ACIT4310 final project

Sebastian T. Overskott

November 2021

Contents

1 Introduction

This report describes the work and results from a graded project in ACIT 4310 - Applied and Computational Mathematics. It consists of analyzing, solving, simulating and interpreting the Fitzhugh-Nagumo model. First, I will describe the origin and use of the explored model. Then I will look at the model with different analyzing tools such as nullclines, stable states, phase plots and bifurcation plots. The main part is the analysis section containing several findings. Finally, I will be presenting some results and a short discussion at the end.

Most of the analysis and solving has been done numerically using Matlab. Some relevant code and results from Matlab will be presented in `verbose` . The script is provided in the appendix.

2 Background and model description

The Fitzhugh-Nagumo model describes an excitable system i.e. a neuron [1]. It was suggested in 1961 by R. Fitzhugh, and the year later by J. Nagumo et. al. It is an 2D simplification of the Hodgkin-Huxley model (HH) which models the neuron impulses of squid giant axons. Implementing the 4D HH on the analog computers that was available at Fitzhugh disposal at the time, was a complex task. This motivated the creation of a simpler model with the same behavior as HH. Due to the new model similarities to the Van der Pol oscillator, Fitzhugh named the new model "Bonhoeffer–Van der Pol oscillator". Around the same time, Nagumo et al. found the same types of equation while designing a "monostable multivibrator" electrical circuit. Today, the name of the model honors both these discoveries.

Here are the Fitzhugh-Nagumos equations as presented in this project:

$$
\begin{aligned}
v' &= v - \frac{v^3}{3} - w + RI_{app} \\
w' &= \frac{v + a - bw}{\tau}, \quad \tau > 0
\end{aligned}
$$

The model is a non-linear system of ODE with two variables, where v is the membrane voltage, w is recovery variable, a, b and τ are parameters, I_{app} is the applied current and R is a constant scaling I_{App}. To simplify, I will set $R = 1$ throughout this project and not include it in equations from now on. Based on the original name, "Bonhoeffer–Van der Pol oscillator", and how this model is described, we should expect some kind of cyclical/oscillating behavior with the right parameters. The Fizhugh-Nagumo model has been called "the prototypical example of an excitable system" [1], and is to this day still explored, with new discoveries being made.

2.1 First run

I used parameters, documented on Wikipedia, that creates the cyclical behavior (I will present the parameter values in detail in the analysis section). I solved the system with the `ode45()` in Matlab. I started with initial conditions $(0, 0)$ and $I_{app} = 0$, and increased the applied current to see what happened. I have plotted the results from three of these runs to illustrate the model behavior.

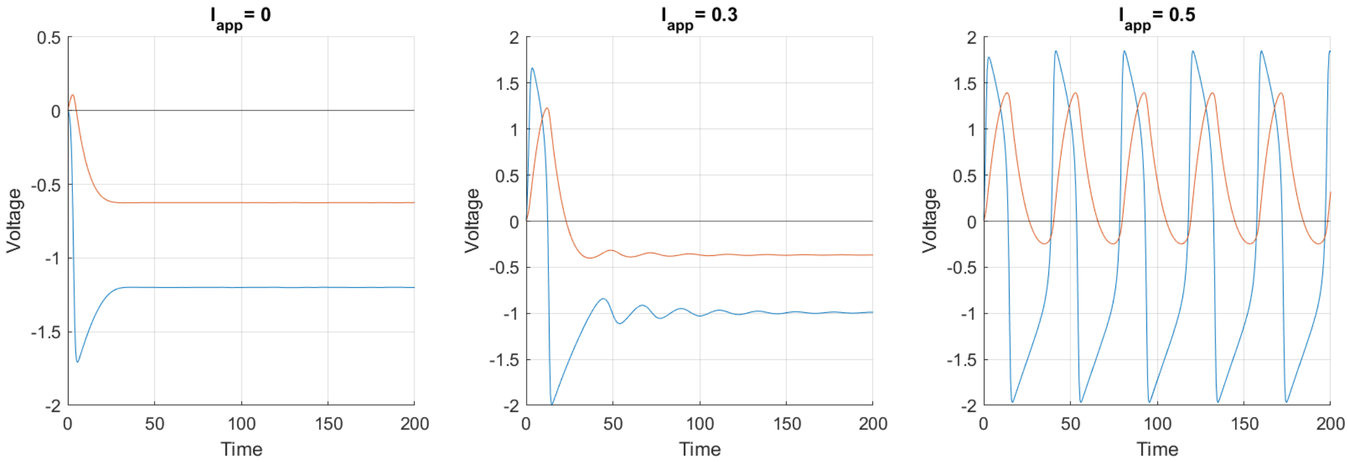

Figure 1: Solutions of the Fitzhugh-Nagumo model

At $I_{app} = 0$ we have an initial movement in membrane potential v (the blue line) before the system returns to what seems to be a stable steady state. At $I_{app} = 0.3$ we see a stronger initial movement of the voltages then some ripples before the system again stabilizes. It also starts to be visible how the recovery potential w (orange line) dampens the membrane potential. At $I_{app} = 0.5$ we see these regular spikes that were expected from the model description and it looks like we have an unstable steady state. I_{app} is changing the steady states. In the next section will we analyze the model and see if we can obtain more information about this behavior.

3 Analysis

We've see that there is interesting behavior when the applied current I_{app} is adjusted. I will analyze the nullclines, steady states and phase orbits to get some understanding on what is happening. For models like this, describing a biological phenomenon, it's common to keep the dimensionality. This is convenient when the model variables and parameter are directly related to physical quantities i.e. voltage. I will use the parameters found at Wikipedia [2] during the analysis, and only adjust the I_{app} to study the bifurcation of the system.

3.1 The parameter values used

If nothing else is stated, these are the values we will use throughout this paper:

Parameter	**Value**
a	0.7
b	0.8
τ	12.5
I_{app}	0.5

Table 1: Parameters and values used.

Here's the solution for given parameters:

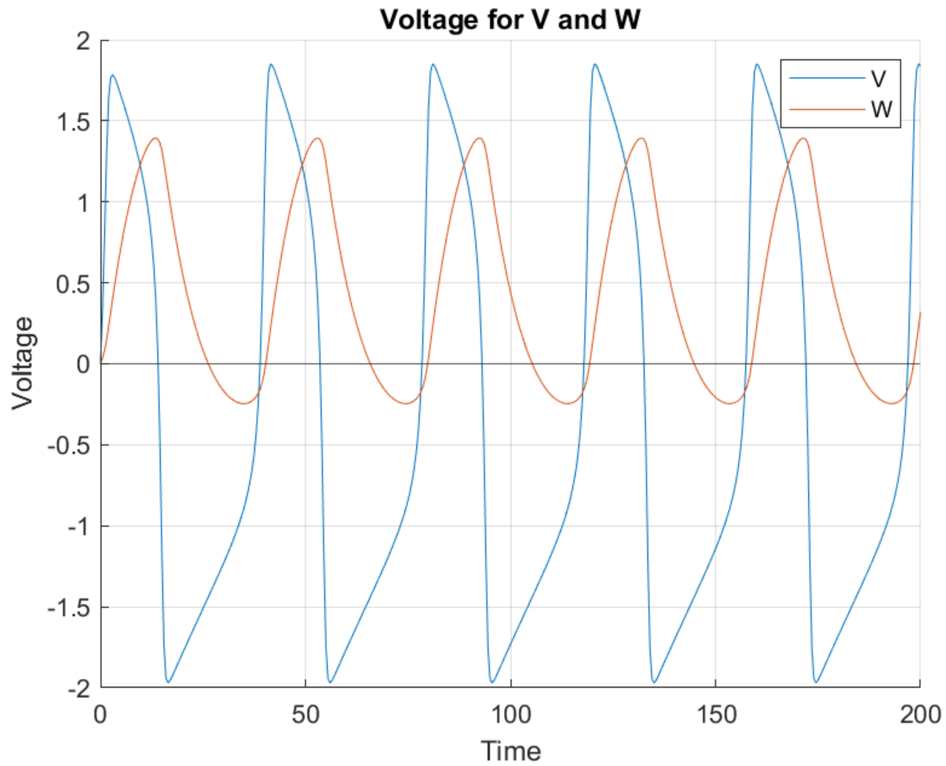

Figure 2: Solution for $a = 0.7, b = 0.8, \tau = 12.5, I_{app} = 0.5$.

