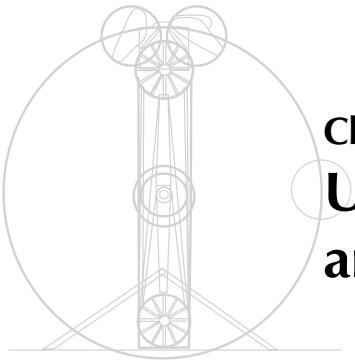

Chapter 7
Using AUTO: Bifurcation and Continuation

For many people, the main reason to use *XPPAUT* is because it provides a fairly simple interface to the continuation package, AUTO (see Doedel, [9]). AUTO now has a graphical user interface in the distributed version, but still requires that you write FORTRAN code to drive it. *XPPAUT* allows you to use most of the common features of AUTO, including following fixed points, periodic orbits, BVPs, homoclinic orbits, and two-parameter continuations. This chapter will show you how to solve a variety of problems with AUTO. Keep in mind that AUTO is a tricky program and can fail in a dramatic fashion. Nevertheless, it is a powerful tool and, even though it is 15 years old, it is still better than most other similar programs when it comes to tracking periodic solutions and solving BVPs.

Many physical and biological systems include free parameters. One of the goals of applied mathematics is to understand how the behavior of these systems varies as the parameters change. This is a numerically challenging task and there is generally no way to systematically explore a system as a function of all its parameters. One of the first things a modeler should do is reduce the number of parameters either by fixing those which are best known (e.g., gravity) or by making the problem dimensionless. Dimensional analysis is very useful and the reader can consult any number of books which describe this technique (see, for example Edelstein-Keshet [10], and Murray [31]). Assuming that you can reduce the free parameters to a manageable number, there are some useful tools for exploring how a dynamical system changes as these parameters vary. The most straightforward technique is called *continuation* in which a particular solution (such as a fixed point or limit cycle) is followed as the parameter changes. AUTO (which is described in this chapter) provides some very powerful algorithms for the continuation of fixed points and periodic solutions to differential equations. The stability of the particular *branch* of solutions is readily obtained by analyzing the linearization. This, too, is automatically accomplished by AUTO. If a fixed point or limit cycle exhibits a change in its stability, this is often a sign that new qualitatively different behavior could occur. For example, new fixed points might emerge from the branch of solutions, or a limit cycle may emerge. These qualitative changes in the local and global behavior are called *bifurcations* and their detection during continuation is the subject of much mathematical research. AUTO provides a number of tools for the automatic detection of bifurcations of fixed points and limit cycles.

A thorough description of local and global bifurcations can be found in the excellent book by Kuznetsov [25].

Consider a differential equation of the form

$$\frac{dx}{dt} = F(x, \mu), \qquad x \in R^n, \tag{7.1}$$

where μ is a parameter. Suppose that (7.1) has a fixed point, (x_0, μ_0). Let $A(\mu_0) = A_0$ be the linearized matrix for F with respect to x about the fixed point (x_0, μ_0). There are two important insights gained by looking at A_0. First, if A_0 is invertible, then we can follow the fixed point x_0 as a function of the parameter μ in some neighborhood of μ_0, and this is the *only* fixed point near x_0. This fact is a consequence of the implicit function theorem. The second point is that if none of the eigenvalues of A_0 lie on the imaginary axis, then the behavior of (7.1) near the fixed point x_0 is the same as the behavior near the origin of the linear differential equation

$$\frac{dy}{dt} = A_0 y. \tag{7.2}$$

In particular, the stability of the two systems is the same. This is why we can determine stability of fixed points by linearizing. However, if one or more eigenvalues of A_0 has a zero real part, then the local behavior of the nonlinear system is not necessarily the same as the linear system. Convince yourself of this fact by looking at the following three systems which all have the same linearization:

$$x' = -y - y^3, \qquad y' = x,$$

$$x' = -y - x^3, \qquad y' = x,$$

$$x' = -y + x^3, \qquad y' = x.$$

Thus, if we vary μ, and if one or more eigenvalues of $A(\mu)$ crosses the imaginary axis, we expect that there will be a *qualitative* change in the behavior of (7.1). The analysis of this change in behavior near the fixed point is the goal of *local bifurcation theory*.

Similar points can be made for discrete dynamical systems

$$x_{n+1} = F(x_n, \mu), \tag{7.3}$$

but the critical set for the eigenvalues of $A(\mu)$ is the unit circle rather than the imaginary axis. Consider a limit cycle solution to (7.1) and take a local Poincaré section. Then, this defines a map and the limit cycle is a fixed point of the map. Thus, the local bifurcations of limit cycles are found by studying the local bifurcations of the associated Poincaré map.

One of the beauties of local bifurcation theory is that the behavior of the systems (7.1) and (7.3) is the same as certain low-dimensional polynomial systems called *normal forms*. We briefly mention the main bifurcations of interest:

- *Bifurcations at a zero eigenvalue for differential equations.* When A_0 has a zero eigenvalue, then new fixed points often arise. These can be transcritical, pitchfork, or saddle-node bifurcations. Only the latter is generic; the other two occur when there

is symmetry. The transcritical and pitchfork are *branch point* bifurcations and AUTO will automatically detect and follow them. The saddle-node, fold, and limit-point bifurcations occur when a branch of solutions bends around.

- *The Hopf bifurcation*. This occurs when A_0 has a pair of imaginary eigenvalues. Generically, a small amplitude limit cycle will emerge from the branch of fixed points.
- *Period-doubling bifurcation*. This occurs for maps when an eigenvalue of A_0 is -1. For a limit cycle, the result is that the period of the limit cycle doubles.
- *Torus bifurcation*. When the eigenvalues of A_0 for a map cross the unit circle at values other than ± 1, $\pm i$, or the cube roots of 1, then a torus bifurcation arises. Chaos and other complex behavior often occur.

AUTO detects all these bifurcations and in some cases can create a two-parameter curve along which the bifurcation occurs.

In the next few sections, we show how to use *XPPAUT* and AUTO together on a variety of differential equations, maps, and BVPs.

7.1 Standard examples

As a first example, we consider the normal form for a cusp bifurcation:

$$x' = a + bx - x^3.$$

Let's pick $a = b = 1$ and let *XPPAUT* find the root by integrating the equation. Make an ODE file for this problem and run it. Integrate it and then integrate it again using the `Initialconds Last` (**I L**) command to pick up where you left off. You should have settled onto a fixed point of about 1.3247. Click on `File Auto` to bring up the **AUTO window**. (See Figure 7.1.) In this window, there are additional menu items and several windows. The lower left shows a circle. The number of crosses inside (outside) the circle corresponds to the number of stable (unstable) eigenvalues for a solution. The two windows at the bottom are, respectively, information about the computed points and a hints or tips window. These bottom windows also tell you the coordinates of the main graphics window. The main window is where the relevant bifurcation diagrams are drawn.

Before you can use AUTO, you must prepare your system for it. You must start your bifurcation analysis from either a fixed point of your model, a periodic orbit, or a solution to a BVP. We have already done this for the present example.

Once you have brought up the **AUTO Window**, you should do the following steps:

- Use the `Parameter` item to tell AUTO the list of five or fewer parameters that you will vary for the session. Choose the names of any of your parameters; the default names are the first five that you have defined in the ODE file.
- Use the `Axes` command to tell *XPPAUT* what parameters are varied and what is to be plotted, as well as the range of the graphs.

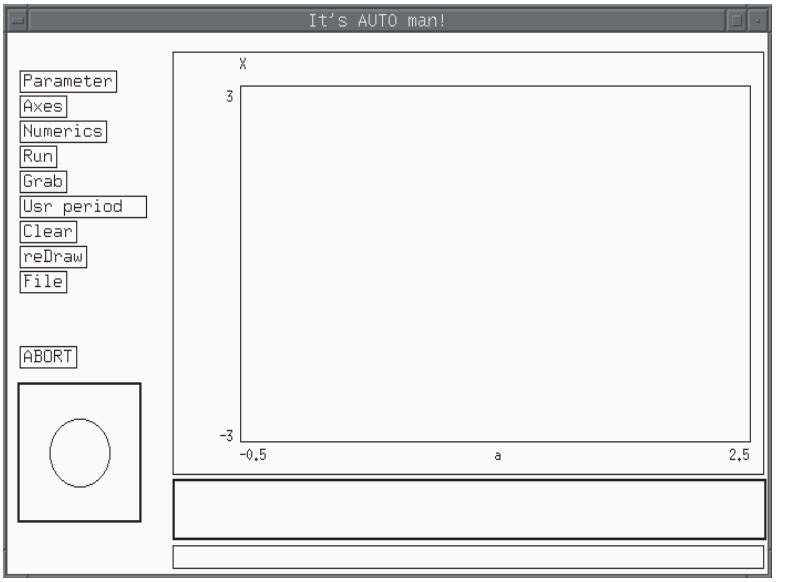

Figure 7.1. *The AUTO window.*

- Use the `Numerics` command to define all the AUTO numerical parameters such as the direction, output options, step size, etc.

- Use the `Run` command to run the bifurcation.

Once you have run a bifurcation, use the `Grab` command and the arrow keys to move around the diagram, and use the `File` command to save things or reset AUTO. The commands `Clear` and `reDraw` just erase and redraw the screen, respectively. The `Usr period` item allows you to tell AUTO to save certain points such as oscillations of a certain period or points when a parameter takes a particular value.

AUTO is essentially independent of *XPPAUT* but there is some communication between them. For example, if you grab a point in the **AUTO Window**, then the state variables are loaded as initial data into *XPPAUT* and the parameters are changed as well. Similarly, when you start fresh in AUTO, the initial fixed point or orbit is computed in *XPPAUT*. Finally, you can import a bifurcation diagram from AUTO into *XPPAUT* and plot this. We will use this feature later to show bursting in a biological model.

