

SIAM WORKSHOP:

XPP

Software for Simulating Differential Equations

Ian Price
peccavo@gmail.com
18 October, 2009

Introduction

XPP came into existence for the analysis of phase planes when studying differential equations. It was later linked to the bifurcation program AUTO. Libraries have been appended in successive upgrades to include various utilities. With an easy to use interface and intuitive files, XPP is ideal for investigating differential equations.

XPP is written and maintained by G. Bard Ermentrout. He has several tutorials and documents by himself and others on his website:

<http://www.math.pitt.edu/~bard/xpp/xpp.html>

Installing XPP

LINUX

This is the easiest option. Either install from a repository (under the name xppaut) or download binary from:
<http://www.math.pitt.edu/~bard/bardware/binary/>

MAC

Also very easy. Get the binary from the website above.
xppaut5.99-macosxintel.tgz or xppaut5.99-macppc.tgz

WINDOWS

Instruction are on:
<http://www.math.pitt.edu/~bard/xpp/ezwin.html>

Getting Started

- Log on to your computer.
- Locate the icon that looks like a computer screen on the bottom toolbar. Click it.
- A window with a command line will open on your computer. This is your terminal. It isn't necessary to run XPP, but it's very useful.

Writing an ode File

The first step is to write a text file with the extension .ode. Our first example will be fisher.ode. Any file editor will work, e.g. emacs, vi, notepad, etc. In the terminal type:

```
emacs fisher.ode &
```

An editing window will now appear in which we can write and save our ode file.

After we write the file, we can launch the file through the GUI, or by launching xppaut and browsing for the file.

First Example File

```
# Comment Line
```

```
# Equations u and v
```

```
u' = v
```

```
v' = -c*v + u*(u^2 - 1)
```

```
# Assign the parameter c
```

```
par c = 2
```

```
# Assign initial condition
```

```
init u=1, v=0
```

```
done
```

Launching XPP

First, save your ode file
Hold CTRL and type X then S.
In the terminal window now type:

```
xppaut fisher.ode
```

A new window should now appear, with a white tool bar to the left and a black display screen.

Using the Interface

Use the mouse to click on the blue boxes up top that read ICs and Param. These are your initial conditions and parameters. You can adjust them from here. Commands can be handled from the left tool bar, or by using hot keys which will be written in parentheses () as we use them.

Run the simulation by using the mouse to click on Initialconds > Go (I G)

Fit the results to your screen by pressing Window/zoom > Fit (W F)

You should see a straight line.

Simple Adjustments

In the Initial Data window, change U to .99
Use the mouse to press Ok, then Go.

What do you see?

Click on the main window and press
Window>Fit (W F) to see everything.

Use the Par window now to change c to .5

What happens now?

Try different values of c using the window.

The Phase Plane

(V 2) Viewaxes > 2D will give you the option of viewing the phase plane. Fill in the chart as below.
Note that the 'Enter' key moves you between fields and the 'tab' key finishes the process.

*X-axis:U	Xmax:2
*Y-axis:V	Ymax:.5
Xmin:-2	Xlabel:U
Ymin:-.5	Ylabel:V

OK Cancel

Nullclines

In your initial data and parameter windows, press 'Default'
This restores your settings to those used in your ode file.

Now click on your main window again, which should say
'V vs U', and press:
(N N) Nullcline New

Green and red lines should appear. These are graphs of
your equations set to zero, i.e. the nullclines.
Where the nullclines intersect are your singular points.

Trajectories

To use the mouse to test an initial condition, type (I M)
Initialcond > Mouse, then click on the screen. To do
this multiple times in succession, type (I I), Initialcond
> Mice. To stop, press Esc.

If a trajectory goes out of range, press Enter to continue.

Press Esc. To clear the graph, press (E) Erase.

Press (N N) to recover the nullclines. To see the flow of
the system, press (D D) Dir.field/flow > Direct Field.
A grid of 10 works effectively for this system.

Singular Points

Where the green and red lines intersect are your singular points. To determine the quality of a singular point, press (S M) Sing pts > Mouse then use the mouse to click on a given intersection. When it asks to print eigenvalues, press yes. Now look at your terminal screen: numerical approximations of your eigenvalues have appeared! (I told you the terminal was useful). If you choose to draw invariant sets, the stable trajectories will appear in blue, and the unstable in yellow by default. Press 'Enter' or 'Esc' to continue after drawing the invariant sets. On the phase plane, stable fixed points appear as circles, unstable as squares, and saddles as triangles.

