

Experimental Soft Matter Physics - Numerical Nonlinear Physics -

Huan Q. Bui

Colby College

PHYSICS & MATHEMATICS
Statistics

Class of 2021

September 15, 2020

Preface

This serves as my notebook for PH333: Experimental Soft Matter Physics. Due to the pandemic, we will be looking at numerical studies of nonlinear phenomena in interdisciplinary physics instead. This course is taught by Professor Jonathan McCoy.

Contents

Preface	2
1 The FitzHugh-Nagumo Model	5
1.1 Introduction & Background	5
1.2 The Model	5
1.3 Numerics & Analysis	5
1.3.1 Solving the Model: Preliminary	5
1.3.2 Nonlinear response to changes initial conditions	8
1.3.3 Nonlinear response to stimulus current: from non-periodic to periodic	11
1.3.4 Excitation block: From periodic to non-periodic	13
1.3.5 Refractory period	14
1.4 Summary	14
2 Turing Patterns (intro)	17
2.1 Introduction	17
2.2 Pattern-forming instabilities	19
2.3 Some strange behavior above Blue curve and below Orange curve	22
2.4 Wavelength from out of nowhere!	22

Part 1

The FitzHugh-Nagumo Model

1.1 Introduction & Background

Because neural signals are electrical in nature, it might be tempting to assume that neurons follow the same mechanism for transmitting signals as electrical conductors do. However, since neurons are poor conductors, the mechanism involving free electrons cannot explain this phenomenon.

It was discovered that neural signal transmission is a non-linear wave propagation of the *action potential* across neurons which occurs when the stimulus a neuron receives exceeds a certain threshold.

Proposed by Hogkin-Huxley and later appeared in a more simplified form as the FitzHugh-Nagumo (FHN) model, the mathematical description for this phenomenon is characteristic of a *excitable system*, a large class of systems in nonlinear dynamics.

[More about experimental and theoretical background here.](#)

1.2 The Model

The FHN model is given by a pair of coupled, first-order differential equations:

$$\begin{cases} \frac{dV}{dt} = V - \frac{V^3}{3} - W + I \\ \frac{dW}{dt} = \epsilon(V + a - bW) \end{cases}$$

where V is a function of time of the potential difference across the cell membrane, and W is a function of time of the recovery process of a neuron following an excitation. The constants a, b, ϵ are the model parameters. ϵ determines the relative rate at which the neuron recovers and the potential changes, while a and b are model constants. The variable I represents the *stimulus current* and can be time-independent. For the most part, we will assume that I is constantly 0, unless otherwise stated.

1.3 Numerics & Analysis

1.3.1 Solving the Model: Preliminary

We will solve the model using the `ode45` differential equation solver in MATLAB. To start, we write the differential equations into MATLAB:

```
function dy = FHN_model(t,y,I,a,b,epsilon)
% model equations
dy = zeros(2,1);
dy(1) = y(1) - (y(1)^3)/3 - y(2) + I;
```

```
dy(2) = epsilon*(y(1) + a - b*y(2));
end
```

Next, we solve! We choose the following values for constants in the model:

$$a = 0.7, \quad b = 0.8, \quad \epsilon = 0.08, \quad I = 0.$$

With this, we can find the equilibrium values for V and W by solving the equations:

$$\frac{dV_{eq}}{dt} = V_{eq} - \frac{V_{eq}^3}{3} + \frac{a + V_{eq}}{b} + I = 0, \quad \frac{dW_{eq}}{dt} = \frac{a + V_{eq}}{b} = 0.$$

We will choose a time span $t \in [0, 50]$. To obtain a numerical solution in MATLAB, one starts by selecting initial values for $V(0)$ and $W(0)$. To obtain a *non-trivial* solution, one must deviate from the equilibrium value, thus let us start with

$$[V_0, W(0)] = [V_{eq}, W_{eq} + 0.01].$$

With these items, we can now let MATLAB do the work:

```
% choose a value for I
I = 0;
% fix model parameters at traditional values
a = 0.7;
b = 0.8;
epsilon = 0.08;
% find equilibrium value of membrane potential
v_eq = fzero(@(v)(v - (v^3)/3 - (a+v)/b + I),0);
% find equilibrium value of recovery variable
w_eq = (a + v_eq)/b;
% pick initial values
y0 = [v_eq, w_eq - 0.01];
% pick a time interval
tspan = [0, 50];
% use the equations you defined
ode = @(t,y) FHN_model(t,y,I,a,b,epsilon);
% solve the equations
options = odeset('RelTol',1e-6);
[t,y] = ode45(ode,tspan,y0,options);
```

The solution is now stored in the variable $[t,y]$, where t is *time* and y contains the values for V and W . We would like to view the solutions. To this end, we will plot the membrane potential $V(t)$ and the recovery variable $W(t)$ respectively against time (t). We will also plot $W(t)$ against $V(t)$ (Figure 1.1). This will give us the *phase portrait* of the solution, provided these initial conditions above (Figure 1.2).

As expected, the membrane potential shows a significant initial increase, followed by a decay to its equilibrium value of -1.994 . Corresponding to this, the recovery variable is low in the beginning (signifying that the neuron might not trigger as well as before if stimulated), then increases and stabilizes to its equilibrium value of -0.6243 . On the phase portrait, it is clear that a small change in W corresponds to a large change in $V(t)$, which fits the description that $V(t)$ is the *fast* variable and $W(t)$ is the *slow* variable. As we will see, we can modify this relationship by changing the value of ϵ .

Now, suppose $I \neq 0$, i.e. there is a nonzero *stimulus current*. In the following two solutions, we observe two markedly different behaviors depending on the magnitude of I . In the first solution, I is nonzero but is

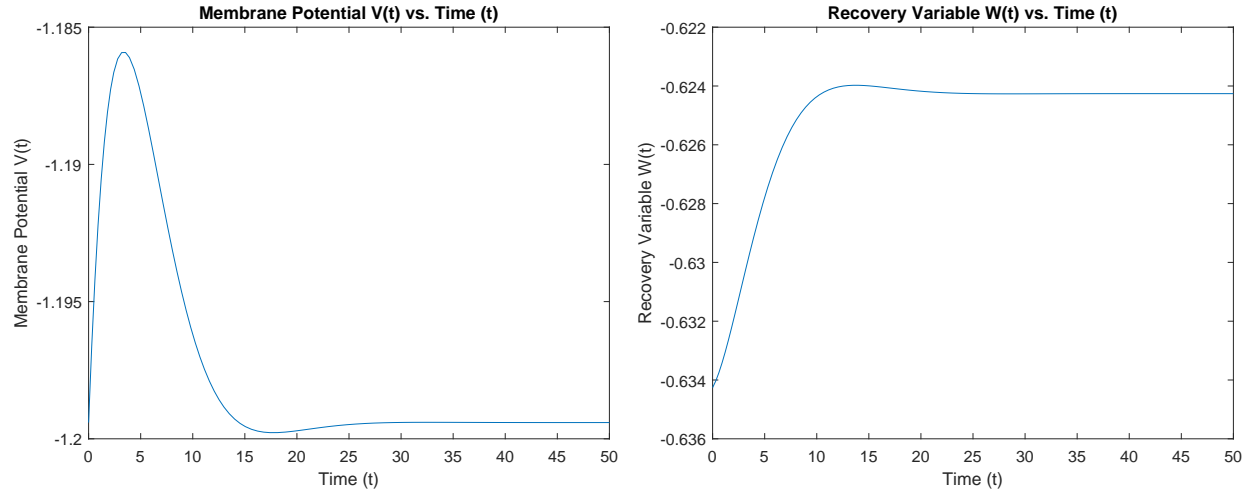


Figure 1.1: SAY SOMETHING HERE

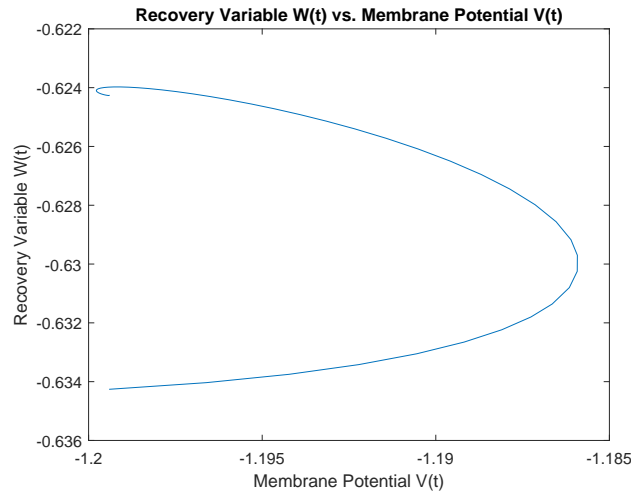


Figure 1.2: FILL IN CAPTION HERE!