With these preliminaries, let's compute the bifurcation diagram for our cusp model. The main parameter is a and the secondary parameter is b. Since there are only two parameters, there is no need to invoke the `Parameter` item in the **AUTO Window**. We eventually want to compute a two-parameter bifurcation diagram; to do this, we must first compute a one-parameter diagram. From here on, all commands that I mention concern the items in the **AUTO Window**. Click on `Axes` and choose `HiLo`; this option plots both the maximum and minimum of orbits which, for the present problem, is irrelevant since there are no periodic orbits. In the resulting dialog box, tell AUTO the variable to plot, the main

parameter (for one-parameter diagrams) and also the second parameter (for two-parameter diagrams) as well as the dimensions of the plot. Fill it in as follows:

Y-axis: X
Main Parm:: a
2nd Parm:: b
Xmin:: -2.5
Ymin:: -3
Xmax:: 2.5
Ymax:: 3

and then click Ok. We have told AUTO that the main bifurcation parameter is a and that X is the variable to plot. The graph will show a on the horizontal axis and x on the vertical axis with a range of $[-2.5, 2.5] \times [-3, 3]$. Next click on the Numerics item. Change only one item in this dialog box, Par Min, the minimum value of the parameter, which you should change from 0 to -2. Par Min and Par Max tell AUTO how far to continue solutions. Then click OK. Now you are ready to run. Click on Run Steady state. (This tells AUTO you are continuing along a steady state or fixed point of the system.) A thick black line will appear across your screen toward the right. If nothing happens, you probably forgot to set a good initial guess for the starting point. You should make sure that you are on a fixed point in *XPPAUT*; for $a = 1$, $b = 1$ the fixed point is about 1.3247. Assuming that you have a curve, you can see that it is pretty boring and that the curve was produced for a increasing.

Click on Grab to move around in the bifurcation diagram. You should see a cross on the screen. (If you don't see a cross on the screen, then, next time you run *XPPAUT*, run it with the -xorfix option which should fix the lack of a cross. By default it runs without requiring this option on Linux and Windows.) Use the left and right arrow keys to move the cross around. The stability circle shows the eigenvalue status. The fact that the drawn curve is solid implies the fixed point is stable; unstable fixed points are drawn with thin lines. AUTO specially marks certain points with small crosses and numbers. You can jump to these by pressing Tab. As you move around the diagram, the text area beneath the diagram gives you a summary of information about the current location such as the value of the parameter, the state variable, the period, and the point type for special points. There are many point types in AUTO. In this example, you will see only EP which means endpoint. The one you never want to see is MX since this means AUTO has had trouble and maxed out on some sort of numerical iteration. Sometimes this can be rectified by changing the numerical parameters. You can exit from Grab mode by clicking Esc which does **not** grab the point, or by clicking Enter which loads the point into memory. *AUTO can generally be restarted only from special points.*

So, lets make this diagram extend to the left. Grab the first point in the diagram (Grab Enter). Click on Numerics and change Ds from 0.02 to -0.02 and click on OK. This tells AUTO you want to change the direction since the sign of the numerical parameter Ds determines the starting direction for AUTO. Now, click on Run. You should now see a sideways cubic curve. In particular, there are two points labeled 3 and 4 occurring at $a = \pm 3.84$. If you click on Grab, you will discover that the two special points are labeled

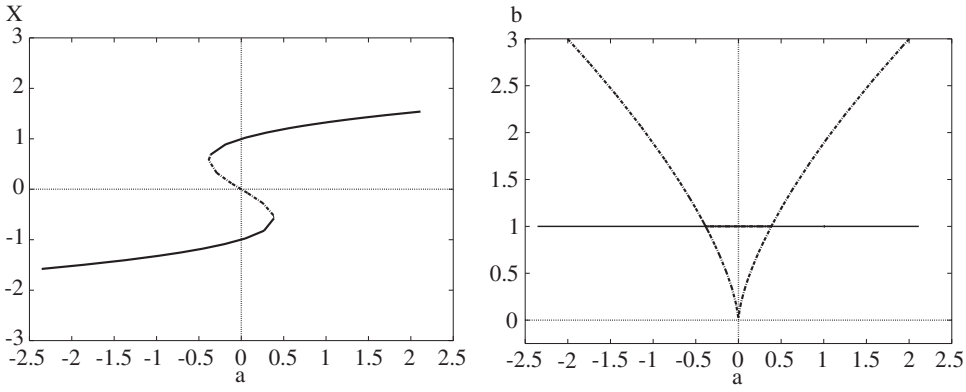

Figure 7.2. *One- and two-parameter bifurcation diagrams for the cusp ODE, $x' = a + bx - x^3$. The one-parameter diagram has $b = 1$. The two-parameter diagram also shows the artifactual line at $b = 1$ from the one-parameter continuation.*

`LP`, which means that they are limit points, also known as fold points. Note that as you move past them, the eigenvalue goes through the unit circle. The curve of fixed points along the middle branch is unstable.

Grab the leftmost limit point 3. We will now track this fold point as a function of the second parameter, b, producing a two-parameter bifurcation curve. Click on `Axis` and select `Two param`. Change `Ymin` from -3 to -0.5. Since the plot type is `Two Par` the vertical axis contains the secondary parameter (`b`) and the horizontal axis contains the main parameter (`a`). Click on `OK`. Click on `Run`. You should see a thin curve that rises and moves to the left. This is a curve of fold points; along this curve in (a, b) space, the original system has a fold.

Once again, invoke the `Grab` item and click on `Tab` until you have selected the limit point 3 again. (Look at the bottom of the **AUTO Window** to see which points you move through.) We now want to change the direction of bifurcation to get the rest of the curve. Click on `Numerics` and change `Ds` from -0.02 to 0.02, thus forcing AUTO to make the main parameter increase. Click on `Run` and you will get the rest of the curve; it is a nice cusp. For each (a, b) inside the cusp, there are three fixed points and, for (a, b) outside the cusp, there is only one fixed point.

Click on `Axes Last 1 Par` and then on `reDraw`. You will get the one-parameter diagram again. Click on `Axes Last 2 par reDraw` and you will see the cusp. This is a shortcut for selecting the most recently defined one- and two-parameter views. Click on `File Postscript` and you can make a PostScript version of the diagram. Figure 7.2 shows the one-parameter and two-parameter diagrams.

7.1.1 A limit cycle

We will start on a limit cycle in this example. The limit cycle is essentially what is called an *isola* since it is not connected to a main branch of fixed points. Here are the equations:

$$x' = xf(R) - y, \qquad y' = yf(R) + x,$$

where $f(R) = 0.25 - a^2 - (R-1)^2$ and $R = x^2 + y^2$. There is one parameter in this problem, a, and for $-.5 < a < .5$ there is a pair of limit cycles, one stable and one unstable. These can be seen by rewriting the system in polar coordinates, in which case it becomes

$$r' = rf(r^2), \qquad \theta' = 1.$$

Thus, when $f(A) = 0$, there is a limit cycle with magnitude $\sqrt{A}$. Starting with the parameter value $a = 0$, we see that $A = 1.5$ so that,

$$x(t) = \sqrt{1.5}\cos t, \qquad y(t) = \sqrt{1.5}\sin t$$

is a solution. (I will leave the verification of this, as well as the stability of the cycle as an exercise for the reader. The reader should also show that for $a = 0$, another solution is $A = 0.5$, leading to an unstable limit cycle. Finally, the reader should prove that $(x, y) = 0$ is an asymptotically stable fixed point for all a.)

Here is how to start AUTO from a limit cycle. Note first that this does not always work; AUTO has much more trouble starting from a periodic solution than from a fixed point:

1. Integrate the equations until a nice limit cycle is formed.
2. Determine the period either by looking at the data browser or by using the mouse to measure peak-to-peak distance in the oscillation. (**Hints:** In the **Data Viewer**, click on the button `Find`, choose a variable, and then pick a really big number. *XPPAUT* will then find the value of the variable closest to this big number, in other words, the maximum. Then click on `Get` which loads this point as an initial condition. Integrate again and look for the time of the next maximum. This is the period. Alternatively, hold the mouse in the window with the left button held down and move it around—this allows you to read off the coordinates at the bottom of the window.)
3. Change the total amount of integration time to the approximate period (`nUmerics Total`) and integrate the equations one more time.
4. Fire up AUTO and set up the bounds for the parameter as well as the graphical viewpoint.
5. In the **AUTO Window**, click on `Run Periodic`.
6. Keep your fingers crossed.

Let's apply these techniques to the isola problem. Here is the ODE file:

```
# an isola of limit cycles
# isola.ode
f(r) = .25-(r-1)^2-a^2
r=x^2+y^2
x'=f(r)*x-y
y'=f(r)*y+x
par a=0
init x=1.224,y=0
@ ylo=-1.5,yhi=1.5
done
```

Run *XPPAUT* and integrate the equations. Use the mouse to see that the period is roughly 6.3 (it is actually 2π). Change the integration time to 6.3 and run again by clicking `Initialconds Last` starting at the end of the last integration—this ensures that the transients are integrated out. Start up AUTO by clicking `File Auto` (**F A**). Since there is only one parameter, you don't have to worry about setting the active parameters. From here on, all commands will be in the **AUTO Window**. Click on `Axes Hilo` and fill in the dialog box as follows:

`Y-axis: X`
`Main Parm:: a`
`2nd Parm:: a`
`Xmin:: -.75`
`Ymin:: -3`
`Xmax:: .75`
`Ymax:: 3`

and click `Ok`. We are plotting `X` versus the parameter `a` in the window $[-.75, .75] \times [-3, 3]$. Click on `Numerics` and set `Par min:-0.75` and `Par max:0.75`, then click on `OK`. Now we are ready to run. But have your mouse poised over the `ABORT` button since AUTO will run in circles until you stop it. (This is because the periodic solution is an isola—an isolated circle of solutions—and AUTO doesn't stop on such branches. When you are ready to run, click on `Run Periodic`. Click on `ABORT` if it looks like AUTO is running in circles. (Another way to prevent this is to set the total number of computed points, `Npts`, to a low value from the `Numerics` menu. However, there is a danger that you won't compute all the points on the diagram.) A pair of ellipses should appear and you should see something like Figure 7.3. The filled dots represent the maximum and minimum values of the stable periodic solution, and the unfilled dots are the max/min values of the unstable periodic branch. A better way to look at this is to view the norm. Click on `Axes Norm` and change `Ymin` to `0`. Then click on `OK` and `reDraw`. Use the `Grab` command and the arrow keys to move around the loop. Note that at $a = \pm 0.5$, there is a limit point of oscillations.

XPPAUT lets you pick up other branches of solutions and put them on the same diagram. Click on `File Clear grab`. This clears any grabbed points and thus allows you to start from a completely different point. From the `Axes Norm` menu, change `Ymin=-.5`. Now, go back into the **Main Window** of *XPPAUT*. Change a to $-.5$ and change the initial conditions to `X=0` and `Y=0`. This is a fixed point. Go back to the **AUTO Window** and click on `Run Steady state`. When prompted to destroy the diagram, click on `No`. This way, you don't delete the part of the diagram that was already computed. You should see a thick horizontal line appear, indicating the existence of a stable fixed point. You should see something like the right panel of Figure 7.3. If you want a PostScript printout of this, click on `File Postscript` and save it with a name of your choice. Click on `File Reset diagram` to get rid of the diagram and all the AUTO junk left on your disk. (Although this is not necessary because these files are deleted when you exit *XPPAUT*, it is always a good idea.) Resetting a diagram is also useful if you want to make changes in the parameters and start fresh.

