15 Plotting

15.1 Introduction to Plotting

Earlier versions of Octave provided plotting through the use of gnuplot. This capability is still available. But, a newer plotting capability is provided by access to OpenGL. Which plotting system is used is controlled by the `graphics_toolkit` function. See Section 15.4.7 [Graphics Toolkits], page 372.

The function call `graphics_toolkit ("fltk")` selects the FLTK/OpenGL system, and `graphics_toolkit ("gnuplot")` selects the gnuplot system. The two systems may be used selectively through the use of the `graphics_toolkit` property of the graphics handle for each figure. This is explained in Section 15.3 [Graphics Data Structures], page 331. **Caution:** The FLTK toolkit uses single precision variables internally which limits the maximum value that can be displayed to approximately 10^{38}. If your data contains larger values you must use the gnuplot toolkit which supports values up to 10^{308}.

15.2 High-Level Plotting

Octave provides simple means to create many different types of two- and three-dimensional plots using high-level functions.

If you need more detailed control, see Section 15.3 [Graphics Data Structures], page 331 and Section 15.4 [Advanced Plotting], page 358.

15.2.1 Two-Dimensional Plots

The `plot` function allows you to create simple x-y plots with linear axes. For example,

```
x = -10:0.1:10;
plot (x, sin (x));
```

displays a sine wave shown in Figure 15.1. On most systems, this command will open a separate plot window to display the graph.

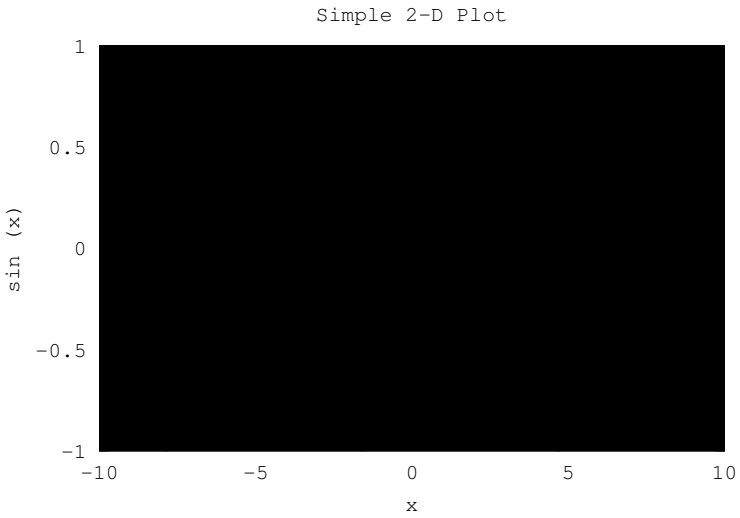

Figure 15.1: Simple Two-Dimensional Plot.

`plot (`*y*`)` [Function File]
`plot (`*x*, *y*`)` [Function File]
`plot (`*x*, *y*, *fmt*`)` [Function File]
`plot (`..., *property*, *value*, ...`)` [Function File]
`plot (`*x1*, *y1*, ..., *xn*, *yn*`)` [Function File]
`plot (`*hax*, ...`)` [Function File]
h `= plot (`...`)` [Function File]

Produce 2-D plots.

Many different combinations of arguments are possible. The simplest form is

```
plot (y)
```

where the argument is taken as the set of y coordinates and the x coordinates are taken to be the range `1:numel (`*y*`)`.

If more than one argument is given, they are interpreted as

```
plot (y, property, value, ...)
```

or

```
plot (x, y, property, value, ...)
```

or

```
plot (x, y, fmt, ...)
```

and so on. Any number of argument sets may appear. The x and y values are interpreted as follows:

- If a single data argument is supplied, it is taken as the set of y coordinates and the x coordinates are taken to be the indices of the elements, starting with 1.
- If x and y are scalars, a single point is plotted.
- `squeeze()` is applied to arguments with more than two dimensions, but no more than two singleton dimensions.

- If both arguments are vectors, the elements of *y* are plotted versus the elements of *x*.
- If *x* is a vector and *y* is a matrix, then the columns (or rows) of *y* are plotted versus *x*. (using whichever combination matches, with columns tried first.)
- If the *x* is a matrix and *y* is a vector, *y* is plotted versus the columns (or rows) of *x*. (using whichever combination matches, with columns tried first.)
- If both arguments are matrices, the columns of *y* are plotted versus the columns of *x*. In this case, both matrices must have the same number of rows and columns and no attempt is made to transpose the arguments to make the number of rows match.

Multiple property-value pairs may be specified, but they must appear in pairs. These arguments are applied to the line objects drawn by `plot`. Useful properties to modify are `"linestyle"`, `"linewidth"`, `"color"`, `"marker"`, `"markersize"`, `"markeredgecolor"`, `"markerfacecolor"`.

The *fmt* format argument can also be used to control the plot style. The format is composed of three parts: linestyle, markerstyle, color. When a markerstyle is specified, but no linestyle, only the markers are plotted. Similarly, if a linestyle is specified, but no markerstyle, then only lines are drawn. If both are specified then lines and markers will be plotted. If no *fmt* and no *property/value* pairs are given, then the default plot style is solid lines with no markers and the color determined by the `"colororder"` property of the current axes.

Format arguments:

linestyle

'`-`'	Use solid lines (default).
'`--`'	Use dashed lines.
'`:`'	Use dotted lines.
'`-.`'	Use dash-dotted lines.

markerstyle

'`+`'	crosshair
'`o`'	circle
'`*`'	star
'`.`'	point
'`x`'	cross
'`s`'	square
'`d`'	diamond
'`^`'	upward-facing triangle
'`v`'	downward-facing triangle
'`>`'	right-facing triangle
'`<`'	left-facing triangle
'`p`'	pentagram
'`h`'	hexagram

color

'`k`'	blacK
'`r`'	Red