XYPLOT-32

User's Guide

© 1987 – 2018 Krishna Myneni and John Kielkopf

Contributors:

Bryan J. Frazar
Helen Kielkopf

Table of Contents

Introduction

XYPLOT is an extensible 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. With its powerful, built-in Forth interpreter, users may write custom modules to add new data manipulation and analysis functions, and even data-acquisition functions accessible from menus. Many of XYPLOT's built-in functions are provided in the form of Forth source modules which are loaded upon startup. For generating publication quality graphs, a module provides the capability to export graphs to Grace.

Features

- Fast and easy to use with minimal fuss in loading and plotting a data set from a file.
- Arbitrarily large data sets may be loaded from files.
- Built-in kForth interpreter allows user to add their own functions to xyplot.
  Provided Forth modules include:
    - Arithmetic between entire data sets
    - Polynomial fitting
    - Histogram
    - Numerical derivative
    - Peak finder
    - Map functions
    - Export and import Grace 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.

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.

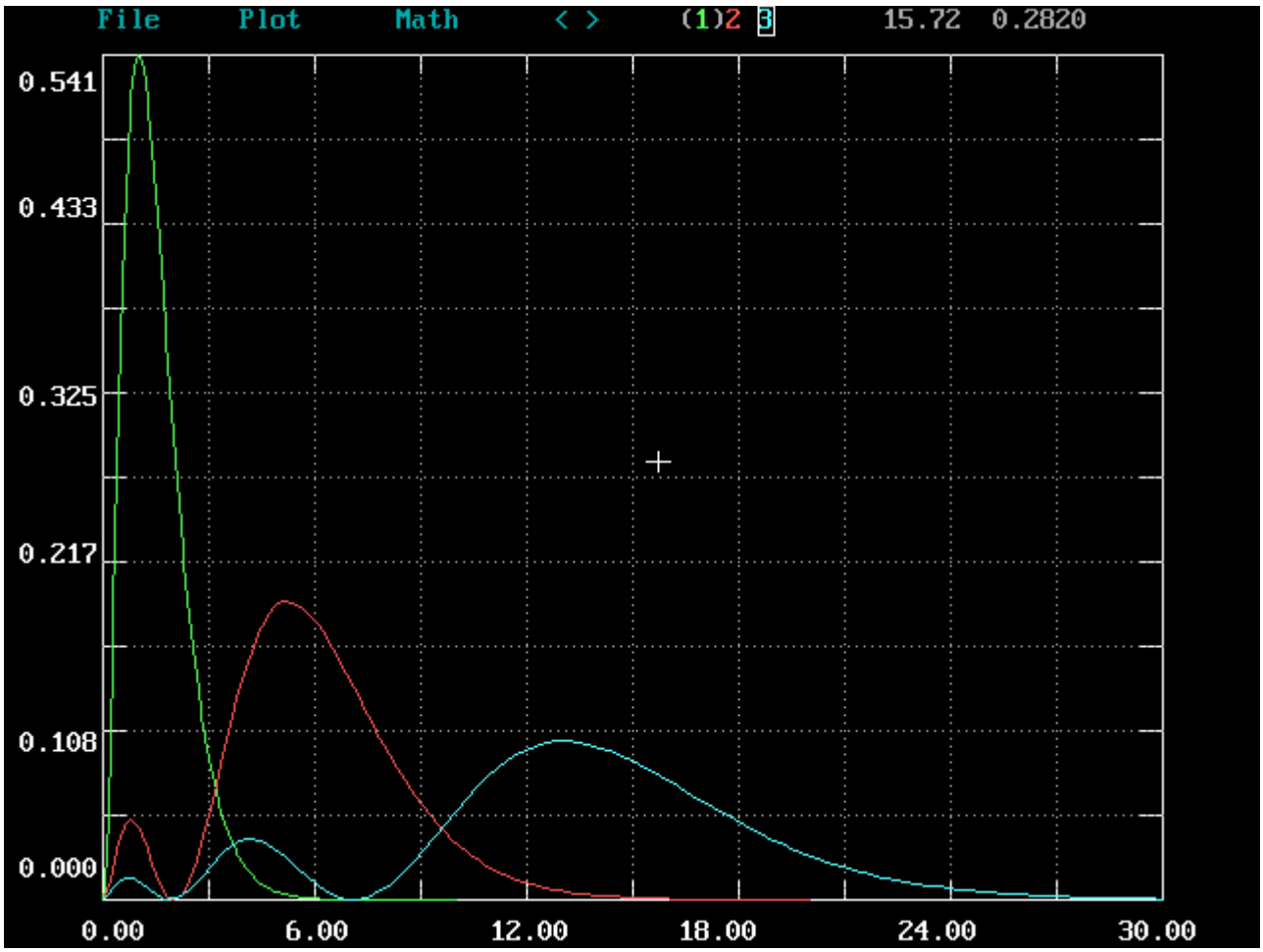
A Guide to the Plot Display

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

- **Plot Cursor** A pointer movable within the plot window.
- **Cursor Coordinates** The *(x, y)* coordinates of the plot cursor.
- **Menus** An index to the commands.
- **Plot List** An index from which plots may be selected for operations.

