Lab 2: Introducing XPPAUT

In biological applications it is quite rare that the solutions of the appropriate differential equations can be obtained using paper and pencil. Thus we typically need to use computer programs to integrate these equations and to use as tools for analyzing their behaviors. In this course we use two types of computer programs: Python, a programming language, and XPPAUT a programming package that is very well suited for the analysis of many of the problems that arise in biology. Note that it is now possible to run XPPAUT within the Python ecosystem (Appendix D); however, we will not do this in the lab.

XPPAUT is a ‘quick and easy’ to use programming package for investigating the differential equations [?]. An advantage of XPPAUT is that it can be used to integrate discrete time systems, ordinary and delay differential equations, stochastic differential equations and partial differential equations. In addition, XPPAUT is particularly well suited for the investigation of the dynamics of excitable systems composed of neurons and cardiac cells. Indeed, it is not uncommon that XPPAUT programs can be located on the Internet for in the Supplemental materials for published papers on journal websites\(^1\). For example, the program we will use in Lab 6 to study the Hodgkin-Huxley equation, hh.ode, can readily be found on the Internet. Can you locate this file? Hint: the program was developed by Micheal Guevara at McGill University. Is it the same program that we have on our website!

1 Background

The lecture introduces the use of XPPAUT for the analysis of linear differential equations. It is very easy to extend the use of XPPAUT to the study of nonlinear differential equations. On the XPPAUT website check

---

\(^{1}\)Typically it is necessary to access these materials through the journal website.
Online Documentation
then check out
Writing ODE files
Reserved words & functions
Common commands

**Browser use:** All of the information needed to program in XPPAUT can be obtained from the XPPAUT website. It is often possible to obtain additional help regarding the use of specific XPPAUT commands by typing, for example, in Google `xppaut heav`. Finally, it is often possible to locate programs on the Internet that can be used.

**Housekeeping:** Since XPPAUT file names have the form `my_name.ode` it is useful to make a directory in the home directory called `mode` (in other words, my ode files). Then we can make subdirectories in `mode` which contain XPPAUT programs used for specific purposes, for example, for Lab 2. This will be useful when we require these programs for other purposes. An advantage of putting all of the data files and other materials necessary to run a program in the same subdirectory, then we don’t need to specify the directory paths in the program since, by default, the computer program always searches for files in the directory in which it is running.

Make sure that you are in your home directory (type `pwd`), then we perform the following steps

```
mkdir mode
cd mode
mkdir Lab2
cd Lab2
```

## 2 Exercise 1: First XPPAUT program

We illustrate the use of XPPAUT by using it to integrate the first–order ordinary differential equation

\[
\frac{dx}{dt} = -kx
\]  

(1)

Use a text editor, for example `emacs`, to create the file `first_ode.ode`. We remind the reader that it is useful to create this file in the directory `Lab2`. All XPPAUT files have the form
# Enter the equation (no spaces)
dx/dt=k*x
# Enter the initial conditions
init x=1
# Enter the parameter values
par k=-0.1
# Change default settings
@ TOTAL=20,dt=0.05,xlo=0,xhi=100,ylo=0,yhi=1.5,maxstor=100000
done

It should be noted that lines that begin with # are comments and are not acted upon by XPPAUT. The purpose of these lines is to make the code more readable for a user. However, you could simply write the program as

dx/dt=k*x
init x=1
par k=-0.1
@ TOTAL=20,dt=0.05,xlo=0,xhi=100,ylo=0,yhi=1.5,maxstor=100000
done

Useful hints:

1. TOTAL refers to the total time for the integration and dt is the step size for the numerical integration. Thus if TOTAL=20 and dt=0.05, there will be $20 \times 20 = 400$ time steps. The parameters xlo, xhi, ylo, yhi refer, respectively, to the dimensions of the x–axis and y–axis of the figure displayed by XPPAUT and not to the variables that appear in the differential equation. The parameter maxstor sets the total number of time steps that will be kept in memory (default value is 5000).

2. Likely the most common mistake made using XPPAUT is to add spaces to the entered equation to “make it look better”. Unfortunately the computer is not impressed! Thus do not add spaces when you type in the equation. A good rule of thumb is that when in doubt, no spaces!

3. The values of the initial conditions and parameters can readily be changed using the on–screen menus provided by XPPAUT. In other words, it really does not matter which values you type into the *.ode program since you can easily change them later.
4. The default integrator for an ordinary differential equation is a 4th-order Runge–Kutta algorithm. This is a common integrator used to study the differential equations that arise in mathematical biology. There are two ways that the integrator can be changed: 1) on the XPPAUT start-up menu, click Numerics, then Method; and 2) in the change default settings part of the ode program type, for example, meth=euler. One of the very useful features of XPPAUT is the ease by which it is possible to compare the effects of different numerical integrators and their parameters on the simulation without writing new programs.

**Running an XPPAUT program:** The XPPAUT program is activated by typing the command

```
xppaut first_ode.ode &
```

In a Linux environment the & is added since we do not want the whole CPU to be involved in doing this calculation. In other words, by adding the & we make it possible to do other computations at the same time.