3.2 Nullclines

In this analysis we will firstly look at the nullclines of the system. We'll see if we can get an idea of the dynamics of the system. We set $v' = 0$ and $w' = 0$ and obtain these functions for the nullclines:

$$0 = v - \frac{v^3}{3} - w + I_{app} \tag{1}$$

$$0 = \frac{v + a - bw}{\tau}, \quad \tau > 0 \tag{2}$$

We have one linear equation and one cubic equation for the nullclines. We now use Matlab to plot v:

$$v'_{nullcline} = v - \frac{v^3}{3} + I_{app} \tag{3}$$

$$w'_{nullcline} = \frac{v + a}{b} \tag{4}$$

It seems like there is one SS in the intersection of the graphs.

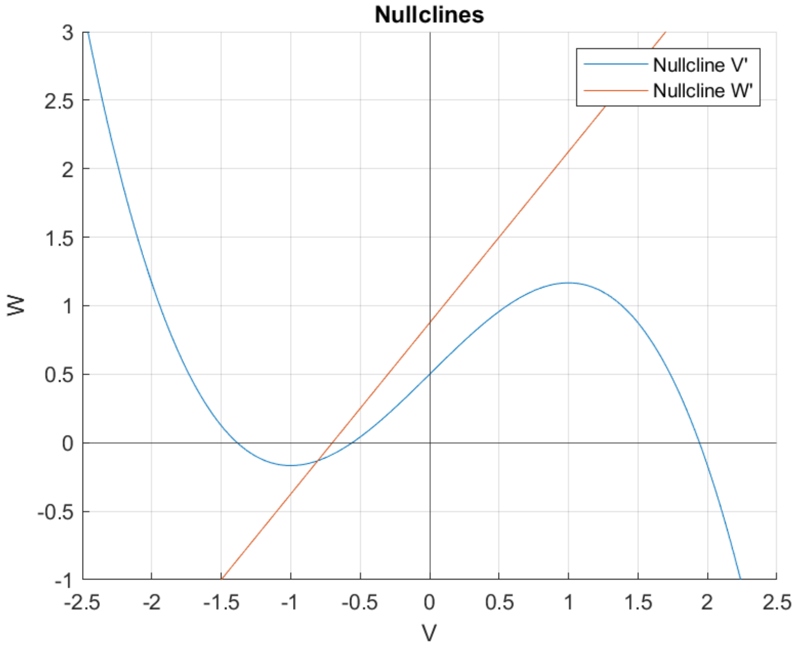

Figure 3: Nullclines for v' and w'.

What about the dynamics?

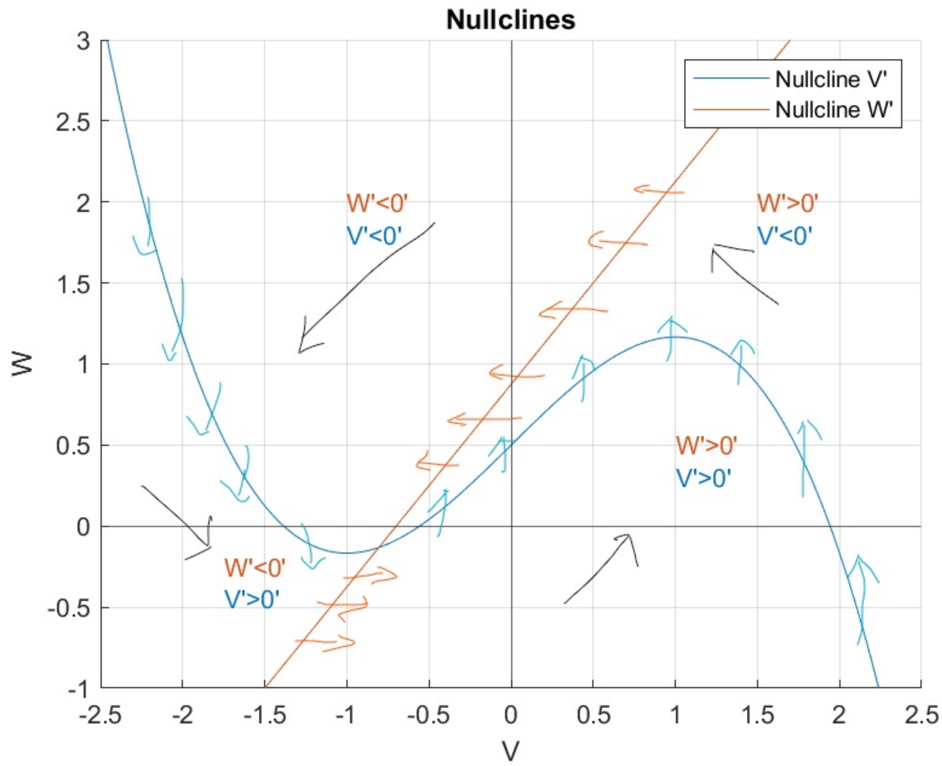

Figure 4: Nullclines with trajectories in v and w.

The SS seems to be a center or a spiral. Let's look at the SS to see if we can find out.

3.3 Steady states

We need to find steady states (SS) for this parameter setting and check the stability. Additionally, we have to understand what kind of SS it is i.e. node, spiral, center or saddle. We set $v' = 0$ and $w' = 0$ again, and rewrite to get this equation set:

$$0 = v - \frac{v^3}{3} - w + I_{app} \tag{5}$$

$$w = \frac{v+a}{b} \tag{6}$$

Which gives us:

$$0 = v - \frac{v^3}{3} - \frac{v+a}{b} + I_{app} \tag{7}$$

We want to use Matlab and the function `roots()` to solve this. The function requires a coefficient vector as an input. By rewriting (7), we obtain:

$$0 = -\frac{1}{3}v^3 + \frac{b-1}{b}v - \frac{a}{b} + I_{app} \tag{8}$$

We then create the coefficient vector and solve:

```
v_coef = [-1/3 0 (b-1)/b -(a/b)+R*I_app];
v_roots = roots(v_coef);
```

The v-value for the SS:

```
v_roots =


   0.4024 + 1.1117i
   0.4024 - 1.1117i
  -0.8048 + 0.0000i
```

We obtain two imaginary solutions and one real solution. We're only interested in the real solutions and ignore the complex solution. That means there is one SS, which was clear from the nullcline plot. Let's submit the result for v into the coefficient vector for $w' = 0$:

$$0 = -bw + a + v \tag{9}$$

Here we use the SS value for v we found previously. This will result in the coefficient vector for w (the index of the `v_real(i)` and `w_roots(i)` is for handling multiple SS):

```
w_coef = [-b a+v_real(i)];
w_roots(i) = roots(w_coef);
```

Here's the w-value is:

```
w_roots =

  -0.1311
```

And if we plot the found values we see that we found the SS (with possible numerical errors):

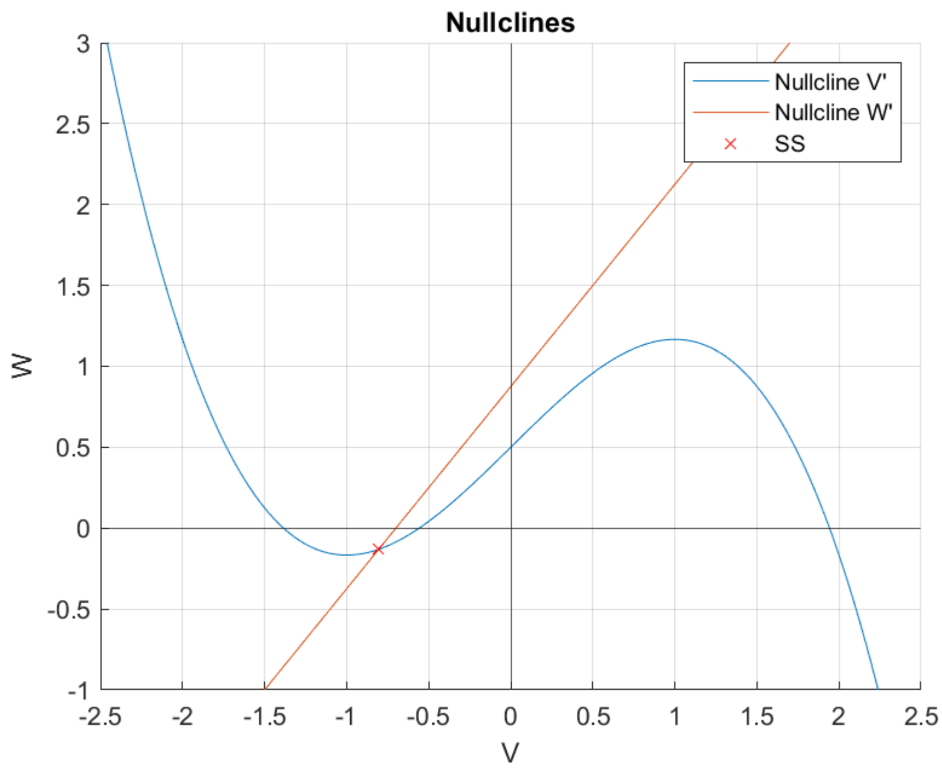

Figure 5: Nullclines with indicated SS

3.4 Jacobian and type of stable state

We now know the SS coordinates for this configuration: (-0.8048, -0.1311). So what kind of SS is this? Let's check! First we need to find the Jacobian of the SS.

