XYPLOT-32

User's Guide

Table of Contents

Introduction

XYPLOT is a plotting and data analysis program for use by students, teachers, scientists, and engineers. It displays the results from your experiments or calculations in an interactive graphical environment. Fast plotting and a myriad of data manipulation and analysis features make XYPLOT an invaluable tool for visualizing and working with data. Under the *Linux* operating system, XYPLOT may also be used as a graphical environment for custom data acquisition applications and for performing animated plot displays.

Features

- Fast and easy to use with minimal fuss in loading and plotting a dataset from a file.
- Arbitrarily large datasets may be loaded from files.
- Powerful built-in kForth programming environment allows user to add their own functions to xyplot.
  Available Forth modules include:
  - Arithmetic between entire data sets
  - Polynomial fitting
  - Histogram
  - Numerical derivative
  - Peak finder
  - Map functions
  - Export and import Grace `.agr` files
- Selection of plot colors and symbols.
- Continuous display of *x*,*y* cursor coordinates.
- Multi-level zoom in/out to examine features of interest.
- Modify *x* or *y* values of a dataset using simple algebraic expressions.
- View and edit dataset headers.
- Compatible with DOS and Windows XYPLOT dataset files and workspace files.

Installation

System Requirements

- x86 GNU/Linux system
- X Windows
- GNU C, C++ compilers and GNU Assembler
- OpenMotif or Lesstif GUI toolkit run-time and development libraries

XYPLOT for Linux is distributed as a compressed tar (Unix Tape Archive) file with the name

`xyplot-x86-linux-x.y.z.tar.gz`

where `x.y.z` is the current version number. The distribution file contains the source code files and a `Makefile` for building the executable. To build **xyplot** on your Linux system, your system requires the GNU C and C++ compilers (version 4.1 or higher is recommended) and the GNU assembler. Linux distributions configured as software development workstations (during installation) will have these development tools installed by default. If not, they may be installed from software groups or individual packages. Also needed for building **xyplot** are a version of the Motif graphical user interface libraries. Either the OpenMotif libraries or the Lesstif libraries may be used. For example, see the Lesstif Installation Notes.

After installing either the OpenMotif or Lesstif packages on your system, you may build **xyplot** by following these steps:

1. Copy the distribution file (`xyplot-x86-linux-x.y.z.tar.gz`) to any desired directory.
2. Change to the directory in which the distribution file resides and unpack the contents:

   tar -zxvf xyplot-x86-linux-x.y.z.tar.gz

   The distribution file will unpack to a new subdirectory called `xyplot-x.y.z`. The unpacked subdirectory will contain all of the xyplot source files (a set of `.h`, `.cpp`, `.c`, `.s`, and `.4th` files), the `Makefile`, and a `README` file, which also contains installation instructions.
3. Change the directory to the new subdirectory and build the **xyplot** executable.

   cd xyplot-x.y.z/

   There are several options for building xyplot, but the simplest is to type:

   make

All of the source files will be compiled/assembled and the executable file, named **xyplot**, will be made. You may notice compiler warnings scroll by as the files are compiled, but these may be ignored. Only if you do not have an executable named **xyplot** at the end of the make process has something gone wrong.

You can examine the file `Makefile` to see the other options for building xyplot. Also, if something does go wrong during the make process, you may need to edit the `Makefile`. For example, the linker may not be able to find the necessary libraries on your system and you may need to redefine the macro `LIBDIRS` in the `Makefile`.

4. To make xyplot accessible to all users on the system, move the executable to a directory in the path for users, *e.g.*:

   mv xyplot /usr/local/bin/

   Note that you must be logged in as superuser in order to move a file into the above system directory. After this is done, any user should then be able to type **xyplot** at a command prompt and be able to execute the program.

5. Each user maintains customized settings in his/her `$HOME/.xyplot` directory. For each user, create the directory as follows:

   mkdir /home/username/.xyplot

   Next, copy the Forth source files `(*.4th)` provided by the distribution into this directory

   cp *.4th /home/username/.xyplot

   When a user executes xyplot, the file `xyplot.4th` from the user's `$HOME/.xyplot` will be loaded. XYPLOT can be customized by appending Forth commands to the end of this file. For example, other Forth files, called modules, may be loaded to provide additional menu functions. The file `smooth.4th` is an example of a module which adds an item called *Smooth* to the *Math Menu*. The smoothing function is written in the Forth programming language and this source code is loaded upon startup from a line in `xyplot.4th`: *include smooth*.

The Basics

This chapter describes some of the elementary operations you can perform with XYPLOT. Advanced topics are covered in the subsequent chapters.

Getting Started

Start XYPLOT from the terminal by typing:

xyplot [*filename1 filename2*] ...

If you include one or more file names on the command line, XYPLOT will read the file(s) and then graph the data in the file(s).

Selecting a File

Click on the *File* menu and select *Open*. Select a file to load by navigating the standard File Open dialog box. You may filter the type of files which are displayed from a list in the dialog box.

