

Plotting

Dr. Mihail

September 25, 2018

Visualize

One of the most important uses of packages like MATLAB is their rich data visualization capability. Visualizing data often yields insights that would otherwise be impossible.

Figures

Graphic visualizations in MATLAB are produced in Figure windows. The MATLAB syntax to create a figure is the following:

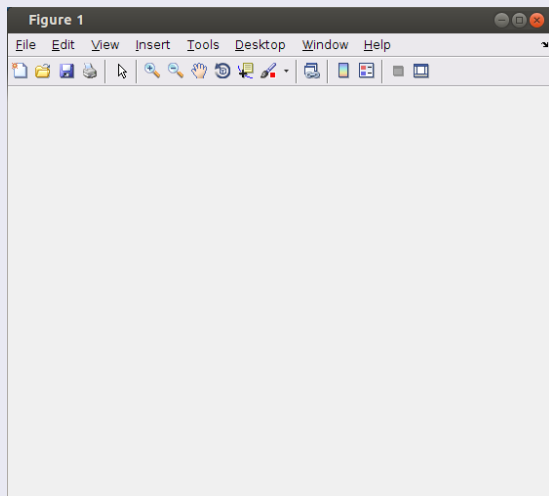
```
figure(figure_number)
```

example:

```
figure(1)
```

will create a new window titled "Figure 1". There can be several Figure windows opened simultaneously. Switching between them is done using the `figure()` function.

Figure Window



Clearing Figure Window

To clear, use the `clf` command. Example:

```
>> figure(1)
>> clf
>>
```

Say we want to plot the function:

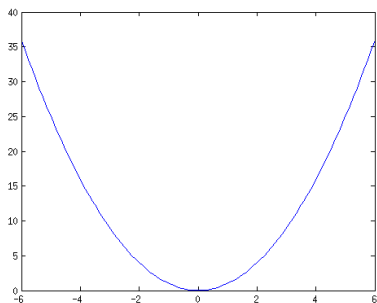
$$y = x^2$$

Entire script

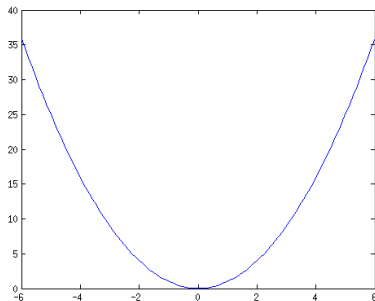
```
x = linspace(-6, 6, 100); % 100 equally spaced x's  
y = x.^2;  
figure(1);clf;  
plot(x, y);
```

1D Function Plot

The result:



1D Function Plot



Notes:

- `plot(x, y)` default behavior is to connect points in order (x_1, y_1) connected to (x_2, y_2) , connected to (x_3, y_3) , etc.
- `x` and `y` can be either column or row vectors

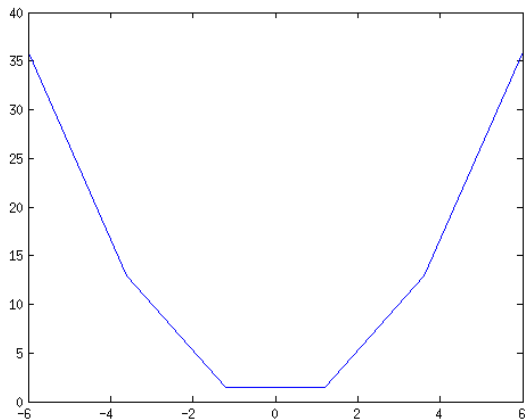
Let's change the number of samples to 6:

```
x = linspace(-6, 6, 6); % 6 equally spaced x's  
y = x.^2;  
figure(1);clf;  
plot(x, y);
```

What will the graph look like?

