parameter in formula below) smaller than 1 we could see in the trajectory that there is a stable limit cycle, or in the other word limit cycle attractor, around the real fixed point:



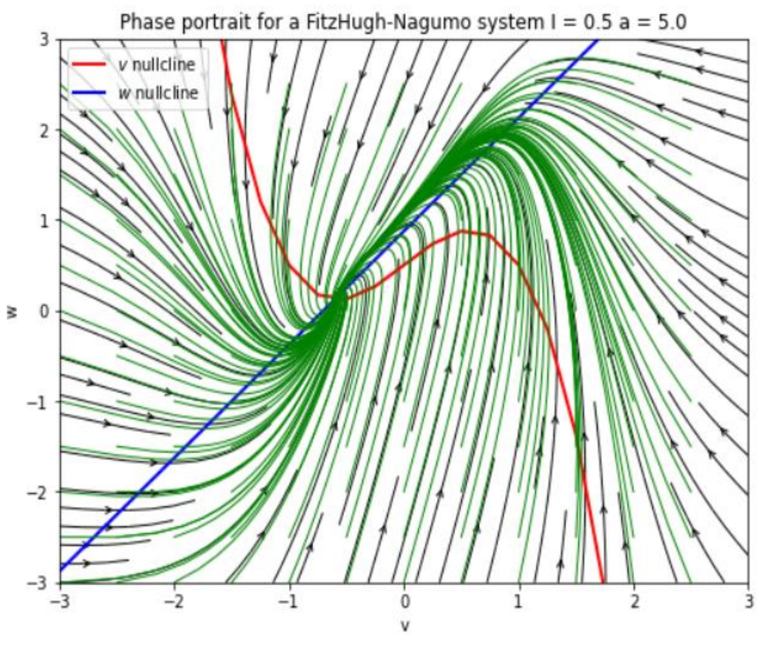
$$\dot{w} = a(v + 0.7 - 0.8w)$$

Fixed point: `(-0.606705831381115, 0.116617710773606)`
We could actually calculate the eigenvalues of the fixed point to see if there's a stable limit cycle or an unstable one.

For the example below, taking I close to 0.3 (calculated in the third question) and a<<1, the eigenvalues are the same as the previous example and the limit cycle should be stable.