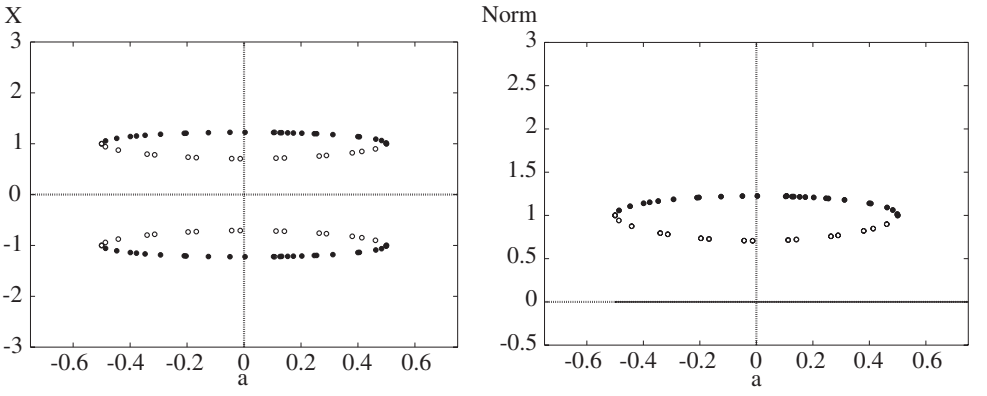

Figure 7.3. *The diagram for the isola example. The left figure shows the maximum and minimum of the stable (filled circles) and unstable (empty circles) limit cycles. The right figure shows the norm of the limit cycles and the stable fixed point at 0.*

7.1.2 A "real" example

The examples presented above are pretty lame—we knew the answers already. Now, lets analyze a real-world example that we have already studied, the Morris–Lecar equations:

$$C\frac{dV}{dt} = I + g_l(E_l - V) + g_k w(E_K - V) + g_{Ca} m_\infty(V)(E_{Ca} - V),$$

$$\frac{dw}{dt} = (w_\infty(V) - w)\lambda_w(V).$$

We considered a dimensionless version of these and will continue to do so here. Here is the ODE file for a single neuron, `ml.ode`:

```
# ml.ode
# morris-lecar; dimensionless
v'=I+gl*(el-v)+gk*w*(ek-v)+gca*minf(v)*(eca-v)
w'=(winf(v)-w)*lamw(v)
par I=0,phi=.333
par ek=-.7,eca=1,el=-.5
par gl=.5,gk=2,gca=1
par v1=-.01,v2=0.15,v3=0.1,v4=0.145
minf(v)=.5*(1+tanh((v-v1)/v2))
winf(v)=.5*(1+tanh((v-v3)/v4))
lamw(v)= phi*cosh((v-v3)/(2*v4))
aux ica=gca*minf(v)*(v-eca)
aux ik=gk*w*(v-ek)
@ total=50,xlo=-.6,xhi=.5,ylo=-.25,yhi=.75
@ xplot=v,yplot=w
done
```

Note that the first two parameters, the current `I`, and the overall rate of the potassium current `phi` are the ones we want to vary. (I have added a pair of auxiliary variables which track the active currents.) Start up *XPPAUT* and integrate this equation using `InitialConds Go` followed by `InitialConds Last` to get rid of all the transients. Click on `File Auto` to bring up AUTO. Since the parameters that we want to vary were the first defined, they will be among the list of usable parameters. Click on `Parameter` in the **AUTO Window** just to check and see that `I` and `phi` are the first two entries. Click on `Ok` to close the dialog box. From here on in, the rest of the commands will be in the **AUTO Window**. Click on `Axes Hilo` to set up the graphics axes. Fill in the dialog box as follows and then close it:

`Y-axis: V`
`Main Parm:: I`
`2nd Parm:: phi`
`Xmin:: -.1`
`Ymin:: -.5`
`Xmax:: .5`
`Ymax:: .5`

Next open the `Numerics` dialog box, change `Par Min` to `-.5` and `Dsmax` to `0.05`; close it. Now you should be OK to run. Click on `Run Steady state`. You should see a cubic like curve with a few special points on it (see Figure 7.4). Click on `Grab` and move through the diagram. You will see limit points (LP) at $I = 0.08326$ and $I = -0.02072$ as well as a Hopf bifurcation point (HB) at $I = 0.20415$. One expects to pick up a branch of periodic solutions at a Hopf bifurcation. When you get to this point, click `Enter` to grab it. Now, you can try to compute the periodic orbit emanating from this point. Click on `Run` and you will get a list of choices. Choose `Periodic` and when you see a sort of smudge at the end of the branch of limit cycles, click on `ABORT` to stop AUTO. You should see something like Figure 7.4. The "smudge" is due to AUTO's difficulty at computing the long period oscillation which terminates on the lower fold point. This shows a number of interesting features that are quite important in bifurcation theory. First, note that the branch of periodic orbits that comes out of the Hopf point is unstable (open circles) and then it turns around at a limit point of oscillations at $I = 0.242$ so that a stable and unstable limit cycle coalesce here. Thus, for a small range of currents, there is bistability between the upper fixed point and the stable limit cycle. The stable limit cycle (solid circles) terminates on the fold, giving rise to a SNIC (saddle-node infinite period cycle). In fact, the frequency of the limit cycle goes to zero as the saddle-node is approached. To see this, click on `Axes Frequency`, fill in the dialog box as follows, and close it:

`Y-axis: V`
`Main Parm:: I`
`2nd Parm:: phi`
`Xmin:: .05`
`Ymin:: 0`
`Xmax:: .3`
`Ymax:: .2`

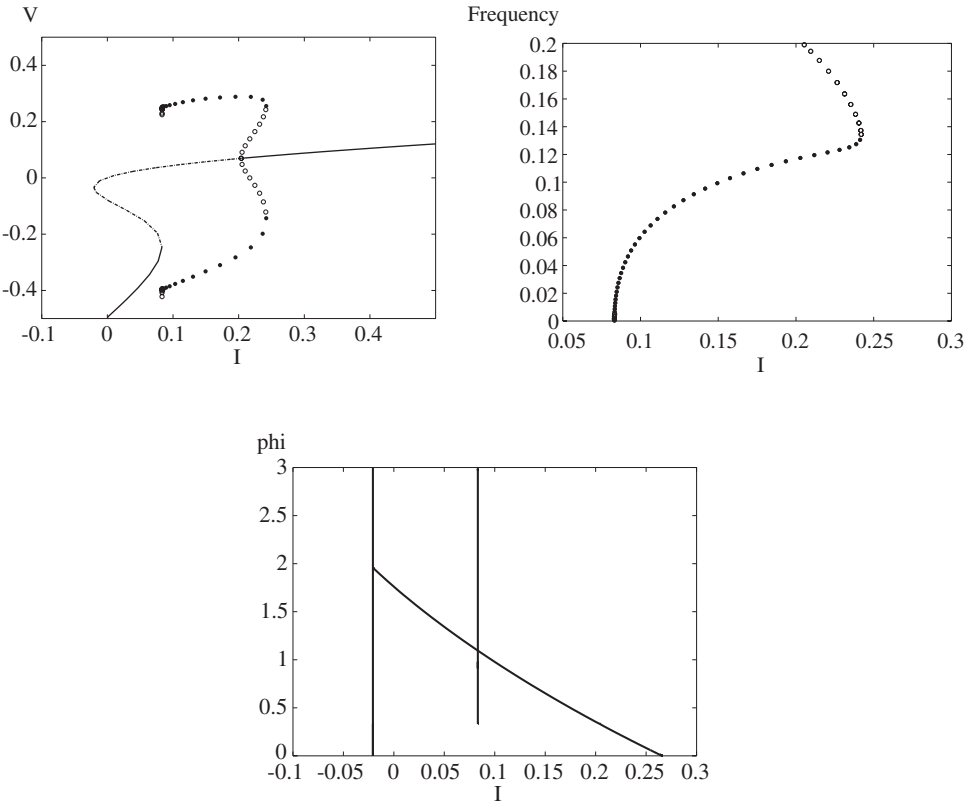

Figure 7.4. *Various diagrams associated with the Morris–Lecar equation. Top left: One-parameter diagram showing fixed points and limit cycles. Top right: Frequency of the periodic orbits. Bottom: Two-parameter diagram showing the curve of Hopf points and (straight line) the lines of saddle-node points. These are independent of the parameter* `phi`.

The lower branch looks just like a square root which is what the theory of this type of bifurcation predicts. Figure 7.4 shows the frequency of both the stable and unstable limit cycles.

We will now look at the two-parameter diagram for this model. Click on `Axes Two par`, fill in the dialog box as follows, and close it:

`Y-axis: V`
`Main Parm:: I`
`2nd Parm:: phi`
`Xmin:: -.1`
`Ymin:: 0`
`Xmax:: .3`
`Ymax:: 3`

This creates a graph with I along the horizontal axis and phi along the vertical axis. Now click on Grab and press the Tab key until you come to the first limit point (LP) which should be labeled 2. (Look at the bottom of the window as you move through the diagram until the desired point is found. Press Enter to grab it.) Click on Run and a vertical line will appear. The line is vertical because the limit point is independent of the parameter phi *which has no effect on the value of the fixed point* but can affect the stability. Again, click on Grab and Tab to the second limit point (LP, labeled 3). Click on Run again and you will see another vertical line corresponding to the leftmost limit point. Within the two vertical lines, there are three fixed points and outside there is one. Next, click on Grab and Tab to the Hopf bifurcation point (HB, labeled 4). Click on Run Two param and you will see a diagonal line going down and to the right. This is the curve of Hopf bifurcation points. Finally, we want to extend the curve of Hopf points to the left. To do this, we must tell AUTO to reverse direction. Click on Numerics and change Ds=0.02 to Ds=-0.02 and close the dialog box. Recall that AUTO uses the sign of Ds to determine in which direction to go. Finally, click on Grab and get that Hopf bifurcation point again. Click on Run Two param and you will see the curve go up and to the left. It crosses the right line of fold points and terminates on the left line. The intersection with the right line of fold points is irrelevant since the Hopf occurs on the upper branch of solutions and the right fold line represents the loss of the lower branch of fixed points. The termination of the curve of Hopf points on the left line of folds signifies a new higher-order bifurcation point, the Takens–Bogdanov bifurcation. There is a double zero eigenvalue; the Hopf bifurcation and the saddle-node have coalesced. This bifurcation often signals the presence of homoclinic orbits. For phi below this curve and between the two vertical lines, the upper branch of fixed points is unstable. Outside the vertical lines, there is only one fixed point. On the right, it is stable above the Hopf curve and unstable below. On the left, the fixed point is always stable. For example, choosing I=.15 phi=.5 is in the regime where there is one fixed point and it is unstable. Since you can readily prove that all solutions stay bounded for this model, this implies that there must be at least one stable periodic solution. Look at the phase-plane in this case and verify that you are right.

