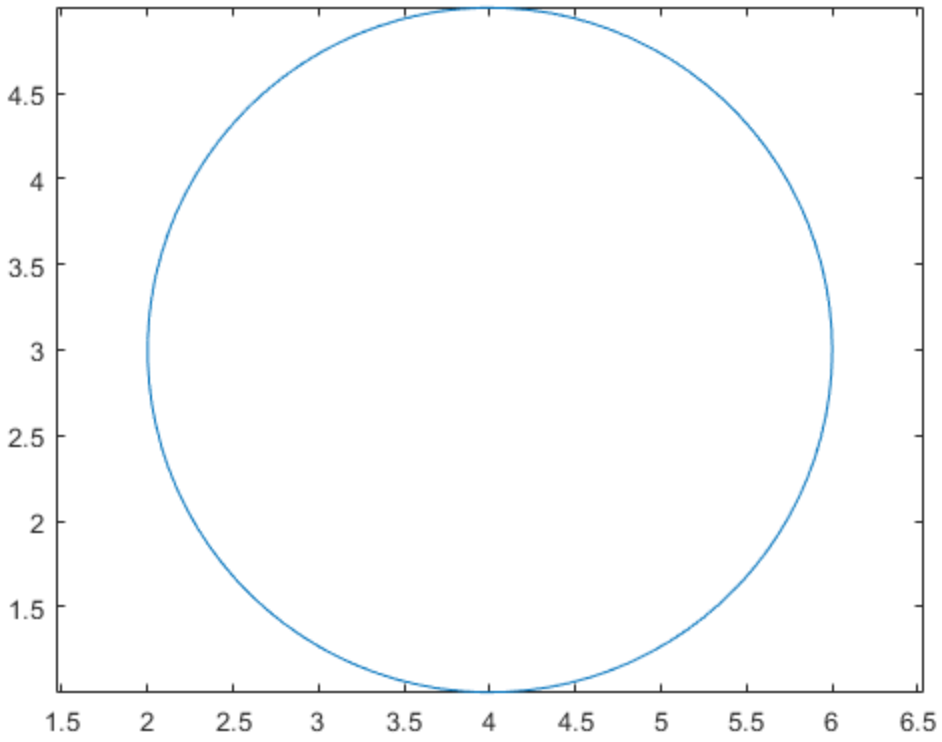
Plot a circle centered at the point (4,3) with a radius equal to 2. Use `axis equal` to use equal data units along each coordinate direction.

Open Live Script

```
r = 2;
xc = 4;
yc = 3;

theta = linspace(0,2*pi);
x = r*cos(theta) + xc;
y = r*sin(theta) + yc;
plot(x,y)
axis equal
```



Input Arguments

collapse all

Y — y values

scalar | vector | matrix

y values, specified as a scalar, a vector, or a matrix. To plot against specific *x* values you must also specify `X`.

Data Types: `single` | `double` | `int8` | `int16` | `int32` | `int64` | `uint8` | `uint16` | `uint32` | `uint64` | `categorical` | `datetime` | `duration`

X — x values

scalar | vector | matrix

x values, specified as a scalar, a vector, or a matrix.

Data Types: `single` | `double` | `int8` | `int16` | `int32` | `int64` | `uint8` | `uint16` | `uint32` | `uint64` | `categorical` | `datetime` | `duration`

`LineSpec` — Line style, marker, and color

character vector | string

Line style, marker, and color, specified as a character vector or string containing symbols. The symbols can appear in any order. You do not need to specify all three characteristics (line style, marker, and color). For example, if you omit the line style and specify the marker, then the plot shows only the marker and no line.