Let:

$$f(v, w) = v - \frac{v^3}{3} - w + I_{app} \tag{10}$$

$$g(v, w) = \frac{v + a - bw}{\tau} \tag{11}$$

and the Jacobian:

$$J = \begin{bmatrix} \frac{\partial}{\partial v} f & \frac{\partial}{\partial v} f \\ \frac{\partial}{\partial w} g & \frac{\partial}{\partial w} g \end{bmatrix} \Rightarrow \begin{bmatrix} 1 - v^2 & -1 \\ \frac{1}{\tau} & \frac{-b}{\tau} \end{bmatrix} \tag{12}$$

Again, we can use Matlab for this by using the function `eig()` on the Jacobian to find the eigenvalues:

```
% Jacobian values
f_v = 1-(v_real^2);
f_w = -1;
g_v = 1/tau;
g_w = -b/tau;

% Jacobian matrix
J = [f_v f_w;
    g_v g_w];

%Jacobian eigenvalues
J_eig = eig(J);
```

The result is:

```
J_eig =

   0.1441 + 0.1915i
   0.1441 - 0.1915i
```

So we see there are two complex eigenvalues. That means it's must be a spiral type SS. Since both of them are positive, it means that this is a unstable point. This complies with what we've seen in the solution (fig: 2). But since we have a cyclical behavior, maybe it will be some kind of stable cycle.

3.5 Phase plane

We can plot the solutions to v and w together to obtain an impression on how the solution moves around the nullclines.

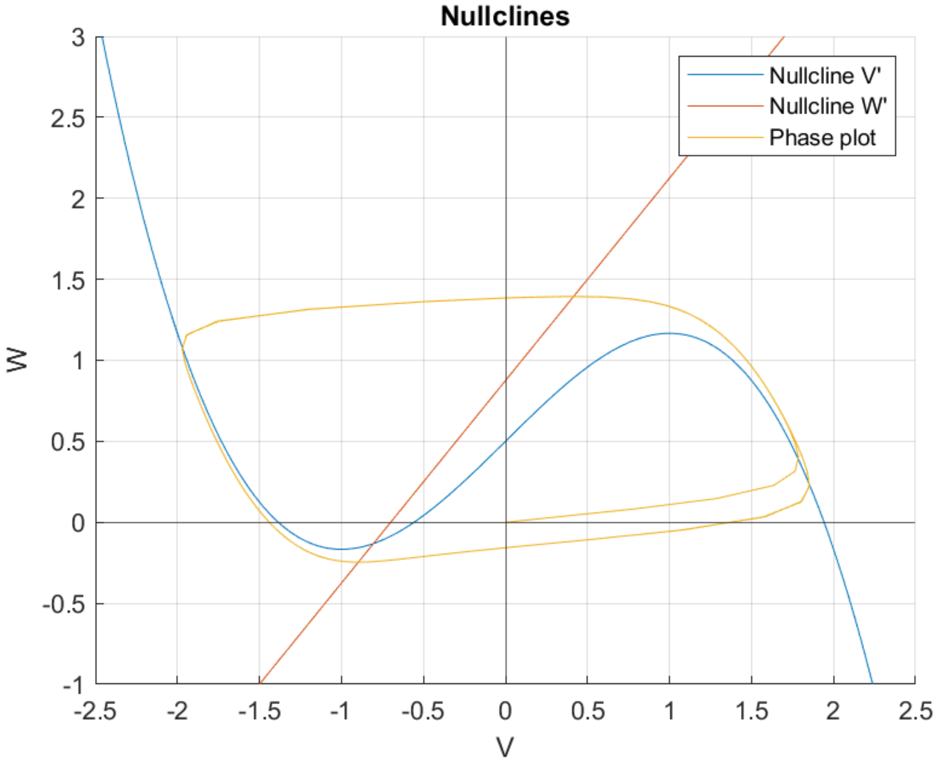

Figure 6: Nullclines with phase plane

We see that there is an oscillation that seems stable, which explains the spiking we observe in the solution. This is an unstable SS with a stable limit cycle.

3.6 Bifurcation

We have a behavior that shows stability and instability. Now, we can look at the bifurcation to see where the stable states changes based on changing I_{app} values. I have experimented with two different approaches to find these values. The first is to inspect the sign of the eigenvalues for different I_{app} values. If the real part of the eigenvalues are negative, the SS is unstable. Vica verca, positive eigenvalues means the SS is stable. If we find the point where the sign changes, we have the point where the SS changes its stability. I have plotted the real part of the two eigenvalues for $I_{app} \in [0, 2]$ along with the intersection points at $v = 0$ in the plot below.

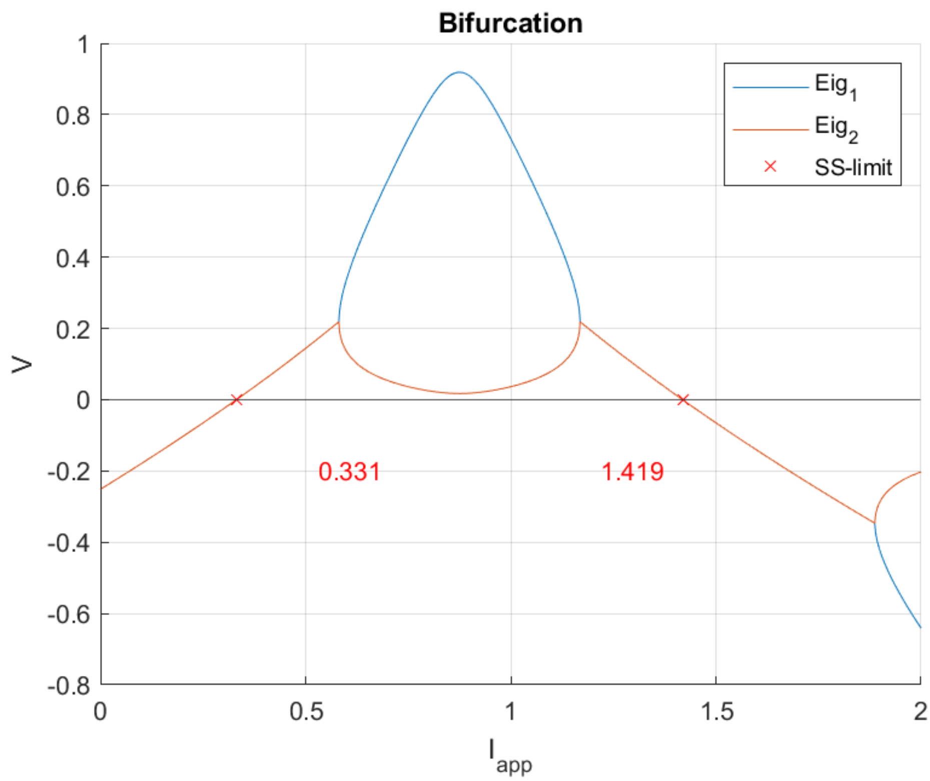

Figure 7: Bifurcation plot of eigenvalues of the Jacobian.

We see that there are some bifurcation both where Re is positive and negative, but both eigenvalues have the same sign everywhere (for this parameter configuration). So now we know that the $I_{app} = 0.331$ is where the SS change from stable to unstable and $I_{app} = 1.419$ is where it changes back again. We can verify this result with my second method. Instead of using the eigenvalues, we can use the trace and determinant of the Jacobian to find the same points.