Here are some hints for computing the complete diagram for a system:

- **ALWAYS** make sure you are starting at a fixed point or clearly defined limit cycle. For fixed points, click on Initialconds Go and then Initialconds Last a few time to get rid of transients. For limit cycles, get a good estimate of the period and integrate **one** full period, no more, no less.
- To navigate the diagram quickly, click on Tab, as this just jumps to a special point; if you get lost in the diagram, use the Axes Fit reDraw sequence to put the entire diagram on the page. Or if there is a really complex part, use the Axes Zoom feature to magnify the given area.
- Follow all branch points; AUTO will generally go back to all branches of fixed points and try to follow these automatically. However, for limit cycles which have branch points such as a pitchfork bifurcation, AUTO does not automatically compute them. Use Grab to move to such points, labeled BP, and then click on Run. Change the sign of Ds in the numerics dialog box to branch off in a different direction.
- Grab all Hopf points and try to find the periodic solutions that arises from them.
- For two-parameter bifurcations, also change the sign of Ds to go in both directions.

- If AUTO fails from the beginning—as indicated by the `MX` label—then you haven't given it a proper starting value. Make sure you are on a good periodic orbit, or at a true fixed point, or have a true solution to the BVP.
- If AUTO fails after partially completing the diagram and you would like to change numerical parameters and have another go at it, destroy the diagram first before continuing: `Grab` the initial starting point and `File Reset diagram` to destroy the diagram.
- If AUTO seems to not be able to continue, change `dsmin` to make it smaller; for periodic orbits and BVPs, change `ntst` and make it larger.
- If AUTO seems to miss a bifurcation point that is clearly there, clear the diagram and make `dsmax` smaller so that AUTO doesn't miss it. Reset the diagram and recompute.

7.1.3 Exercises

1. Set `phi=1.2` and compute the one-parameter bifurcation for the `ml.ode` diagram using I as the parameter. Make sure that `Par Min` is `-0.5`. Note how the periodic orbit intersects the middle branch of fixed points at a homoclinic orbit. Set I to be near this intersection and look at the phase-plane of the Morris–Lecar model. See if you can understand the global dynamics for this value of I. Next observe that for $I = 0.072$ the system appears to be tristable. There are two stable fixed points and one stable periodic orbit. Draw the complete phase-portrait with $I = 0.072$. Include unstable periodics which you can compute by integrating backwards.

2. Compute the bifurcation diagram, using the current as a parameter for the dimensionless Morris–Lecar equations and using the parameter values in the above example, but setting `v3=.0166`, `v4=.25`, `phi=.333` and looking in the window $-0.5 < v < 0.5$ and $0 < I < 0.5$. Make sure that you start at a fixed point when $I = 0$. Compute the branch(es) of periodic solutions emanating from the Hopf bifurcation. Plot the frequency of the periodic orbits as a function of the current, I. For a bonus problem, compute the two-parameter curve of Hopf bifurcation points using `phi` as the second parameter—window the two-parameter graph with `0 < phi < 2`. Thus show that if $\phi < \phi^*$, then there are two Hopf bifurcations. Find ϕ^*.

3. Compute the one-parameter bifurcation diagram for the Brusselator

$$u' = a - (b+1)u - vu^2, \qquad v' = bu - vu^2$$

with $a = 1$, $b = 1.5$, $u = 1$, $v = 1.5$, and treating b as the bifurcation parameter. In `Numerics` set `Par max` to be 4. Window the diagram with $0 < u < 6$ and $0 < b < 4$.

4. This one is for bifurcation jockeys only. Explore the bifurcation diagram of the coupled Brusselators

$$u_1' = f(u_1, v_1), \tag{7.4}$$

$$v_1' = g(u_1, v_1) + d(v_2 - v_1), \tag{7.5}$$

$$u_2' = f(u_2, v_2), \tag{7.6}$$

$$v_2' = g(u_2, v_2) + d(v_1 - v_2), \tag{7.7}$$

$$f(u, v) = a - (b + 1)u + vu^2, \tag{7.8}$$

$$g(u, v) = bu - vu^2 \tag{7.9}$$

with $d = 0.2$, $a = 1$, and letting b be the bifurcation parameter. In order to complete this one, you should follow all special points, in particular, branch points of periodic orbits, labeled `BP`. For these, you may have to grab them twice and change the direction of the continuation (`Ds`) in the `Numerics` menu. A number of other parameters have to be changed in the `Numerics` menu for AUTO, namely, the number of collocation points used to track periodic orbits, `Ntst=30`, the minimum step size `Dsmin=1e-5`, and the maximum step size `Dsmax=0.1`. The diagram should be drawn with $0 < b < 5$ and $0 < u_1 < 7$. As with many internal parameters, most of the AUTO parameters can be set with options in the ODE file. Here is my version of the above equations:

```
# bruss2.ode
# two Brusselator equations
f(u,v)=a-(b+1)*u+v*u^2
g(u,v)=b*u-v*u^2
u1'=f(u1,v1)+du*(u2-u1)
v1'=g(u1,v1)+dv*(v2-v1)
u2'=f(u2,v2)+du*(u1-u2)
v2'=g(u2,v2)+dv*(v1-v2)
par b=1.5,a=1,du=0,dv=.2
init u1=1,u2=1,v1=1.5,v2=1.5
@ ntst=30,dsmin=1e-5,dsmax=0.1,parmin=0,parmax=5,autoxmin=0
@ autoxmax=5,autoymin=0,autoymax=7
done
```

Note that, a few years ago, this computation and some discussion of the resultant system would have been considered a decent research problem. Now it's merely an exercise in a book.

7.2 Maps, BVPs, and forced systems

XPPAUT allows you to study continuation of discrete dynamical systems, BVPs, and, with some preparation, periodically forced systems. The use of AUTO with maps is rather restricted since all one can do is follow branch points and two-parameter continuations of limit points and Neimark–Sacker points. (Recall that the Neimark–Sacker bifurcation is the discrete analogue of the Hopf bifurcation.)

7.2.1 Maps

XPPAUT makes some aspects of tracking the behavior of maps pretty easy, mainly because it allows you to study the continuation of $f^k(x)$, the kth iterate of the map, rather than just the map. For, example, consider the logistic map

$$x_{n+1} = ax_n(1 - x_n) \equiv f(x, a)$$

with parameter a. If we were to run this with AUTO, it would say that at $a = 3$ there was a Hopf bifurcation with period 2. Since this is really a period-doubling bifurcation, we could try to follow it. However, AUTO cannot continue from "Hopf" points of maps. Thus, how could we follow this apparent period 2 point? Since this period 2 point arises from the primary branch of solutions, if we look at $f^2(x, a) = f(f(x, a), a)$, then the period 2 orbit is just a branch of the second iterate and AUTO will follow it. Thus, if we wanted to find all the periodic orbits dividing 8, we could just look at the 8th iterate of the map with a as the parameter. Here is our version of the logistic map:

```
# logisticmap.ode
# the classic logistic map
x(t+1)=a*x*(1-x)
par a=2.8
init x=.64285
@ total=200,meth=disc
done
```

Run *XPPAUT* with this. We have started pretty close to the fixed point. Click on `nUmerics` in the main *XPPAUT* window and change `nOutput` from 1 to 8 and exit the numerics menu. When AUTO is run within *XPPAUT*, it looks at the numeric parameter `nOut` and uses this to iterate the map `nOut` times before returning the value of the map. Thus, it is possible to use *XPPAUT* to continue branches of periodic points of maps rather easily. We will thus look at the 8th iterate of the map. Fire up AUTO `File Auto`. Click on `Axes` and choose `Hi Lo`. Fill in the dialog box as follows and then click `Ok`:

Y-axis: X
Main Parm: a
2nd Parm: a
Xmin: 2.5
Ymin: 0
Xmax: 4
Ymax: 1

Click on `Numerics` in the AUTO window and change `Dsmax` from 0.5 to 0.05 and change `Par max` from 2 to 4. Now click on `Run` and you will see a rather messy looking diagram. There are three distinct branch points at $a = 3, 3.449, 3.544$ corresponding to the period 2, 4, and 8 bifurcations. You should see a figure like Figure 7.5.

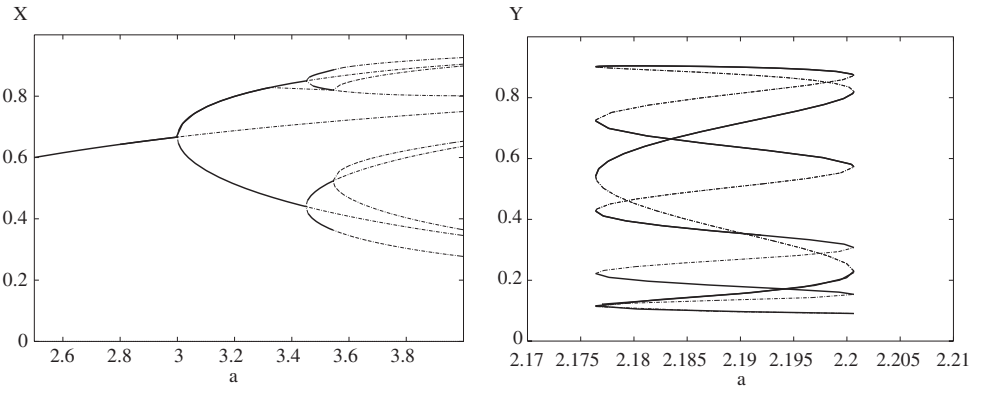

Figure 7.5. *Bifurcation diagram for period 8 orbits of the logistic map (left) and period 7 orbits for the delayed logistic map (right).*

We can use a similar trick to find a set of period 7 orbits in a two-dimensional map, the "delayed logistic" map

$$x_{n+1} = y_n, \qquad y_{n+1} = a x_n (1 - y_n) + \epsilon.$$

Here, there is an additional parameter ϵ which will play no role in our analysis. At $a = 2.177$ there is a period 7 orbit that starts at $(x, y) = (.1115, .12111)$. We will let AUTO find the entire branch of solutions for this model. Here is the ODE file:

```
# delayed logistic map
# illustrates a period 7 continuation for the delayed
# logistic map
y'=x
x'=a*x*(1-y)+eps
par a=2.177,eps=0
init x=.1115,y=.12111
@ meth=discrete,total=100,nout=7
@ autoxmin=2.17,autoxmax=2.21,autoymin=0,autoymax=1
@ dsmax=.025,dsmin=.00001,parmin=2,parmax=2.5
done
```

I have taken the liberty of setting some of the AUTO parameters so that you can just run this with no hassle. I have also set `nout=7` so that AUTO looks at the 7th iterate of the map. Start up *XPPAUT* and iterate it a few times to make sure you have no transients. Then, click on `File Auto` and, in the **AUTO Window**, click on `Run` to compute the diagram. It looks complicated but, in reality, it simply shows that for $2.1764 < a < 2.20$ there are two period 7 points; one is stable and the other is unstable. The reason there appears to be many branches is that for a period 7 point, there are 7 possible starting values. Figure 7.5 shows the diagram.