Now press ENTER. The computer screen will look like that shown in Figure 1.
Six additional windows can be opened (ICs, BCs, Delay, Param, Eqns, Data) by clicking on the appropriate icons. Since our goal is to integrate an ordinary differential equation we need to open the menus ICs and Param. Click on these icons and use your mouse to arrange the computer screen so that it looks like that shown in Figure 2.

To run the program click ‘OK’ in the ‘Initial Data’ window and ‘Parameters’ window and then ‘Go’ in the ‘Parameters’ window. The solution appears in the main window as it is computed (see Figure 3).

Questions to be answered:

1. In class we showed that the solution of (1) decreases exponentially when \( k < 0 \) and increases exponentially when \( k > 0 \). Do your simulations agree with this expectation?

2. The mathematical solution of (1) is \( x(0) \exp(kt) \), where \( x(0) \) denotes the initial value (i.e., the number that appears in the IC window). Explain how can you verify that the numerically estimated solution has this form (Hint: what is the \( 1/e \) time)?
Figure 3: The X versus T plot was produced by first clicking on ‘OK’ on the ICs panel and then ‘Go’ on the Parem panel.

3 Exercise 2: Second-order DDEs

In order to use XPPAUT to integrate a second (and higher) order differential equation it is necessary to rewrite the differential equation as a system of two (2) first-order differential equations, i.e.

\[
\begin{align*}
\frac{dx}{dt} &= f(x, y) \\
\frac{dy}{dt} &= g(x, y)
\end{align*}
\]

where \( f \) and \( g \) are functions of \( x \) and \( y \).

Questions to be answered:

1. Write the second–order differential equation

\[
\frac{d^2 x}{dt^2} + b \frac{dx}{dt} + cx = 0
\]

as a system of two first–order differential equations.

2. Write an XPPAUT to integrate this second–order differential equation. Call this program \texttt{second_ode.ode} and put this file into the directory \texttt{\mode\Lab2}. A screen shot of the program we wrote is shown in Figure 4.
3. What is the characteristic equation and what are its roots?

4. Determine values of the parameters $b, c$ so that both roots are negative real numbers, complex numbers with negative real parts, and a pair of purely imaginary numbers. In each case use your XPPAUT program to compute the solutions. Are the solutions qualitatively the same or do they differ? If they differ, then what are the differences?

4 Exercise 3: Phase plane representations

Up to now we have plotted $x$ or $y$ versus time. In these plots, time appears explicitly (namely, it is the x–axis). However, there is another way to plot the data, namely $x$ versus $y$. This type of plot is referred to as a phase plane plot and is characterized by the fact that time does not appear explicitly.

After running XPPAUT, click on Viewaxes, then 2D (see Figure 4). The
Figure 5: The phase plane solution in the right panel was produced by clicking on the ‘OK’ Tab on the ‘2D View’ shown in Figure 4.

table that opens identifies the x–axis of the plot as time and the y–axis as the value of x. The ranges are those we entered in the last line of the program second_ode.ode. To obtain the phase plane representation, change the values so that y is plotted on the y–axis and x is plotted on the x–axis. You will also need to adjust the ranges in the two axis so that they span both positive and negative numbers, e.g.

\[ x_{lo}=-5, x_{hi}=5, y_{lo}=-5, y_{hi}=5 \]

Press ‘OK’ to obtain the phase plane representation (Figure 5).

1. For the values of \( b, c \) determined above, determine the phase plane for a pair of negative real numbers, a pair of complex numbers with negative real part and a pair of purely imaginary numbers. In our lectures we will describe the shape of the phase plane representations as a spiral, a focus and a center. Can you guess which pair of eigenvalues corresponds to which type of phase plane representation?

**Problem:**

In lecture we discussed the conditions for steady state for the consecutive reaction

\[
A \xrightarrow{k_1} B \xrightarrow{k_2} C, \quad (3)
\]
1. Write down the differential equations that describe this consecutive reaction as a system of three first-order differential equations.

2. Write a XPPAUT program to integrate these equations. Do the time courses of A, B, and C resemble those we discussed in Chapter 3 on *Mathematics as a Laboratory Tool*?

3. Show that as the ratio \( k_2/k_1 \) becomes larger, the steady state condition for B becomes more reasonable.

4. Keep track of the values of \( k_1, k_2 \) you used in your simulations. In Lab 3 we will show you how to write a Python program with *matplotlib* to make a figure. At that time we will ask you to produce a figure that shows these results. Note that you can save yourself some time by clicking on Data, then Write to save the data file after you run each simulation.

**Deliverable:**

Use *Lab2_template.tex* to prepare the lab assignment.

**Assignment hints:**

- A useful LaTeX environment for writing up your lab assignment is the *verbatim environment*, namely `\begin{verbatim} ... \end{verbatim}`. Whatever is placed within this environment appears in the LaTeX document as written.

- You can save a figure in XPPAUT by clicking on Graphic stuff, then Postscript. This makes a *.ps* figure which is likely not going to be much use to you since we can’t use such a figure in LaTeX and we don’t have a postscript printer. A very powerful software package for dealing with figures is called *ghostscript*. It is freely available and can be easily downloaded and installed from the Internet on your own computer. Using this software package it is possible to convert a *.ps* figure to a *.pdf* figure, by typing the command

  `ps2pdf name.ps`
References