small, representing a small stimulus current. In the second, I corresponds to a large stimulus. In Figure 1.4, the black line represents the solution for $I = 0$. Further, to illustrate the significance of this observation, we will solve the system for $t \in [0, 200]$.

We note that when I is small, the solution does not significantly deviate from the solution at $I = 0$. The difference is mainly a shift in the equilibrium value of both W and V , resulting in a “phase shift” on the phase portrait. The profile of the phase trajectory remains roughly the same. However, when the stimulus current I is sufficiently high, both the membrane potential and recovery variable exhibit large oscillations. This corresponds to a significantly larger phase trajectory on the phase portrait, as is evident in Figure 1.5.

While this might be of mathematical interest, we must also put the amplitude of I and the corresponding oscillation amplitudes into physical perspective. It is possible that the large deviations from the rest solution are not observable in the real world because stimulus currents of this magnitude might be deleterious to the neurons, i.e. they are too strong to only “stimulate” the nerve cells. It is also possible that different behaviors occur beyond this regime. In any case, we will return to this discussion in the subsequent sections.

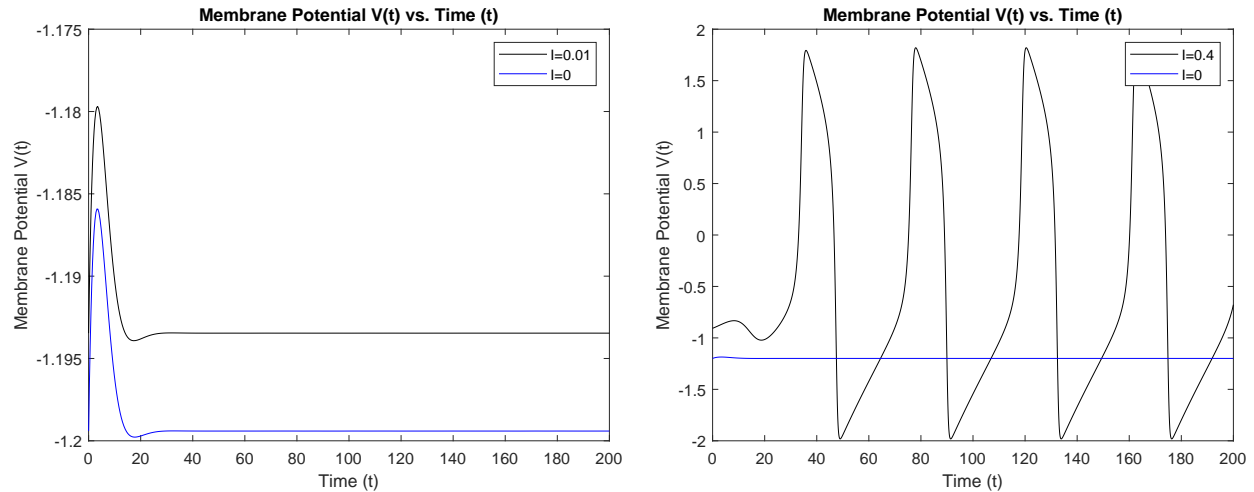


Figure 1.3: SAY SOMETHING HERE

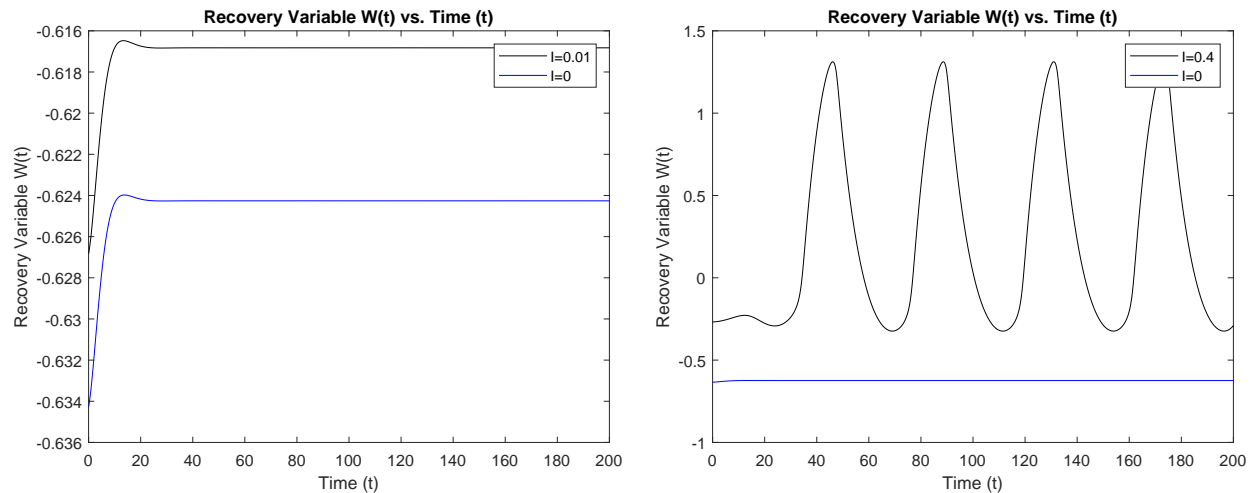


Figure 1.4: SAY SOMETHING HERE

1.3.2 Nonlinear response to changes initial conditions

It is often useful to study how solutions to an initial value problem change depending on the initial conditions. In some cases, small perturbations to the initial conditions can result in markedly different solution curves.

To see whether the FitzHugh-Nagumo model exhibits this behavior, we will create an array of initial conditions resulting from small perturbations from the equilibrium value. We will then numerically solve the model for each such initial condition and observe how the solution curves change accordingly.

