

# unofficial OctPlot documentation to OctPlot Version 0.3.5

Author:

**Michel D. Schmid**

Send bugs, comments or suggestions to:

[michel\\_d\\_schmid@gmx.net](mailto:michel_d_schmid@gmx.net)

Date: 2006-02-18, Saturday

# Contents

<b>1</b>	<b>OctPlot</b>	<b>2</b>
1.1	activating OctPlot	2
<b>2</b>	<b>OctPlot Commands</b>	<b>3</b>
2.1	area	3
2.2	axis	3
2.3	bar, barh	3
2.4	cla	6
2.5	clf	6
2.6	colorbar	6
2.7	colormap	6
2.8	contour	6
2.9	contourc	6
2.10	freqz	6
2.11	gca	6
2.12	gcf	6
2.13	grid	6
2.14	hold	6
2.15	ishold	6
2.16	line	6
2.17	loglog	6
2.18	patch	6
2.19	pcolor	7
2.20	plot	7
2.21	print	7
2.22	semilogx	7
2.23	semilogy	7
2.24	specgram	7
2.25	stem	7
2.26	subplot	8
2.27	text	11
2.28	surface	11
2.29	title	11
2.30	xlabel	11
2.31	ylabel	11
<b>3</b>	<b>OctPlot Properties</b>	<b>12</b>
3.1	set & get	12
3.2	figure properties	12
3.2.1	Background color of figures	12
3.3	Axes properties	13
3.3.1	XLim, YLim and ZLim	13

3.4	LineStyle (line specifiers) . . . . .	13
3.5	Colors . . . . .	15
3.5.1	ColorSpec (color specifiers) . . . . .	15

# Chapter 1

## OctPlot

### 1.1 activating OctPlot

After starting Octave, enter at the command prompt **toggle\_octplot**. This will unactivate gnuplot. For future plots, in this Octave session, OctPlot will be used.

## Chapter 2

# OctPlot Commands

### 2.1 area

### 2.2 axis

### 2.3 bar, barh

**Syntax:**

```
bar(y)
bar(x, y)
bar(x, y, width)
bar(x, y, ColorSpec)
bar(x, y, [r g b])
bar(x, y, width, ColorSpec)
bar(x, y, width, [r g b])
h = bar(...)
```

```
barh(y)
barh(x, y)
barh(x, y, width)
barh(x, y, ColorSpec)
barh(x, y, [r g b])
barh(x, y, width, ColorSpec)
barh(x, y, width, [r g b])
h = barh(...)
```

**Description:**

A bar chart displays the values in a vector as horizontal or vertical bars.

**bar(y)** draws one bar for each element in *y*. The x-axis scale ranges from 1 to `length(y)` when *y* is a vector.

**bar(x,y)** draws a bar for each element in *y* at locations specified in *x*, where *x* is a monotonically increasing vector defining the x-axis intervals for the vertical bars.

**bar(...,width)** sets the relative bar width and controls the separation of bars within a group. The default width is 0.8, so if you do not specify *x*, the bars within a group have a slight separation. If width is 1, the bars within a group touch one another.

`bar(..., ColorSpec)` sets the *FaceColor* of bar. Default color is "r" (red). Also possible is an rgb-tripl `[1 1 0]` or `[0.1 0.4 0.7]`.

`barh(...)`, and `h = barh(...)` create horizontal bars. `y` determines the bar length. The vector `x` is a monotonic vector defining the *y-axis* intervals for horizontal bars. The *y-axis* will be drawn from bottom to top.

**Example 1:**

Plot a bell shaped curve:

```
x = -2.9:0.2:2.9;  
bar(x,exp(-x.*x))  
set(gcf,'Color','w')  
set(gca,'XLim',[-2.5 2.5])
```

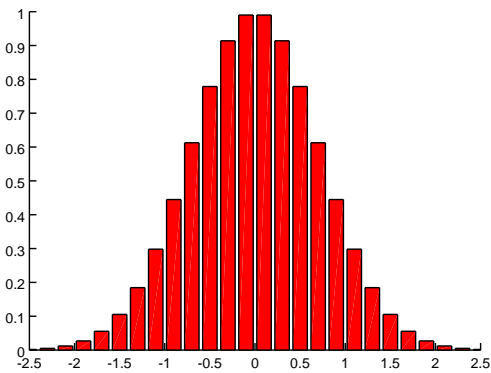


Figure 2.1: bar plot

**Example 2:**

Plot a horizontal bell shaped curve :

```
x = -2.9:0.2:2.9;  
barh(x,exp(-x.*x))  
set(gcf,'Color','w')  
set(gca,'YLim',[-2.5 2.5])
```