We will now do a two-parameter bifurcation analysis of the above problem. You can either edit all the parameters in the above file or simply load in the stripped down version:

```
# del_log2.ode
# delayed logistic map
# simple 2 parameter stuff
y'=x
x'=a*x*(1-y)+eps
par a=1.5,eps=0
init x=.333,y=.333
@ meth=discrete,total=100
done
```

Run this model, then start up AUTO. Use the `Axes Hi Lo` and change them so the `Xmin=0`, `Ymin=-1`, `Xmax=4`, `Ymax=1`. In AUTO `Numerics`, change `Par max` to 4. Then run the continuation. You should see a so-called Hopf point labeled `2`. Note that it says the period is 6. This may mean that there is a period 6 point lurking nearby.

Let's draw the two-parameter diagram using this heretofore worthless parameter ϵ. Grab the Hopf point by clicking on `Grab`, moving to it, and then clicking `Enter`. Then click `Axes Two par` and click on `Ok` in the dialog box—the window is fine. Click on `Run` and you will see a curve of Hopf points in the `eps-a` plane and another lower curve which is tangent to this curve at a point approximately $a = 2.8$ and $\epsilon = -0.3$. Along the lower curve, there is an eigenvalue of $+1$, and along the upper curve are eigenvalues on the unit circle but which are not real.

7.2.2 Exercises

1. The Ricker model is another discrete population model that is often used:

$$x_{n+1} = ax_n e^{-x_n}.$$

   Starting at $a = 1.2$ and $x = 0.18232$, compute the sequence of period doublings up to period 8 using the 8th iterate. The parameter a should be allowed to range between 0 and 20 and the population x line between 0 and 8.

2. The modified Nicholson–Bailey system models a predator–prey interaction:

$$x_{n+1} = x_n \exp(r(1 - x_n/k) - ay_n), \qquad y_{n+1} = x_n(1 - \exp(-ay_n)),$$

   where r, k, a are positive parameters. Start at $r = .5$, $k = .5$, $a = 1$, $x = 1$, $y = 0$ and create the bifurcation diagram for $0 < k < 5$. Note that k is the carrying capacity and represents the amount of prey that exists when there are no predators. Solve the system when $k = 3.2$. Does the resulting "periodic" solution have a period roughly as predicted from the Hopf point? Set $k = 5$, $r = 1.6$, and verify that there is a period 7 orbit starting at roughly $x = .376$, $y = .198$. Continue this period 7 orbit holding r fixed and letting $4.5 < k < 5.5$ be the bifurcation parameter. Plot the prey x in a window $0 < x < 6$ and finally set `Dsmax=0.1`. You will get a nice corkscrew!

7.3 BVPs

XPPAUT is able to solve BVPs as we saw earlier in the book. However, the method used by *XPPAUT* depends on shooting which can be wildly difficult when the desired evolution equation has large positive eigenvalues. Furthermore, *XPPAUT* neither tracks solutions through turning points nor finds branch points, which are often indicators of nontrivial solutions. Thus, if you can write your problem as an autonomous boundary value problem on the interval [0, 1] and find a simple starting solution, then AUTO is probably your best bet for tracking a solution. We will examine a few cases.

EXAMPLE 1. Here is an example form the original AUTO manual that involves branch switching:

$$u' = v, \quad v' = -(\pi r)^2 u + u^2, \qquad u(0) = u(1) = 0.$$

Here is an ODE file that has all the required AUTO parameters set for you:

```
# bvp1.ode
# using AUTO to find the branches of a BVP
par r=0
u'=v
v'=-(r*pi)^2*u+u^2
b u
b u'
@ total=1,dt=.01
@ dsmax=.2,ds=.2,autoxmin=0,autoxmax=6,autoymin=0,autoymax=100
@ ntst=5,parmax=5
@ meth=cvode,bound=10000,xhi=1,ylo=-50,yhi=50
done
```

Run *XPPAUT* and integrate this once to get the trivial solution loaded. Click on `File Auto` to bring up the AUTO window. In the **AUTO Window**, click on `Run Bndry Value` and you will see a series of labeled points along the bottom of the figure. Click on `Numerics` and change `Dsmax` to 10 and then click on `Ok`. `Grab` the first branch point (`BP`) (labeled 2). Then click on `Run` again; AUTO automatically switches when asked to continue from branch points. You will see a branch rising up from this point. When AUTO is done, grab the next branch point labeled 3 and continue this. Do this for the branch points at $r = 3, 4$ as well. Now, click on `Numerics` and change `Ds` to -0.2 to change directions, and click on `Ok`. `Grab` each of the branch points and run again. For some of them, it will appear like nothing has happened; this is because the maximum value of u is the same on each branch so that they are not distinguishable. Click on `Axes Hi` and change the `Y-axis` to `v` instead of `u`, click on `Ok`, and then `reDraw`. You will see a slightly different view.

Trick: Distinguishing branches in AUTO

Here is a trick that you can use to avoid the symmetry problem in graphing the curves of solutions. As we saw in the above example, the things that AUTO plots—maximum and norm of a solution—cannot always lead to distinctions. So here is a way to fool AUTO

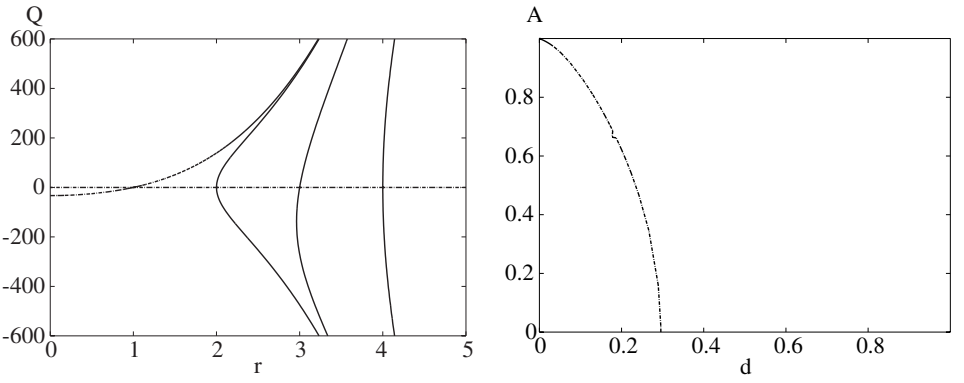

Figure 7.6. *Solution to the BVPs from Example 1 (left) and Example 2 (right).*

into letting you plot something else in the bifurcation diagram. For the above problem, the value of v at the origin is zero for the trivial branch of solutions, but for the solutions that emanate from each branch point it is nonzero and it is necessarily different. (The reason for this follows from the uniqueness theorem for ODEs. Since $u(0) = 0$, if $v_1(0) = v_2(0)$ are two different branches, then the solutions are identical.) Thus, we introduce a dummy differential equation

$$q' = 0$$

with a boundary condition $q(0) = v(0)$. Then the solution to this dummy equation satisfying the additional boundary condition is just $q(t) = v(0)$ for all t. Add the following two lines to the above ODE file:

```
q'=0
b q-v
```

The dimension of the system is a bit larger but this is a triviality. Figure 7.6 shows the result of this trick.

EXAMPLE 2. Here we revisit the example studied in Chapter 4 and which arises in the analysis of a reaction-diffusion equation on a disk. Recall that the BVP is

$$0 = A(1 - A^2) + d(A'' - (A/r)' - Ak^2),$$

$$\Omega = qA^2 + d(k' + k/r - 2kA'/A),$$

with boundary conditions $A'(1) = k(1) = 0$, and

$$\lim_{r\to 0} \frac{A(r)}{r} = A'(0) < \infty, \qquad \lim_{r\to 0} \frac{k(r)}{r} < \infty.$$

We handled the behavior as $r \to 0$ by forming a Taylor series near the origin. We let r_0 be close to zero and start the problem at $r = r_0$. This problem is nonautonomous, so we

introduce the differential equation

$$r' = 1$$

with boundary condition $r(0) = r_0$. As before, Ω is a free parameter so we write this as an ODE also. Here is the new ODE file which is autonomous and thus suitable for AUTO:

```
# gberg_auto.ode
#
init a=0.00118  ap=1.18  k=0  omeg=-0.19,r=.001
# domain is reasonably small sqrt(1/d)
par d=0.177  q=0.5, r0=.001
# the odes...
a'=ap
ap'=a*k*k-ap/r+a/(r*r)-a*(1-a*a)/d
k'=-k/r-2*k*ap/a-(omeg+q*a*a)/d
# extras to make it autonomous
omeg'=0
r'=1
# the boundary conditions
# at r=0
bndry  a-r*ap
bndry  k
bndry r-r0
# at r=1
bndry  ap'
bndry  k'
# set it up for the user
@ xhi=1.001,dt=.01,total=1.001,ylo=0
@ parmax=1,autoxmax=1,autoxmin=0,autoymax=1,autoymin=0
done
```

We have added a few setup parameters for AUTO. I have also computed a good starting point through trial and error. The parameter of interest is d which is like the diffusion constant; if d increases, the effective domain size decreases and, as $d \to 0$, the domain becomes infinite. *XPPAUT* can only follow a branch with d as a parameter in a limited range; thus, we will use AUTO to get a fuller picture. Run *XPPAUT* with this ODE file and integrate it once to load the initial solution. Click on `File Auto` to get the **AUTO Window**. In the **AUTO Window**, click on `Run Bdry value` and then, when the curve hits the horizontal axis, click on `ABORT` to stop it. Click on `Numerics` and change `Ds` to -0.02 and click `Ok` so that we can continue in the opposite direction. `Grab` the starting point (labeled 1). Run again and, when the curve gets fairly close to the vertical axis, `ABORT` the calculation. You will see a parabola that intersects the horizontal axis at about $d = 0.295$. This tells us that if the diffusion is too great, it is impossible to support rotating waves. Similarly, as d gets smaller, the magnitude of the rotating waves increases; they seem to exist in arbitrarily large domains.