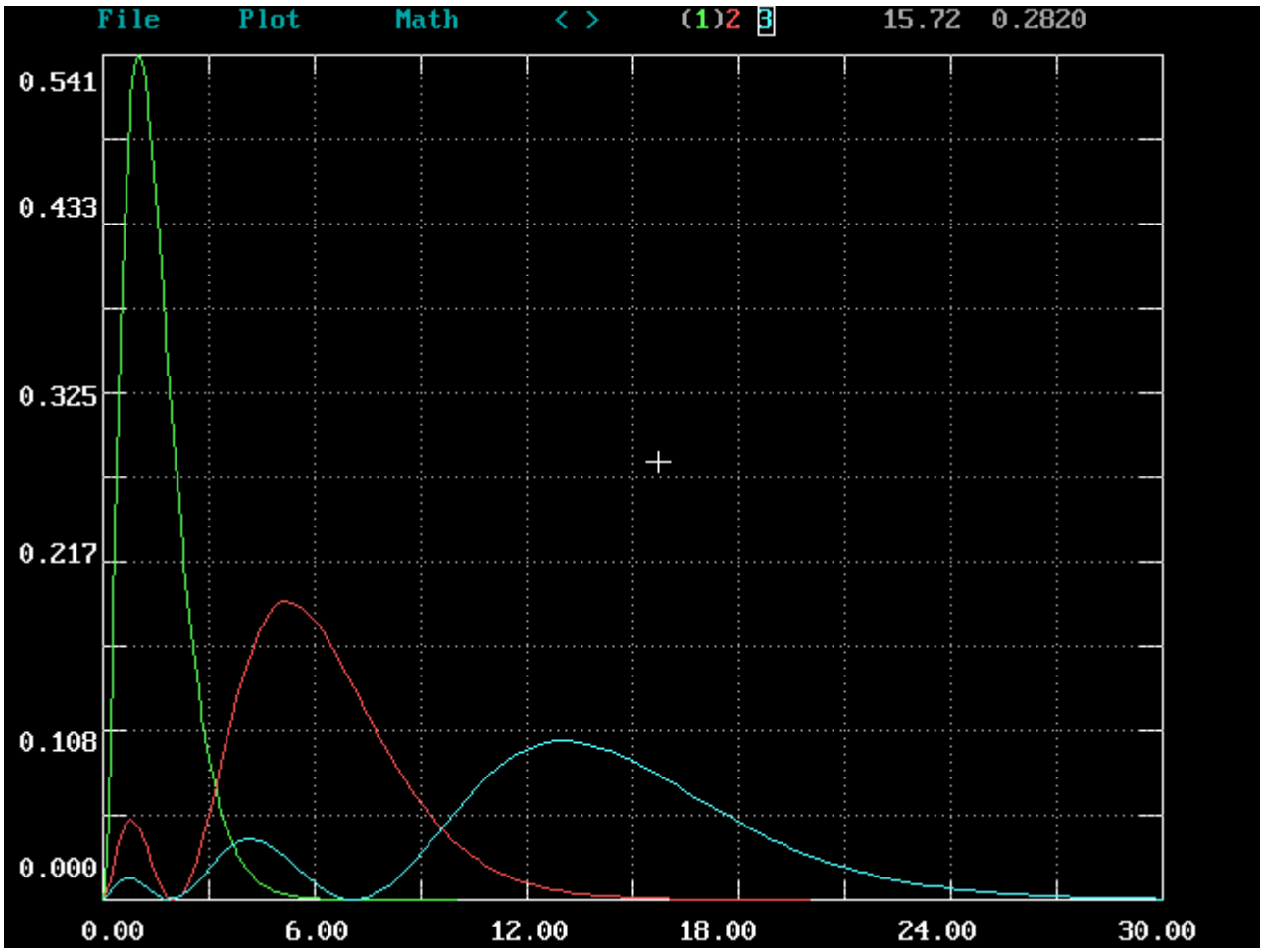
A Guide to the Plot Display

A plot of the data in the file that you have selected appears on the *plot display*. Look at the following image for

- **Plot Cursor** A pointer movable within the plot window.
- **Cursor Coordinates** The *(x, y)* coordinates of the plot cursor.
- **Menu** An index to the commands.
- **Plot List** A color coded index to the data.

Each of these has a special function. As much space as possible on the screen is allocated to the plot so that the data will be displayed at high resolution. Data is plotted within the plot window. When new data are loaded, the limits will be adjusted automatically to show you the entire plot, however, you may change the limits with the mouse, or through the menu.

A frame, its inner edges marked with fixed tics, surrounds the plot window. A dotted *grid* may be superimposed on the frame. The grid is toggled on or off by the **G** key. In this sense the plot display behaves as a piece of graph paper that sets the scales on its axes automatically to show the data that you have.



Setting the Plot Window Limits

XYPLOT automatically sets the plot window limits from the minimum to the maximum values in all the data. You may specify new window limits by

- defining an expansion box with the plot cursor
- manually entering new window limits

The first method is a quick way to get an expanded view of a region of interest. You can inspect fine detail in whatever region interests you. The second method sets the limits precisely. It is useful because in XYPLOT *the x limits of the plot window also serve to select data* for operations such as storing parts of a dataset, calculating areas, or making a polynomial fit. This is in analogy to selecting text for editing in a word processing program.

Setting Window Limits With the Plot Cursor

You may zoom in on part of the plot by using the pointer to define a box. Move the cursor to point at the lower left corner of a region you want to see in more detail. Hold down the left pointer button and move the pointer to define the box. Release the button and the graph will be redrawn with the new window limits. The portion of the plot window that had been inside the box will now fill the entire plot window.

Setting the Window Limits by Input

Press the **E** key and you will be prompted for the new window limits with a line reading:

Enter new window limits *x1, y1, x2, y2*:

where *(x1, y1)* is the lower left corner and *(x2, y2)* is the upper right corner of the window. You need not specify all of the new coordinates. For example, to change only *y1* and *y2*, you could enter

,y1,,y2

Cancelling the Limits

Every time you change the window limits XYPLOT makes a record of the previous window limits. Several levels of expansion are retained. To return to the previous level, press **Esc**.