WinPP for Neophytes

1 Introduction

This is a very brief introduction to get you started using WinPP. There is considerably more information in the manual (Readme.html) which is included in the folder containing the program. You should begin by running some of the sample files and experimenting with them. The best way to become proficient in writing your own programs is to imitate the examples given in the files in the manual and in this introduction and paying attention to the comments.

WinPP is a program designed to solve a variety of kinds of dynamical systems. We will consider only a few of these and will restrict our attention to

(i) first order ordinary differential equations,

(ii) first order difference equations (discrete dynamical systems),

(iii) first order differential–difference equations,

(iv) systems of two first order ordinary differential equations.

The basic idea is to create an input file with any text editor. As input files are quite short and simple, Notepad is a convenient editor for this purpose. It is also possible to define plot parameters such as the size of the plotting window and the interval over which the problem is to be solved. This input file gives WinPP the functions and equations, parameters, and initial data to be used. Parameters, initial data, plotting variables and integration variables may also be changed while the program is open.

To start WinPP, simply click on its icon and then use the file selector window which opens to choose the input file. In addition to the main plotting window, several other windows will open for initial conditions, boundary conditions, and parameters. There is also a browser window, which stores numerical data. You will probably have little need for the boundary condition and browser windows initially and it is convenient to minimize them to get them out of the way. It is not possible to close them except by exiting from the program. Since you may wish to change parameters and initial conditions, it is convenient to move these two windows to the bottom of the screen and resize the main window so that these three windows are visible.

2 Creating input files

Input files for WinPP have the extension .ODE. A line which begins with the symbol # is a comment line. There should be no spaces between numbers and equals signs in definitions of parameters or initial values. An input file should contain the following elements:

(1) a specification of the equation, with the form $dx/dt = f(x)$ for an ordinary differential equation, and $x(t+1) = f(x)$ for a difference equation. In a differential - difference equation, the form $f(delay(x, \tau)$ is used to include a term $f(x(t-\tau))$. For a system of differential equations there would be two equations, preferably on separate lines, one for each variable. Functions may be entered into equations directly or a line defining a function can be used, such as $f(x) = exp(-x)$, and then the the name of the function can be used in the specification of the equation.

(2) a line declarating parameters, beginning with the word parameter (or the abbreviation p). This line should include values for each parameter, which serves as a default value.

(3) a line giving a default initial value, which may be in the form $x(0) =$, or the form $initx =$.

(4) (optional) a line declaring values for some of the plotting values, such as total (for the length of the interval on which the solution is to be calculated)

(5) the last line of the file must read "done".

There are certain reserved words, including sin, cos, exp, log, ln, pi, which should not be used except with a standard meaning. A complete list of reserved words may be found in the manual.

The program creates a file called WPP.LOG whenever WinPP is run. It contains error messages, such as messages to tell you that there is something wrong with an input file so that the file may be editted.

3 Running WinPP

Once you have opened the program and an input file and have placed the windows to your satisfaction, click RUN and then GO. You can change parameters by editting the parameter window and clicking on OK, and then run the program. Alternately, you can edit the parameter window and click on GO. You will obtain the new plot in the same window with any previous plots; to obtain just the new plot, click GRAPHICS and then ERASE. Initial values can be changed in the same way.

Other options on the RUN menu include LAST, which runs the program again using the final value as initial value, MORE TIME, which allows integration over a longer interval (but does not automatically change the window to show the extended solution, and EQUILIBRIA, which can be used to approximate equilibria. To change integration parameters, click NUMERICS and then INT PARS. Integration parameters include TOTAL (the interval for the independent variable over which the solution is to be calculated), choices for the integration method and error tolerances. Delta T is the step size for fixed step integration methods and the output for adaptive integration methods. TSTART is the starting time (default value 0). MAXDELAY tells the program how much time to allocate for delay equations. The default value is 0, which is fine exept for differential–difference equations where the delay should be no greater than MAXDELAY. To change any of the parameters, click on its box and edit.

The GRAPHICS menu sets up the display. VIEW gives a submenu for choosing between two– and three– dimensional plots (for systems). In a two–dimensional plot, the variables XLO, XHI, YLO, and YHI define the limits of the graph. For the examples described here, you will almost always use a two–dimensional plot. For first order equations of any kind, this will be automatic, but for a sytem of ordinary differential equations you will need to choose which variables to plot from the VIEW menu. This is done by editting the first line of the VIEW PARAMETERS menu to give the variables on each axis. You can plot x vs. t, y vs. t, or y vs. x. For a two–dimensional system, it is better to set the view parameters from the GRAPHICS menu, rather than in the input file.