7.3.1 Homoclinics and heteroclinics

A recent version of AUTO includes a library of routines called HOMCONT which allow the user to track homoclinic and heteroclinic orbits. *XPPAUT* incorporates some aspects of this package. The hardest part of computing a branch of homoclinics is finding a starting point. Consider a differential equation

$$x' = f(x, \alpha),$$

where α is a free parameter. Homoclinics are codimension one trajectories; that is, they are expected to occur only at a particular value of a parameter, say $\alpha = 0$. We suppose that we have computed an approximate homoclinic to the fixed point $\bar{x}$ which has an n_s-dimensional stable manifold and an n_u-dimensional unstable manifold. We assume $n_s + n_u = n$, where n is the dimension of the system. The remaining discussion is based on Champneys, Kuznetsov, and Sandstede [7]. The way that a homoclinic is computed is to approximate it on a finite interval, say $[0, P]$. We rescale time by $t = Ps$. We double the dimension of the system so that we can simultaneously solve for the equilibrium point as the parameters vary. We want to start along the unstable manifold and end on the stable manifold. Let L_u be the projection onto the unstable subspace of the linearization of f about the fixed point and let L_s be the projection onto the stable space. Then we want to solve the following system:

$$\begin{aligned} \frac{dx}{ds} &= Pf(x, \alpha), \\ \frac{dx_e}{ds} &= 0, \\ f(x_e(0)) &= 0, \\ L_s(x(0) - x_e(0)) &= 0, \\ L_u(x(1) - x_e(1)) &= 0. \end{aligned} \tag{7.10}$$

Note that there are $2n$ differential equations and $2n$ boundary conditions: n for the equilibrium, n_s at $s = 0$, and n_u at $s = 1$. There is one more condition required. Clearly, one solution to this BVP is $x(s) \equiv x_e(s) \equiv \bar{x}$ which is pretty useless. However, any translation in time of the homoclinic is also a homoclinic so we have to somehow define a phase of the homoclinic. Suppose that we have computed a homoclinic $\hat{x}(s)$. Then we want to minimize the least-squares difference between the new solution and the old solution to set the phase. This leads to the following integral condition:

$$\int_0^1 \hat{x}'(s)(\hat{x}(s) - x(s))\, ds = 0. \tag{7.11}$$

This is *one* more condition which accounts for the need for an additional free parameter.

XPPAUT allows you to specify the projection boundary conditions and, by setting a particular flag "on" in AUTO, you can implement the integral condition. Since the *XPPAUT* version of AUTO does not allow you to have more conditions than there are differential

equations, you should pick one parameter which will be slaved to all the other ones you vary and which will satisfy a trivial differential equation

$$\alpha' = 0.$$

This will be varied by AUTO to satisfy the conditions (7.10), (7.11).

Here is an example of continuing a homoclinic in two dimensions:

$$x' = y, \qquad y' = x(1-x) - ax + \sigma xy.$$

When $(a, \sigma) = (0, 0)$, there is a homoclinic orbit. (Prove this by integrating the equations; this is a conservative dynamical system.) For small a it is possible to prove that there is a homoclinic orbit for a particular choice of $\sigma(a)$ using Melnikov methods (see Guckenheimer and Holmes [18]). We now write the equations as a five-dimensional system using σ as the slaved parameter and introducing a parameter P for the period:

$$\begin{aligned} x' &= Pf(x, y), \\ y' &= Pg(x, y), \\ x_e' &= 0, \\ y_e' &= 0, \\ \sigma' &= 0, \end{aligned}$$

where $f(x, y) = y$, $g(x, y) = x(1-x) - ax + \sigma xy$. The boundary conditions are

$$\begin{aligned} 0 &= f(x_e, y_e), \\ 0 &= g(x_e, y_e), \\ 0 &= L_s(x(0) - x_e, y(0) - y_e), \\ 0 &= L_u(x(1) - x_e, y(1) - y_e), \end{aligned}$$

along with the integral condition. *XPPAUT* has a defined function for the projection boundary conditions called `hom_bcs(k)`, where `k=0,1,...,n-1` corresponding to the total number required. You do not need to be concerned with ordering at this point as long as you get them all and you give XPP the required information. Here is the ODE file:

```
# tsthomi.ode
f(x,y)=y
g(x,y)=x*(1-x)-a*y+sig*x*y
x'=f(x,y)*per
y'=g(x,y)*per
# auxiliary ODE for fixed point
xe'=0
ye'=0
#
# free parameter
```

```
sig'=0
# (xe,ye) are the fixed points at the ends
b f(xe,ye)
b g(xe,ye)
# project off the fixed point from unstable manifold
b hom_bcs(0)
# project onto the stable manifold
b hom_bcs(1)
par per=8.1,a=0
init x=.1,y=.1
@ total=1.01,meth=8,dt=.001
@ xlo=-.2,xhi=1.6,ylo=-1,yhi=1,xp=x,yp=y
done
```

The only new feature is the projection conditions. *XPPAUT's boundary value solver will not work here since there are more equations than conditions and it doesn't know about the integral condition*. I have set the total integration time to 1 and have added the additional parameter `per` corresponding to the parameter P in the differential equation. I use the Dormand–Prince order 8 integrator, as it is pretty accurate. I have also set the view to be the (x, y)-plane. Note that this is a pretty rough approximation of the true homoclinic. We will use AUTO to improve this before continuing in the parameter a. Run *XPPAUT* with this ODE file and integrate the equations. You will get a rough homoclinic pretty far from the fixed point. Click on `File Auto` to get to the **AUTO Window**. Now click on `Axes Hi`. Choose `xmin=0, xmax=50, ymin=-6, ymax=6`, and also select `sig` as the variable in the y-axis. Click on `OK` and bring up the **AUTO Window** `Numerics` dialog box. Change `Ntst=35, Dsmin=1e-4, Dsmax=5, Par Max=50, EPSL=EPSU=EPSS=1e-7` and click `OK`. Now, before you run the program, click on `Usr Period` and choose `3` for the number. We want AUTO to output at particular values of the parameter `per` corresponding to P. When the dialog box comes up, fill in the first three entries as `per=20, per=35, per=50`, respectively, and click `OK`. This forces AUTO to output when P reaches these three values. Now, click on `Run` and choose `Homoclinic`. A little dialog box appears. Fill it in as follows and click `OK`:

Left Eq: Xe
Right Eq: Xe
NUnstable: 1
NStable: 1

You must tell AUTO the dimension of the stable and unstable manifolds as well as the fixed point to which the orbit is homoclinic. (Note that if you ever fill this in wrong or need to change it, you can access it from the **Main Window** menu under `Bndry Value Homoclinic`.) Once you click on `OK`, you should see a straight line across the screen as the homoclinic approximation gets better. Click on `Grab` and grab the third point corresponding to the point `Per=35`. For fun, in the **Main Window** click on `Initial Conds Go` and you will see a much better homoclinic orbit.

Now that we have a much improved homoclinic orbit, we will continue in the parameter a as desired. First, let's make sure we get the orbits when $a = -6, -4, -2, 2, 4, 6$.

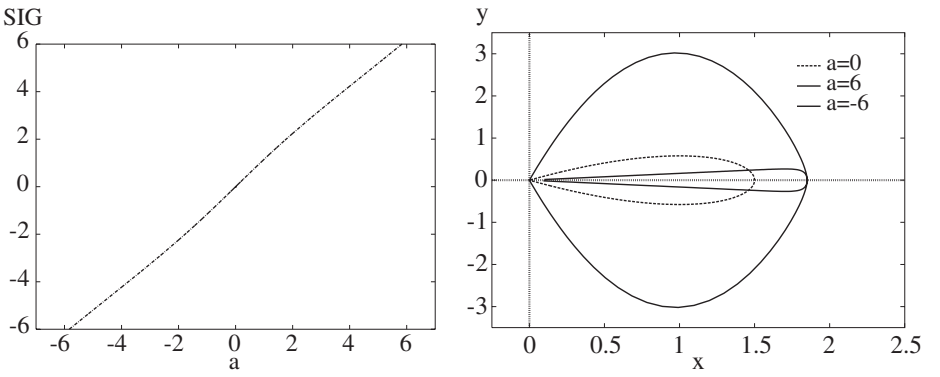

Figure 7.7. *Left: The continuation of the homoclinic orbit for the example problem. Right: Sample orbits computed for* $a = 0, \pm 6$ *in the* (x, y) *phase-plane.*

We will click on `Usr Period` and choose `6`. Type in the following in the first six entries: `a=-6`, `a=-4`, `a=-2`, `a=2`, `a=4`, `a=6`, and click `Ok`. Click on `Axes Hi` to change the axes and the continuation parameter. Change the `Main Parm` to `a`, `Xmin=-7`, `Xmax=7`, and click `OK`. Click on `Numerics` and change `Par Min=-6`, `Par Max=6`, and then click `OK`. Now click on `Run` and you will see a line that is almost diagonal. When done, click on `Grab` again, watch the bottom of the AUTO window until you see `Per=35`, and click `Enter`. In the **AUTO Window** `Numerics` menu, change `Ds=-.02` to change directions, and click `Ok`. Now click `Run` and there will be another diagonal line going in the opposite direction. Click on `Grab` and grab point number 7 corresponding to `a=6`. In the **Main Window**, click on `Initial Conds Go` and you will see a distorted homoclinic. It is not that great and could probably be improved by continuing with `Per`. Grab the point labeled 11 (`a=-6`) and in the **Main Window** try to integrate it. It doesn't look even close. This is because the homoclinic orbit is unstable, and shooting (which is what we are doing when we integrate the equation) is extremely sensitive to the stability of the orbits. In the **AUTO Window**, click on `File Import Orbit` to get the orbit that AUTO computed using collocation. In the **Main Window**, click on `Restore` and you will see a much better version of the homoclinic orbit. This is because collocation methods are not sensitive to the stability of orbits! In fact, you can verify that the fixed point (0,0) is a saddle-point with a positive eigenvalue λ_u and a negative of λ_s, whose sum is the trace of the linearized matrix, $-a$. The sum of the eigenvalues is called the **saddle-quantity** and, if it is positive (for us, $a < 0$), then the homoclinic is unstable. Figure 7.7 shows the continuation and some representative orbits.