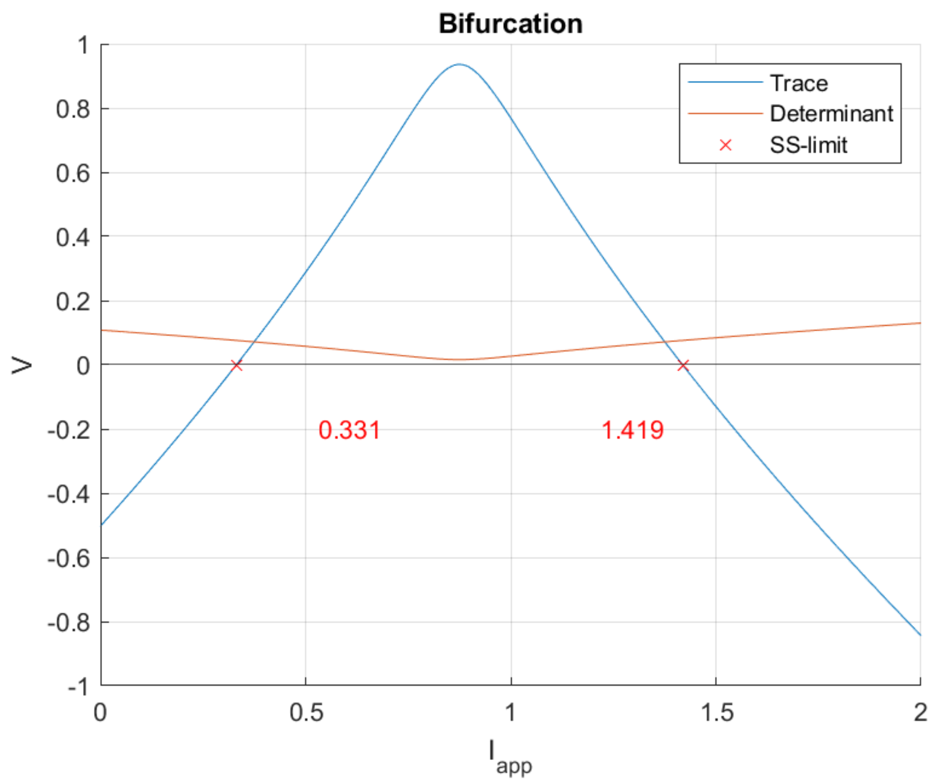

Figure 8: tr and det of the Jacobian

The blue line is the trace of the Jacobian for different I_{app} values. If it is positive, we have instability. The line intersects the X-axis in the same point as the eigenvalue method. The determinant value in orange is required to be positive for this method to be valid, which we see it is for all $I_{app} \in [0, 2]$.

4 Numerical methods and schemes

The approach to analyze and solve the model in this project has relied heavily on numerical approximations. It is therefore fair to study the schemes used, together with other schemes for comparison. There are several numerical schemes we can use to approximate the solution of ODEs. Here we will go through three well known schemes: Eulers method, Heuns method and Runge-Kutta fourth order(RK4).

4.1 Comparison of three common schemes

The solver used in this project is Matlabs ode45, which is considered to have a better accuracy than the three previously mentioned (under the hood, ode45 is a 4-5 order Runge-Kutta with some smart time stepping). Let's use ode45 as the "accurate answer", and compare the other three method to this. We will use the Matlab functions for Euler, Heuns and RK4 provided in the course ACIT4310. We plot the three numerical solutions with increasing time steps to see how they behave.

At the lowest time step (fig: 9), there's obvious differences between the schemes, especially at the peaks. When we increase the number of time steps, the differences decrease. At 1001 time steps (fig: 11) the difference is almost indistinguishable at this pot resolution.