To implement this in MATLAB, we will first set $I = 0$, the time span back to $[0, 50]$ and add the following lines to the existing code:

```
% create a vector containing a series of perturbation sizes
steps = 0.1:0.01:0.27;
% steps = sort([steps steps_critical]);
% create another vector in which to store responses
responses = zeros(1,length(steps));
```

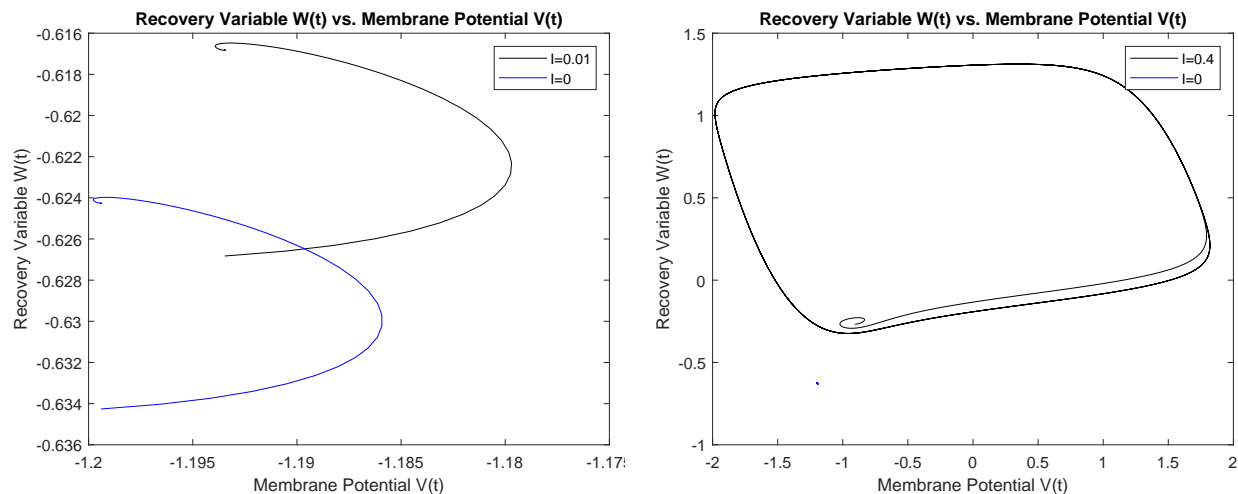



Figure 1.5: SAY SOMETHING HERE

```
% loop over all perturbation sizes

hold on
for j = 1: length(steps)
% pick initial values
y0 = [v_eq, w_eq - steps(j)];
% solve the equations
[t,y] = ode45(ode,tspan,y0,options);
% find maximum in membrane voltage response
responses(j) = max(y(:,1)) - v_eq;
V(t) vs. t
figure(1);
plot(t, y(:,1));
title('Membrane Potential V(t) vs. Time (t)')
xlabel('Time (t)')
ylabel('Membrane Potential V(t)')
hold on
% W(t) vs. t
figure(2);
plot(t, y(:,2));
title('Recovery Variable W(t) vs. Time (t)')
xlabel('Time (t)')
ylabel('Recovery Variable W(t)')
hold on
% phase trajectories
figure(3);
plot(y(:,1), y(:,2));
title('Phase Trajectories W(t) vs. V(t)')
xlabel('Membrane Potential V(t)')
ylabel('Recovery Variable W(t)')
hold on
end
```

There is a noticeable “bifurcation” in solutions; particularly, while for small perturbations in W solutions curves are concentrated in a small neighborhood of the solution at $\epsilon = 0$, sufficiently large perturbations result in a set of solutions characterized by a large initial spike. For example, the “bifurcation” in the membrane potential $V(t)$ solutions in Figure 1.6 is apparent at values of t around 5 and 10. With the initial solution marked with the black dashed line, we can see that for perturbations $\epsilon \approx 0.18$, the membrane potential exhibits a large spike up to approximately 1.8, followed by a proportionally marked depression as low as -2.3 in stark contrast.

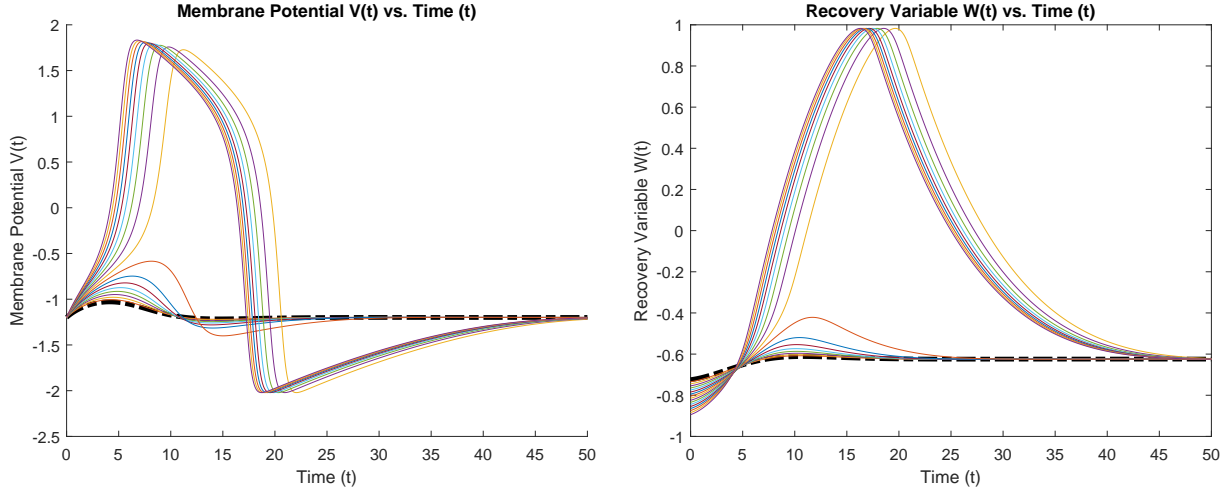


Figure 1.6: SAY SOMETHING HERE

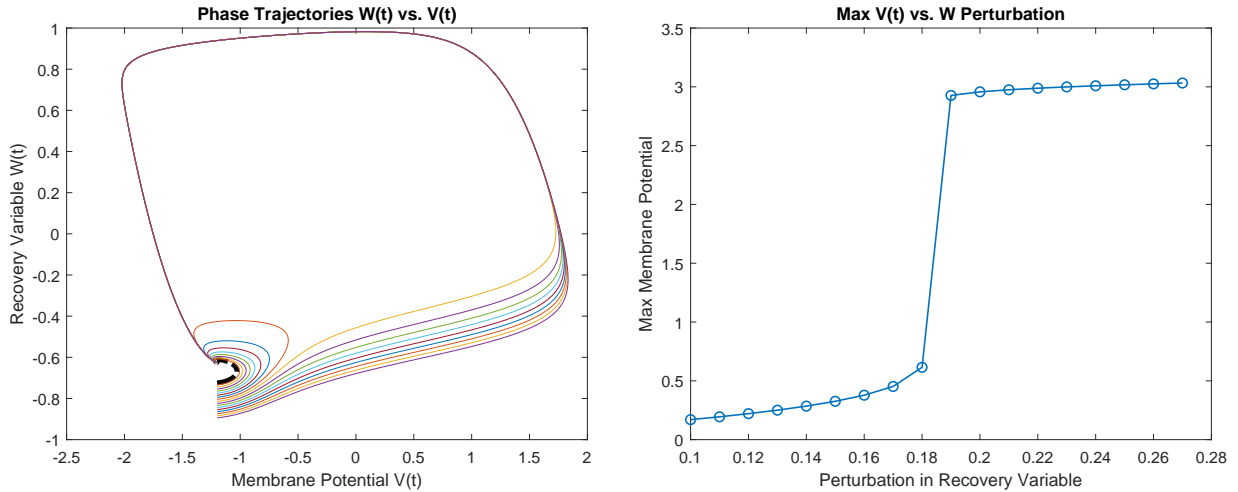


Figure 1.7: SAY SOMETHING HERE

The “bifurcation” is not clear on the phase portrait. This is due to the fact that the phase trajectory is much more sensitive to initial conditions near $\epsilon \approx 0.18$, which we will refer to as the *threshold* perturbation. To better resolve this “no man’s land” where solutions are highly unstable, as is evident in phase portrait in Figure 1.7, we will consider a set of closely spaced perturbations and find the associated solutions.

Figure 1.8 provides a clear visualization of what happens to the membrane potential near the threshold perturbation. At small t , all phase trajectories near threshold behave similarly. However, when t is sufficiently large, some solutions tend toward an increase in V when W increase, while some tend toward a decreasing V . In both cases, the change in V is always higher than the change in W , in general. Thus, this further

illustrates the description that V is a *fast* variable.