Heteroclinics

We now describe how to find heteroclinic orbits. The methods are the same except that we must track two *different* fixed points. Thus, we need an additional n equations for the other fixed point. As with homoclinic orbits, we go from the unstable manifold to the stable

manifold. In this case, the "left" fixed point is the one emerging from the unstable manifold and the "right" fixed point is the one going into the stable manifold. Thus, the dynamical system is

$$\begin{aligned}
\frac{dx}{ds} &= Pf(x,\alpha),\\
\frac{dx_{left}}{ds} &= 0,\\
\frac{dx_{right}}{ds} &= 0,\\
f(x_{left}(0)) &= 0,\\
f(x_{right}(1)) &= 0,\\
L_s(x(0) - x_{left}(0)) &= 0,\\
L_u(x(1) - x_{right}(1)) &= 0.
\end{aligned}$$

The only difference is that we have the additional n equations for the right fixed point and the n additional boundary conditions. It is important that you give good values for the initial conditions for the two fixed points since they are different and you need to converge to them. The classic bistable reaction-diffusion equation provides a nice example of a heteroclinic. We examined this in equation (6.1) in the previous chapter both by integrating the discretized partial differential equation and by shooting in the ODE arising from the traveling wave assumption. The equation is

$$-cu' = u'' + u(1-u)(u-a)$$

which we rewrite as a system

$$\begin{aligned}
u' &= u_p \equiv f(u,u_p),\\
u_p' &= -cu_p - u(1-u)(u-a) \equiv g(u,u_p).
\end{aligned}$$

The fixed point $(1,0)$ has a one-dimensional unstable manifold and $(0,0)$ as a one-dimensional stable manifold. Recall that we found that, when $a = 0.25$, the velocity c was approximately $c \approx 0.34375$. We will now use a different method to compute the solution—solve the approximate BVP. We seek a solution heteroclinic from $(1,0)$ to $(0,0)$. For $a = 0.5$ and $c = 0$, there is an exact solution joining the two saddle-points. (Prove this by showing that $u_p^2 + u^4/2 - 2u^3/3 + u^4/2$ is constant along solutions when $a = 0.5$, $c = 0$.) We will use this as a starting point in our calculation. Here is the ODE file:

```
# tstheti.ode
# a heteroclinic orbit
# unstable at u=1, stable at u=0
f(u,up)=up
g(u,up)=-c*up-u*(1-u)*(u-a)
```

```
# the dynamics
u'=f(u,up)*per
up'=g(u,up)*per
# dummy equations for the fixed points
uleft'=0
upleft'=0
uright'=0
upright'=0
# the velocity parameter
c'=0
# fixed points
b f(uleft,upleft)
b g(uleft,upleft)
b f(uright,upright)
b g(uright,upright)
# projection conditions
b hom_bcs(0)
b hom_bcs(1)
# parameters
par per=6.67,a=.5
# initial data
init u=.918,up=-.0577,c=0
# initial fixed points
init uleft=1,upleft=0,uright=0,upright=0
@ total=1.01,dt=.01
@ xp=u,yp=up,xlo=-.25,xhi=1.25,ylo=-.75,yhi=.25
# some AUTO parameters
@ epss=1e-7,epsu=1e-7,epsl=1e-7,parmax=60,dsmax=5
@ dsmin=1e-4,ntst=35
done
```

I have added a few AUTO numerical settings so that I don't have to set them later. Run *XPPAUT* with this ODE file and integrate the equations. We will now continue this approximate heteroclinic in the parameter `per`. Fire up AUTO (`File Auto`) and click on `Axes Hi`. Put `c` on the y-axis and make `Xmin=0,Xmax=60,Ymin=-2,Ymax=2`. Then click `OK`. As above, we will also keep solutions at particular values of `per` by clicking on `Usr period 3`, choosing `per=20,per=40,per=60`, and then `OK`. Now we are ready to run. Click on `Run Homoclinic`. When the dialog box comes up, set the following:

Left Eq: uleft
Right Eq: uright
NUnstable: 1
NStable: 1

and then click `OK`. You should see a nice straight line across the screen. `Grab` the point labeled `per=40` and then click on `File Import Orbit`. Look at it in the **Main**

Window by clicking `Restore` and freeze it. Now in the **AUTO Window**, click on `Axes Hi` and change the `Main Parm` to `a` and `Xmax=1`. Click `OK` and then click on `Numerics` to get the `Auto Numerics` dialog box. Change `Dsmax=0.1,Par Max=1`, and click `OK`. Now click on `Usr Period 4` and make the four user functions `a=.75,a=.9,a=.25,a=.1`, and click `OK`. Now click on `Run` and watch a line being drawn across the screen. This is the velocity c as a function of the threshold a. Click on `Grab` and, looking at the bottom of the screen, wait until you see `per=40` and then click `Enter`. Now, open the `Numerics` dialog box and change `Ds=-.02` to go in the other direction. Click on `Run` and you should see the rest of the line drawn across the screen. Click on `Grab` and move to the point labeled `a=0.25`. Click on `File Import Orbit` and plot this in the **AUTO Window**. Amazingly, there is essentially no difference in the two plots! Can you explain why? Notice the value of c at the point $a = .25$. It is very close to the value we computed by shooting in Chapter 6.

7.3.2 Exercises

1. A nonlinear PDE that arises in the study of pattern formation has the form

$$u_t = -u_{xxxx}/\pi^4 - au_{xx}/\pi^2 - u + r\tanh(u)$$

with boundary conditions

$$u(0,t) = u_{xx}(0,t) = u(1,t) = u_{xx}(1,t) = 0.$$

One steady-state solution to this is $u = 0$. If a is large enough, the zero state loses stability as r increases, and the resulting solution is a stationary spatial pattern. Since this solution branches from the origin, we can use AUTO to try to find a nontrivial branch of solutions. The steady-state solution is a fourth order nonlinear problem

$$u'''' = \pi^4(r\tanh(u) - u + au''/\pi^2)$$

with

$$u(0) = u''(0) = u(1) = u''(1) = 0.$$

We can rewrite this as a four-dimensional system and then write an ODE file for it. Here is a possible ODE file where I have set up AUTO for you as well:

```
# boundary value problem
# 0 = - (u_xxxx/pi^4+au_xx/pi^2)-u+r tanh(u)
# u,u_xx = 0 at x=0,1
u'=up
up'=upp
upp'=uppp
uppp'=pi^4*(r*tanh(u)-u-a*upp/(pi*pi))
par r=1,a=2
@ total=1.01,xhi=1,meth=cvode,bound=1000
@ dsmax=2,parmax=20,autoxmin=0,autoxmax=20
```

```
@ autoymin=-1,autoymax=4
b u
b u'
b upp
b upp'
done
```

Run this in *XPPAUT* and integrate once. Then look at the bifurcation diagram with r as the parameter. Follow any branches of solutions that arise. Grab a point on a nontrivial branch (not too far away from the branch point) and, back in the **Main Window** of *XPPAUT*, integrate the solution showing that it is nonconstant.

2. Here is another nonlinear BVP:

$$u'' = -r\exp(u), \quad u(0) = u(1) = 0,$$

where r is a parameter. Write this as a pair of first order equations

$$u' = v, \qquad v' = -r\exp(u).$$

Starting with $r = 0$ use AUTO to continue the solution $u = 0$ for $0 < r < 10$. Window the y-axis in the bifurcation diagram between 0 and 20. Show that if $0 < r < r^*$, there are two solutions to the BVP and, if $r > r^*$, there appears to be none.

3. Consider the nonautonomous equation

$$u''(x) = -r\exp(ux^2), \quad 0 < x < 1, \quad u(1) = u(0) = 0.$$

By introducing the additional equation $x' = 1$, $x(0) = 0$, solve this using AUTO over the interval $0 \le r \le 20$.

4. Using the file `ml.ode` that we studied in Exercise 1 of section 7.1.2, find the homoclinic orbit and continue this, using the two parameters `I,phi`. Do this by adding an equation for the applied current

$$I' = 0$$

and treating `phi` as a parameter. You should start at the long period point that terminates the branch of periodic orbits. This is a difficult exercise.

7.4 Periodically forced equations

To solve periodically forced equations, one can always use the same trick we used in the previous section and increase the dimension by one. For example,

$$x' = f(x, t),$$

where f is periodic in t with period 1. We want to look for periodic solutions, so we solve

$$x' = f(x,s), \quad s' = 1, \quad x(0) - x(1) = 0, \quad s(0) = 1.$$

However, this method does not say anything about the stability of the solutions. Instead, what I do is introduce the two-dimensional dynamical system

$$u' = u(1-u^2-v^2) - \omega v, \qquad v' = v(1-u^2-v^2) + \omega u.$$

This admits an asymptotically stable periodic orbit $(u(t), v(t)) = (\cos\omega t + \phi, \sin(\omega t + \phi))$, where ϕ is an arbitrary phase-shift. Thus, to solve a periodically driven system, I merely append this new system and couple $u(t)$ to it. Here is an example of a periodically forced bistable system

$$x' = x(1-x)(x-a) + c\cos\omega t.$$

For use with AUTO, we create the following ODE file:

```
# bistabfor.ode
# periodically forced bistable system
x'=x*(1-x)*(x-a)+c*u
u'=u*(1-u^2-v^2)-w*v
v'=v*(1-u^2-v^2)+w*u
init u=1,x=.043
par c=.1,a=.25,w=1
@ dt=.01,total=6.281,xlo=0,xhi=6.5
@ dsmax=.05,parmin=-1,parmax=1
@ autoxmax=1,autoxmin=-1
@ autoymin=-.5,autoymax=2
done
```

I have set it up for AUTO. Run *XPPAUT* with this file. Integrate the equations once. Now click on `File Auto` to get AUTO started. The diagram is set up so that the forcing strength c is the main parameter. Click on `Run Periodic` to pick up the periodic solution. Click on `ABORT` to stop it from infinitely looping. You should see a little four-leaf clover pattern. Remember that this is showing the maximum and minimum of each periodic solution. There are two such solutions; one is stable and one is unstable. Click on `Grab` to load the first point. Click on `Enter` immediately afterwards. Now in the **AUTO Window** click on `File Clear grab`. This makes AUTO forget the point it just grabbed. However, the point is still loaded into *XPPAUT*. In the **Initial Data Window** of *XPPAUT*, change the initial value of x to 1. In the **Main Window** of *XPPAUT*, click on `Initialconds Go` and then, when that is done, click `Initialconds Last`. We now have another periodic solution where x is large and not near 0. Returning to the **AUTO Window**, click on `Run Periodic` and a new branch of solutions will appear. The point of clearing the AUTO grab point is that AUTO can start fresh from a new point. Grab the starting point of this most recent computation—this may require pressing the `Tab` key several times—until the cross is on the first point in the upper branch. Change the sign of `Ds` in the `Numerics` dialog box of the **AUTO Window**. Click on `Run Extend` to get the rest of the branch. You should see something like Figure 7.8.