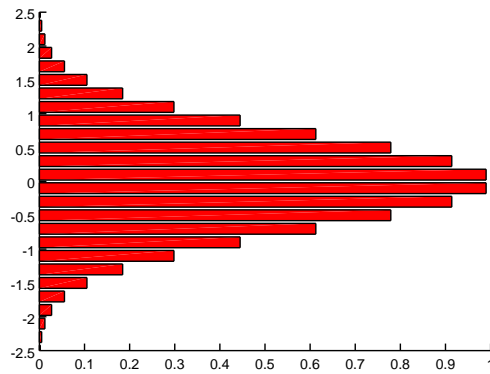


Figure 2.2: horizontal bar plot

**Example 3:**

Plot a horizontal bell shaped curve colored with rgb-tripl :

```
x = -2.9:0.2:2.9;  
barh(x,exp(-x.*x),[0.25 0.5 0.75])  
set(gcf,'Color','w')  
set(gca,'YLim',[-2.5 2.5])
```

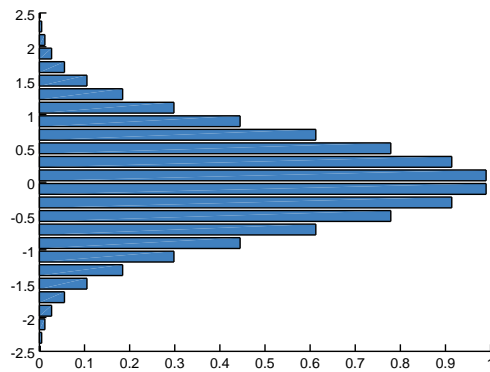


Figure 2.3: horizontal bar plot

- 2.4 cla
- 2.5 clf
- 2.6 colorbar
- 2.7 colormap
- 2.8 contour
- 2.9 contourc
- 2.10 freqz
- 2.11 gca
- 2.12 gcf
- 2.13 grid
- 2.14 hold
- 2.15 ishold
- 2.16 line
- 2.17 loglog
- 2.18 patch

**Syntax:**

```
patch(x,y)
patch(x,y,args)
handle = patch(...)
```

**Description:**

patch is the low-level graphics function for creating patch graphics objects. A patch object is one or more polygons defined by the coordinates of its vertices.

**patch(x,y)** adds the filled two-dimensional patch to the current axes. Standard color is red.

**patch(x,y,args)** adds the filled two-dimensional patch to the current axes and applies the remainder of the arguments. These arguments are taken as pairs; ... *'PropertyName', 'PropertyValue', ...*

**Remarks:**

If you do not specify a color when plotting more than one line, plot automatically cycles through the colors in the order specified by the current axes ColorOrder property. blue red green yellow cyan magenta black white



**Properties:**

blabla

## 2.19 pcolor

## 2.20 plot

The command **plot** can be used like described below:

**Syntax:**

```
plot(y)
plot(x,y)
h = plot( ... )
```

**Description:**

**plot(Y)** plots the columns of Y versus their index if Y is a real number. Complex numbers are not allowed.

**plot(X1,Y1,...)** plots all lines defined by Xn versus Yn pairs. If only Xn or Yn is a matrix, the vector is plotted versus the rows or columns of the matrix, depending on whether the vector's row or column dimension matches the matrix. If the matrix is nxn, columns are preferred.

**plot(X1,Y1,LineStyle,...)** plots all lines defined by the Xn,Yn,LineStyle triples, where LineSpec is a line specification that determines line type, marker symbol, and color of the plotted lines. You can mix Xn,Yn,LineStyle triples with Xn,Yn pairs: plot(X1,Y1,X2,Y2,LineStyle,X3,Y3).

**h = plot(X1,Y1,X1,Y2)** returns the handles to line objects. In this case, there are two handles in h. The first handle in h ( access with h(1) ) will hold the handle for line (X1,Y1) and so on.