You should be able to count three fixed points on the screen. But suppose this was not given, how can we find all singular points? Press (S C) to search randomly. Press Enter for all the default chooses to start. When finished, look at the terminal again.

Save Settings

By now we have made various changes to the screen setup, parameters, and initial conditions. All these changes can be saved as set files.

Press (F W) File > Write Set to save your settings as a set file. You can name the set file whatever you like as long as it ends in .set

You can load the set file in the future by pressing (F R) File > Read Set.

Graphics

(G P) will create a postscript (.ps) files that you can print.

An alternative is to export the data and open it with another program. (G O) will create a dat file. Call it fisher.dat
Note this will only save your current trajectory.

Go to your terminal and type Ctrl+Shift+T to open another terminal. In the new terminal, type: matlab.

Once matlab opens, type:
A=dlmread('fisher.dat', '');

Note that there's a single space between the single apostrophes.

Now use the command:
plot(A(:,1), A(:,2))

To close matlab, type: exit.

Comparing Graphs - Freeze

Go back to the XPP window. To see a time course of a single trajectory, press (X) Xi vs t and then type U in the display window. A single trajectory appears.

Type (G F F) to freeze this graph. When the window appears, change the color from 0 (white) to 1 (red) so it stands out, then click OK.

Now change the parameter or initial conditions, run it-- the old graphs are still there and you can compare the change.

Comparing Graphs – More Windows!

To compare your current trajectory to another component of the system, type (M C) Makewindow > Create. An identical window will appear. Press (X) and then delete the U written in the window and type V then 'Enter'.

Recall, typing (W F) will reset the window to a viewable range. You can also compare the U trajectory to the phase plane by typing (V 2), and confirming the previous settings. (M K Y) Makewindow > Kill All > Yes will close the extra windows.

Quitting XPP

Say it's time to close shop. Save your save your set file as before (F W), export your data (G O), and your postscript (G P).

(F Q Y) File > Quit > Yes will close XPP. The little box with an X in the top right corner is just for decoration.

Example 2 -- brusselator.ode

To write the new file, follow the following instructions:

In your text editor, hold CTRL and type x f.

Type brusselator.ode and press ENTER. You should see this happening in the bottom of your window.

Type the file as seen in the right hand column. Then, hold CTRL and type x s.

Now in your terminal, launch xpp by typing:

xppaut brusselator.ode

```
# brusselator.ode
# 17 Oct, 2009
x' = a + x^2*y - (b+1)*x
y' = b*x - x^2*y

par a=1, b=1
init x=1, y=1

# Settings. You need a space after the @
@ total=60

# allocate memory, set bounds and time step
@ maxstor=500000
@ bounds=1e7
@ dt=.05

# set up the initial screen
@ xp=t, yp=x, xlo=0, xhi=60, ylo=0, yhi=5

done
```

Phase Plane and Parameters

- You can view the trajectory by typing (I G) as before.
- View the phase plane by typing (V 2) then entering X and Y between 0 and 5.
- Type (N N) to view the nullclines.
- To get information on the singular point, type (S M) and click your mouse on the intersection of the nullclines. It should be stable.
- Click on the blue Param box and change b to 2.5
- Press (E) in the main window to erase and redraw the nullclines (N N). Type (S M) then click the mouse on the intersection of the nullclines. It should be unstable.
- What happened?

Bifurcation Analysis-- AUTO

Here, we attempt to explain the change in the quality of the fixed point through numerical bifurcation analysis.

- Change b back to 1. It's easiest to start AUTO from a stable fixed point.
- Open the ICs window. We'll be watching this.
- In the main window press (I G). From here, press (S G) to converge to the singular point.
- Another method is to iteratively type (I L) Initialcond> Last until the numbers in the ICs window have a 5 decimal tolerance.
- Once you're sure you've converged on the fixed point, type (F A) File > Auto, and another window will launch. It's AUTO man!

AUTO Settings

Whenever you launch AUTO, you will want to handle the first three items in the left hand column: Parameter, Axes, Numerics, in that sequence before hitting Run.

