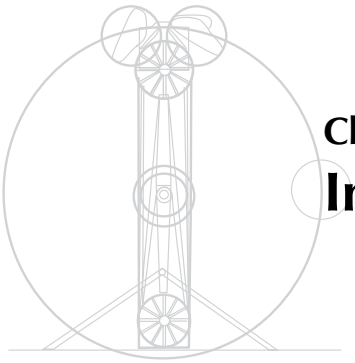

Chapter 1
Installation

Like all Holmes's reasoning the thing seemed simplicity itself when it was once explained.
—Sir Arthur Conan Doyle, *The Adventure of the Stockbroker's Clerk*, The Strand Magazine, March 1893.

Installation of *XPPAUT* is done either by downloading the source code and compiling it or downloading one of the binary versions. I will give sample installations for UNIX®, Windows®, and Mac OS X operating systems. If you are totally clueless at compiling source code, it is best to either have your system administrator install it for you or download a precompiled binary for your computer. There are compiled versions available for Linux®, SUN™, HP™, Windows, and Mac OS X operating systems.

1.1 Installation on UNIX

1.1.1 Installation from the source code

Create a directory called `xppaut` and change to this directory by typing:

```
mkdir xppaut
cd xppaut
```

Step 1. Download the compressed tarred source code `xppaut_latest.tar.gz` into this directory from one of the two following URLs:

- http://www.math.pitt.edu/~bard/xpp/xpp.html
- http://www.cnbc.cmu.edu/~bard/files.html

Step 2. Uncompress and untar the archive:

```
gunzip xppaut_latest.tar.gz
tar xf xppaut_latest.tar
```

This will create a series of files and subdirectories.

Step 3. Type

```
make
```

and lots of things will scroll by, including occasional warnings (that you can safely ignore). If you get no errors, then you probably have succeeded in the compilation. If the compilation stops very quickly, then you probably will have to edit the Makefile according to the architecture of your computer. Check the README file and the Makefile, both of which have suggestions for many platforms.

Step 4. If you have successfully compiled the program, then the file `xppaut` will be in your directory. To check, type

```
ls xppaut
```

If you see something like `xppaut*` listed, then you have succeeded. If you don't see this, then the compilation was unsuccessful. Consult the README file for a variety of possible fixes. Also, see the numerous comments in the Makefiles included with the package. I have not yet found a computer on which I cannot compile the program. Common problems are the wrong path to the X Window libraries, and nonexistence of `ranlib`, among others.

Step 5. Once you have compiled the program, just move the executable file to some place in your path. (The usual location is `/usr/local/bin` but you must have root privileges to do this.) *XPPAUT* needs no environment information.

1.1.2 Installation from binaries

Some binaries are available at one or both of the above URLs. You should download these binaries as well as the source code above. The source code includes many examples as well as the *XPPAUT* reference manual. Download a binary, e.g., `xppaut4.6_hpux.gz`, and uncompress it with the command `gunzip xppaut4.6_hpux.gz`, and copy it to the desired directory. You will notice that the binaries are missing things like the example files and the documentation. Download the source code to get these.

1.1.3 Additional UNIX setup

In some systems, the zooming feature and cursor movement do not always work properly. In these systems, you will want to call in *XPPAUT* with an additional command line argument, e.g.,

```
xppaut -xorfix file.ode
```

This will usually fix these problems. By default, *XPPAUT* comes up with only the main window visible. (You can always make the other windows visible by clicking on the top row of buttons on the main window.) You may want to have *XPPAUT* come up with all the windows visible; to do this, add the command line argument in `-allwin`. You can use the `alias` command in your shell to call in *XPPAUT* with these command line arguments. Alternatively, create a text file called, e.g., `myxpp`, with the following line in it:

```
/usr/local/bin/xppaut %1 %2 -allwin -xorfix
```

Save the file and make it executable by typing `chmod +x myxpp`. Now if you call in `myxpp`, it will have the two command line options enabled.

1.2 Native Version for Windows 95, 98, 2000, and Windows NT

To run the program in the Windows environment just download the program `winpp.zip` into a folder, say **wpp**, and then use WinZip® or a similar program to unzip the file. Create a shortcut to `winpp`. This version does not have all the features of the full version. Furthermore, the interface is quite different. Most of the equation files will work for this version and most of the standard features are extant. There is a binary X version for Windows that is identical to the full UNIX version and I recommend you use that instead, as this book describes the X version. (See the next section.)

