XPPAUT Tutorial

The free software package **XPPAUT**, written by Bard Ermentrout at the University of Pittsburgh, is a great tool for models consisting of ordinary differential equations. It's freeware that can be downloaded from his website, and it is available for any computer platform (even for iPad and iPhone). This short introduction will tell you some of the basic XPP commands using one example. Bard's web site has a tutorial and many files that you can download to learn more and get more practice.

This tutorial takes you through some basic XPP commands for solving systems of ordinary differential equations, and then commands for the generation of a bifurcation diagram using the AUTO software as implemented in XPPAUT. To illustrated this process, a model of oscillatory glycolysis developed in 1972, called the **Goldbeter-Lefever model**, is used. This describes in a very simple way how the molecule ATP is converted to ADP by the enzyme phosphofructokinase. This enzymatic function is described by "Φ". The ATP input rate to the system is ν and the ADP degradation rate is η. Parameter τ is a time constant.

The Goldbeter-Lefever model (*Biophysical Journal*, vol. 12, pp. 1302–1315, 1972) is

$$\begin{aligned} \frac{dATP}{dt} &= [\nu - \Phi(ATP, ADP)]/\tau \\ \frac{dADP}{dt} &= [\Phi(ATP, ADP) - \eta ADP]/\tau \end{aligned}$$

where $\Phi(ATP, ADP) = ATP(1 + k\,ADP)^2$, and ν, η, $k = 20$, and $\tau = 500$ are parameters. Various values for ν and η will be used.

The computer code for the Goldbeter-Lefever model, **Goldbeter.ode**, can be downloaded from my website under Course Software. The program starts with a title. This is just a comment, using the comment character %. Next comes a block of code setting the XPP parameters that tell XPP what to do. They say to use the CVODE method (a solver that is really good with stiff systems of equations) to numerically solve the ODE with initial step size of *dt*. The simulation should go over *total* time units. Parameters *maxstor* and *bounds* are there for technical reasons and you should never have to change them. Parameters *xp* and *yp* are just the names of the variables that should go on the x- and y-axis, respectively. The lower and upper bounds for the axes are *xlo*, *xhigh*, *ylo*, and *yhi*. The final parameter is the most important; it tells xpp whether or not to beep every time a simulation is complete. For the sake of sanity I set *bell*=0 (or *bell*=off in some versions of XPP). If you want to irritate your friends then set *bell*=1 and do a lot of simulations.

Next are initial conditions followed by a list of parameters and their values. You can change these values once you have started up XPP. Then there is an algebraic

expression for the phosphofructokinase rate, followed by the differential equations for ATP and ADP.

Running XPP ode files

There are a number of ways to run XPP, depending on the computer platform. You may just click on the name of the XPP application and Open the file Goldbeter.ode, or if using unix or linux you would type the command "xppaut Goldbeter.ode" or (if the executable is in the Applications folder), "/Applications/xppaut Goldbeter.ode". Notice the buttons on the top of the XPP screen. Clicking on these will open other windows that will allow you to do things like change the values of the system parameters or initial conditions.

Once XPP is running you should see a screen with coordinate axes. It should say "X vs T" at the top. To run a simulation (i.e., to solve the differential equation) click on Initialconds and Go. Once you get the hang of this you'll want to type instead of click. In this case you would type "ig". You should see how ATP evolves in time. To see what the ADP time course looks like you can open another window by clicking Makewindow and Create or typing "mc". [Note that this new window becomes the active window, as indicated by the small white or black dot in the left corner of the new window.] To change what is plotted in this new window click on Viewaxes and 2D (or type "v2"). Put ADP on the y-axis. While you are at it, you can change Ymax to 2 in the Viewaxes window (since ADP never goes beyond s). Click on each window and type "r" (Restore) to redraw the curve on the window.

Now let's see what happens when the initial condition is changed from ATP=0.5 to 0. To do this click on ICs at the top of the screen. Change 0.5 to 0 and click OK. Note that XPP does a funny thing here. For it to accept the change you need to click OK, then press Esc (escape) on your keyboard, and then click OK again (maybe even twice). You will know that the change has been accepted when the little black vertical bar next to the changed number goes away. In the future I'll just say "click OK". Now re-run starting from the new initial condition by clicking on a window to make it active, clicking Erase (or typing "e"), and then typing "ig".

Now we'll change a parameter value ν, the input to the ATP equation. To make change, click on the Param button at the top of the screen and change ν from 10 to 1, followed by Ok. Run the simulation starting with the original initial conditions (Default). What happens? What is the value of ATP at the end of the simulation? To find out, click the middle button on the mouse then the coordinates show up in the bottom left corner of the main XPP window. Just locate the cursor over the curve and click. If you don't have a middle button then maybe it is time to consider investing in a mouse with one!

