ISETL Graphics Guide

This document describes the graphics package for ISETL. If you are not familiar with ISETL, refer to the *ISETL Getting Started Guide*.

Using the Graphics Facility

The graphics facility is invoked by calling one of six graphics routines in an ISETL expression. The routines (commands) are:

The specific syntax for each command is given in the next section, beginning on page 2.

- plot
- graphpar
- graphpol
- graphR
- graphs
- vectors

Once a graph is constructed, you may study the graph by interacting with it. You may interact with the graphics window in the following ways:

These features are described beginning on page 5.

- read coordinates of a point
- zoom in
- add a grid
- enter data in the window
- increase the resolution (in graphR only)
- print the graphics window

Graphics and Windows

Three windows are available during a graphics activity.

Note: Once the drawing of the graph(s) is complete, the graphics activity is still operating. It is waiting for an interactive feature command or a quit <q>.

- Graphics Window
- Execution Window
- Memory Usage Window

When the graph is completed, the Graphics window comes to the front. The Execution and Memory Usage windows and any other open windows remain behind the Graphics

You can size and position these windows using the normal operations for your system. You may bring the other windows to the foreground at any time during an activity by clicking on the desired window, but the Graphics window remains active during the entire activity.

Do not close any window while an activity is in operation.

Scales for your Graphs

Your graph will be drawn in the Graphics window using a set of coordinates automatically determined by the routine you are using. Arguments may adjust the scale before graphing and individual scales can be zoomed interactively.

Rectangular Coordinates of the Cursor

The title bar of the Graphics window displays the rectangular coordinates of the cursor position during graphics activities. Coordinates are only displayed for locations inside the graphics window.

Interacting With Graphics

You may perform as many graphics features as you like while the graph is activated. Graphics interactive features are engaged by single key strokes made while the cursor is in the graphics window.

Exiting The Graphics Facility

Press the single key <q>.

The Execution window will come to the front, a prompt will appear, and the graph becomes inactive. You will still be able to view the graph by clicking in the graphics window. After pressing <q> you may no longer use the interactive features to change the graph.

Closing the graphics window while it is active will exit the graphics facility, remove the graphics window from you screen, and dump error messages into the Execution window. You will not be able to view the graph after closing the graphics window. It is best to quit the graphics facility by pressing <q>.

How to tell the Drawing is Complete

You can tell a drawing is complete by two signals

- The graphics window moves to the front of the screen and stays there.
- The cursor changes from an arrow or I-bar to a set of cross hairs (+). Its coordinates appear at the top of the screen.

Available Graphics Routines

plot

Syntax:

```
plot(f);
plot(f,a,b);
plot(f,a,b,c,d);
```

plot(f); This command will produce a graph with default horizontal and vertical scales. To plot more than one curve on one set of axes, use square brackets around the objects to be plotted (e.g. `plot([f,g]);`).

plot(f,a,b); This command allows you to set the horizontal scale to $[a, b]$ and uses the default scale for the vertical axis. (a, b must be numbers)

plot(f,a,b,c,d); This command allows you to set both the horizontal $[a, b]$ and vertical $[c, d]$ scales. (a, b, c, d must be numbers)

The argument f can be any of the following:

- A func which accepts a number and returns a number.
- A tuple whose components are numbers.
- An smap whose domain and range are sets of numbers.
- An mmap whose domain and range are sets of numbers.
- A tuple whose components are any of the above.

Details:

For f as a func, the default scales are:

- $[-10, 10]$ on the horizontal axis.
- Vertical scale determined by the maximum and minimum values of f on the interval $[a, b]$.

A sampling procedure will partition the domain interval into a set of domain points. The func f will be evaluated at these points and the resulting coordinate pairs will be plotted. The sampling procedure begins with a partition of the interval into 20 evenly spaced points. ISETL tests for curvature and increases the sampling rate as the graph appears to be curving.

For f as a tuple of numbers, the default scales are:

- $[0, n]$ on the horizontal axis where n is the number of components in the tuple f.
- Vertical scale is determined by the largest and smallest components of f.

The horizontal axis is partitioned into domain points which are the positive integers in $[0..n]$. The range values are the corresponding components of f. This produces ordered pairs which are plotted.

For f as an smap or mmap, the default scale is to simply plot the pairs in the map. Scales are determined by the maximum and minimum values in the domain and range of the map.

graphpar

Syntax: `graphpar(F,a,b);`

The argument F is a tuple of two funcs ($F = [f, g]$) each of which accepts a number and returns a number. The arguments a, b are numbers.

Details:

graphpar plots a function given by a parametric equation. The domain of the two funcs are sampled on the interval $[a, b]$ for values t and the pairs of values $[f(t), g(t)]$ are plotted.

graphpol

Syntax: `graphpol(f,a,b);`

The argument f is a func which accepts a number and returns a number. The arguments a, b are numbers.

Details:

A sampling procedure divides the interval $[a, b]$ into evenly spaced points. The func f is evaluated at these points and ordered pairs $[t, f(t)]$ are plotted in polar coordinates.

graphR

Syntax:

```
graphR(f);
graphR(f,a,b);
graphR(f,a,b,c,d);
```

The argument f is a func which accepts two numbers as argument and returns a number.

graphR(f); This command produces a graph with the default scale of $[-10, 10]$ on the horizontal and vertical axes.

graphR(f,a,b); This command produces a graph with scale $[a, b]$ on the horizontal axis and the default scale $[-10, 10]$ on the vertical axis. (a and b must be numbers)

graphR(f,a,b,c,d); This command produces a graph with the scale of $[a, b]$ on the horizontal axis and the scale of $[c, d]$ on the vertical axis. (a, b, c, d must be numbers).

