

LOCUS User's Guide

Steve Scott
March, 2005

Last updated November 22, 2005

LOCUS is a database utility written in IDL that generates scatter plots from SQL database tables generated by the FILL_DB utility. It is based on a utility of the same name written for TFTR about twenty years ago by Jane Murphy.

New features!

- (5/25/2007) You can not plot multiple yidevices.
- (3/22/2006) You can now plot error bars for the x- and/or y- data.
- (11/22/2005) Fixed a bug that improperly computed complicated expressions (those having more than three variables).
- (11/22/2005) Added support for a new function: absolute value.

Use

To run LOCUS, type the following

```
IDL  
.run /home/sscott/locus/locus.pro  
locus
```

You will be prompted to enter

- The name of the database
- The name of the table
- The name of the parameter file which was used by FILL_DB to define the names of each of the columns in the database

Conventions and Syntax in LOCUS

1. Typically, LOCUS issues a succinct prompt such as

```
SET? X, Y, C, S, G, Ex, J, Lookup, PL, Q >>
```

to remind you what your options are at any level in the program. You are expected to respond with one of the listed options, e.g. 'Y' to define the variable to be plotted on the y-axis or 'PL' to plot the data.

If you are unsure what the various prompts mean, simply hit <cr> in response to the prompt. LOCUS will then type out a longer description of each available option.

2. If you are in the process of defining a list of items, e.g. a list of constraints, entering a semi-colon (";") terminates the list. For example,

```
SET? X, Y, C, S, G, Ex, PL, Q >> C
Constraint? [ ] ip>0.5
Constraint? [ ] bt>4.2
Constraint? [ ] ;
```

tells LOCUS to define two constraints, and the final semi-colon terminates entry.

3. Otherwise, entering a semi-colon causes LOCUS to return from a lower-level menu to its parent menu.
4. If you have already defined a list of items, e.g. a list of constraints, then entering a comma causes LOCUS to retain the current value of that item. For example, if you re-visit the constraints menu after defining the two constraints in the example above, you could type

```
SET? X, Y, C, S, G, Ex, PL, Q >> c
Constraint? [ ip>0.5 ] ,
Constraint? [ bt>4.2 ] wtot>47.
Constraint? [ ] ;
```

This would retain the original constraint on the plasma current and replace the original constraint on toroidal field with a constraint on plasma stored energy.

5. The x- and y-axes can be simple variable names such as ip, bt etc. or they can be arithmetic functions of the variables. Recognized functions include

```
+      -      *      /      ^
exp    log    abs    sqrt
sin    cos    tan
asin   acos   atan

e.g. ip*(log(bt))/(4.7/wtot^1.4)
```

The usual fortran & IDL rules regarding the precedence of order operation prevail. The use of parentheses is encouraged to clarify what operations are done first.

Spaces are not allowed when defining the x- and y-axes.

6. At the present time, arithmetic functions are not allowed in constraints:

ip>0.56	allowed
log(bt)>0.5	not supported
ip>bt	not supported

7. Syntax of constraints: the following operators and syntax are supported when defining constraints:

ip=0.47	
ip>0.47	
ip<0.47	
ip<=0.47	
ip>=0.47	
ip<>0.47	(not equal to)
(ip>0.47)and(bt>4.2)	
(ip>0.47)or(bt>4.2)	

Effectively, LOCUS just passes the constraint directly to SQL as a character string. So LOCUS should correctly process any operator that is known to SQL ... but I haven't tested this feature extensively.

8. Constraints and symbols:

Constraints are conditions imposed on ALL data that is to be displayed.

You can define additional conditions through the Symbols option – this allows you to define conditions for one or more symbols. For example,

SET? X, Y, C, S, G, Ex, PL, Q >> c

Constraint? []	ip>0.47
Constraint? []	;

SET? X, Y, C, S, G, Ex, PL, Q >> s

Symbol? []	bt>4.2
Symbol? []	bt<3.8
Symbol? []	;

would cause LOCUS to plot two different symbols. The first set would satisfy (ip>0.47)and(bt>4.2) and the second set would satisfy (ip>0.47)and(bt<3.8).

Features

1. You can define plot limits with the **g** option at the highest menu level. Otherwise, both the x- and y- graph limits will be chosen to include all of the selected data.
2. The Extras menu provides access to other customization features of LOCUS:
 - symbol icon (square, triangle, ...)
 - symbol color
 - symbol size
 - symbol thickness

 - filename: this defines a template name when you dump a plot into a file. By default, the template name is "locus" and the plots will be named locus_1.ps, locus_2.ps, ...

 - expression table: you can store frequently-used expressions in a table. This feature will be documented at a later date.
3. You can record comments about the previous plot using the 'Journal' option. This feature lets you type in an arbitrary number of comment lines. Your comments are added to the plot file, with up to 25 lines of comments per page. If you are prolific and enter more than 25 lines, new comment pages are opened as needed.
4. The 'Lookup' feature allows you to determine what a given LOCUS variable represents. You specify the variable name, and LOOKUP returns the line in the parameter file that is associated with this variable, which includes the full device specification.

POST PLOT Options

1. After a plot is generated, LOCUS offers the following list of post-plot options:

Enter Post-Graph Command (<cr>, I, D, ?, ...) []:

If you enter <cr> you will be returned to the main menu. Type ? to get a complete listing of the available options.

2. Identify mode (I): in this mode, you click on a datapoint, and then click again somewhere inside the graph – usually, somewhere close to the original datapoint. LOCUS will print out the shot number at the location of the second click.

You continue indefinitely in this mode – i.e. selecting a datapoint with the first click and then defining where to print the shot number with the second click – until you double-click outside the confines of the plot.

3. Polynomial fit mode: if you enter a single-digit integer in response to the post-graph prompt, LOCUS will fit a polynomial of that order to each set of data for each symbol.
4. Other functional forms such as a Gaussian fit are available. Consult the list in response to **?** for a complete listing.
5. Dump plot contents to a file (**D**): this merely causes the contents of the last displayed graph to be written to a postscript file.
6. Dump plot contents to an ascii file (**W**). You can selectively write the contents of the x-array, the y-array, or both to a file. The data will be written (appended) to an ascii file locus_dump.txt

Plotting Error bars (new feature 3/22/2006)

You can plot error bars for the y-data and/or x-data by defining a column that represents the error bar using the 'xerr' and 'yerr' commands.

To eliminate the plotting of these error bars, invoke the 'xerr' or 'yerr' option and define the column name as 'none'.

Plotting multiple y-devices (new feature 5/25/2007)

You can now plot multiple y-devices using the 'my' option.

Note: you can't simultaneously ask for both multiple symbols and multiple y-devices. So if you invoke the 'my' option and define one or more y-devices, then the number of symbols will be set to zero. Similarly, if you invoke the 's' option to define symbols, and if you define one or more symbols, then the number of y-devices will be set to zero.

Limitation: there is currently no support for error bars in the multiple y-device option.