

```
>> v=[1,2,3]
```

```
v =
```

```
1      2      3
```

```
>> w=[4 4 4]
```

```
w =
```

```
4      4      4
```

```
>> v.*w
```

```
ans =
```

```
4      8     12
```

```
>> v*w
```

Error using *
Incorrect dimensions for matrix multiplication. Check that the number of columns in the first matrix matches the number of rows in the second matrix. To perform elementwise multiplication, use '.*'.

Related documentation

```
>> v.^3
```

```
ans =
```

```
1      8     27
```

```
>> a=[1 2 3;4 5 6]
```

```
a =
```

```
1      2      3  
4      5      6
```

```
>> b=[4 1 3;7 9 1]
```

```
b =
```

```
4      1      3  
7      9      1
```

```
>> a.*b
```

```
ans =
```

```
4      2      9  
28     45      6
```

```
>> a*b'
```

```
ans =
```

```
    15    28  
    39    79
```

```
>> a./b
```

```
ans =
```

```
    0.2500    2.0000    1.0000  
    0.5714    0.5556    6.0000
```

```
>> format long e
```

```
>> a./b
```

```
ans =
```

```
    2.5000000000000000e-01    2.0000000000000000e+00    1.0000000000000000e+00  
    5.714285714285714e-01    5.555555555555556e-01    6.0000000000000000e+00
```

```
>> %x=[val_in:passo:val_fin]
```

```
>> x=[-pi:0.1:pi]
```

```
x =
```

```
Columns 1 through 3
```

```
   -3.141592653589793e+00   -3.041592653589793e+00   -2.941592653589793e+00
```

```
Columns 4 through 6
```

```
   -2.841592653589793e+00   -2.741592653589793e+00   -2.641592653589793e+00
```

```
Columns 7 through 9
```

```
   -2.541592653589793e+00   -2.441592653589793e+00   -2.341592653589793e+00
```

```
Columns 10 through 12
```

```
   -2.241592653589793e+00   -2.141592653589793e+00   -2.041592653589793e+00
```

```
Columns 13 through 15
```

```
   -1.941592653589793e+00   -1.841592653589793e+00   -1.741592653589793e+00
```

```
Columns 16 through 18
```

```
   -1.641592653589793e+00   -1.541592653589793e+00   -1.441592653589793e+00
```

```
Columns 19 through 21
```

-1.341592653589793e+00	-1.241592653589793e+00	-1.141592653589793e+00
------------------------	------------------------	------------------------

Columns 22 through 24

-1.041592653589793e+00	-9.415926535897929e-01	-8.415926535897928e-01
------------------------	------------------------	------------------------

Columns 25 through 27

-7.415926535897928e-01	-6.415926535897931e-01	-5.415926535897930e-01
------------------------	------------------------	------------------------

Columns 28 through 30

-4.415926535897929e-01	-3.415926535897928e-01	-2.415926535897928e-01
------------------------	------------------------	------------------------

Columns 31 through 33

-1.415926535897931e-01	-4.159265358979303e-02	5.840734641020706e-02
------------------------	------------------------	-----------------------

Columns 34 through 36

1.584073464102067e-01	2.584073464102068e-01	3.584073464102069e-01
-----------------------	-----------------------	-----------------------

Columns 37 through 39

4.584073464102070e-01	5.584073464102071e-01	6.584073464102067e-01
-----------------------	-----------------------	-----------------------

Columns 40 through 42

7.584073464102068e-01	8.584073464102069e-01	9.584073464102070e-01
-----------------------	-----------------------	-----------------------

Columns 43 through 45

1.058407346410207e+00	1.158407346410207e+00	1.258407346410207e+00
-----------------------	-----------------------	-----------------------

Columns 46 through 48

1.358407346410207e+00	1.458407346410207e+00	1.558407346410207e+00
-----------------------	-----------------------	-----------------------

Columns 49 through 51

1.658407346410207e+00	1.758407346410207e+00	1.858407346410207e+00
-----------------------	-----------------------	-----------------------

Columns 52 through 54

1.958407346410207e+00	2.058407346410207e+00	2.158407346410207e+00
-----------------------	-----------------------	-----------------------

Columns 55 through 57

2.258407346410207e+00	2.358407346410207e+00	2.458407346410207e+00
-----------------------	-----------------------	-----------------------