In a three dimensional plot, the limits are determined by XLO, XHI, YLO, YHI, ZLO, ZHI. There is also an option to let the program determine these limits, and it is recommended that you choose this option for three–dimensional plots. For a problem in which the variables are necessarily positive, such as a population model, set the minimum value to zero. Then in a harvested model, where it is possible for a value to reach zero in finite time, this is indicated by a message on the graph that the variable went out of bounds at a specified time.

To draw different curves in the same window, such as x and y as functions of t in a two–dimensional system, click on GRAPHICS, then ADD/EDIT. Choose which variables to plot (making sure that the low and high values of each of the variables you are plotting are the same) scrolling through the curve list using ADD NEW. When all the desired curves are entered, click DONE and then RUN and GO.

For two–dimensional systems, the PHASE PLANE menu includes the options NULLCLINES, which draws the null clines (red for the first variable and green for the second variable), DIRECT FLD, which draws a direction field of line elements, and FLOW, which draws a collection of orbits. It is possible to use the choice GRID PARAMETERS to set the grid for these.

Warning: Difference equations are treated in the form $x(t+1) = f(x(t))$ and the result is plotted for all t, rather than being treated in the form $x_{n+1} = f(x_n)$ and plotted only for integer values n, possibly with straight line segments joining the plotted points.

To edit the right hand side of the equation, click on FILE and then EDIT RHS. To leave the program click FILE and then EXIT. To produce a print-out click on FILE and then PRINT. This creates a PostScript file which may be viewed and printed using GhostView, or may be sent directly to a PostScript printer for printing. It seems that the PostScript files are always rotated through 90 degrees, but Ghostview has an option for changing the orientation. However, if you want to insert a figure in a TeX or LaTeX file, you will need to include instructions to rotate. You may also click on FILE and then COPY to copy the current graphics window to the clipboard for pasting into another document.

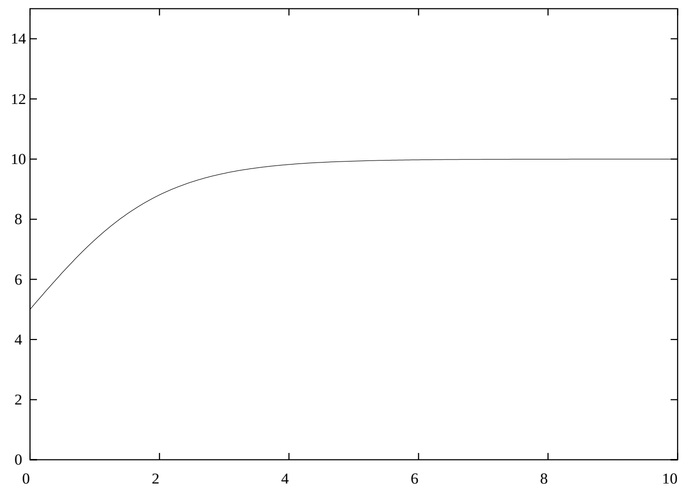
4 Examples

Here are four examples, with input files, graphic output for the default settings of parameters and initial values, some comments, and some suggestions for other things to try with each example.

I Logistic

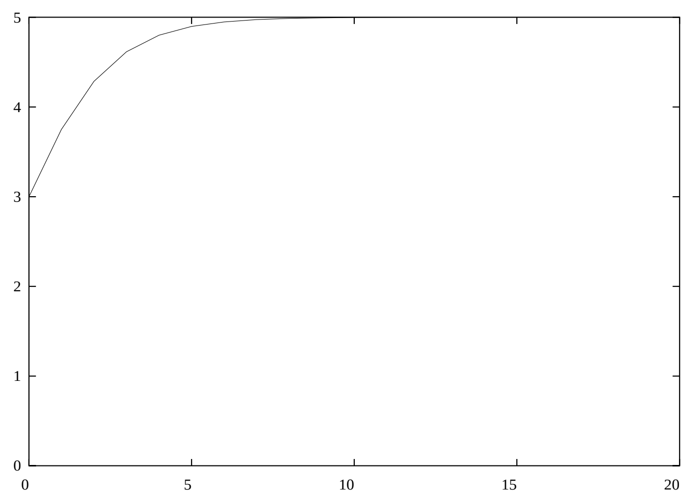
```
#logistic
dx/dt=r*x*(1-x/K)-H
par r=1,K=10,H=0
x(0)=5
@XLO=0,XHIGH=10,YLO=0,YHI=15
done
```

This is a first order ordinary differential equation. You should practice with this example by varying the parameters r, K, and H, rescaling the plotting window if necessary. A negative value of H corresponds to a stocking of the system; try some negative values. Also, plot solutions with several initial values between 0 and 15 in the same figure.