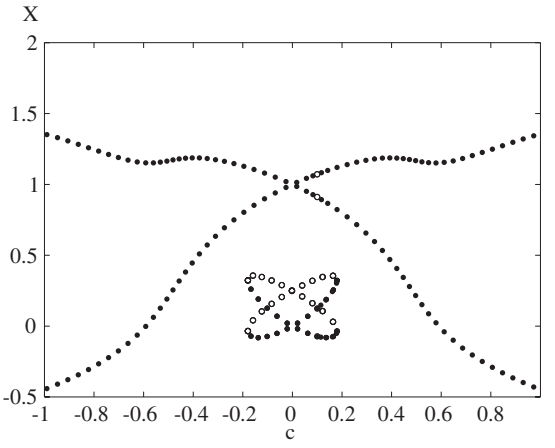

Figure 7.8. *Bifurcation diagram for the periodically driven bistable system.*

7.4.1 Exercises

1. Periodically forced oscillators lead to remarkably complex behavior. Here, we will look at the periodically driven Fitzhugh–Nagumo equations (see Chapter 3). Here is the appropriate ODE file:

```
# fhnfor.ode
# Fitzhugh-Nagumo equations with sinusoidal forcing
v'=I+v*(1-v)*(v-a) -w+c*u
w'=eps*(v-gamma*w)
u'=u*(1-u^2-z^2)-f*z
z'=z*(1-u^2-z^2)+f*u
init u=1,v=.0505,w=.1911
par c=.05,f=.8
par I=0.2,a=.1,eps=.1,gamma=.25
@ xp=V,yp=w,xlo=-.25,xhi=1.25,ylo=-.5,yhi=1,total=7.86
@ dsmax=.05,parmin=-.5,parmax=1
@ autoxmax=1,autoxmin=0,autoymin=-1.5,autoymax=1.5
done
```

Run this a few times to make sure you are locked on a periodic. Then Run AUTO using the above parameters. You should see that the periodic solution loses stability at a point which AUTO marks as `PD`—this means that there is a period-doubling instability. Grab this point and click on `Run`, choosing `Period doubling` from the menu choices. Be ready to click on the `ABORT` key since it will circle around the period 2 orbit.

2. Find a problem in a research paper that involves periodic forcing and analyze the bifurcation behavior. Or, see what kind of phenomena you can find by adding a term like

$$c \cos \omega t$$

to the Morris–Lecar equations.

7.5 Importing the diagram into *XPPAUT*

In this section, I will describe a nice geometric idea that underlies the complex firing patterns of some neurons. Many neurons exhibit what is known as bursting behavior which consists of bouts of rapid spiking followed by period of relative quiescence. A typical voltage trace is shown in Figure 7.9. The model used to produce this is the Morris–Lecar model (previously described) but is such that the applied current satisfies a differential equation depending on the potential:

$$C\frac{dv}{dt} = I - g_L(V - V_L) - g_{Ca}m_\infty(v)(V - V_{Ca}) - g_K w(V - V_K),$$

$$\frac{dw}{dt} = \phi\frac{w_\infty(v) - w}{\tau_w(v)},$$

$$\frac{dI}{dt} = \epsilon(v_0 - v)$$

with parameters given below in the associated ODE file:

```
# mlsqr.ode
dv/dt = ( I - gca*minf(V)*(V-Vca)-gk*w*(V-VK)-gl*(V-Vl))/c
dw/dt = phi*(winf(V)-w)/tauw(V)
dI/dt = eps*(v0-v)
v(0)=-41.84
w(0)=0.002
minf(v)=.5*(1+tanh((v-v1)/v2))
winf(v)=.5*(1+tanh((v-v3)/v4))
tauw(v)=1/cosh((v-v3)/(2*v4))
param eps=.001,v0=-26,vk=-84,vl=-60,vca=120
param gk=8,gl=2,c=20
param v1=-1.2,v2=18
param v3=12,v4=17.4,phi=.23,gca=4
@ dt=1,total=4000,meth=cvode
@ xp=I,yp=v,xlo=25,xhi=45,ylo=-60,yhi=20
done
```

This is set up in the (I, v)-plane to illustrate how the current I increases and then decreases as the voltage either moves through an apparent steady state or oscillates. Plot I against t to see that the oscillations are almost nonexistent for I. This is because ϵ is small and so

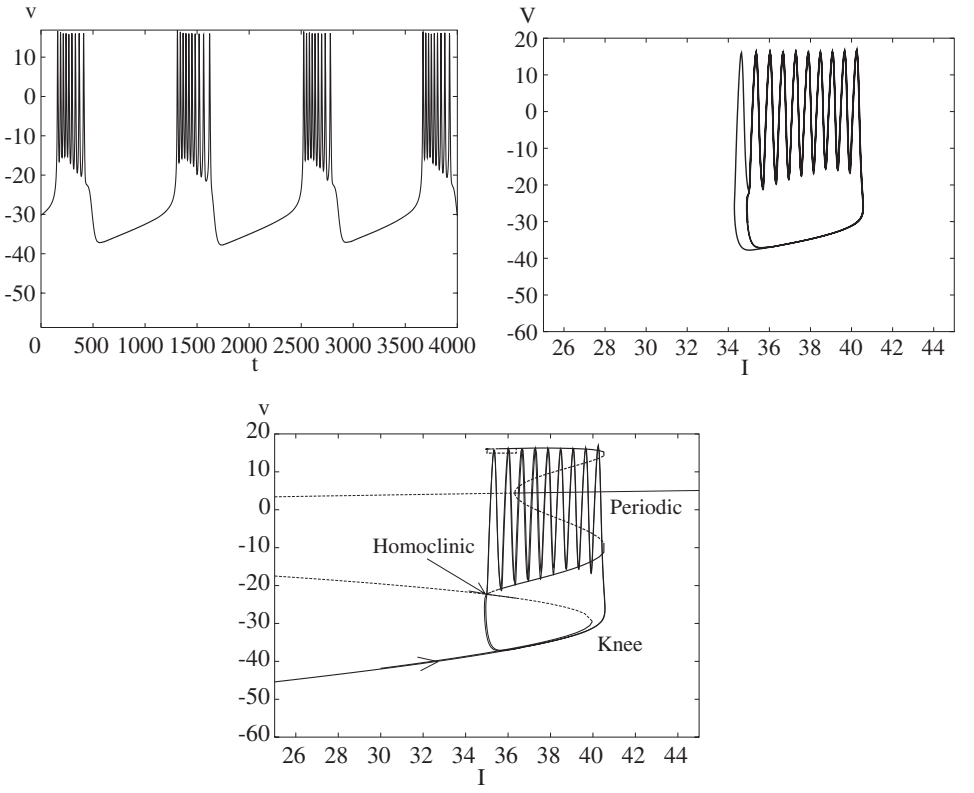

Figure 7.9. *Bursting in a modified membrane model. Top left: The voltage as a function of time during a burst. Top right: The slow parameter I and the voltage. Bottom: Superimposition of the bifurcation diagram in the* (I, v)*-plane.*

the I equation is essentially satisfying

$$\frac{dI}{dt} = \epsilon(v_0 - \langle v \rangle),$$

where $\langle v \rangle$ is the average of $v(t)$.

Since I is slowly varying, this suggests treating I as a parameter and studying the behavior of the (v, w) system as I varies. The file `ml2dhom.ode` will do the trick:

```
# ml2dhom.ode
dv/dt = ( I - gca*minf(V)*(V-Vca)-gk*w*(V-VK)-gl*(V-Vl))/c
dw/dt = phi*(winf(V)-w)/tauw(V)
v(0)=-41.84
w(0)=0.002
minf(v)=.5*(1+tanh((v-v1)/v2))
winf(v)=.5*(1+tanh((v-v3)/v4))
tauw(v)=1/cosh((v-v3)/(2*v4))
```

```
param i=30,vk=-84,vl=-60,vca=120
param gk=8,gl=2,c=20
param v1=-1.2,v2=18
param v3=12,v4=17.4,phi=.23,gca=4
@ total=150,dt=.25,xlo=-60,xhi=60,ylo=-.125,yhi=.6,xp=v,yp=w
@ dsmax=2,parmin=-30,parmax=60
@ autoxmin=-10,autoxmax=60,autoymin=-60,autoymax=40
done
```

I have set this up specifically for AUTO so that it is ready to analyze the system. (The virtues of hindsight.) The last two lines of options set up the AUTO window and the numerical continuation. Run *XPPAUT* and integrate this a few times using `Initialconds Go` and then the `Initialconds Last` commands. Then click on `File Auto` and click on `Run Steady state` in the **AUTO Window**. You should see a cubic-like curve of steady states. Click on `Grab` and follow the branch until you find the Hopf bifurcation, labeled `HB`, at the bottom of the **AUTO Window**. Click `Enter` to grab this point. Now, click on `Run Periodic` to track the branch of periodic solutions emerging from the Hopf bifurcation. This branch terminates on the middle branch at a homoclinic orbit. The picture will be similar (but not identical) to Figure 7.4. Now for the cool part. We will save this diagram and then import it into the bursting file so that the (I, v) dynamics can be plotted superimposed on the bifurcation diagram. In the **AUTO Window**, click on `File Write pts` and pick a filename such as `mlhom.dat` to save the points of the diagram. Exit *XPPAUT*. Run *XPPAUT* with the file `mlsqr.ode` which is the previously viewed bursting equation. Integrate the equations to get something that looks like the top right of Figure 7.9. Now click on `Graphics Freeze Bif. Diag.` and choose the previously saved diagram, e.g., `mlhom.dat`. Voila! You will see the nice burst superimposed on the bifurcation diagram. This explains the mechanism for bursting. When the potential v is below v_0, the current I increases, causing the voltage to follow the lower stable fixed point. This continues until the "knee" is reached and the potential jumps up to the branch of periodic solutions. The average voltage is greater than v_0 so that I begins to decrease until it hits the homoclinic point. The potential then drops to the stable rest state where it repeats the cycle.

7.5.1 Exercise

Repeat the above trick with the canonical elliptic burster

$$u' = u(\lambda + R - R^2) - v + c,$$

$$v' = v(\lambda + R - R^2) + u + c,$$

$$\lambda' = \epsilon(R_0 - R),$$

$$R = u^2 + v^2.$$

Try $\epsilon = .01$, $R_0 = 0.3, c = 0.01$, and integrate for a total of 1000 with `dt=0.25` using `qualrk` as the method. Plot the (λ, u)-projection in a window $-0.4 < x < 0.4$ and

$-1.2 < y < 1.2$. Next write an ODE file for the (u, v) system with λ as a parameter. Start with $\lambda = -.4$ and find a steady state. Then track the fixed point as λ increases using AUTO. (**Hint:** Set `Dsmax=0.1` and `Par Min=-0.5` in the AUTO numerics dialog box.) Find the Hopf bifurcation and obtain the periodic solution. Write the points to a file. Then superimpose them on the elliptic burster (λ, u)-plane.