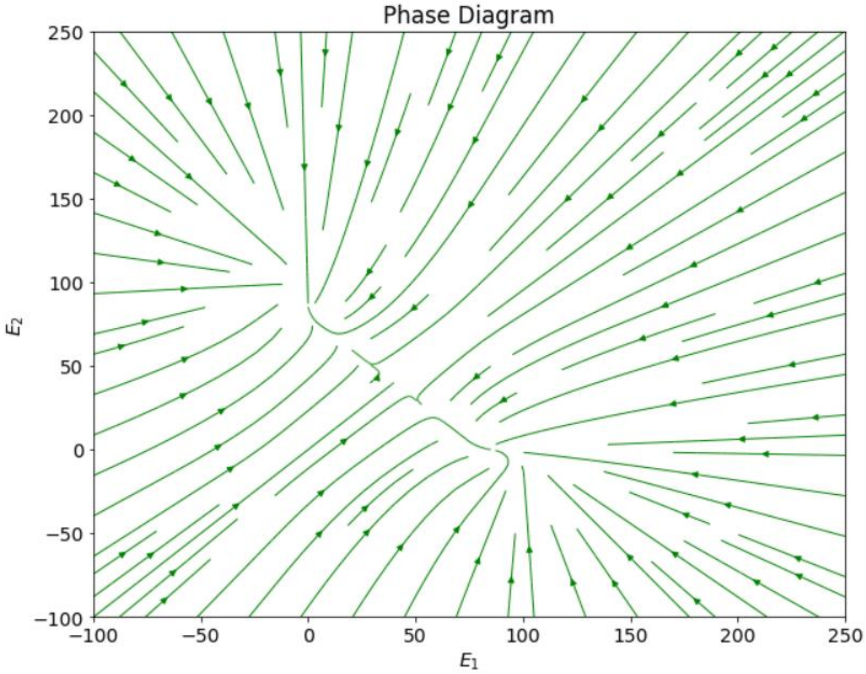


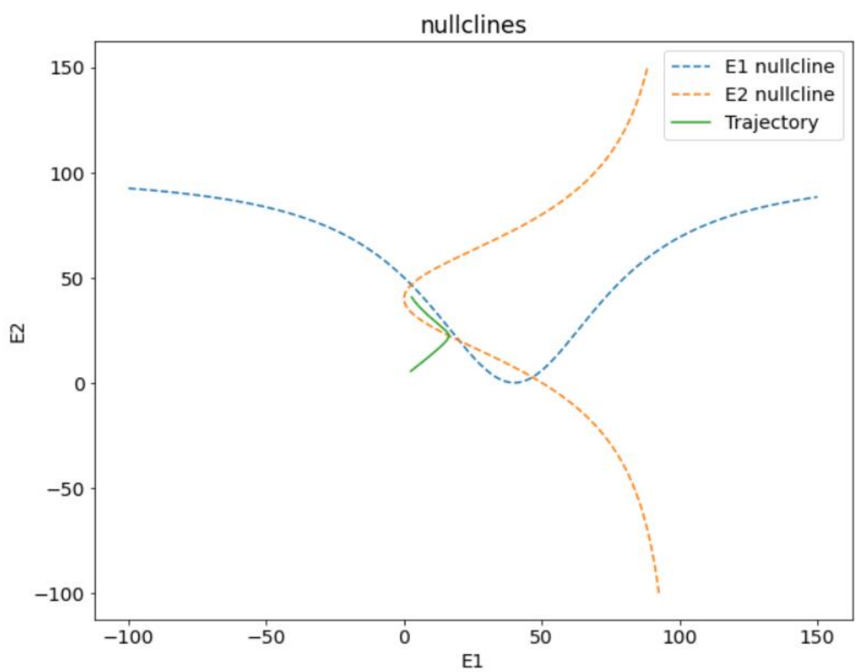
For normal value of I and much bigger than 1 we could see in the trajectory that there is a stable limit cycle around the same real fixed point too:



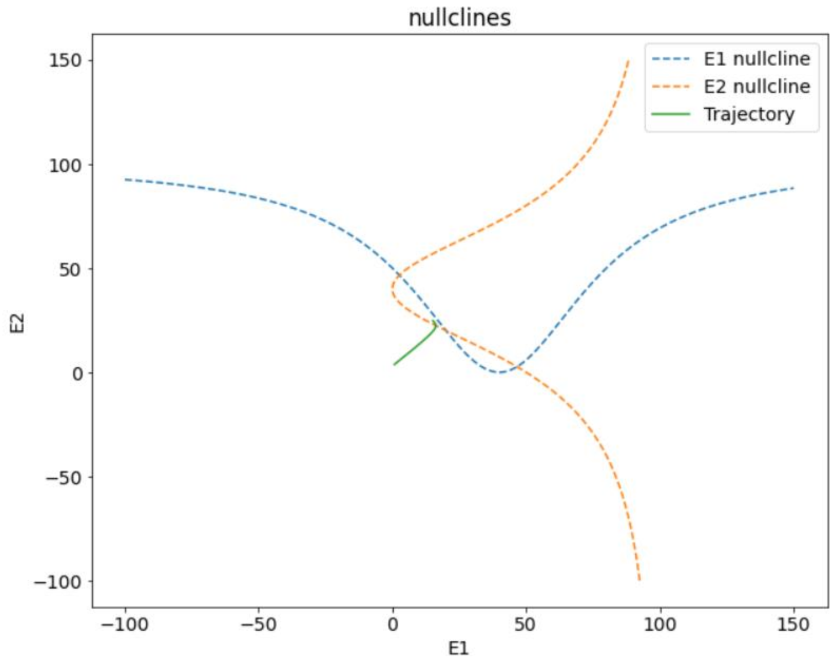
2.1)

- For E_0= [2.5, 5.55]:

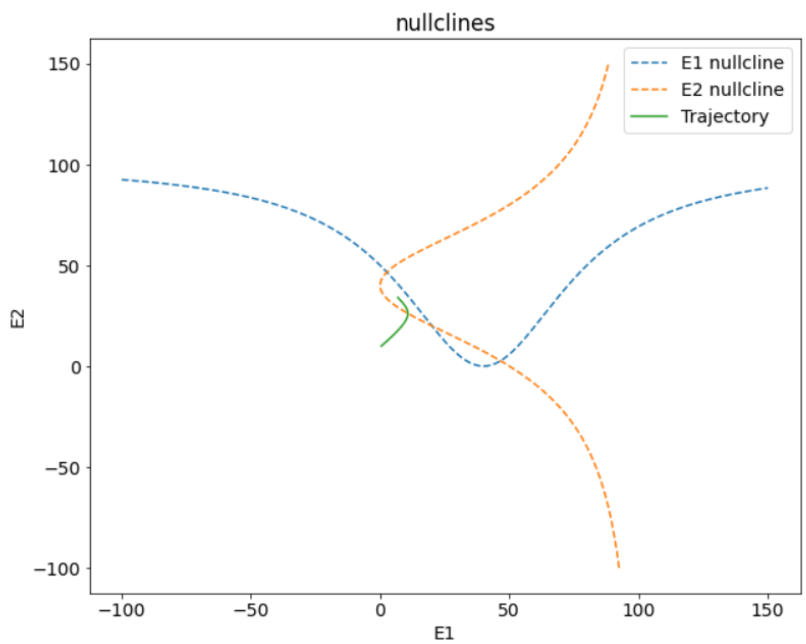




➢ For E_0= [0.75, 3.75]:



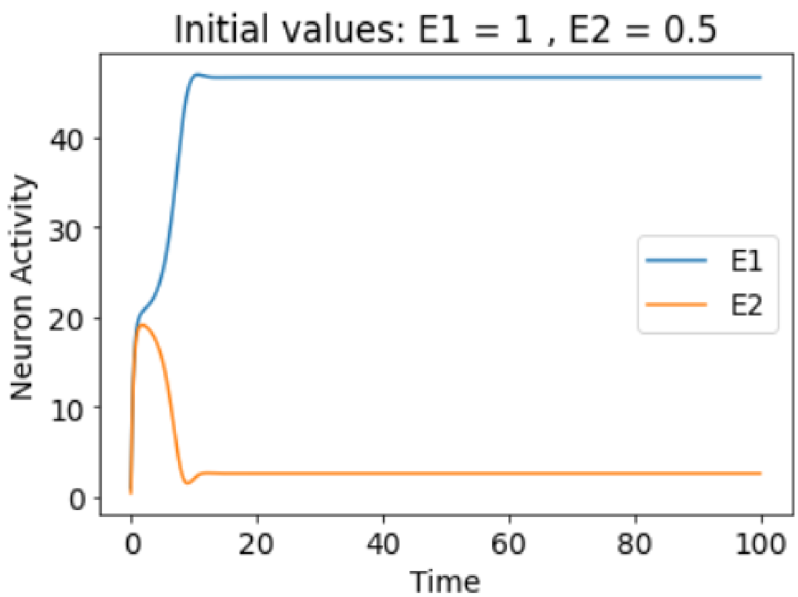
➢ For E_0= [0.5, 10]:

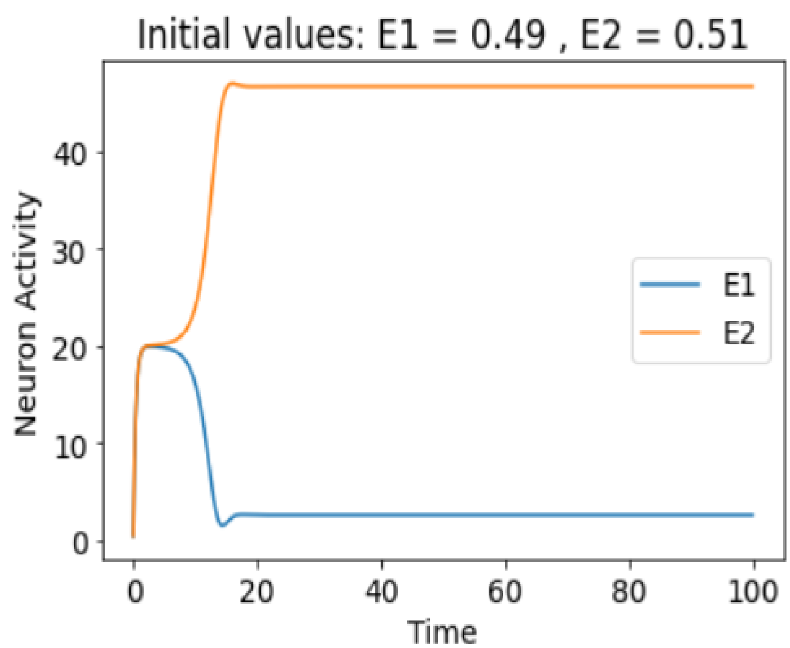


2.2)

For initial values of E1 = 1 and E2 = 0.5, both neurons move toward activation, but E1 activates faster than E2. As time goes on, E1 reaches an upper limit and E2 reaches a lower limit.