Example: `'--or'` is a red dashed line with circle markers

Line Style	**Description**
`-`	Solid line
`--`	Dashed line
`:`	Dotted line
`-.`	Dash-dot line
Marker	**Description**
`'o'`	Circle
`'+'`	Plus sign
`'*'`	Asterisk
`'.'`	Point
`'x'`	Cross
`'_'`	Horizontal line
`'|'`	Vertical line
`'s'`	Square
`'d'`	Diamond
`'^'`	Upward-pointing triangle
`'v'`	Downward-pointing triangle
`'>'`	Right-pointing triangle
`'<'`	Left-pointing triangle
`'p'`	Pentagram
`'h'`	Hexagram
Color	**Description**
`y`	yellow
`m`	magenta
`c`	cyan
`r`	red
`g`	green
`b`	blue
`w`	white
`k`	black

`ax` — Target axes

`Axes` object | `PolarAxes` object | `GeographicAxes` object

Target axes, specified as an `Axes` object, a `PolarAxes` object, or a `GeographicAxes` object. If you do not specify the axes and if the current axes are Cartesian axes, then the `plot` function uses the current axes. To plot into polar axes, specify the `PolarAxes` object as the first input argument or use the `polarplot` function. To plot into a geographic axes, specify the `GeographicAxes` object as the first input argument or use the `geoplot` function.

Name-Value Pair Arguments

Specify optional comma-separated pairs of `Name,Value` arguments. `Name` is the argument name and `Value` is the corresponding value. `Name` must appear inside quotes. You can specify several name and value pair arguments in any order as `Name1,Value1,...,NameN,ValueN`.

Example: `'Marker','o','MarkerFaceColor','red'`

The chart line properties listed here are only a subset. For a complete list, see Line Properties.

`'Color'` — Line color

`[0 0.4470 0.7410]` (default) | RGB triplet | hexadecimal color code | `'r'` | `'g'` | `'b'` | ...

Line color, specified as an RGB triplet, a hexadecimal color code, a color name, or a short name.

For a custom color, specify an RGB triplet or a hexadecimal color code.

- An RGB triplet is a three-element row vector whose elements specify the intensities of the red, green, and blue components of the color. The intensities must be in the range `[0,1]`; for example, `[0.4 0.6 0.7]`.
- A hexadecimal color code is a character vector or a string scalar that starts with a hash symbol (#) followed by three or six hexadecimal digits, which can range from `0` to `F`. The values are not case sensitive. Thus, the color codes `'#FF8800'`, `'#ff8800'`, `'#F80'`, and `'#f80'` are equivalent.

Alternatively, you can specify some common colors by name. This table lists the named color options, the equivalent RGB triplets, and hexadecimal color codes.

Color Name	Short Name	RGB Triplet	Hexadecimal Color Code	Appearance
`'red'`	`'r'`	`[1 0 0]`	`'#FF0000'`	
`'green'`	`'g'`	`[0 1 0]`	`'#00FF00'`	
`'blue'`	`'b'`	`[0 0 1]`	`'#0000FF'`	
`'cyan'`	`'c'`	`[0 1 1]`	`'#00FFFF'`	
`'magenta'`	`'m'`	`[1 0 1]`	`'#FF00FF'`	
`'yellow'`	`'y'`	`[1 1 0]`	`'#FFFF00'`	
`'black'`	`'k'`	`[0 0 0]`	`'#000000'`	
`'white'`	`'w'`	`[1 1 1]`	`'#FFFFFF'`	
`'none'`	Not applicable	Not applicable	Not applicable	No color

Here are the RGB triplets and hexadecimal color codes for the default colors MATLAB® uses in many types of plots.

RGB Triplet	Hexadecimal Color Code	Appearance
`[0 0.4470 0.7410]`	`'#0072BD'`	
`[0.8500 0.3250 0.0980]`	`'#D95319'`	
`[0.9290 0.6940 0.1250]`	`'#EDB120'`	

RGB Triplet	Hexadecimal Color Code	Appearance
[0.4940 0.1840 0.5560]	'#7E2F8E'	
[0.4660 0.6740 0.1880]	'#77AC30'	
[0.3010 0.7450 0.9330]	'#4DBEEE'	
[0.6350 0.0780 0.1840]	'#A2142F'	

Example: `'blue'`

Example: `[0 0 1]`

Example: `'#0000FF'`

`'LineStyle'` — Line style

`'-'` (default) | `'--'` | `':'` | `'-.'` | `'none'`

Line style, specified as one of the options listed in this table.