1D Function Plot

The choppy result:

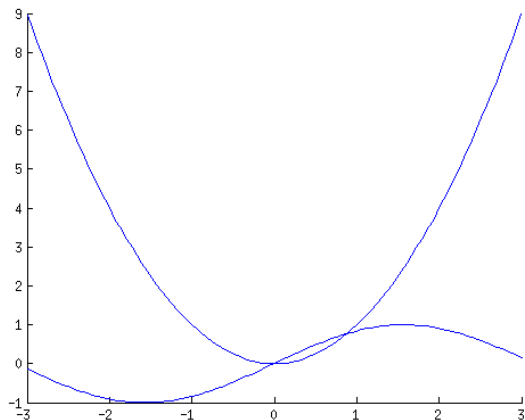


Multiple graphs same figure

To graph multiple plots on the same figure, use `hold on;`. This will hold on to existing graphics in the Figure window until `hold off;` is called. For example:

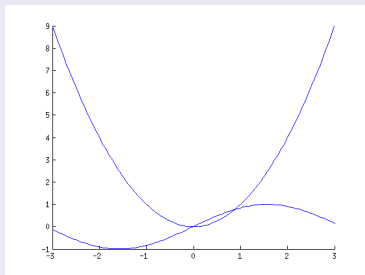
```
x = linspace(-3, 3, 100);  
y1 = x.^2;  
y2 = sin(x);  
figure(1);clf;hold on;  
plot(x, y1);  
plot(x, y2);
```

Two 1D Function Plots



Note how MATLAB automatically sets the x-limits and y-limits.

Two 1D Function Plots



Each of the two functions has 100 data points. We can change the default behavior of `plot` using the following syntax `plot(x, y, LineSpec)`, where `LineSpec` is a string that sets the line style, marker symbol and color. For more info, type `>> doc plot`.

`plot(x, y, LineSpec)`

Line style, marker symbol, and color, specified as a string. The elements of the string can appear in any order, and you can omit one or more options from the string specifier. If you omit the line style and specify the marker character, then the plot shows only the marker and no line.

If Y is a matrix and you specify a color with `LineStyle`, then all lines are plotted using the specified color. If you specify a marker type or line style and do not specify a color, then MATLAB cycles through the predefined color order.

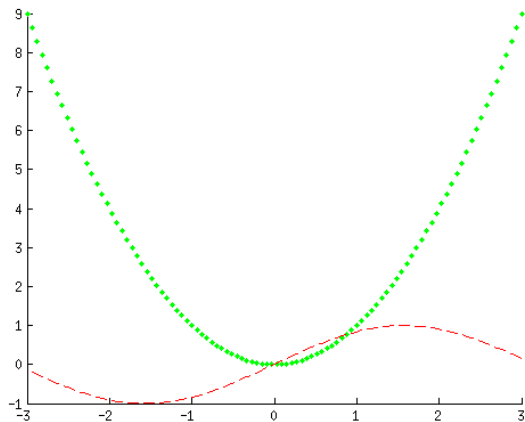
Example: Example: `'--or'` is a red dashed line with circle markers

Multiple graphs same figure

Let's change our previous example, make quadratic dotted green and the sine wave a red dashed line.

```
x = linspace(-3, 3, 100);  
y1 = x.^2;  
y2 = sin(x);  
figure(1);clf;hold on;  
plot(x, y1, '.g');  
plot(x, y2, '--r');
```

Two 1D Function Plots



Plot Properties

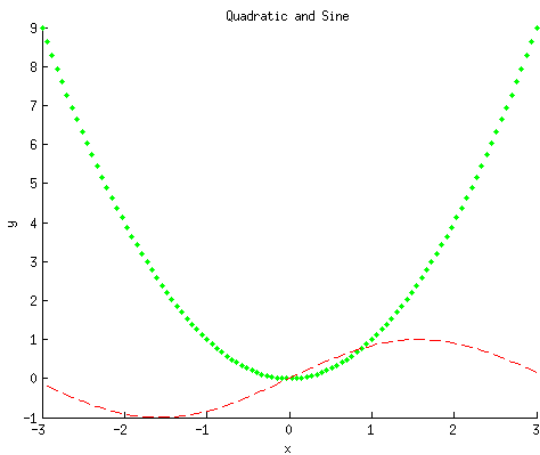
Useful plots will have their axes labeled and a title. In MATLAB:

- Title: `title('example title')`
- X-Axis Label: `xlabel('example x-label title')`
- Y-Axis Label: `ylabel('example y-label title')`

Example:

```
x = linspace(-3, 3, 100);
y1 = x.^2;y2 = sin(x);
figure(1);clf;hold on;
plot(x, y1, '.g');
plot(x, y2, '--r');
xlabel('x'); ylabel('y');title('Quadratic and Sine');
```

Two 1D Function Plots

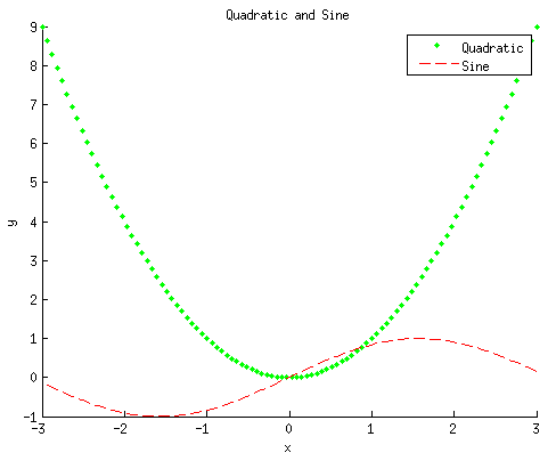


What about a legend? In MATLAB:

`legend({'name of plot1', 'name of plot 2', ...});`. Example:

```
x = linspace(-3, 3, 100);  
y1 = x.^2;y2 = sin(x);  
figure(1);clf;hold on;  
plot(x, y1, '.g');  
plot(x, y2, '--r');  
xlabel('x'); ylabel('y');title('Quadratic and Sine');  
legend({'Quadratic', 'Sine'});
```

Two 1D Function Plots

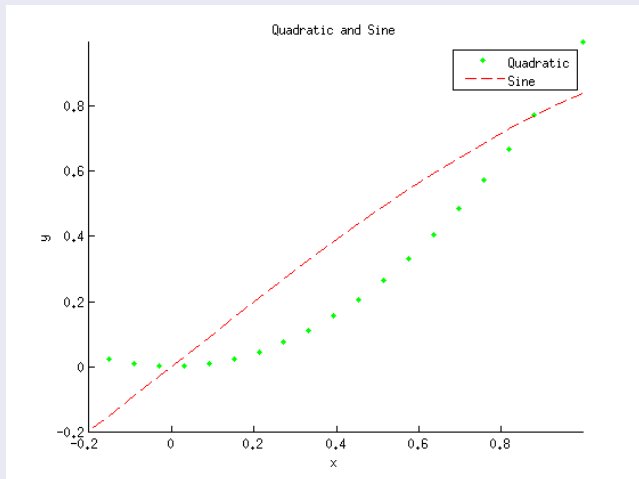


Plot Properties

What about x and y-axis limits legend? In MATLAB:
`xlim([lower, upper])` and `ylim([lower, upper]);`. Example:

```
x = linspace(-3, 3, 100);  
y1 = x.^2;y2 = sin(x);  
figure(1);clf;hold on;  
plot(x, y1, '.g');  
plot(x, y2, '--r');  
xlim([-0.2 1]);ylim([-0.2 1]);  
xlabel('x'); ylabel('y');title('Quadratic and Sine');  
legend({'Quadratic', 'Sine'});
```

Two 1D Function Plots



Other plots

- polar: polar plot using polar coordinates
- bar, barh, bar3, bar3h, pie, pie3
- use **doc function_name** or your textbook to find out more about them.

Dividing up the Figure window

- Done using `subplot(m, n, p)`
- Figure window split into a grid of size m by n , and p is the current subwindow, numbered from left to right, top to bottom