One feature that is sometimes helpful is the ability to freeze curves. This will keep them on the screen even after you have erased everything else. Let's see how this can

be useful. First, freeze the ATP time course that you just generated. Click Graphic Stuff, then Freeze, and then again Freeze (or type "gff"). This brings up a box that allows you to enter the color of your frozen curve. This is coded by integers, so just type an integer other than 0 and your curve will be frozen with some color, but you have to type "r" (or click Restore) to see it. Next set $\nu = 10$ and re-run. This will trace out a new curve with an oscillatory trajectory. Freeze it with a different color. Do it again with $\nu = 20$ and again freeze the new curve. How to get rid of frozen curves? Click Graphic stuff, then Freeze and then Delete. This allows you to remove one frozen curve at a time.

Printing out your work

There are two ways you can print out a solution curve. One is to click Graphic stuff and then Postscript. This opens a window that allows you to name the postscript file and put it in the directory of your choice. Another window comes up first, just click Ok here. On the second window that pops up click on the home icon and you will then see your directories.

The second option is to export the data itself, rather than a picture of the graph. This is much more flexible since it allows you to make a figure using any graphing program that you wish to use, so the figures will always look better than with the first option. To do this, click on the Data button at the top of the screen. This brings up the Data Viewer, which shows columns of numbers for the time, variable(s), and auxiliary variables. These columns of numbers are what get saved when you then click on Write. This brings up another box that allows you to save the data in the directory of your choice.

You can make some nice graphs with Matlab. I have put two M-files on my web site (under Course Software) that are templates for making figures with Matlab. The first one (**plot_single.m**) reads in two columns of data from the data file *data.dat*. You can modify the M-file to read in whichever columns you are interested in plotting. The second one (**plot_double.m**) sets up two panels, one on top of the other. You can plot two columns of data in one panel and two others in the other.

Quitting XPP

This single command gets its own heading since it is so important (almost as important as Bell Off). To quit XPP elegantly click File Quit Yes, or simply type "fqy".

Making bifurcation diagrams

While still in an active XPP session, click on File and Auto. This brings up the Auto software. In this window, click on Parameters to enter the bifurcation parameter (this is Par1, which is already the parameter we want). Click Ok. Then click Axes and hI-lo. Currently ATP is on the y-axis and ν (the bifurcation parameter) is on

the x-axis. You can adjust the dimensions of the bifurcation axes in this window. Set Xmax=70. Click Ok. Next click on Numerics. In this window you can adjust some of the numerical parameters used by Auto to make the bifurcation diagram. The important ones to know are Nmax, NPr, Ds, Dsmax, Par Min, and Par Max. The diagram is made by solving a boundary value differential equation with variable step size. Nmax is the maximum number of steps to take, NPr tells Auto how often to print a label (NPr=50 means put a label after each 50 steps), Ds is the initial step size, and Dsmax is the maximum allowed step size. Par Min is the minimum value for the bifurcation parameter and Par Max is the maximum value. Set Nmax=500, NPr=550, Par min=0, and Par max=70. Use the default values for Ds and Dsmax. [Note: Auto will put labels at every bifurcation point it finds, so setting NPr > Nmax we ensure that any label will be an endpoint of the branch or a bifurcation point.]

We are now ready to generate the bifurcation diagram. This must start from an equilibrium point, so in the Parameter window of XPP (not Auto) set $\nu = 0$ and type "il" several times, so that the system becomes very close to the stable equilibrium. This point will automatically be fed into Auto, and can serve as the starting point for the bifurcation diagram. Back in the Auto window click on Run and Steady State. A bifurcation diagram of steady states, a *stationary branch*, should be drawn out, with endpoints and bifurcation points labelled. [If this does not work it probably means the initial point was not sufficiently close to the stable equilibrium, so go back to the XPP Parameter window, set $\nu = 0$ and try again.] The portion of the bifurcation diagram representing stable equilibria is in bold, the portion representing unstable equilibria is thin.

You can trace out the periodic branch of the bifurcation diagram by clicking Grab, then pressing Tab, then pressing Return. This Tab will move you from one labelled point to the next. The periodic branch starts at the Hopf bifurcation with label 2, so one Tab should get you there (information on the labelled point will show up at the bottom of the Auto window). Now click Run and Periodics. The periodic branch should emerge, connecting up with the Hopf bifurcation at label 3. You can print out the full bifurcation diagram by clicking File and Postscript. This will generate a postscript file that you can then print.

Next, try generating a bifurcation diagram with η as the bifurcation parameter (start with $\eta = 1$). Set $\nu = 10$. [To remove the old bifurcation diagram click File and reset diagram.] Note that there should be two Hopf bifurcations. Grab one of these and generate a periodic branch.