- Parameter: Change par1 to b and par2 to a, since we want to work with parameter b.
- Axes: Click on hi-lo
- Set Xmin (parameter value) and Ymin to -1.
- Set Xmax and Ymax to 5. Then press OK, or tab.
- Note, like XPP, pressing ENTER lets you cycle through options, while pressing TAB accepts changes and closes the window.

Auto Settings

Fill in the numerics settings as seen on the right.

Ntst controls how many mesh points AUTO uses.

Nmax is the max.

NPr controls how often AUTO randomly samples.

Par Min should match Xmin and Par Max should match Xmax.

Press OK

Ntst:20	Dsmax:0.1
Nmax:200	Par Min:-1
NPr:30	Par Max:5
Ds:0.01	Norm Min:0
Dsmin:0.001	Norm Max:1000
Ncol:4	EPSU:0.0001
EPSSL:0.0001	EPSS:0.0001

Ok Cancel

AUTO!

Everything is set. Press RUN > Steady State and hope for the best.

A thick line should appear that changes to a thin line. Where this change occurs is our bifurcation. Click Grab. You can either use your arrow keys or the tab button. Click the tab button until you land on the point, labeled HB for Hopf bifurcation. When you land on it, press ENTER. Now, hit RUN > Periodic. Solid dots representing stable limit cycles now appear. Click grab and play around with the window.

AUTO!

But suppose we're dealing with a decreasing parameter rather than increasing. We can handle that quite easily. Go to numerics, and change Ds to -.01. Click on Grab, and get to the point labeled 1. Press Enter. Now hit Run > Steady State again. The solid line will continue to the left now. Unfortunately, there are no interesting dynamics.

If you want to change the parameter from XPP, go to File > Clear Grab. Then head back to XPP and change the parameters there. You'll see the parameters will be set to where they were last in AUTO. You can find a fixed point or limit cycle here and examine the new fixed point in AUTO.

Saving AUTO

There are three main ways of saving your AUTO settings.

- 1) Postscript – this creates a graph you can print out.
- 2) Write pts – this creates a data file (basically a spread sheet) that can be read into matlab or excel.
- 3) Load diagram – this creates a file that can only be read by AUTO that you may want to use at some later point

Numerics

Close AUTO and return to XPP.
Change the window to view X by pressing (X), make sure X in the top window and hit Enter.
Make sure the Param window is open.
Change b to 5. Ok. Go. What do you see?
Change b to 8. Ok. Go. How about now?
You can refine the mesh by going pressing (U D)
nUmeric > Dt and changing to 0.01. Hit ENTER, then ESC to return to the main menu. It should converge now.
Change b to 10. Ok. Go. Is decreasing our Dt alone going to be enough as we increase b?

Numerics - CVODE

XPP uses the Runge-Kutta Method by default. Our system is obviously stiff-- that is it changes very slowly in some places and very quickly in others, so we need a stiff solver.

Type (U M C) nUmeric > Method > Ccode to use Ccode. Use the settings:

Relative tol: 1e-8

Abs. Toler: 1e-8

Banded: 0

Press ESC to get back to the main menu and watch it run!

To hard code ccode, go to your editor window and type:
@ method=ccode, tol=1e-8, atol=1e-8

Optional: Parameter Fitting

Parameter fitting is a challenge to all modelers. While XPP is not the ideal setting for parameter fitting, it does have a built in Levenberg-Marquadt solver that works well for a small number of parameters. This, however, requires the use of data file with equally spaced points.

You can find a sample file on the website. This data was taken from fisher.ode and had noise added. Let's see how good a solution we can find.

Parameter Fitting

Close your current XPP window by typing (F Q Y).
Make sure your data file, pfit.dat, is in your directory by typing ls then ENTER.

Type xppaut fisher.ode to launch XPP.
Type (U H I) nUmeric > stocHastic > flt data to get to the correct screen.

Enter the window as seen on the next slide and hit OK.
The answer will be in your terminal.
Take a look at the data file pfit.dat and the way we fill out the screen to try to get the logic down.

Parameter Fitting

File:pfit.dat	NCols:2
Fitvar:U	To Col:2
Params:c	Params:
Tolerance:0.0001	Epsilon:1e-06
Npts:3	Max iter:40

Ok Cancel

Questions?