1.3 X Window version on Windows

This is the recommended way to run the program in the Windows environment. It is only slightly more difficult to install. It does not use the Windows API but works identically to the UNIX version. **Note.** If you have used only an X Window emulator to log into another machine, this may be a bit of a surprise. You can run local programs which are properly compiled X Window programs right on your PC with the X emulator running. *You do not have to be on a network to run this program in the Windows environment.*

Before you download *XPPAUT* into a Windows environment, you should have the X Window emulator. There are a number of emulators available at a low cost or as demos. There are at least three that are very inexpensive, are pretty simple to install, and take up very little disk space.

WinaXe: The demo version runs for 30 minutes at a time and the full version costs $90.
URL: http://www.labf.com/index.html

X-Win32: The demo lasts for 30 days. I use this product at home. Prices range from $60 for students to $249 for corporations.
URL: http://www.starnet.com/productinfo/

MI/X: This is the smallest and has the fewest features. The demo lasts for 15 days. The cost is $25.
URL: http://www.microimages.com/

Many universities have site licenses for X servers such as Exceed™. For information on Exceed, go to http://www.hummingbird.com/products/nc/exceed

Here are the steps for installing *XPPAUT* in Windows:

Step 1. Create a folder `tmp` that will be a temporary directory. Create another folder called `xpp`.

Step 2. If you have an X Window emulator already, then skip this step. Otherwise, you should install one of the above X servers from the `tmp` directory or the desktop. I have an old version of the MI/X server available to download. If you want to try it, here is how:

- Download the two files into the `tmp` directory: `runme1st.exe` and `file000.bin`.
- Run the program `runme1st.exe` to install an X Window server onto your computer. This server needs only a few megabytes of disk space. Test the installation by clicking on the START menu and running the program found under TNT Lite.

Step 3. Download the file `xpp4win.zip` into the folder `xpp`. Unzip this with the WinZip utility. There will be a number of files including `xppaut.exe`. Note that there are two dynamic link libraries (DLLs) in the zipped file, so, if you want to move `xppaut.exe` to a different directory, you should move the DLLs there as well. To make it available everywhere, you can copy `xppaut.exe`, `cygwin1.dll`, and `libX11.dll` into your Programs directory or any other directory in your path.

Step 4. Test your download.

Step 4.A. Start your X server.

Step 4.B. Open the MS-DOS® prompt from the START menu. Change to the `xpp` directory. (In Windows 2000, this is called Command Prompt. It is available from the Start/Applications menu. If you cannot find it, click on `Run` and type in `command.com`.)

Step 4.C. Now you have to tell X where to send the display.

If you are on a network. Type `set DISPLAY=mypc:0.0` where `mypc` is the name of your PC on the network.

If you are not on a network. Type `set DISPLAY=127.0.0.1:0.0`

Note that, even on a network, the second command usually works.

Step 4.D. You are now ready to run. Type `xppaut lecar.ode` and *XPPAUT* should fire up in the X Window. If not, then check that you have started the X server and set the DISPLAY correctly. Note that, if you get the error `Can't open display`, then you should try to find out the name of your PC, as that is probably the problem. Another possibility is that your X server won't let your PC host the display. Look for something that allows you to set HOSTS in your X server and set the host to your display name.

Step 4.E. If you are successful, exit *XPPAUT* by clicking on the `File` and then the `Quit` entry and answer `Yes`.

Note. My home computer is not on a network, so I have just created a batch file `xpp.bat` and included in it a line that sets the DISPLAY for me:

```
set BROWSER=c:\Program Files\Netscape\Communicator\Program\netscape.exe
set XPPHELP=c:\xpp\help\xpphelp.html
set DISPLAY=127.0.0.1:0.0
set HOME=c:\xpp
C:\xpp\xppaut %1 %2 %3
```

Note that this also sets some other environmental variables.

1.4 Installation on Mac OS X

Installation on Macintosh computers running OS X is possible by downloading the source code for *XPPAUT* and then compiling it using the software development tools provided for the new OS. In addition, you will need to download the X development libraries to compile it. The following steps were helpfully provided to me by Chris Fall and James Sneyd. I have managed to test them on one laptop and everything seems to work. A Mac OS X binary can be found on the *XPPAUT* homepage if you don't want to compile it yourself.