Line Style	Description	Resulting Line
`'-'`	Solid line	————
`'--'`	Dashed line	- — — — —
`':'`	Dotted line	.....
`'-.'`	Dash-dotted line	—·—·—·—
`'none'`	No line	No line

`'LineWidth'` — Line width

`0.5` (default) | positive value

Line width, specified as a positive value in points, where 1 point = 1/72 of an inch. If the line has markers, then the line width also affects the marker edges.

The line width cannot be thinner than the width of a pixel. If you set the line width to a value that is less than the width of a pixel on your system, the line displays as one pixel wide.

`'Marker'` — Marker symbol

`'none'` (default) | `'o'` | `'+'` | `'*'` | `'.'` | ...

Marker symbol, specified as one of the values listed in this table. By default, the object does not display markers. Specifying a marker symbol adds markers at each data point or vertex.

Value	Description
`'o'`	Circle
`'+'`	Plus sign
`'*'`	Asterisk
`'.'`	Point
`'x'`	Cross

Value	Description
`'_'`	Horizontal line
`'|'`	Vertical line
`'square'` or `'s'`	Square
`'diamond'` or `'d'`	Diamond
`'^'`	Upward-pointing triangle
`'v'`	Downward-pointing triangle
`'>'`	Right-pointing triangle
`'<'`	Left-pointing triangle
`'pentagram'` or `'p'`	Five-pointed star (pentagram)
`'hexagram'` or `'h'`	Six-pointed star (hexagram)
`'none'`	No markers

`'MarkerIndices'` — Indices of data points at which to display markers

`1:length(YData)` (default) | vector of positive integers | scalar positive integer

Indices of data points at which to display markers, specified as a vector of positive integers. If you do not specify the indices, then MATLAB displays a marker at every data point.

> **Note**
>
> To see the markers, you must also specify a marker symbol.

Example: `plot(x,y,'-o','MarkerIndices',[1 5 10])` displays a circle marker at the first, fifth, and tenth data points.

Example: `plot(x,y,'-x','MarkerIndices',1:3:length(y))` displays a cross marker every three data points.

Example: `plot(x,y,'Marker','square','MarkerIndices',5)` displays one square marker at the fifth data point.

`'MarkerEdgeColor'` — Marker outline color

`'auto'` (default) | RGB triplet | hexadecimal color code | `'r'` | `'g'` | `'b'` | ...

Marker outline color, specified as `'auto'`, an RGB triplet, a hexadecimal color code, a color name, or a short name. The default value of `'auto'` uses the same color as the `Color` property.

For a custom color, specify an RGB triplet or a hexadecimal color code.

- An RGB triplet is a three-element row vector whose elements specify the intensities of the red, green, and blue components of the color. The intensities must be in the range `[0,1]`; for example, `[0.4 0.6 0.7]`.
- A hexadecimal color code is a character vector or a string scalar that starts with a hash symbol (#) followed by three or six hexadecimal digits, which can range from `0` to `F`. The values are not case sensitive. Thus, the color codes `'#FF8800'`, `'#ff8800'`, `'#F80'`, and `'#f80'` are equivalent.

Alternatively, you can specify some common colors by name. This table lists the named color options, the equivalent RGB triplets, and hexadecimal color codes.

Color Name	Short Name	RGB Triplet	Hexadecimal Color Code	Appearance
`'red'`	`'r'`	`[1 0 0]`	`'#FF0000'`	

Color Name	Short Name	RGB Triplet	Hexadecimal Color Code	Appearance
'green'	'g'	[0 1 0]	'#00FF00'	
'blue'	'b'	[0 0 1]	'#0000FF'	
'cyan'	'c'	[0 1 1]	'#00FFFF'	
'magenta'	'm'	[1 0 1]	'#FF00FF'	
'yellow'	'y'	[1 1 0]	'#FFFF00'	
'black'	'k'	[0 0 0]	'#000000'	
'white'	'w'	[1 1 1]	'#FFFFFF'	
'none'	Not applicable	Not applicable	Not applicable	No color

Here are the RGB triplets and hexadecimal color codes for the default colors MATLAB uses in many types of plots.