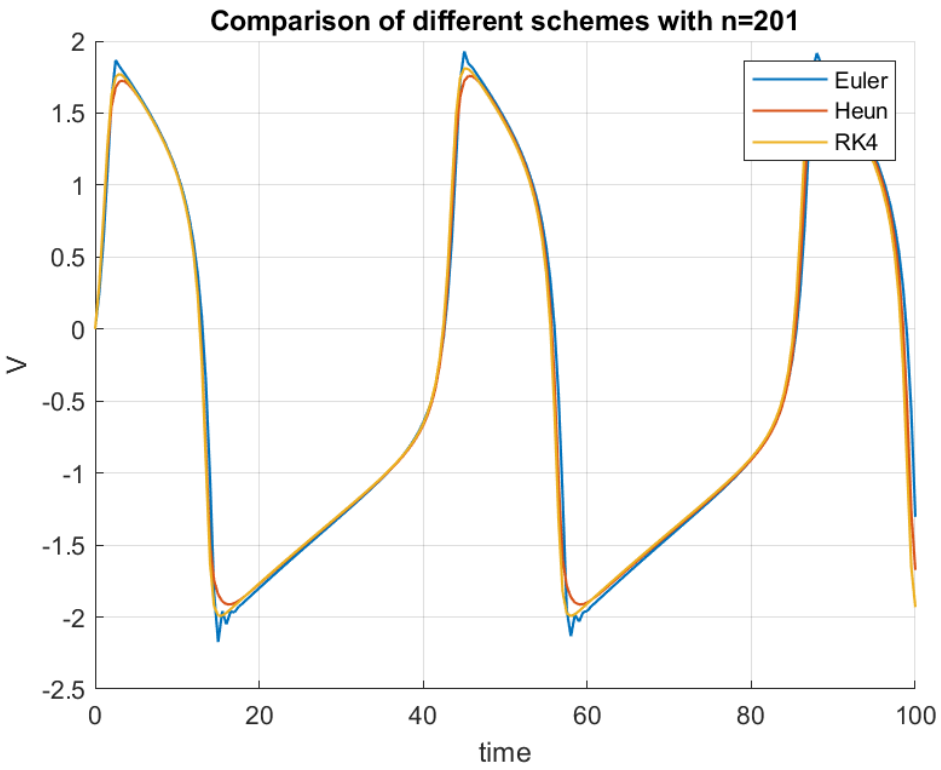

Figure 9: The three schemes with 201 time steps

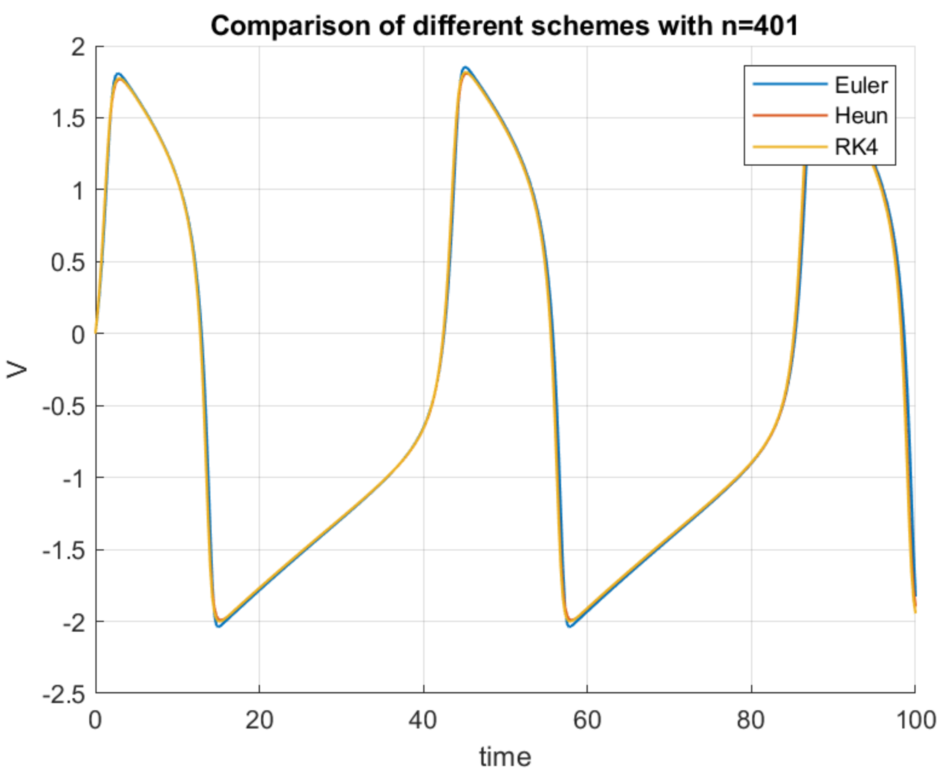

Figure 10: The three schemes with 401 time steps

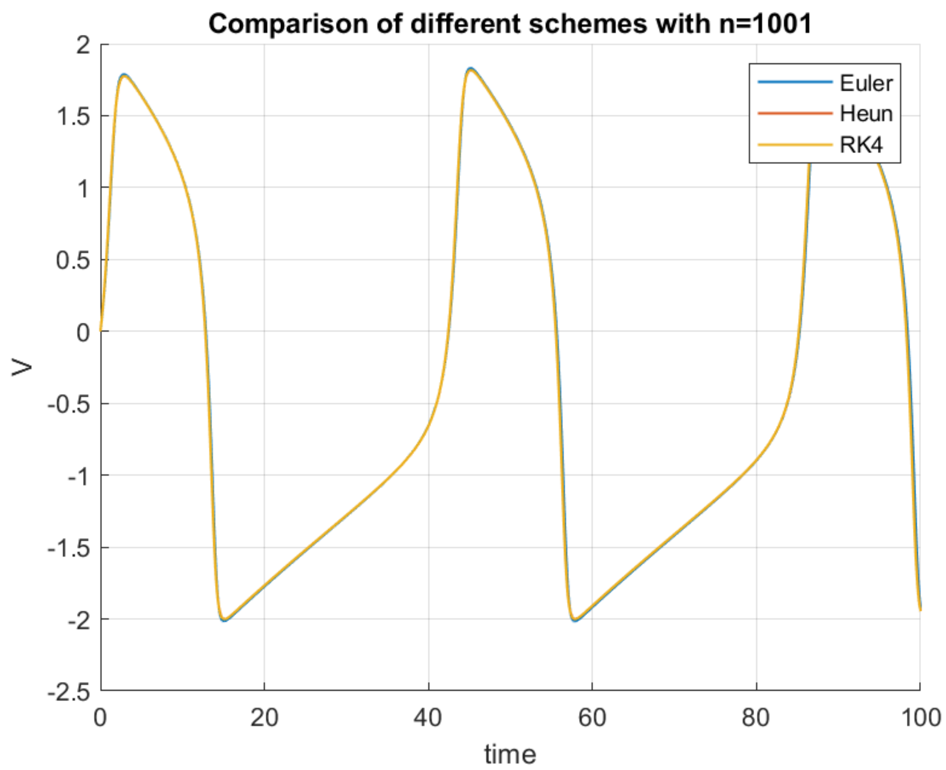

Figure 11: The three schemes with 1001 time steps

4.2 Error plot

To have a better view at the scaling, I will plot the differences from ode45 on a logarithmic plot, using a modified version of the `TestConvergenceODE.m` from ACIT4310. Here we calculate the mean value of the absolute difference between one of the three schemes and ode45.

Here the differences are more obvious. Runge Kutta is by far the most effective, at least while we have a small number of time steps. Euler is the least accurate for all time step numbers. Heuns is roughly as good as Euler at fewer time steps, but gets close to RK4 when the number of time steps is large. When the time steps gets really small, RK4 is flattening out. This is probably due to the fact that ode45 also is an RK4-5 (with improvements as mentioned earlier).

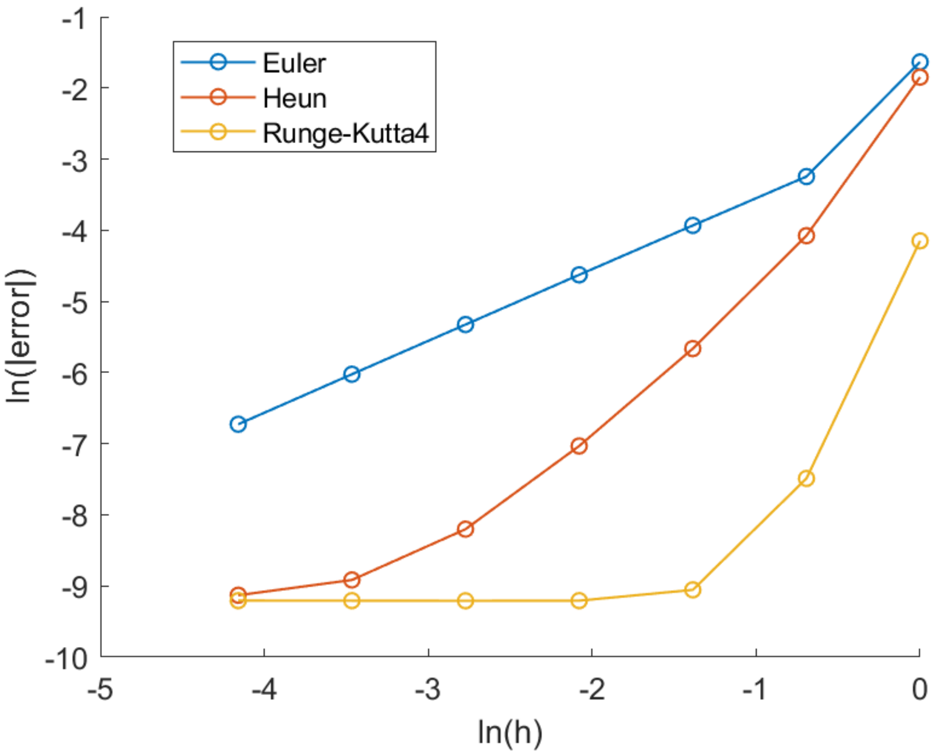

Figure 12: Error form ode45

5 Results and discussion

After analysing the model and understanding the dynamics, we now have a good idea on how the model behaves. We see that we get stable limit cycles for certain values of I_{app}. In a stable state, the system quickly finds the equilibrium. The other case (and in my opinion more interesting) is where the system is unstable. In this case, we can observe regular oscillation or spikes of the membrane potential. We see from the plots that this occurs when the linear nullcline crosses the qubic nullcline somewhere between its local extrema. This could be used to i.e. model a neuron receiving a signal and start spiking impulses.

5.1 Applied current

The most interesting part in this paper is how we can alter I_{app} parameter. We found that we expect a stable behaviour for $I_{app} < 0.33$ and $1.42 < I_{app}$, and unstable behavior with stable limit cycles everywhere else. I ran the script for four different values of I_{app} and the result (6.2) corresponds to our predictions from the analysis. You get a stable steady state when you are outside the boundaries, and stable oscillation when in the unstable area.

5.2 Potential three SS situation

I found a parameter setting by trial and error that gives us three SS in the model. If we use the same parameters but change $b = 2$, we get the SS situation plotted below.

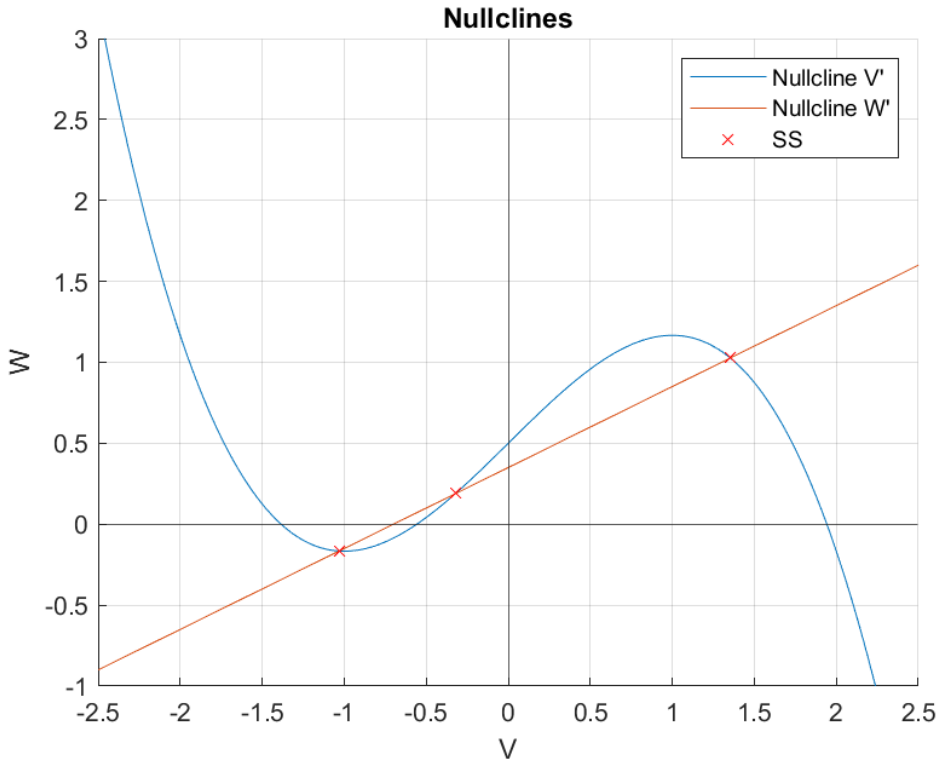

Figure 13: Situation where we have three SS, where $b = 2$.

I tried to use the same techniques as we did earlier with one SS. I found the three steady states, and calculated their eigenvalues:

```
J_eig =


   -0.6780
   -0.3144


J_eig =


  -0.1118 + 0.2787i
  -0.1118 - 0.2787i
```

```
J_eig =

     0.8140
    -0.0779
```

Here it seems like we have one stable node, one unstable spiral, and one unstable saddle point. The saddle point seems curious as I would have expected it to be a stable point. I therefore tested some different initial conditions to check its behavior.