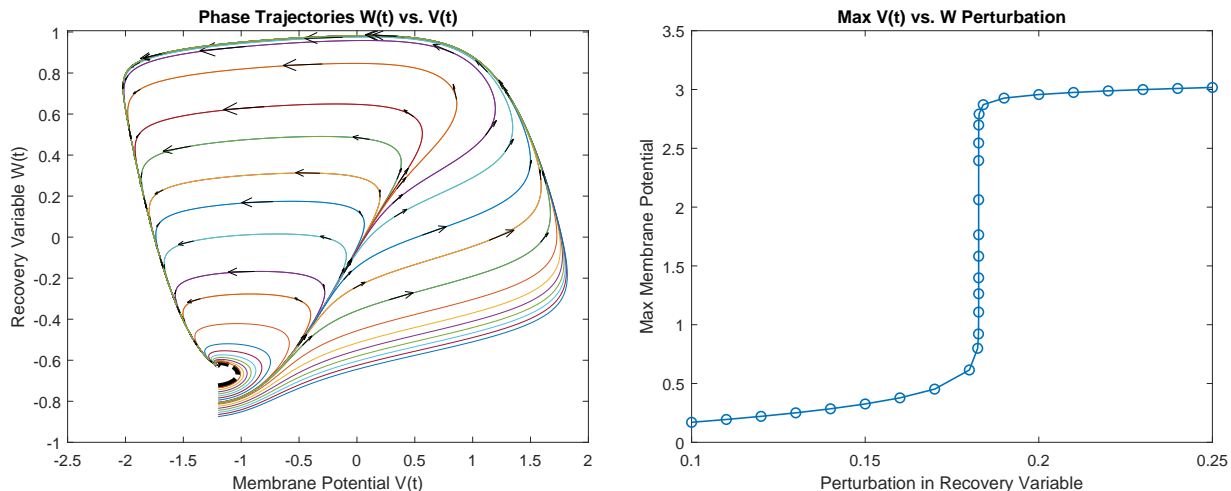


Figure 1.8: SAY SOMETHING HERE

Figure 1.9 provides a visualization of how solutions evolve as a function of the perturbation amplitude. In the previous plots, we saw a clear “bifurcation” in solutions. Here, we resolve how solutions change as we move past the threshold perturbation. It is clear that while a “bifurcation” is apparent in the case where the perturbation step size is linear, solutions in fact continuously and very rapidly morph as we cross the threshold. This illustrates the fact that the aforementioned “bifurcation” is an illusion due to the sensitivity of the system to changes in initial conditions.

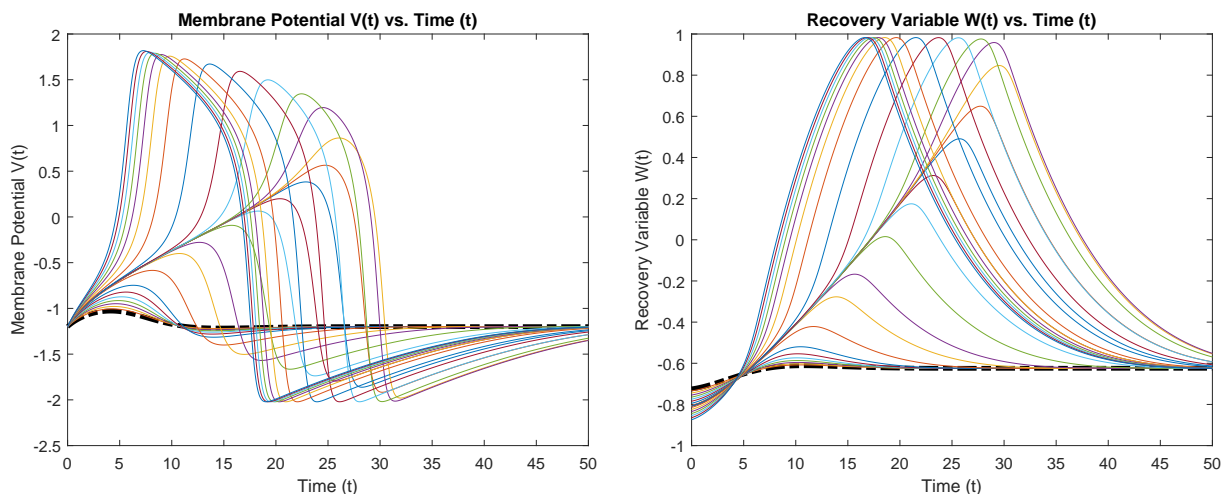


Figure 1.9: SAY SOMETHING HERE

1.3.3 Nonlinear response to stimulus current: from non-periodic to periodic

We have an almost identical description of the system as in the previous case when the stimulus current is assumed to be constant. As discussed earlier, there are two regimes for the values of I : when the stimulus current is sufficiently large, oscillations in the membrane potential occurs.

Now, a fascinating aspect of the model as far as the stimulus current is concerned is that there appear to be two types of threshold values for I : there is one threshold for I beyond which oscillations occur,

and another one beyond which the membrane potential spikes. Because these thresholds coexist, when I is continuously varied from a small to a large value, we observe *bands* of solution curves rather than a roughly uniform distribution of these curves.

To illustrate this, we first consider the regime where I is small (Figures 1.10 and 1.11). We observe a similar behavior in phase space where there is seemingly a “bifurcation” surrounding the threshold value for I . It also makes sense that large values for I correspond to a large spike in the potential curve. The phase portrait to almost identical to what we had before.