However, for initial values of E1 = 0.49 and E2 = 0.51, both neurons move to standstill, and as time progresses, E1 reaches a lower bound and E2 reaches an upper bound. These changes in the graph are shown by the difference in the height and descent of the graph after several compounding times.





2.3)

*Fixed Point (Only real): (20, 20)

- E_1 nullcline $= \dfrac{M(k_1 - 3E_2)^N}{\sigma^N + (k_1 - 3E_2)^N} = E_1$
- E_2 nullcline $= \dfrac{M(k_2 - 3E_1)^N}{\sigma^N + (k_2 - 3E_1)^N} = E_2$

$$f(E_1, E_2) = -E_1 + \frac{M(k_1 - 3E_2)^N}{\sigma^N + (k_1 - 3E_2)^N}$$

$$g(E_1, E_2) = -E_2 + \frac{M(k_2 - 3E_1)^N}{\sigma^N + (k_2 - 3E_1)^N}$$

$$L = \begin{bmatrix} \partial f/\partial E_1 & \partial f/\partial E_2 \\ \partial g/\partial E_1 & \partial g/\partial E_2 \end{bmatrix}$$

$$L = \begin{bmatrix} -1 & -3MN\sigma^N(k_1 - 3E_2)^{N-1} \\ -3MN\sigma^N(k_2 - 3E_1)^{N-1} & -1 \end{bmatrix}$$

$N = 2$, $M = 100$, $\sigma = 120$, $k_1 = k_2 = 120$

- $\tau = \text{trace}(L) = -2$

- $\Delta = \det(L) = 1 + 9M^2N^2\sigma^{2N}(k_2 - 3E_1)^{N-1}(k_1 - 3E_2)^{N-1}$

$$\Delta = 9(10^4)(4)(120)^4(120 - 3\times 20)^2 + 1$$

$$(E_1 = 20, E_2 = 20) \Rightarrow \Delta \simeq 268{,}73 \times 10^{15}$$

$$\lambda_1 = \frac{\tau + \sqrt{\tau^2 - 4\Delta}}{2} = \frac{-2 + \sqrt{4 - 4\times 26{,}8\times 10^{16}}}{2}$$

$$\lambda_2 = \frac{\tau - \sqrt{\tau^2 - 4\Delta}}{2}$$

$$\lambda_1 \simeq -1 + 5{,}18\times 10^8 j$$

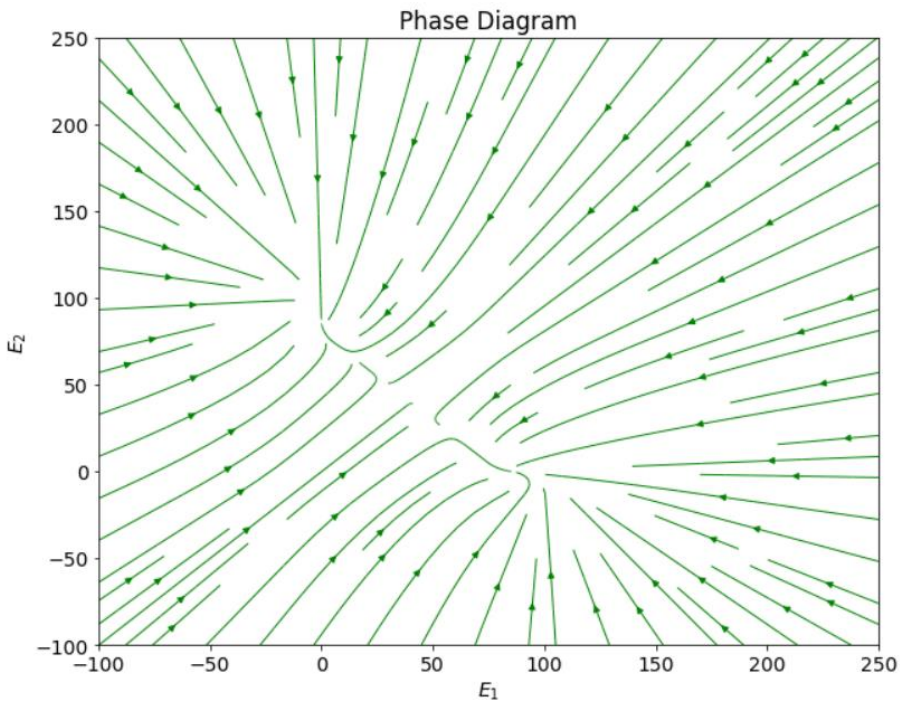
$$\lambda_2 = -1 - 5{,}18\times 10^8 j$$