1. Make sure you get and install the full Developer Kit for Mac OS X. This is how you get the cc compiler.
2. Install XFree86 on OS X. Download from

   ftp://ftp.xfree86.org/pub/XFree86/4.1.0/binaries/Darwin-ppc/

   Make sure (no matter what the Install file says) that you also get the `Xprog.tgz` bundle. You need it.
3. Get the *xpp* source code and put it in a directory of your choice. I'll assume you've called it `xpp`. Untar the archive.
4. You will need to make the following changes in the MAC system directories, as I think there is a bug in their header files:
   - copy `/usr/include/dirent.h` to your xpp directory. I'll assume you've called it `dirent.h` locally.
   - copy `/usr/include/sys/dirent.h` to your xpp directory (giving it a new name, obviously. I called it `sysdirent.h`).
   - In the file `read_dir.c` change the `#include<dirent.h>` statement to call in your local copy of `dirent.h`, not the one in `/usr/include`.
   - In your local copy of `dirent.h`, change the `#include<sys/dirent.h>` statement to call in your local copy of `sysdirent.h`.
   - In your local copy of sysdirent.h, change the following lines:

```
u_int32_t d_fileno;           /* file number of entry */
u_int16_t d_reclen;           /* length of this record */
u_int8_t  d_type;           /* file type, see below */
u_int8_t  d_namlen;       /* length of string in d_name */
```

to the following new lines (These occur in the `struct dirent` declaration.):

```
unsigned long d_fileno;          /* file number of entry */
unsigned short d_reclen;         /* length of this record */
unsigned char d_type;         /* file type, see below */
unsigned char d_namlen;     /* length of string in d_name */
```

Save the file.

5. In the Makefile use the following options:

```
CC= cc
CFLAGS=  -O -DAUTO -DCVODE_YES  -I/usr/X11R6/include
LDFLAGS= -L/usr/X11R6/lib
AUTLIBS= -lf2c -lX11 -lm
LIBS= -lX11 -lm
OTHERLIBS= libcvode.a libf2cm.a
```

Note. In the subdirectories `cvodesrc` and `libI77` make sure that `CC=cc`.

Finally, in the main directory type `make` and everything should go fine. From here, follow the steps in the UNIX installation on pages 1–2.

1.5 Environment variables

XPPAUT makes use of certain environment variables. In UNIX, these are set in files such as `.bashrc`. In order to use the online help, *XPPAUT* needs to know the starting help file. For example, in my Linux shell environment, this is `/home/bard/xpppaut/help/xpphelp.html`. *XPPAUT* also needs to know the name of your browser since it calls in the browser to display the help. Thus, in my `.bashrc` file, I have the two lines

```
export XPPHELP=/home/bard/xppnew/help/xpphelp.html
export BROWSER=/usr/bin/netscape
```

These steps should work for OS X since this new Mac operating system is UNIX. In Windows, you can do the same thing using the "set" command as follows:

```
set BROWSER=c:\Program Files\Netscape\Communicator\Program\netscape.exe
set XPPHELP=c:\xpp\help\xpphelp.html
```

I usually include all this in a batch file that also sets up other parameters, which are discussed below.

1.5.1 Resource file

The other environment variable that *XPPAUT* makes use of is the `HOME` directory. *XPPAUT* looks here for the file `.xpprc`. Each time *XPPAUT* is run, it loads the options defined in `.xpprc`. These are described in Appendix B. The resource file `.xpprc` just contains

global options that you might want to have for every ordinary differential equation (ODE) that you run. For example, here is a short option:

```
# my xpprc file
@ but=quit:fq
@ maxstor=50000,bell=0
@ meth=qualrk,tol=1e-6,atol=1e-6
# thats it
```

This automatically puts a quit button on the top bar, allocates 50,000 storage points (instead of the default 5000), makes the default integrator the adaptive Runge–Kutta, and turns off the bell.

In Windows, you set the `HOME` directory and in that directory create a file called `.xpprc`. It will be the same form as in the UNIX version. I usually make a batch file which does everything for me automatically. I repeat my batchfile `xpp.bat` here:

```
set BROWSER=c:\Program Files\Netscape\Communicator\Program\netscape.exe
set XPPHELP=c:\xpp\help\xpphelp.html
set DISPLAY=127.0.0.1:0.0
set HOME=c:\xpp
c:\xpp\xppaut %1 %2 %3
```

This sets the display for X and also tells *XPPAUT* where to look for the resource file, the browser, and the first page of the help file. Then it calls in *XPPAUT*.