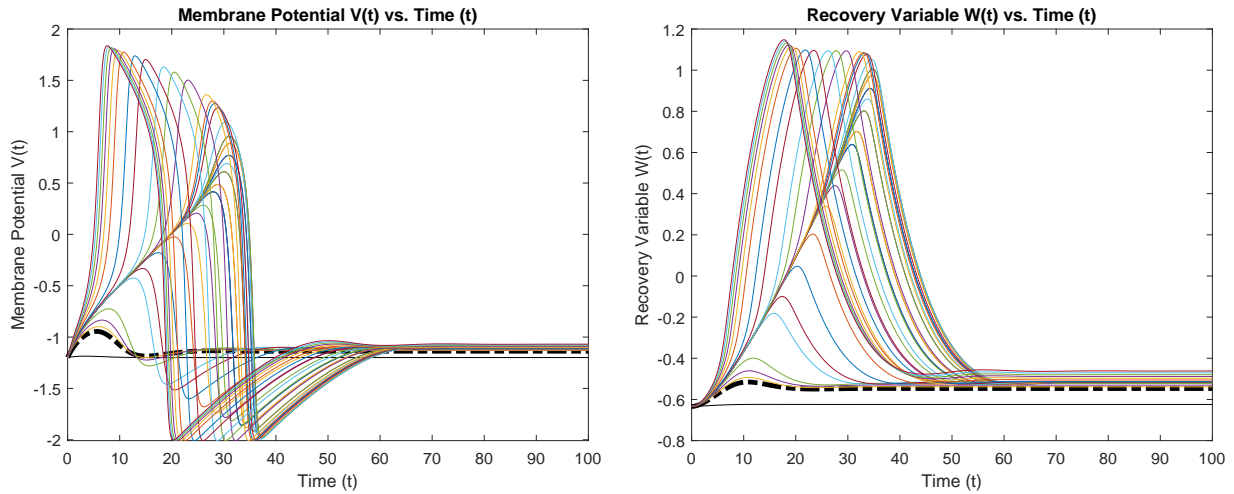


Figure 1.10: SAY SOMETHING HERE

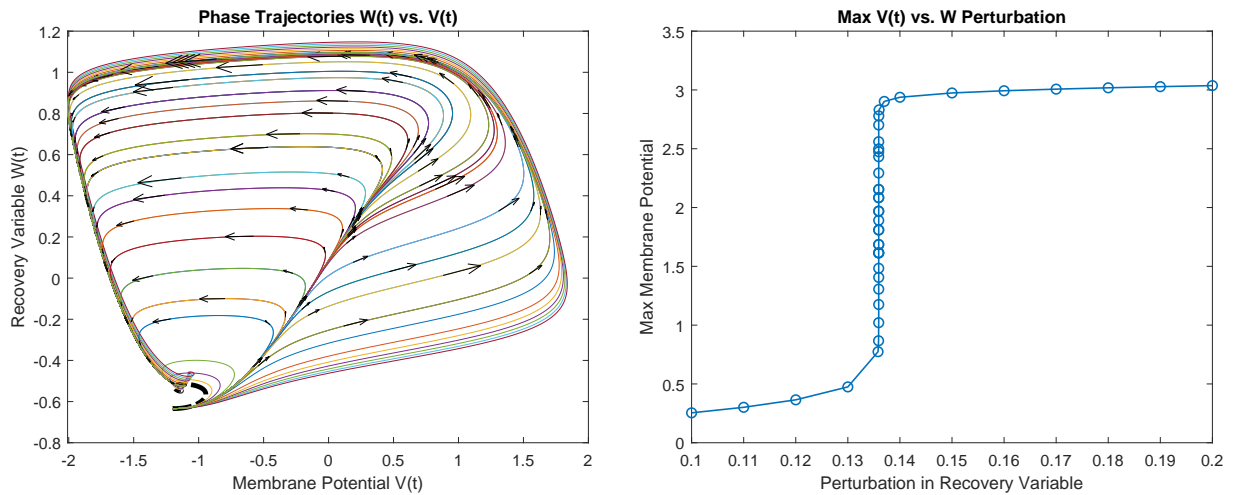


Figure 1.11: SAY SOMETHING HERE

However, when we extend the range of I , we begin to observe oscillations in the potential curves. What’s interesting is that, at each spike, we can observe an apparent “bifurcation” in the solution curves. These are also caused by certain threshold values for I in the range where I is high enough to generate the spikes leading up to the spike in consideration. Once again, because the solution curves and phase trajectories are very sensitive to I around this threshold value, we observe the *banding* of potential curves, just as we have before.

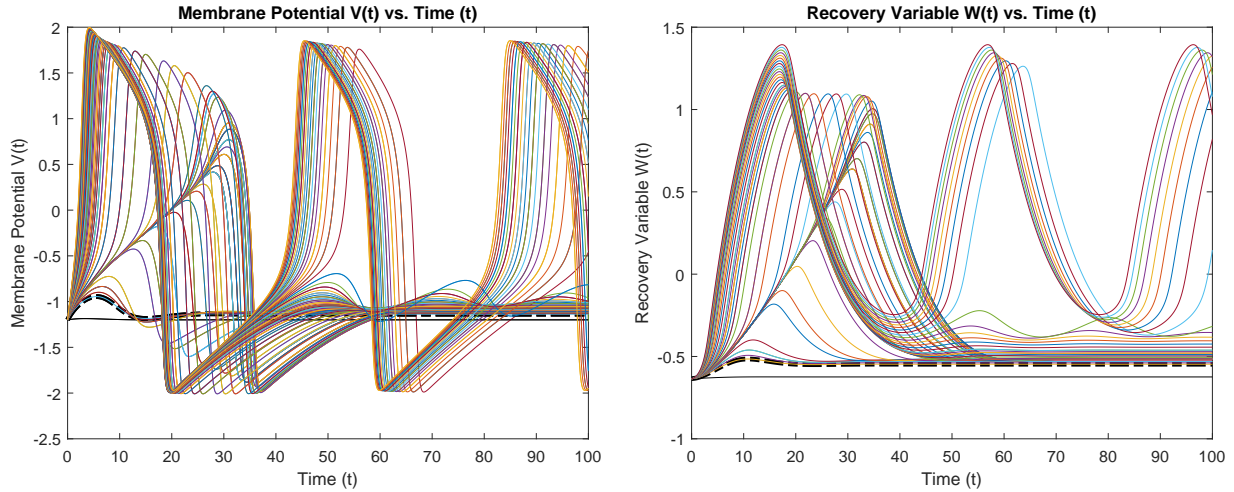


Figure 1.12: SAY SOMETHING HERE

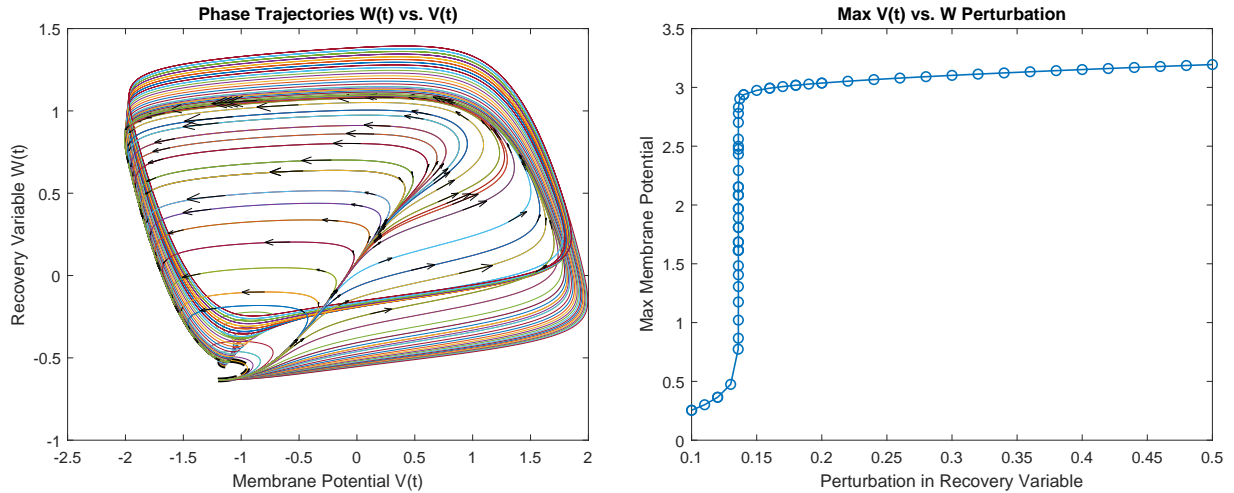


Figure 1.13: SAY SOMETHING HERE

1.3.4 Excitation block: From periodic to non-periodic

As the stimulus current continues to increase, we observe what is called *excitation block*, where the oscillations in the membrane potential induced by sufficiently high currents cease to occur *because of excessive excitation*. Figures 1.14 captures this phenomenon. A large value for I corresponds to a large initial spike. However, we can see that when the initial spike is too high, the potential curve fails to oscillate. Notice, based on the potential curves in Figure 1.14, that there exists a threshold value of I beyond which this blockage occurs.

A qualitative explanation for this can be obtained from analyzing the phase portrait shown in Figure 1.15. The subfigure on the right is the the phase portrait, superimposed with equilibrium trajectories. The equilibrium solution is the intersection of the line and the cubic curve. Now, notice that when I is small and near zero, the intersection on the lower left is a stable point. When I is increased, the intersection moves up and to unstable points until I is high enough. In this situation, the equilibrium point is brought up to the upper right, which is once again a stable point (hence the spiraling action in the phase trajectory).