✓ Focus

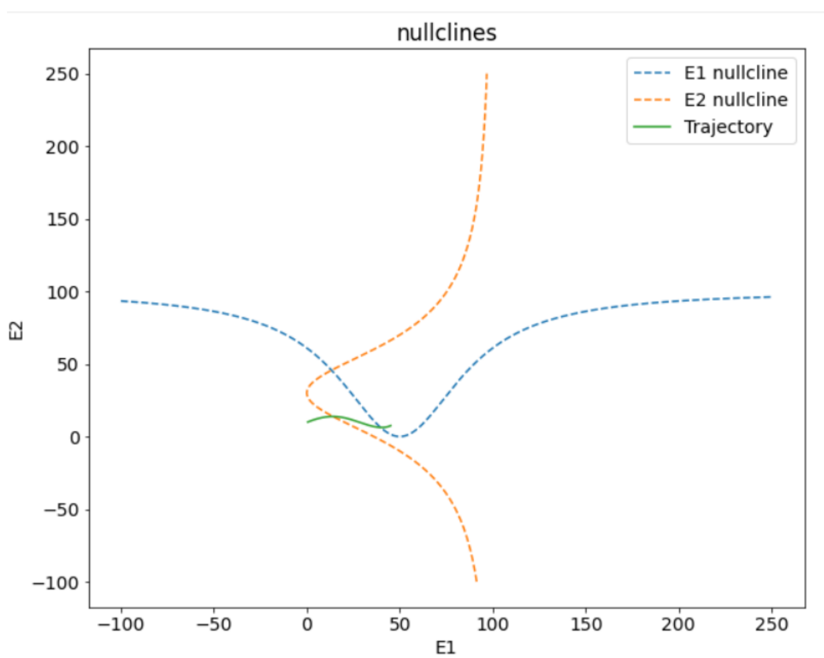
$\sigma = -1 < 0 \Rightarrow$ stable ✓

2.4)

- Phase diagram for $k_1 = 150\,, k_2 = 90$

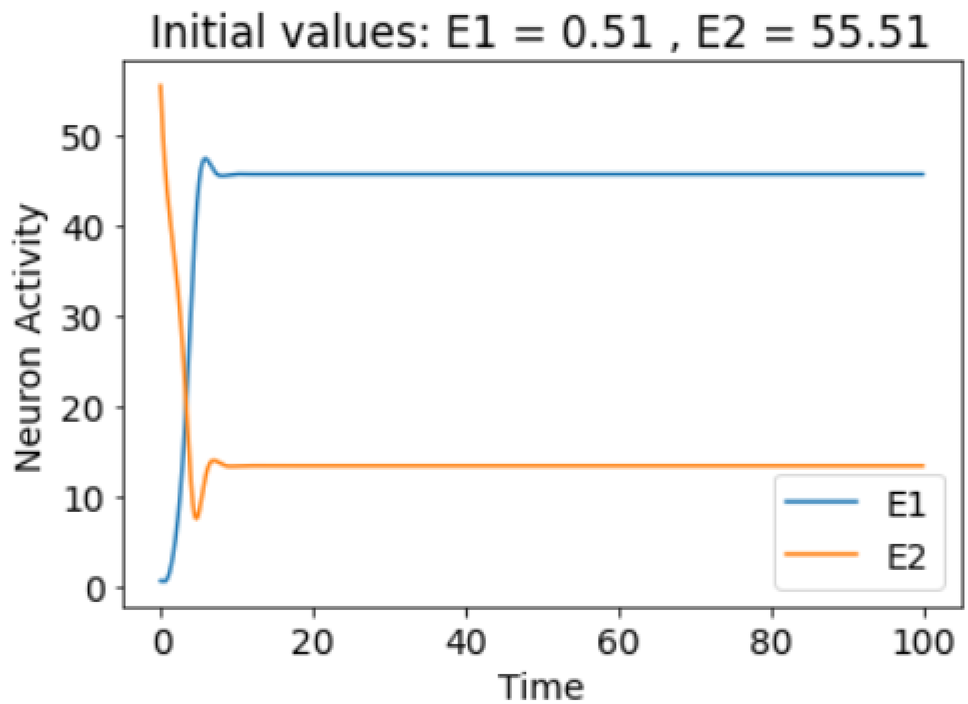


- Nullclines for $k_1 = 150\,, k_2 = 90$



- The written code is implemented to simulate a biological system with two neutrons named E1 and E2. This system consists of a feedback circuit in which neutron E1 is activated if the average activity of neutron E2 is more than 3 units. And neutron E2 is activated if the activity of neutron E1 is more than 3 units.