II Verhulst

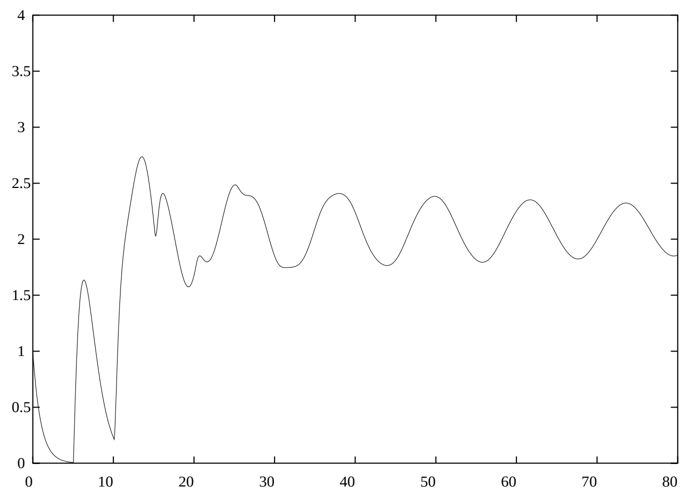
```
#Verhulst
x(t+1)=r*x/(x+A)-H
par r=10,A=5,H=0
x(0)=3
@XLO=0,YLO=0,YHI=8
done
```



This is a difference equation. You should make the same kind of variations as for the previous example. To make the problem look more like a difference equation, you should set Delta T=1 in the NUMERICS–INT PARS menu.

III Nisbetgurney

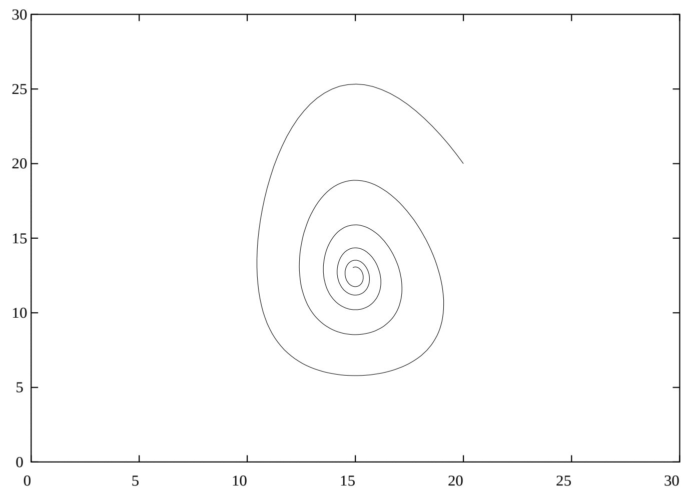
```
#nisbetgurney
f(x)=x*exp(-x/A)
dx/dt=r*f(delay(x,tau))-d*x-H
par A=0, r=8,d=1,tau=5,H=0
x(0)=1
@XLO=0,XHI=80,YLO=0,YHI=4,delay=20,total=80
done
```



This is a differential–difference equation representing a delayed–recruitment population model. Note the way a delayed value is entered into the file. You should try varying the delay τ. Also, take several values of H increasing from zero and see what happens to the solution. Determine the largest value of H for which the solution remains positive for all time.

IV Predprey

```
#predprey
f(x)=r*x*(1-x/K)
g(x)=x/(x+A)
dx/dt=f(x)-y*g(x)
dy/dt=s*y*A*(g(x)-g(J))-H
par r=1,s=1,K=30,A=10,J=15,H=0
init x=20,y=20
done
```



This is a two–dimensional system representing a predator–prey population system. The display is of the phase plane; you should plot each of the variables as a function of time as well. Remember to set the range of each variable in the GRAPHICS/VIEW menu before running each plot. Change the value of J to 5 and see whether the phase plane plot or the plot of individual variables against gives you more information. Try changing J to 40. Also use the PHASEPLANE menu to plot nullclines and the flow. For each of the values 15 and 5 for J, increase H from zero to the value for which the y–variable goes out of bounds, signalling the extinction of the y (predator) population.

5 Exercise

Write a program to handle the“delay–logistic” equation

$$x'(t) = rx(t)\big(1 - \frac{x(t-\tau)}{K}\big) - H$$