Details:

A sampling procedure selects $[x, y]$ pairs from the rectangle determined by the scales on the horizontal and vertical axes.

The func f is evaluated at these pairs. If $f(x, y)$ is close to zero, a point is plotted at $[x, y]$.

Better resolution is achieved by pressing <r> in the Graphics window (see details on page 6).

graphs

Syntax: `graphs(f,g,a,b);`

The arguments f and g are funcs which accept a number and return a number. The arguments a, b are numbers.

Details:

The scale on the horizontal axis is $[a, b]$ and the vertical scale is determined by the maximum and minimum values of either func.

A sampling procedure partitions $[a, b]$ into 60 evenly spaced points. Values are sampled and pairs $[x, y]$ satisfying

$$\min(f(x), g(x)) = y = \max(f(x), g(x))$$

are plotted.

vectors

Syntax:

```
vectors(S);
vectors(S, arrow);
```

The argument S is a set each of whose elements are either pairs $[x, y]$ of numbers or pairs of pairs $[[u, v], [x, y]]$ of numbers.

In the first case each pair represents a vector from the origin to $[x, y]$. In the second case, each tuple represents a vector from $[u, v]$ to $[x, y]$.

The argument `arrow` is a Boolean value (either true or false).

Details:

The vectors in S are drawn on an appropriate scale.

If the argument `arrow` is not present or is present and has the value true, then arrowheads will be drawn on the vectors.

If the argument `arrow` is present and has the value false then the arrowheads will not be drawn.

Interacting with graphics

After drawing is complete the following graphics features may be activated.

To activate a feature, the cursor must be on the graphics window. Most features may be repeated and mixed with other features.

Pressing <q> disables the features. Always deactivate the features with <q>, *not* by closing the graphics window.

Reading coordinates of a point:

This feature is available with all graphics routines.

- Place the cursor at the desired point in the Graphics window.
- Click the mouse.

  A rectangular coordinate pair will appear in the Execution window. graphpol points will be converted to polar coordinates.

Zooming in:

This feature is available in the graphR, graphs, plot, and vectors graphics routines.

This feature is not reliably working on the Windows platform. Be prepared for less than exact zooming.

- Place the cursor on the Graphics window and press the <z>.
- Move to another position and press <z> again.

The two cursor positions are taken as opposite corners of a rectangle. After pressing <z> a second time, this rectangle appears on the screen.

- Press <z> again to redraw the graph on the scale given by the rectangle.

  In the graphs routine, the graph is redrawn with the horizontal scale determined by the rectangle. The vertical scale is determined by the maximum and minimum values of the graphs on the domain interval.

Seeing more of the graph:

This feature is available in the plot routine only.

- To double the horizontal scale and redraw the graph, press <X>.
- To double the vertical scale and redraw the graph, press <Y>.
- To double both scales and redraw the graph, press <Z>.

Adding a grid to the graphics window:

This feature is available in all graphics routines.

- To place a grid on the Graphics window, press <g>.
- To remove the grid, press <g> again. This can be repeated.

Increasing the resolution:

This feature is only for graphR.

- To increase the resolution of the curve, press <r>. Each time you do this, the curve will be sharper and more accurate, but drawing time increases by 2–4 times over the previous graph.

Entering information on the graphics window:

This feature is available in all graphics routines.

Save this feature for last, just before printing the graph! If you hit any key (except <enter> or <q>) after entering information, the graph is redrawn and the labels are lost! On the Macintosh keyboard, you may use <return> or <enter>.

- Place the cursor where you want to type on the Graphics window and press <enter>.
- Type desired information exactly as you want it (editing your text is not possible in this feature).
- Press <enter> to resume normal operation.
- You may repeat this operation on as many parts of the graph as needed.

Printing the Graphics Window

Graphs in ISETL for Windows

Simply use the **Print...** option in the **File** menu.

Graphs in ISETL on a Macintosh

Printing graphs from ISETL on the Macintosh is somewhat complicated. Follow these steps:

- Get the graph ready to print. Add your group name, fix the scales, get the resolution (in graphR), etc.
- Take a screen snapshot by pressing the Shift, Command (the Apple key, it is next to the space bar), and 3 keys at the same time. This places a file called 'Picture 1' on the top level of the hard drive. The picture files are numbered consecutively.
- Leave ISETL running and switch back to the desktop (also known as the Finder) by clicking on the desktop where it is visible. Double-click on the Picture file to open it in SimpleText.
- Choose **Page Setup...** in the File menu and set the orientation to landscape (the sideways man). Click OK
- Choose **Print...** from the file menu. Return to ISETL by using the Applications menu in the top right corner.

Warning: Do not use the Print command in the File menu when the Graphics window is active! This will cause the printer to hang, stopping all printing until the job is killed.

Common Errors with Graphics

"Cannot read from: OM" errors

The error message `Cannot read from: OM` sometimes occurs when `plot` is called. This means that ISETL is not pointing to its home directory. To correct this, open the axis.stl file found in the ISETL directory. This points ISETL back. You may then close axis.stl and re-run your `plot` command.

No cursor (cannot type) in Execution window

The Graphics window is still active. Go back to the graphics window and type a <q>. Be sure to use a lowercase q; check that Caps Lock is not on.

The 'q' shows up on the graph

You have enabled the text entering mode. Press <enter>, and then <q> to quit the interactive graphics. You will need to re-run the graphing command in order to erase the extra letters on the graph before printing.

Unable to reopen saved graphics window

Do not save the graphics window. Save the code that produced the graph and re-run that. The graphics utility is designed for interactive use, not for portability.

Credits: J. Dautermann, D. Breidenbach, J. Cottrill, J. Kleiman

December 2001