Documentation for xmk22

R.O. Gray

1 Introduction

The program xmk22 is a simple X-Windows based spectral plotting program designed for use in spectral classification on Linux and UNIX platforms. xmk22 is distributed as a tarred gzipped file of sources which must be compiled with a "C" compiler such as gcc.

2 Compiling and Installing xmk22

To compile xmk22 on your machine, you must have on your system the X-Windows development libraries. If you do not have these libraries, you will get a number of errors referring to X-Windows functions used by xmk22 when you try to compile. Assuming that you do have the X-Windows development libraries installed, untar and ungzip the distribution file xmk22.tar.gz with the following command:

```
tar -xvzf xmk22.tar.gz
```

This will create a new directory xmk22. Inside that directory you will find the source code for xmk22 and a file named Makefile. This makefile assumes that your C compiler is "gcc". Make certain that you have gcc installed on your system. If instead you have a native compiler that you can invoke with the command "cc", you can edit the Makefile and replace the top line with the line:

```
CC = cc
```

To compile, issue inside the xmk22 directory the command:

```
make
```

Hopefully, the program will now compile without errors or warnings. If successful, the executable program will be named xmk22 (without extension). You should place this either in your local bin directory or in /usr/local/bin or some equivalent location on your PATH.

If you do run into problems at this point, the most likely reason will be that you do not have the X-Windows development libraries installed on your system. For most recent distributions of Linux, these libraries will be the xorg-dev libraries, and you can use a package manager such as aptitude or synaptic to find the xorg development package and install it. For UNIX systems, consult your computer administrator or your manuals.

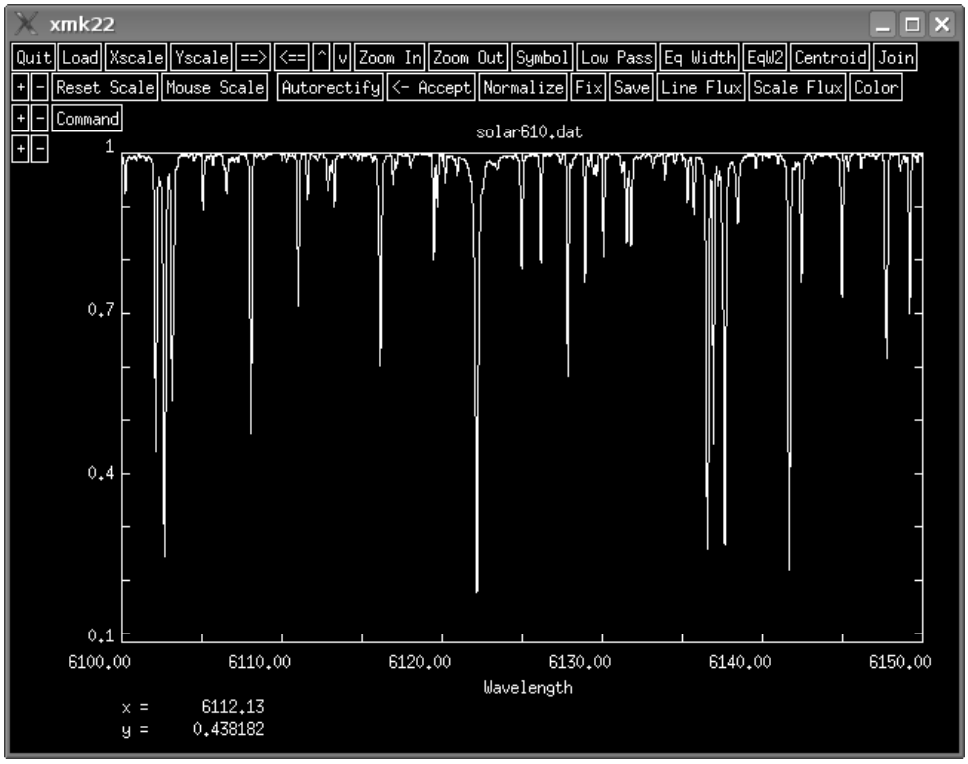

Figure 1: The appearance of `xmk22` when it is used to display the sample spectrum file `solar610.dat`. The rectangles along the top of the frame are buttons which can be activated using the mouse.

If you know that you have installed the X-Windows development libraries, but the compiler still does not seem to be able to find them (it will complain that it cannot find references to a number of functions beginning in "X"), this probably means that the line in the Makefile

```
LDFLAGS = -lm -lX11 -L/usr/X11R6/lib
```

does not correctly point to your X-Windows library. Find the correct path to your library and replace the reference `/usr/X11R6/lib` with that path. For any other problem, please contact me, at `grayro@appstate.edu` and I will try to help you to the best of my ability.