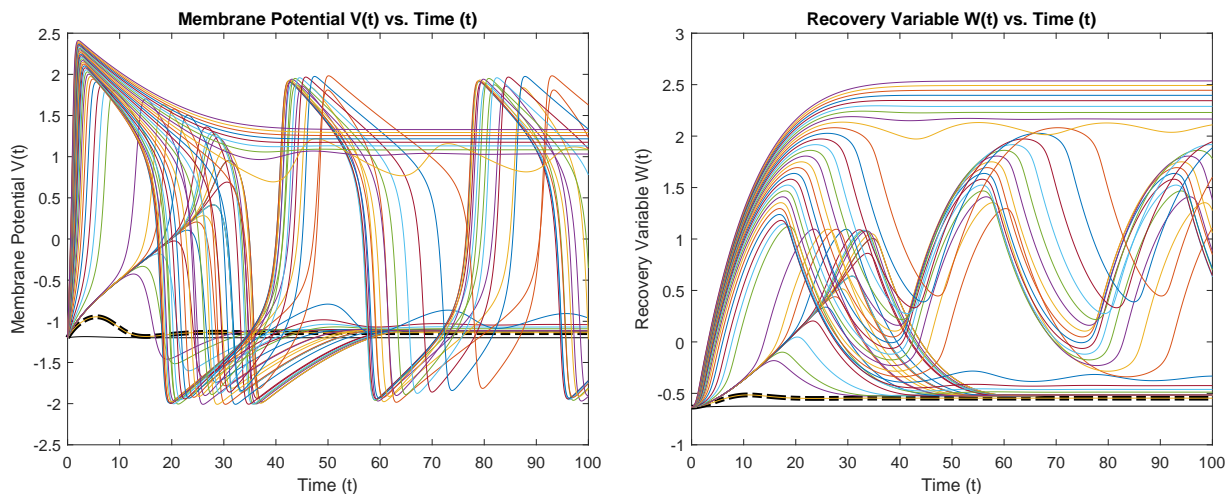


Figure 1.14: SAY SOMETHING HERE

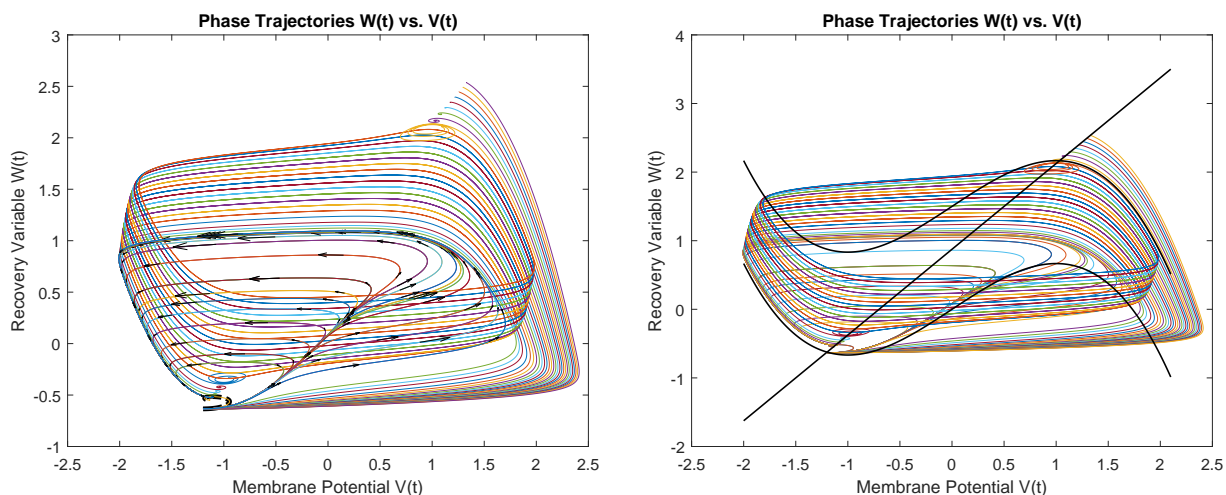


Figure 1.15: SAY SOMETHING HERE

1.3.5 Refractory period

In this section we study the refractory period. In particular, we are interested in the relationship between the time we have to wait following a spike and the ease with which another spike can be generated. There are two approaches to answering this question, as we have seen: either through perturbations in the initial value of W or through perturbing I . We will be looking at perturbations in I instead.

Figure 1.16 shows various potential curves for different values of I . Small spikes are due to small I , and vice versa. Notice that the refractory period is loosely proportional to the size of the spike: A small stimulus creates a small spike, and the neuron quickly recovers after; but when a large stimulus creates a large spike, the neuron takes significantly longer to recover.

1.4 Summary

What did we learn today?

- Excitable systems. The FHN model is an example of such a system.

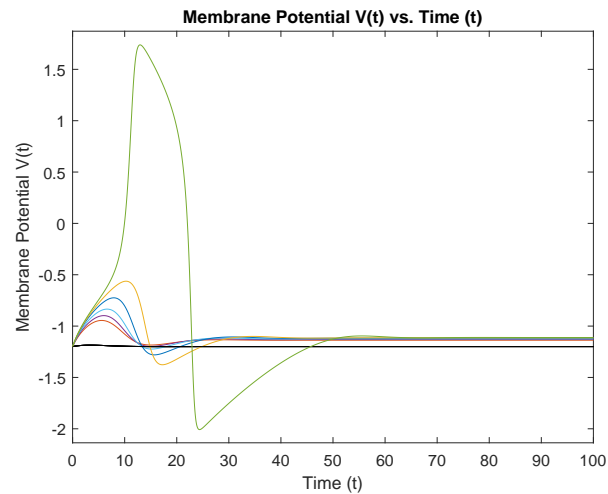


Figure 1.16: SAY SOMETHING HERE

- Nonlinear response: It is clear that there are various regimes in which the system carries distinctive behaviors.
- Numerics in MATLAB. We used `ode45` extensively to generate numerical solutions to the model. We also explored a number of plotting functionalities to generate (hopefully interesting and insightful) visualizations.

Part 2

Turing Patterns (intro)

2.1 Introduction

In 1917, D'Arcy Wentworth Thompson published *On Growth and Form*, emphasizing the significance of physical and mathematical principles in pattern formation in nature. Many scientists were inspired thereafter, including Alan Turing. Turing published a paper titled “The Chemical Basis of Morphogenesis” exploring a system of PDEs now known as a *reaction-diffusion system*. A model describing two chemical concentrations $U(x, t), V(x, t)$ has the form

$$\begin{cases} \partial_t U = f(U, V) + D_U \partial_x^2 U \\ \partial_t V = g(U, V) + D_V \partial_x^2 V \end{cases}$$

where f, g are reaction terms.