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

Canceling 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**.

How To Use Files

XYPLOT offers a unique graphical environment for working with sets of *(x, y)* data. The XYPLOT workspace treats each set as a single entity. You may think in terms of operations on sets, rather than on individual numbers. This structure allows you to use fast simple procedures to perform powerful operations. Sets of data and their visual representations constitute the *workspace*.

XYPLOT can accommodate arbitrarily large data sets. XYPLOT may read a file on disk and place the information in a data set, or it may create a file and save the contents of a data set on disk. This chapter describes the program's use of data set files.

The Structure of Data Set Files

XYPLOT reads data from and stores data to plain text files. These are the files that appear sensibly displayed on the screen when you type

type filename.dat

or, under Linux,

cat filename.dat

for example. To be more precise, they are known as ASCII files, which stands for the widely adopted **A**merican **S**tandard **C**ode for **I**nformation **I**nterchange. Information saved in files of this sort may be read by almost any computer.

Data files contain numbers arrayed in one or more columns preceded by an optional header.

1. A single column of data represents the *y* values with *x*, understood, a running index from one to the number of points.
2. A two column file represents *(x, y)* pairs. The first column is *x*.
3. With files of three or more columns, you are prompted to choose a pair of columns representing *x* and *y*.

Many mathematical operations require that the values of *x* are either in ascending or descending order in that file. If the data are not in this order, a `Sort` function may be used to reorder the points.

If lines of text are included at the beginning of the file and enclosed between the `/*` and `*/` symbols, XYPLOT interprets the information as a descriptive *header*. For example, the data might be measurements of thermocouple voltages at different times. The header could be used to save whatever information you want to record about how the measurements were made. When you apply a calibration to convert the voltages to temperatures, a record of the calibration will be inserted

automatically into the header. You may save the calibrated measurements in a new file together with the modified header.

The example below illustrates a data file with two columns. Use your favorite text editor to create this or other data files or write them with other programs. A valid data file has the following characteristics:

- A header may be present, but is not required.
- The first line of a header, if present, must be `/*`
- After the header text, the last line of the header must be `*/`
- Numbers may be written in the following formats:
    - integer (`20`)
    - floating point (`20.0`)
    - exponential (`2.0e1`)

- Columns may be separated by spaces, commas, or tabs.

[Beginning of file]

```
/*
Test data to illustrate file format.
Parabola
*/

-3.0 9.0
-2.0 4.0
-1.0 1.0
0.0 0.0
1.0 1.0
2.0 2.0
3.0 9.0
```

[End of file]

In the example above the numbers are in floating point format, and the column delimiter is a single space.

An alternative style of header may be used in which each header text line is preceeded by the # character. This style is useful for its compatibility with other graphing programs such as Grace. Also, users may write Forth modules for loading their own custom format data files, such as binary data files, into xyplot.

How To Select and Modify Data

The XYPLOT Database

The XYPLOT *database* holds each data set loaded from a file in memory. There is no restriction on the size of an individual data set or on how many data sets may be loaded into the database. Data sets may be *Duplicate*d or *Delete*d from XYPLOT's database using built-in functions.

A data set consists of (*x, y*) pairs and additional descriptive text information in a header. A plot is automatically created for a data set loaded from a file into XYPLOT's database. Each plot in the plot display is linked to a single data set. However, each datas et in the database does not have to be linked to an any plot in the display, or a single data set can be linked to multiple plots.

Selecting Data Sets

Data sets are selected for mathematical and other operations by selecting their corresponding plots from a *Plot List*. This means that a data set in XYPLOT's database must appear as a plot for any operation to be performed upon it. The *active plot* represents the data set upon which single set operations are performed. The active plot is identified by a box or bracket around its number in the *Plot List*.

Two-set operations are specified by selecting the active plot and the *operand plot* in the *Plot List*. The operand plot is identified by parentheses surrounding its plot number in the *Plot List.*

If a previously loaded data set does not appear as a plot, you may use the *Pick* function from the *Plot Menu* to make a plot of the data set. *Pick* may also be used to see a list of data sets stored in the database.