3 Some Basic Instructions

The program `xmk22` is designed to plot spectra that are in an ascii format with two columns, the first column the wavelength data, the second the flux or intensity data. An example file, `solar610.dat`, is included in the distribution to show you the correct format. The wavelength and flux/intensity data should be delimited by white spaces or a comma, but not by tabs.

To start `xmk22` type "`xmk22`" at the command prompt. The program will respond with a prompt to enter the name of a spectrum file or files. Enter the name of the example spectrum `solar610.dat` and hit return. A window should appear and display the spectrum (see Figure 1).

The background should be black and the foreground white as illustrated. If this is not the case, it is possible that your desktop (KDE, GNOME, etc.) is substituting colors. Go to the setup for your desktop (for KDE, it is the Control Center) and then look under "Colors". You should see a checkbox (for KDE it reads "Apply colors to non-KDE applications"). Uncheck that box, logout and log back in, and run `xmk22` again. This time, hopefully, the colors should be as shown in Figure 1. If you don't like those colors and would prefer others, you can edit the two lines

```
#define DEFAULT_BGCOLOR          "black"
#define DEFAULT_FGCOLOR          "white"
```

in the function `initapp.c` and then recompile.

`xmk22` interacts with the user via the mouse and a terminal window. For instance, if you use your mouse to click on the "Load" button, a prompt will appear in the terminal window asking for the name of a spectrum file. Most other buttons, such as Xscale, Yscale, Zoom In, Zoom Out, are self explanatory. The $\Rightarrow$, $\Leftarrow$, $\wedge$, and $\vee$ buttons enable you to scroll the spectrum. The "+" and "–" buttons on the lefthand side enable one to make fine adjustments in the position of the spectrum with respect to the wavelength scale, handy when displaying two superimposed spectra.

`xmk22` can display multiple spectra. At the prompt in the terminal window, enter the names of the spectra you want to display as a string concatenated with "+" signs, thus:

`spectrum1.flx+spectrum2.flx+spectrum3.flx`

`xmk22` will then prompt you for an offset. If you want the spectra to be superimposed, enter 0. `xmk22` will display all the spectra with the color white, but if you click on the Color button, different colors will be assigned to the different spectra.

Here are short explanations for some of the other buttons:

Symbol: This can be used to represent each point in the spectrum with a symbol. There is a choice of 4 symbol types; you will be prompted to make that choice in the terminal window.

Low Pass: Applies a low-pass filter to the spectrum. See the prompt in the terminal window.

Eq Width: A facility to measure the equivalent width of an absorption line. After clicking on this button, the cursor changes from a cross to ⇓. Position the cursor at a continuum point just to the left of the absorption line, leftclick, position at a continuum point to the right, leftclick, and the equivalent width will appear in the terminal window. The equivalent width is given in mÅ (assuming the wavelength scale is in Å), and is calculated using trapezoidal integration.

EqW2: Is a slightly more sophisticated equivalent width facility, suitable for somewhat noisy spectra. Four clicks are required in this case, two on either side of the absorption line. Two clicks to the left indicate the range over which the average continuum level will be determined, and likewise the two clicks on the right will determine the average continuum level there. The equivalent width is then calculated via trapezoidal integration.

Centroid: Can be used to calculate the centroid of an emission line. Click the button. A prompt in the terminal window will ask for a pixel radius for calculating the centroid. Center the cursor over the emission line and click. The centroided wavelength will then be calculated and printed to the terminal window.

Reset Scale: Resets scaling of the wavelength and flux axes.

Mouse Scale: Zoom in to a rectangular area defined by the cursor. Move the cursor to the lower lefthand corner of the rectangle, left click, then move to the upper righthand corner, left click again, and the view will be zoomed into the defined area.

Autorectify: A *rectified* spectrum is one in which all the continuum points are at unit intensity. The autorectify facility assists the user in rectifying the spectrum. The underlying algorithm is a simple one which attempts to locate suitable continuum points in the spectrum. Manual assistance is often required, especially in concave regions of the spectrum. A new continuum point may be added by locating the cursor at the

intended point and left clicking. A continuum point may be removed by centering the cursor on the offending point and right clicking. When satisfied, click on the Accept button and the spectrum will be replaced with its rectified form; the result is output to a file with the original name but with an extension of "`.rec`". The rectified file will also have a comment line at the end of the file (the first character of which is a # sign) noting the details of the rectification.

Normalize: This facility will normalize the spectrum to unity at a single point. The normalized spectrum will be saved with the extension "`.nor`", and a comment line at the end of the file will note the details of the normalization.