RGB Triplet	Hexadecimal Color Code	Appearance
[0 0.4470 0.7410]	'#0072BD'	
[0.8500 0.3250 0.0980]	'#D95319'	
[0.9290 0.6940 0.1250]	'#EDB120'	
[0.4940 0.1840 0.5560]	'#7E2F8E'	
[0.4660 0.6740 0.1880]	'#77AC30'	
[0.3010 0.7450 0.9330]	'#4DBEEE'	
[0.6350 0.0780 0.1840]	'#A2142F'	

`'MarkerFaceColor'` — Marker fill color

`'none'` (default) | `'auto'` | RGB triplet | hexadecimal color code | `'r'` | `'g'` | `'b'` | ...

Marker fill color, specified as `'auto'`, an RGB triplet, a hexadecimal color code, a color name, or a short name. The `'auto'` option uses the same color as the `Color` property of the parent axes. If you specify `'auto'` and the axes plot box is invisible, the marker fill color is the color of the figure.

For a custom color, specify an RGB triplet or a hexadecimal color code.

- An RGB triplet is a three-element row vector whose elements specify the intensities of the red, green, and blue components of the color. The intensities must be in the range `[0,1]`; for example, `[0.4 0.6 0.7]`.
- A hexadecimal color code is a character vector or a string scalar that starts with a hash symbol (#) followed by three or six hexadecimal digits, which can range from `0` to `F`. The values are not case sensitive. Thus, the color codes `'#FF8800'`, `'#ff8800'`, `'#F80'`, and `'#f80'` are equivalent.

Alternatively, you can specify some common colors by name. This table lists the named color options, the equivalent RGB triplets, and hexadecimal color codes.

Color Name	Short Name	RGB Triplet	Hexadecimal Color Code	Appearance
'red'	'r'	[1 0 0]	'#FF0000'	

Color Name	Short Name	RGB Triplet	Hexadecimal Color Code	Appearance
'green'	'g'	[0 1 0]	'#00FF00'	
'blue'	'b'	[0 0 1]	'#0000FF'	
'cyan'	'c'	[0 1 1]	'#00FFFF'	
'magenta'	'm'	[1 0 1]	'#FF00FF'	
'yellow'	'y'	[1 1 0]	'#FFFF00'	
'black'	'k'	[0 0 0]	'#000000'	
'white'	'w'	[1 1 1]	'#FFFFFF'	
'none'	Not applicable	Not applicable	Not applicable	No color

Here are the RGB triplets and hexadecimal color codes for the default colors MATLAB uses in many types of plots.

RGB Triplet	Hexadecimal Color Code	Appearance
[0 0.4470 0.7410]	'#0072BD'	
[0.8500 0.3250 0.0980]	'#D95319'	
[0.9290 0.6940 0.1250]	'#EDB120'	
[0.4940 0.1840 0.5560]	'#7E2F8E'	
[0.4660 0.6740 0.1880]	'#77AC30'	
[0.3010 0.7450 0.9330]	'#4DBEEE'	
[0.6350 0.0780 0.1840]	'#A2142F'	

`'MarkerSize'` — Marker size

6 (default) | positive value

Marker size, specified as a positive value in points, where 1 point = 1/72 of an inch.

`'DatetimeTickFormat'` — Format for `datetime` tick labels

character vector | string

Format for `datetime` tick labels, specified as the comma-separated pair consisting of `'DatetimeTickFormat'` and a character vector or string containing a date format. Use the letters `A-Z` and `a-z` to construct a custom format. These letters correspond to the Unicode® Locale Data Markup Language (LDML) standard for dates. You can include non-ASCII letter characters such as a hyphen, space, or colon to separate the fields.

If you do not specify a value for `'DatetimeTickFormat'`, then `plot` automatically optimizes and updates the tick labels based on the axis limits.

Example: `'DatetimeTickFormat','eeee, MMMM d, yyyy HH:mm:ss'` displays a date and time such as `Saturday, April 19, 2014 21:41:06`.

The following table shows several common display formats and examples of the formatted output for the date, Saturday, April 19, 2014 at 9:41:06 PM in New York City.