Columns 58 through 60

2.558407346410207e+00 2.658407346410207e+00 2.758407346410207e+00

Columns 61 through 63

2.858407346410207e+00 2.958407346410207e+00 3.058407346410207e+00

```
>> x=[-pi:0.1:pi];
```

```
>> y=sin(x);
```

```
>> plot(x,y)
```

```
>> help plot
```

```
plot   Linear plot.
```

```
plot(X,Y) plots vector Y versus vector X. If X or Y is a matrix,
then the vector is plotted versus the rows or columns of the matrix,
whichever line up. If X is a scalar and Y is a vector, disconnected
line objects are created and plotted as discrete points vertically at
X.
```

```
plot(Y) plots the columns of Y versus their index.
```

```
If Y is complex, plot(Y) is equivalent to plot(real(Y),imag(Y)).
```

```
In all other uses of plot, the imaginary part is ignored.
```

Various line types, plot symbols and colors may be obtained with
`plot(X,Y,S)` where `S` is a character string made from one element
from any or all the following 3 columns:

b	blue	.	point	-	solid
g	green	o	circle	:	dotted
r	red	x	x-mark	-.	dashdot
c	cyan	+	plus	--	dashed
m	magenta	*	star	(none)	no line
y	yellow	s	square		
k	black	d	diamond		
w	white	v	triangle (down)		
		^	triangle (up)		
		<	triangle (left)		
		>	triangle (right)		
		p	pentagram		
		h	hexagram		

For example, `plot(X,Y,'c+:')` plots a cyan dotted line with a plus
at each data point; `plot(X,Y,'bd')` plots blue diamond at each data
point but does not draw any line.

`plot(X1,Y1,S1,X2,Y2,S2,X3,Y3,S3,...)` combines the plots defined by
the `(X,Y,S)` triples, where the `X`'s and `Y`'s are vectors or matrices
and the `S`'s are strings.

For example, `plot(X,Y,'y-',X,Y,'go')` plots the data twice, with a
solid yellow line interpolating green circles at the data points.

The `plot` command, if no color is specified, makes automatic use of
the colors specified by the axes `ColorOrder` property. By default,
`plot` cycles through the colors in the `ColorOrder` property. For

monochrome systems, plot cycles over the axes `LineStyleOrder` property.

Note that RGB colors in the `ColorOrder` property may differ from similarly-named colors in the (X,Y,S) triples. For example, the second axes `ColorOrder` property is medium green with RGB [0 .5 0], while `plot(X,Y,'g')` plots a green line with RGB [0 1 0].

If you do not specify a marker type, plot uses no marker.
If you do not specify a line style, plot uses a solid line.

`plot(AX,...)` plots into the axes with handle `AX`.

`plot` returns a column vector of handles to lineseries objects, one handle per plotted line.

The X,Y pairs, or X,Y,S triples, can be followed by parameter/value pairs to specify additional properties of the lines. For example, `plot(X,Y,'LineWidth',2,'Color',[.6 0 0])` will create a plot with a dark red line width of 2 points.

Example

```
x = -pi:pi/10:pi;
y = tan(sin(x)) - sin(tan(x));
plot(x,y,'--rs','LineWidth',2,...
      'MarkerEdgeColor','k',...
      'MarkerFaceColor','g',...
      'MarkerSize',10)
```

See also `plottools`, `semilogx`, `semilogy`, `loglog`, `plotyy`, `plot3`, `grid`, `title`, `xlabel`, `ylabel`, `axis`, `axes`, `hold`, `legend`, `subplot`, `scatter`.

Documentation for `plot`

Other functions named `plot`

```
>> plot(w, y, 'ro', 'linewidth', 3)
Error using plot
Vectors must be the same length.
```

```
>> plot(x, y, 'ro', 'linewidth', 3)
>> xlabel('asse x')
>> ylabel('asse y')
>> x = [-5:0.1:5];
>> y = exp(-x.^2);
>> plot(x, y, 'g', 'linewidth', 3)
>> x = linspace(0.001, 2, 10000);
Unrecognized function or variable 'linspace'.
```

Did you mean:

```
>> x = linspace(0.001, 2, 10000);
>> y = x.*sin(1./x);
>> plot(x, y, 'g', 'linewidth', 3)
>> x = linspace(0.1, 100, 1000);
>> y = sqrt((100.*(1-0.01.*x.^2).^2+0.02.*x.^2)./( (1-x.^2).^2+0.1.*x.^2));
```

```
>> plot(x,y,'g','linewidth',3)
>> a=max(y)
```

```
a =
```

```
3.130974289258856e+01
```

```
>> format long
>> b=min(y)
```

```
b =
```

```
0.014272926670735
```

```
>> loglog(x,y,'g','linewidth',3)
>>
>>
```