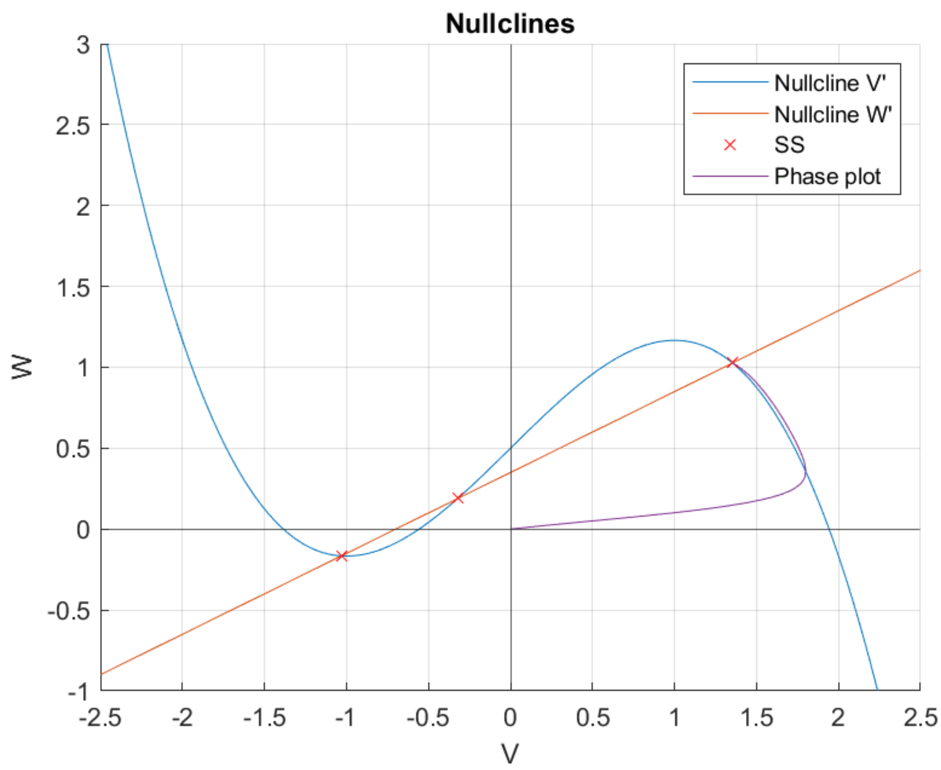

Figure 14: Run with initial conditions [0 0], where $b = 2$.

The lowest point behaves as a steady SS. The middle point is definitely unstable, but the last point seems stable. Here I assume the code I used contains bugs, which could explain the differences between calculations and plots. The adjustment of parameter b can change the stability of the system, but in what way and for which values? Unfortunately, I did not find the time to get to the bottom of this problem. The script I made works for analysing

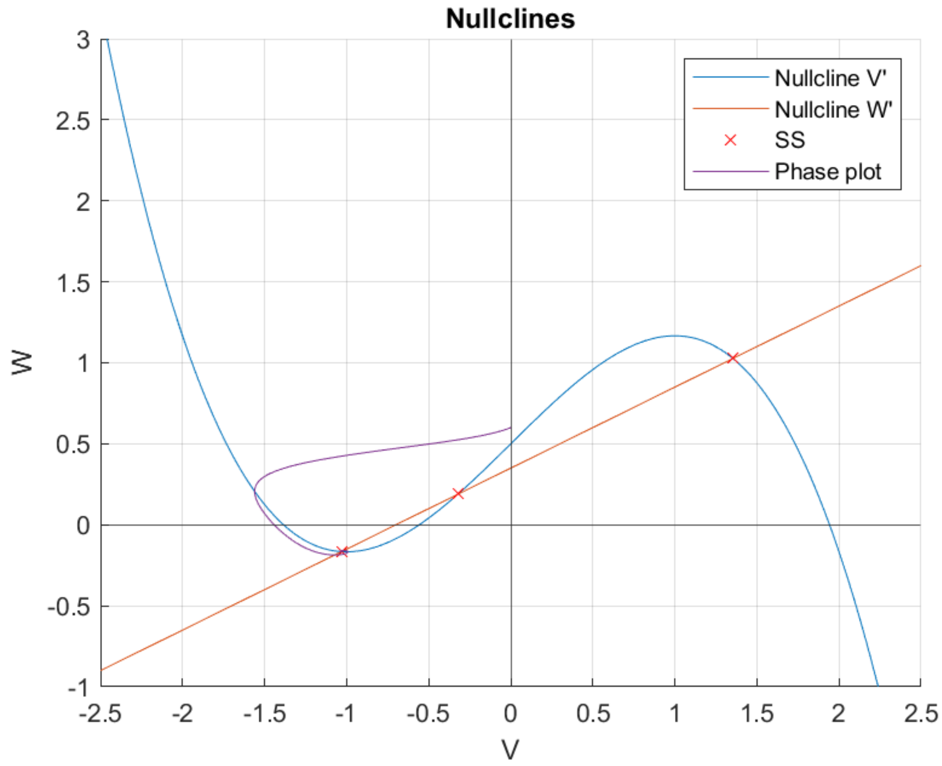

Figure 15: Run with initial conditions [0 0.6]. $b = 2$

systems with one SS. It also works well for finding and plotting several SS, but comes short when it needs to find the bifurcation of these states. This would be an obvious task to do if this work should continue. It would also be exiting to find real world examples of the Fitzhugh-Nagumo model and see how well it performs.

6 Appendix

6.1 Matlab code

The Matlab code used in this project can be found on github: `https://github.com/Overskott/ACIT4310---final-project`

6.2 Stable state results

Plots over solutions for different I_{app} values around the change of stability is steady states. $I_{app} = [0.32, 0.34, 1.41, 1.43]$. *(Plots starts on next page).*