For this lab, we look at this system when

$$f(U, V) = a - (b + 1)U + U^2 V, g(U, V) = bU - U^2 V.$$

We will assume that $a = 2, b = 3$. By setting $f(U^*, V^*) = g(U^*, V^*) = 0$, we find the equilibrium solution:

$$U^* = a, \quad V^* = \frac{b}{a}.$$

The MATLAB code is provided by Professor McCoy. The method used to solve this problem is the finite difference method.

```
function [u,v,x,t] = Turing(a,b,nt)

% Let's assume the diffusion constant in the first equation equals 1
% and choose the diffusion constant in the second equation to be:
d = 5;

% Now assume our spatial domain has length L
% and subdivide it into nx steps
L = 2*pi*10;
nx = 400; dx = L/nx;
x = 0:dx:nx*dx;

% Next, we need to choose a time step and how long to run the code
dt = 5*(dx^2);
t = 0:dt:nt*dt;
```

```

% Set up results arrays, adding a phantom node on either side
u = zeros(nt+1,nx+3);
v = zeros(nt+1,nx+3);

% The initial condition will be the steady state, plus some noise
u(1,2:nx+2) = a + rand(1,nx+1)*a/100;
v(1,2:nx+2) = b/a + rand(1,nx+1)*b/a/100;

% Enforce boundary conditions
u(1,1) = u(1,2); u(1,nx+3) = u(1,nx+2);
v(1,1) = v(1,2); v(1,nx+3) = v(1,nx+2);

% Create matrix that defines step k in terms of step k+1
ru = dt/(dx^2);
Lu = toeplitz([1+2*ru, -ru, zeros(1,nx-1)]);
Lu(1,1) = 1+ru; Lu(nx+1,nx+1) = 1+ru;
rv = d*dt/(dx^2);
Lv = toeplitz([1+2*rv, -rv, zeros(1,nx-1)]);
Lv(1,1) = 1+rv; Lv(nx+1,nx+1) = 1+rv;

% Now loop
for j = 1:nt

f = a - (b + 1)*u(j,2:nx+2) + (u(j,2:nx+2).^2).*v(j,2:nx+2);
g = b*u(j,2:nx+2) - (u(j,2:nx+2).^2).*v(j,2:nx+2);

rhs_u = (u(j,2:nx+2) + dt*f)';
rhs_v = (v(j,2:nx+2) + dt*g)';

u(j+1,2:nx+2) = linsolve(Lu,rhs_u)';
v(j+1,2:nx+2) = linsolve(Lv,rhs_v)';

u(j+1,1) = u(j+1,2);
u(j+1,nx+3) = u(j+1,nx+2);
v(j+1,1) = v(j+1,2);
v(j+1,nx+3) = v(j+1,nx+2);

end

% Drop phantom nodes
u = u(:,2:nx+2);
v = v(:,2:nx+2);

end

```

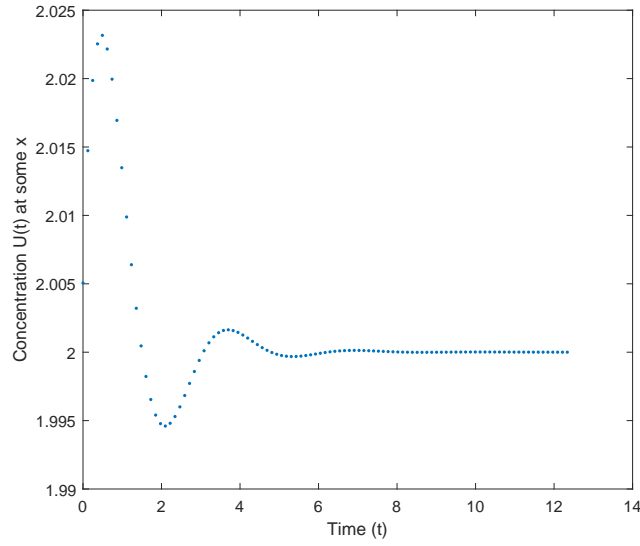
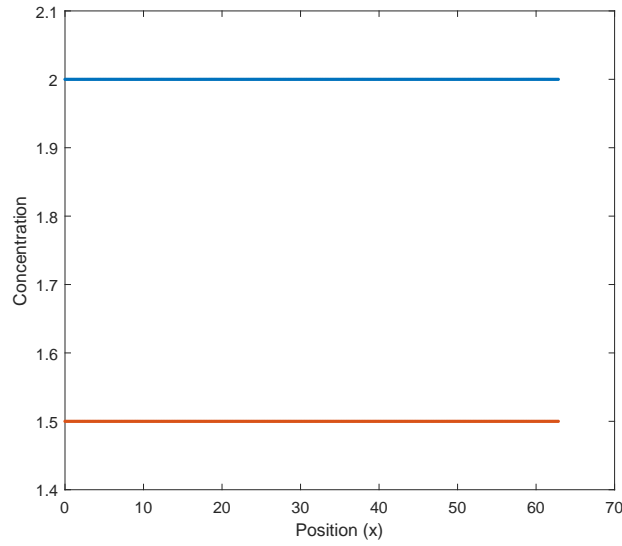
The figure belows shows how the concentration of U evolves in time at a certain position.

We can also see how the concentrations distribute in space

Let's assume now that $a = 2$ and $b = 4$ and solve the model again:

We can also see how the concentrations distribute in space

What changed? To find out, we explore the next section.

Figure 2.1: $a = 2$ and $b = 3$ Figure 2.2: $a = 2$ and $b = 3$

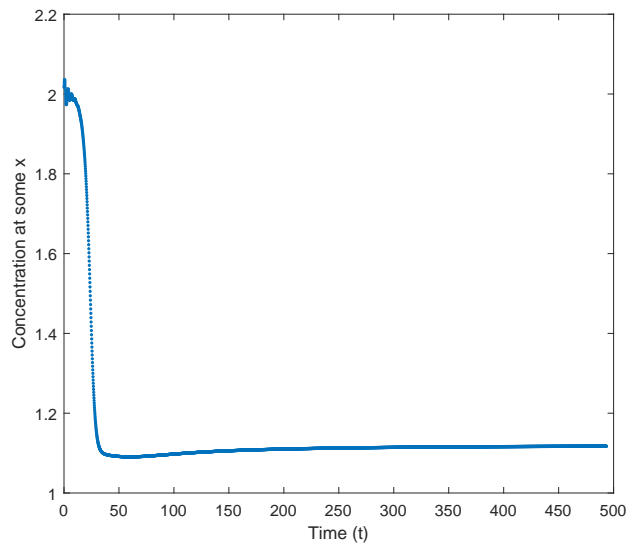
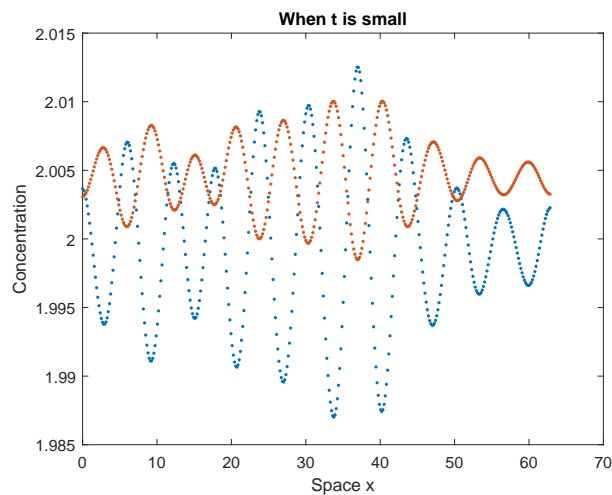
2.2 Pattern-forming instabilities

Interesting things occur near the uniform steady states. In particular, for small departures from such states, $u = U - U^*$ and $v = U - U^*$, the model takes the generic linear form

$$\partial_t \begin{bmatrix} u \\ v \end{bmatrix} = M \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} D_U & \\ & D_V \end{bmatrix} \begin{bmatrix} \partial_x^2 u \\ \partial_x^2 v \end{bmatrix}.$$

In the absence of diffusion, the fate of small departures from the steady state is determined entirely by the matrix M . Whenever M has both negative trace and positive determinant,

$$\tau = M_{11} + M_{22} < 0, \quad \Delta = M_{11}M_{22} - M_{12}M_{21} > 0.$$

Figure 2.3: $a = 2$ and $b = 4$ Figure 2.4: $a = 2$ and $b = 4$, t “small”

We describe the steady state as asymptotically stable: all nearby phase trajectories converge at the steady state. For the particular nonlinear model introduced in the previous section, one finds:

$$M = \begin{bmatrix} b - 1 & a^2 \\ -b & -a^2 \end{bmatrix}.$$

It is easy to see that the steady state is asymptotically stable if $b < 1 + a^2$.