**Remarks:**

If you do not specify a color when plotting more than one line, plot automatically cycles through the colors in the order specified by the current axes ColorOrder property. blue red green yellow cyan magenta black white

## 2.21 print

## 2.22 semilogx

## 2.23 semilogy

## 2.24 specgram

## 2.25 stem

**Syntax:**

```
stem(y)
stem(x, y)
stem(..., 'fill')
stem(..., LineSpec)
stem(x, y, ..., [r g b], ...)
stem(x, y, ..., [r g b], ...)
```

```
stem(x, y, ..., [r g b], ..., [r g b])
h = stem(...)
```

**Description:**

A bar chart displays the values in a vector as horizontal or vertical bars.

**bar(y)** draws one bar for each element in y. The x-axis scale ranges from 1 to length(y) when y is a vector.

**bar(x,y)** draws a bar for each element in y at locations specified in x, where x is a monotonically increasing vector defining the x-axis intervals for the vertical bars.

**bar(...,width)** sets the relative bar width and controls the separation of bars within a group. The default width is 0.8, so if you do not specify x, the bars within a group have a slight separation. If width is 1, the bars within a group touch one another.

**bar(...,'color specifier')** sets the *FaceColor* of bar. Default color is "r" (red). Also possible is an rgb-tripl *[1 1 0]* or *[0.1 0.4 0.7]*.

**barh(...)**, and **h = barh(...)** create horizontal bars. y determines the bar length. The vector x is a monotonic vector defining the *y-axis* intervals for horizontal bars. The *y-axis* will be drawn from bottom to top.

## 2.26 subplot

**Syntax:**

```
subplot(m,n,p)
h = subplot(m,n,p)
```

**Description:**

Create and control multiple axes.

**subplot(m,n,p)** creates m x n axes. p defines which will be used actually.

**h = subplot(m,n,p)** subplot returns the handle of the axes of the current subplot.

**Example 1:**

Plot a bell shaped curve and a bell shaped stem:

```
x = -2.9:0.2:2.9;  
subplot(1,2,1);  
bar(x,exp(-x.*x));  
set(gca,'XLim',[-2.5 2.5]);  
subplot(1,2,2);  
stem(x,exp(-x.*x), "fill");  
set(gca,'XLim',[-2.5 2.5]);  
set(gcf,'Color','w');
```

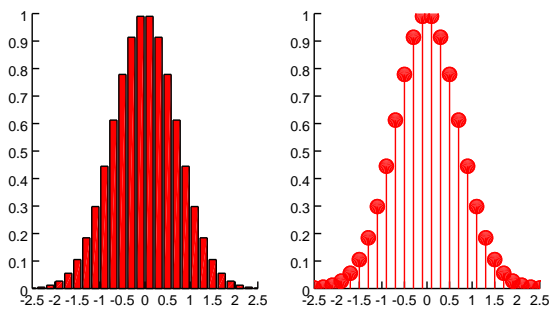


Figure 2.4: 1x2 subplot

**Example 2:**

Plot some bell shaped curve and a pie plot :

```
x = -2.9:0.2:2.9;
subplot(2,2,1);
bar(x,exp(-x.*x),[0.2 0.2 1]);
set(gca,'XLim',[-2.5 2.5]);
subplot(2,2,2);
stem(x,exp(-x.*x), "fill", 'm');
set(gca,'XLim',[-2.5 2.5]);
subplot(2,2,3);
plot(x,exp(-x.*x), "r-");
set(gca,'XLim',[-2.5 2.5]);
subplot(2,2,4);
x = 1:5;
pie(x);
set(gcf,'Color','w');
```

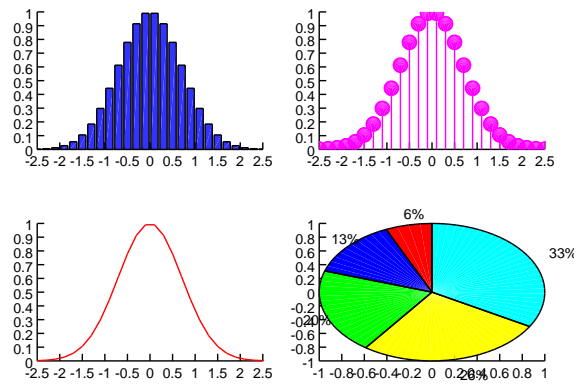


Figure 2.5: 4x4 subplot

**2.27** text

**2.28** surface

**2.29** title

**2.30** xlabel

**2.31** ylabel

## Chapter 3

# OctPlot Properties

### 3.1 set & get

#### Modifying Properties:

You can set and query graphics object properties in one way:  
The set and get commands enable you to set and query the values of properties.