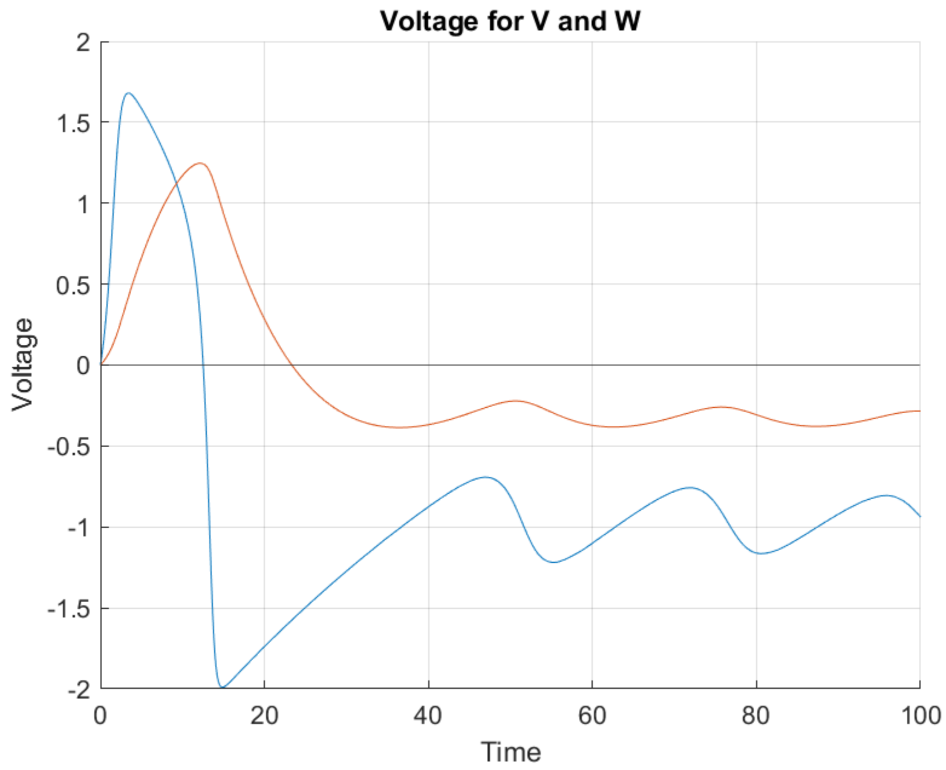

Figure 16: Stable solution for $I_{app} = 0.32$

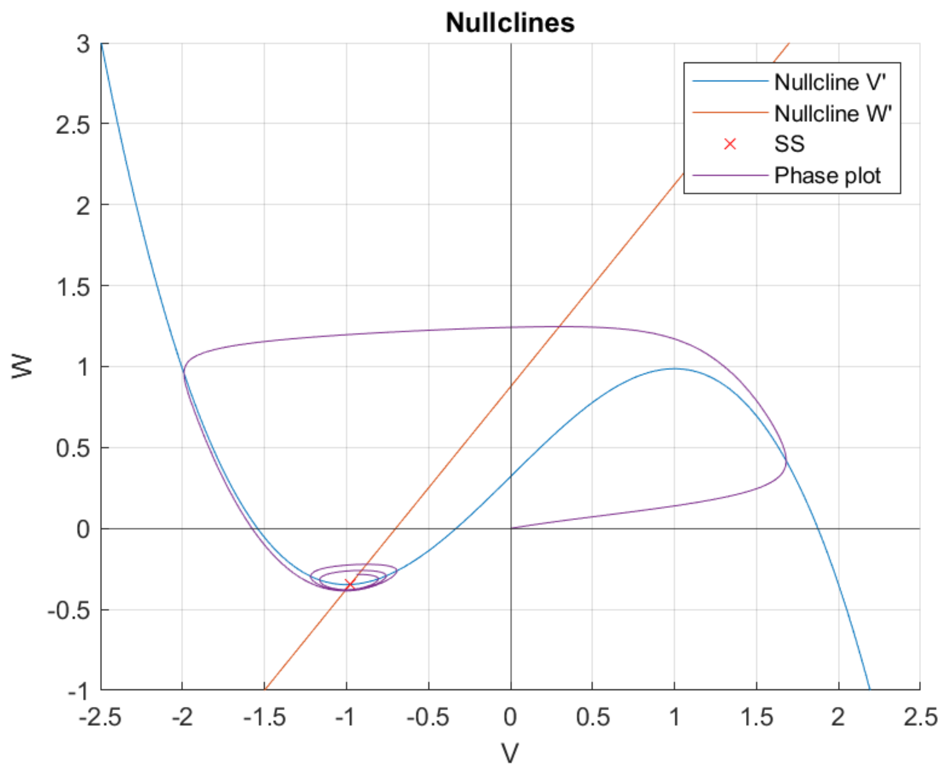

Figure 17: Nullcline with phase plot for $I_{app} = 0.32$

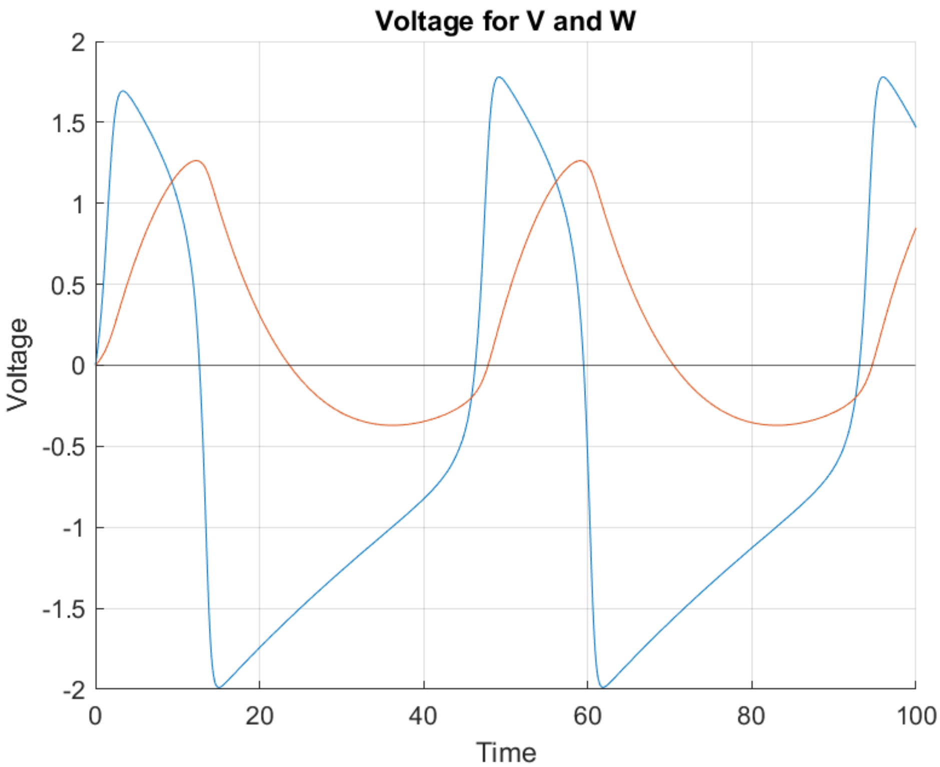

Figure 18: Unstable solution for $I_{app} = 0.34$

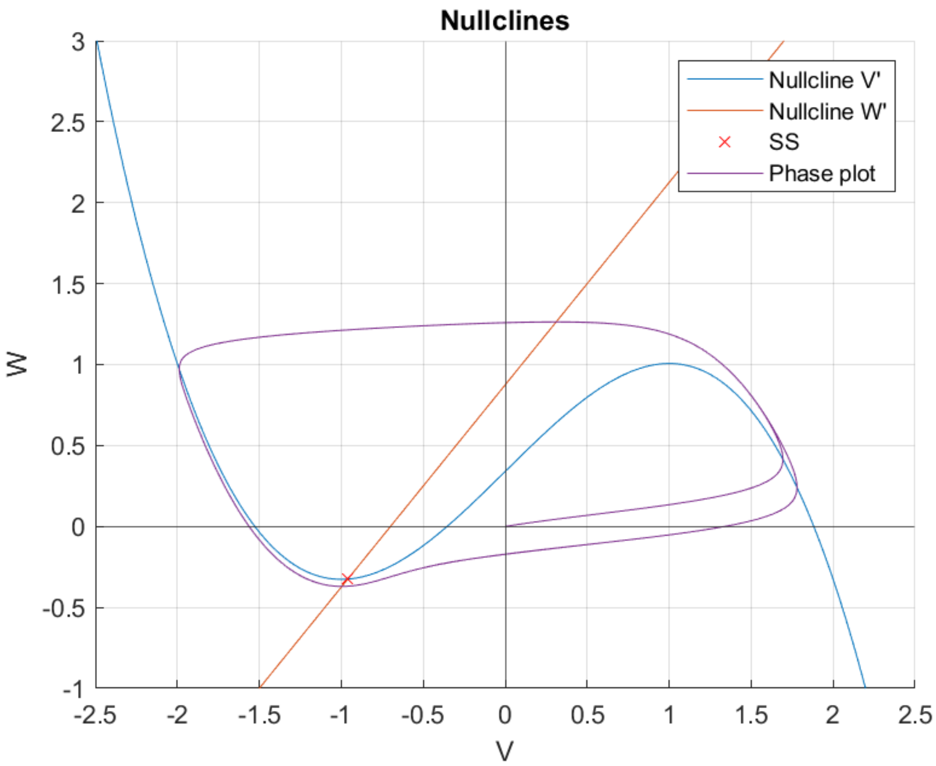

Figure 19: Nullcline with phase plot for $I_{app} = 0.34$

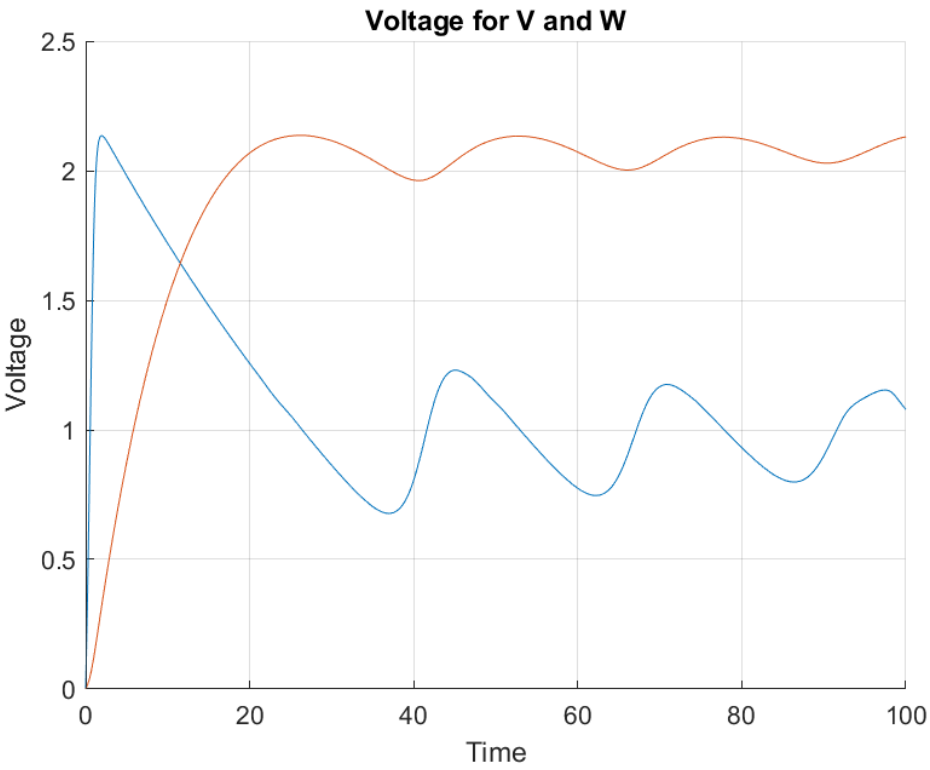

Figure 20: Stable solution for $I_{app} = 1.41$

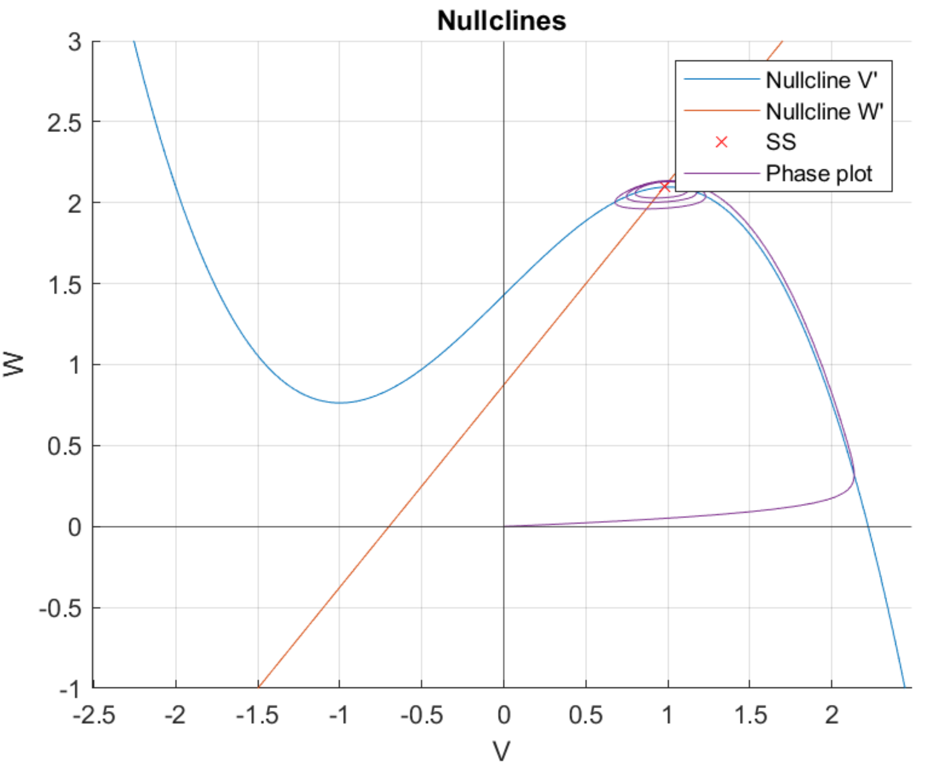