Surprisingly, it turns out that diffusion can also destabilize things! This occurs whenever

$$(D_v M_{11} + D_U M_{22})^2 > 4D_U D_V \Delta.$$

Combining this with what we have before, we find the instabilities occur when

$$b < b_H = 1 + a^2, \quad b > b_T = (1 + \sqrt{D_U/D_V})^2.$$

To see the region where this happens we look at the following plot:

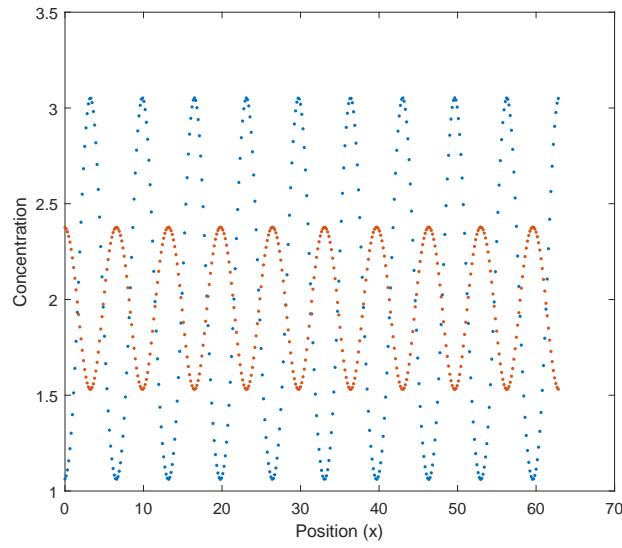
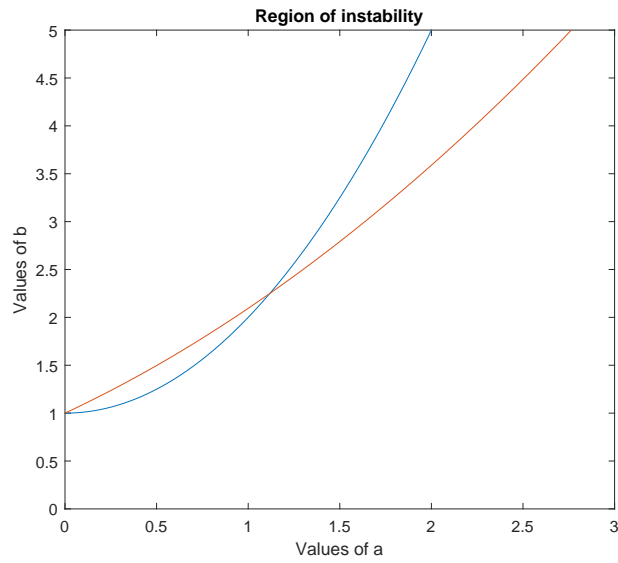
Figure 2.5: $a = 2$ and $b = 4$, t “large”

Figure 2.6: SAY SOMETHING HERE

The region of instability is defined to be the range of b that is under the blue curve and above the orange curve, for any value of a . Obviously, from some values of a , there is either no instability, or things get complicated.

We see that this explains the stark difference observed in the previous section. In the first case where $a = 2, b = 3$, we see no instability. When $a = 2, b = 4$ however, we got instability.

2.3 Some strange behavior above Blue curve and below Orange curve

Nothing interesting occurs when $b \gg 1 + a^2$. When a is small and b is not too large so that b is between the blue curve and the orange curve, we see that the concentrations oscillate in time, but are uniform in space, at all times.

At all times, the concentrations are uniform in space:

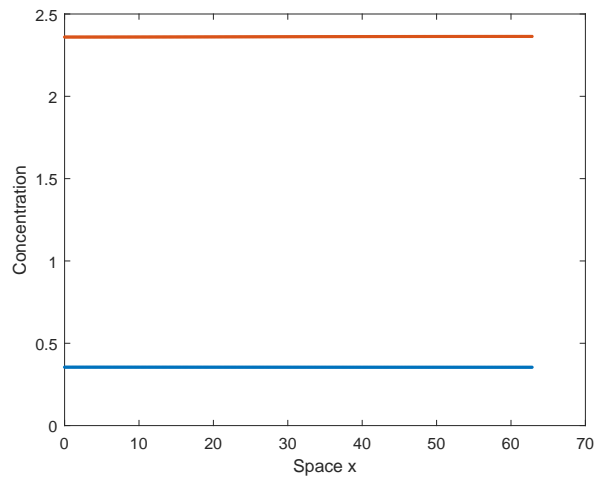


Figure 2.7: SAY SOMETHING HERE

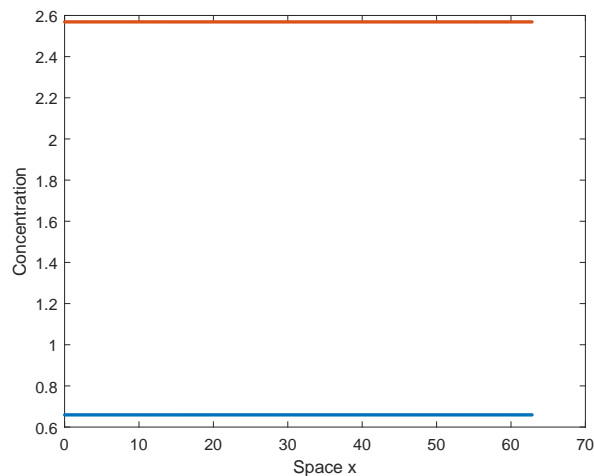


Figure 2.8: SAY SOMETHING HERE

But at a fixed point in space, the concentrations oscillate:

2.4 Wavelength from out of nowhere!

One of the more important aspects of the Turing instability mechanism is its spatial structure. Turing patterns emerge from a selection process in which spatially periodic perturbations with preferred wavelengths grow, while all other decay exponentially.

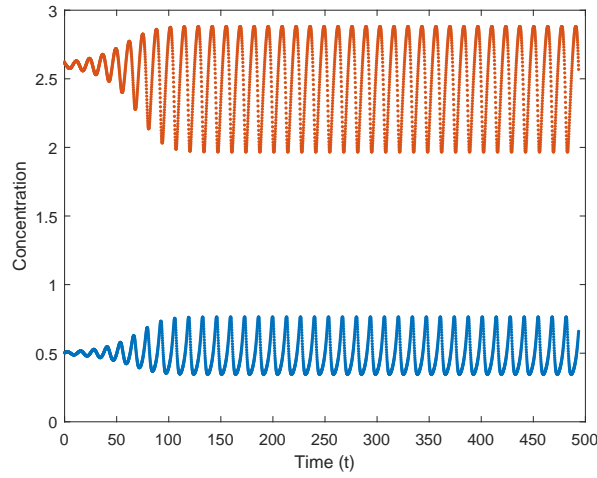


Figure 2.9: SAY SOMETHING HERE

It turns out that the fastest growing mode has a wavenumber k given by

$$k^2 = \frac{1}{2} \left(\frac{M_{11}}{D_U} + \frac{M_{22}}{D_V} \right).$$

At the critical point for instability,

$$k_c^2 = \frac{a}{\sqrt{D_U D_V}}.$$

Looking at Figure 2.5 again. We find that the oscillation has a wavelength of about $2\pi \times 10/10 = 2\pi$. This corresponds to a wavenumber of 1. Meanwhile, at the critical point for instability, $k_c = 2/\sqrt{4} = 1 = k$. So, in this case, the wave numbers match.

However, this obviously does not always happen. For example, it is possible to have small deviations from the critical wavenumber. For each set of model parameters values a, b, D_U, D_V , there will be a *band* of wavenumbers around the critical value which survive to the steady state.