#### Syntax:

```
set(handle,'PropertyName','PropertyValue');  
returnValue = get(handle,'PropertyName');
```

### 3.2 figure properties

#### Modifying Properties:

You can set and query graphics object properties in one way:  
The set and get commands enable you to set and query the values of properties.

#### Syntax:

```
set(gcf,'PropertyName','PropertyValue');
```

#### 3.2.1 Background color of figures

```
set(gcf,'Color','w');
```

This will change the background color to white. 'w' can be any other color defined in [link muss noch hin...](#)

```
set(gcf,'Color',[0.5 0.5 0.5]);
```

This will change the background color to gray. All other combinations of the rgb color model are also valid between the range of 0 to 1.

## 3.3 Axes properties

### 3.3.1 XLim, YLim and ZLim

```
set(gca,'XLim',[xmin xmax]);  
set(gca,'YLim',[ymin ymax]);  
set(gca,'ZLim',[zmin zmax]);
```

## 3.4 LineSpec (line specifiers)

### Description:

LineSpec is not a command; it refers to the ways in which you specify line properties in Octplot:

LineSpec can be used in several different commands. Dependent what LineSpec specifies, it will change the look&feel of lines like line style, color and marker style.

### Example 1:

Plot a standard line:

```
x = 1:10;  
plot(x)  
set(gcf,'Color','w')
```

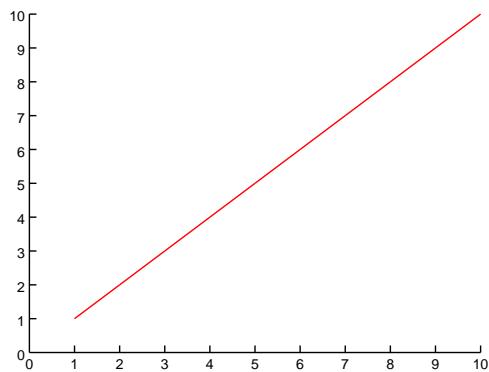


Figure 3.1: line plot

**Example 2:**

Plot a blue dotted line:

```
x = 1:10;  
plot(x,'b:');  
set(gcf,'Color','w')
```

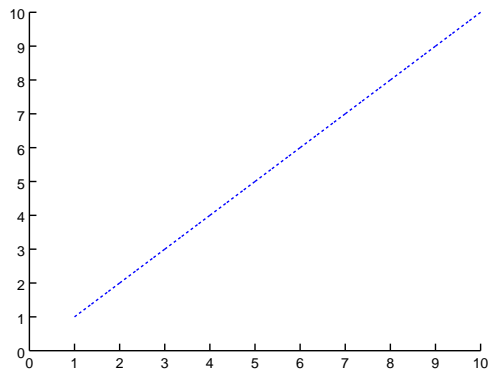


Figure 3.2: line plot with changed line color and line style. It doesn't matter if the LineSpec is written 'b:' or ':b'. The same line plot will be created!

**Example 3:**

Plot a blue dotted line with squared markers:

```
x = 1:10;  
plot(x,'b-s');  
set(gcf,'Color','w')
```

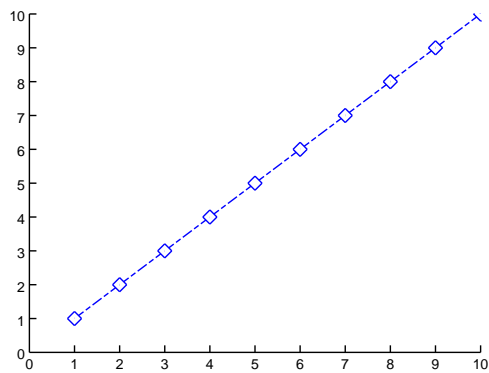


Figure 3.3: Also here, the order of the line specifiers doesn't matter.



## 3.5 Colors

### 3.5.1 ColorSpec (color specifiers)

**Description:**

ColorSpec is not a command; it refers to the three ways in which you specify colors in Octplot:

rgb-tripl	short name	long name
[0 0 0]	k	black
[1 0 0]	r	red
[1 1 0]	y	yellow
[0 1 0]	g	green
[0 1 1]	c	cyan
[1 0 1]	m	magenta
[1 1 1]	w	white

Table 3.1: Table of color specifications

# Bibliography

- [1] John W. Eaton  
GNU Octave Manual, Edition 3, PDF-Version, February 1997
- [2] Shai Ayal  
Octplot help and Octplot m-files