The resulting diagram shows the changes in the activity of E1 and E2 neutrons over time. This diagram shows that E1 was active at the beginning of the work, and then with increasing time, its activity decreases, and instead, E2 neutron activity increases to create a reasonable balance of two neutrons.



- Fixed point for $k_1 = 150\,, k_2 = 90$
- (60,0)

$$\Delta = 1 + 9M^2N^2\sigma^{2N}(k_2 - 3E_1)^{N-1}(k_1 - 3E_2)^{N-1}$$

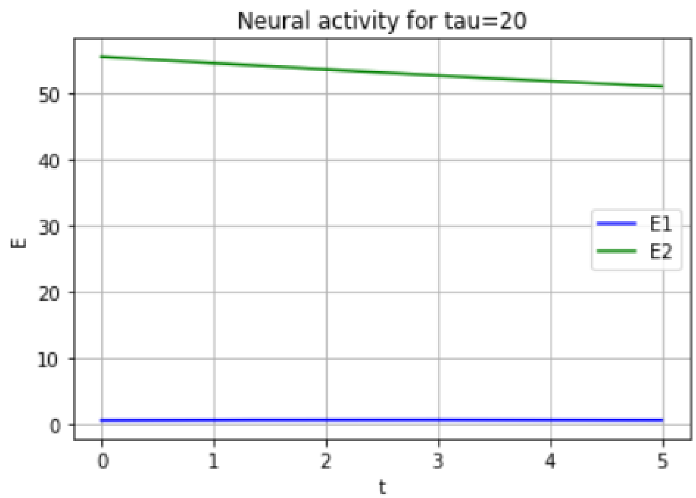
$$\Delta \cong 100.7 * 10^{16}$$

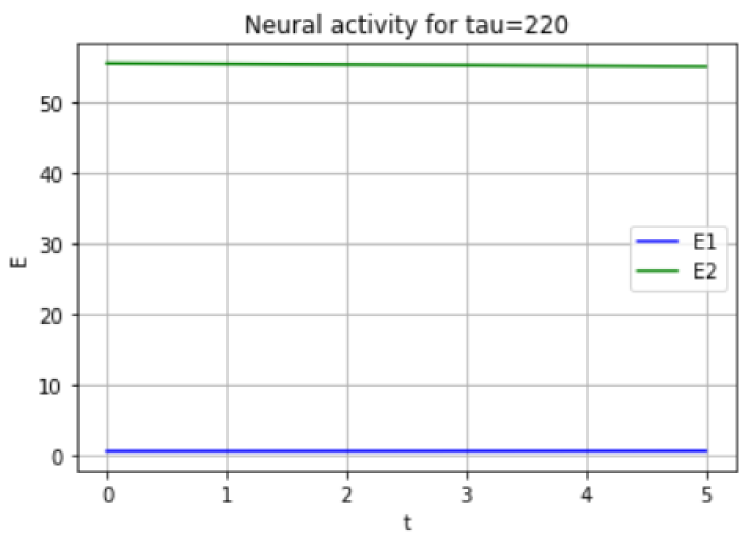
$$\lambda_1 = \frac{\tau + \sqrt{\tau^2 - 4\Delta}}{2} = 0$$

$$\lambda_2 = \frac{\tau - \sqrt{\tau^2 - 4\Delta}}{2} = -2$$

If it's node it's stable , if it's saddle , it's always unstable.

2.5)





The resulting graphs show that increasing the time constant value of tau decreases the rate of changes of E1 and E2. As a result, the balance in the neural activity graph with tau=220 is generally similar to the neural activity graph with tau=20; However, the speed of changes in the value of E1 and E2 in the graph with tau=220 is lower.

2.6)

In the decision making scenario in WTA network, the point to be considered is that only one active or dominant neuron can be produced as output, so in this network, only one neuron is selected and activated and other neurons are automatically They return to inactive mode. This feature of the WTA network is suitable for applications where only one pattern decision is required in the presence of several different patterns.