Figure 21: Nullcline with phase plot $I_{app} = 1.41$

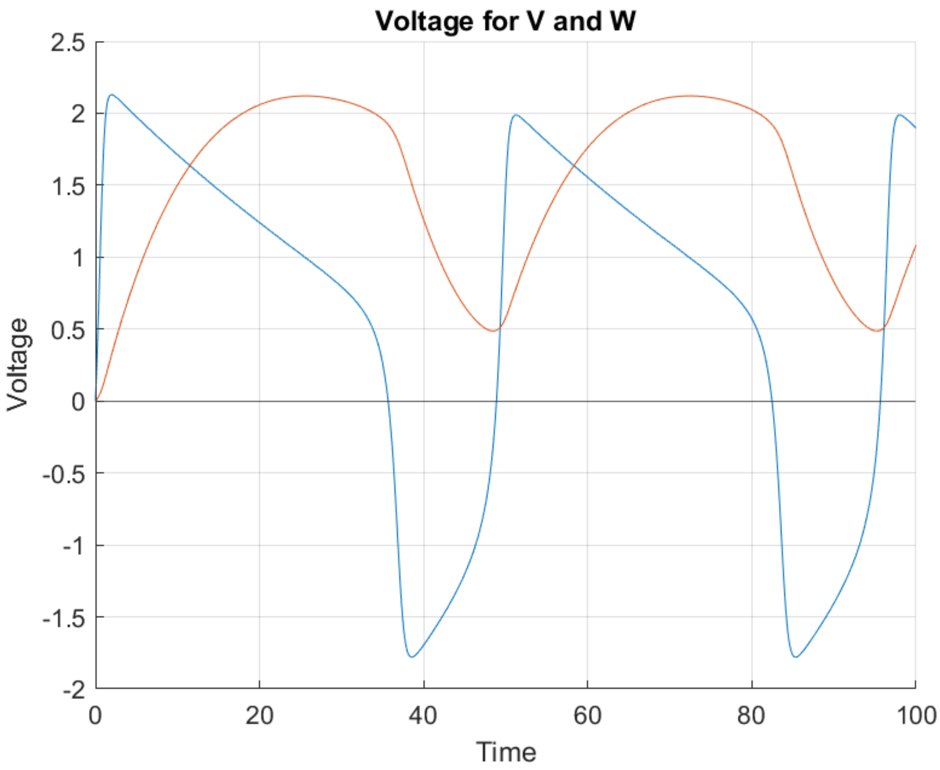

Figure 22: Unstable solution for $I_{app} = 1.43$

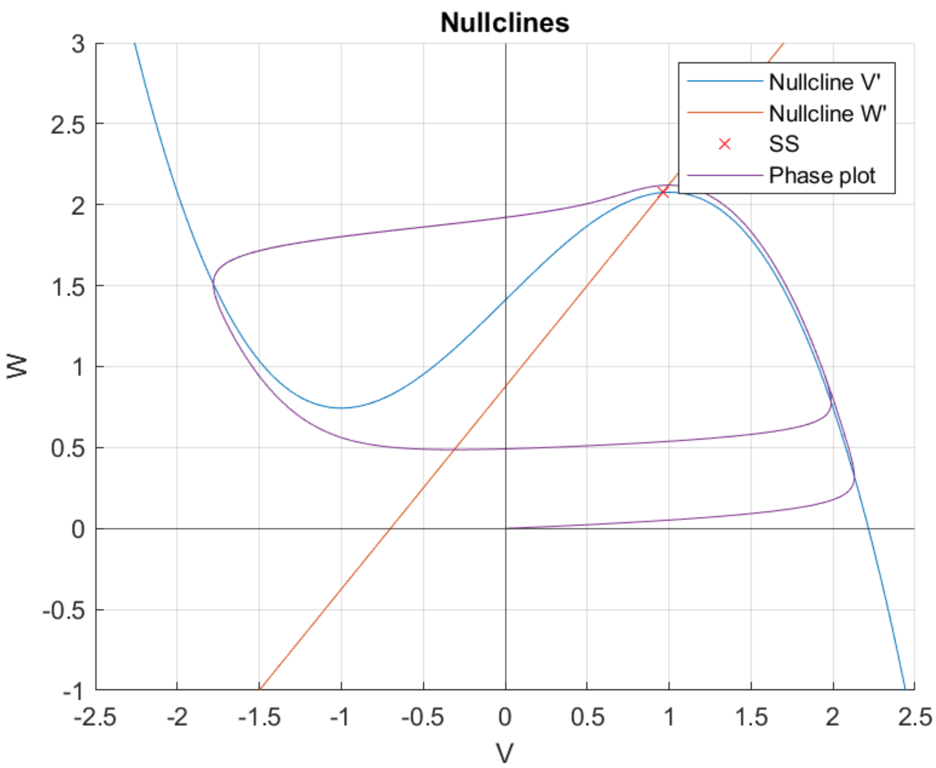

Figure 23: Nullcline with phase plot $I_{app} = 1.43$

List of Figures

References

[1] E. M. Izhikevich and R. FitzHugh. *FitzHugh-Nagumo model*. revision #123664. 2006. DOI: 10.4249/scholarpedia.1349.

[2] *Fitzhugh–Nagumo Model*. URL: https://en.wikipedia.org/wiki/FitzHugh\%E2\%80\%93Nagumo_model.