Plotting with MATLAB

Introduction

MATLAB allows plotting lines , surfaces, … using predefined functions,

The simplest and most commonly used plotting function is :

plot(x,y),

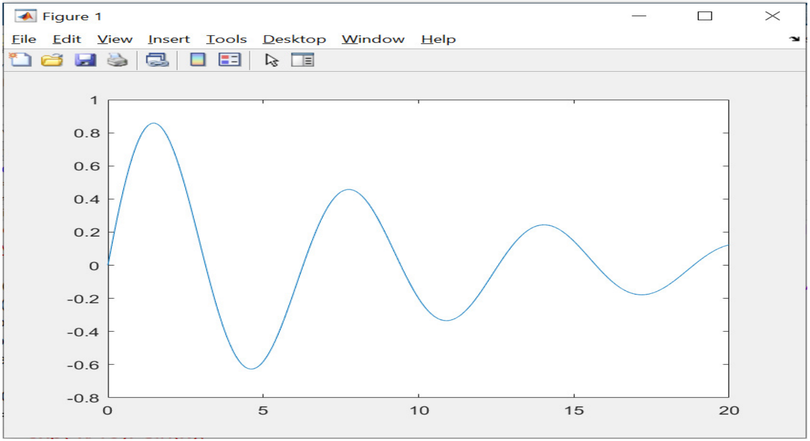
where x and y are simply vectors containing the x and y coordinates of the data to be plotted.

Example : plot $f(x) = e^{-x/10} \sin(x)$

```
>> x = 0:0.1:20;
>> y = exp(-x/10).*sin(x);

>>plot(x,y)
```

The vectors containing the x and y data must be the same length.

1. When MATLAB executes a plotting command, a new Figure Window opens with the plot in it.
2. It offers a graphical user interface and a figure toolbox to control plots

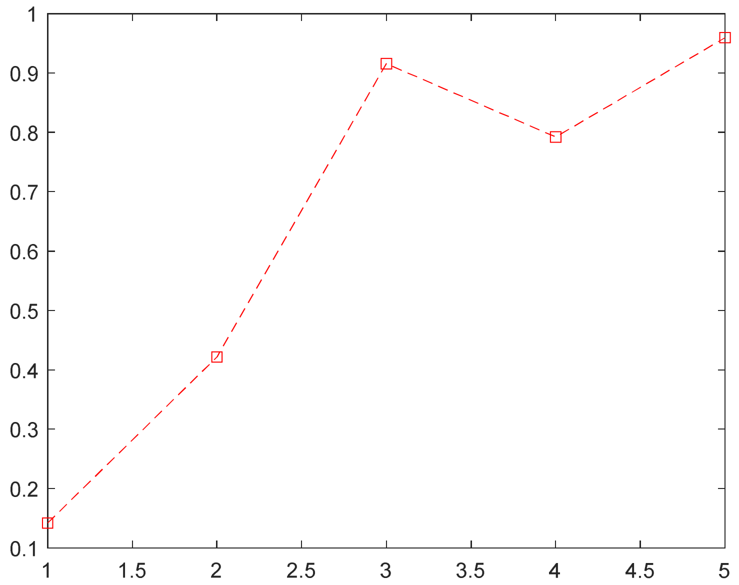


Line plots

Change the line color, marker style, and line style by adding a string argument

plot(x_axis_data , y_axis_data, 'color/marker/linestyle')

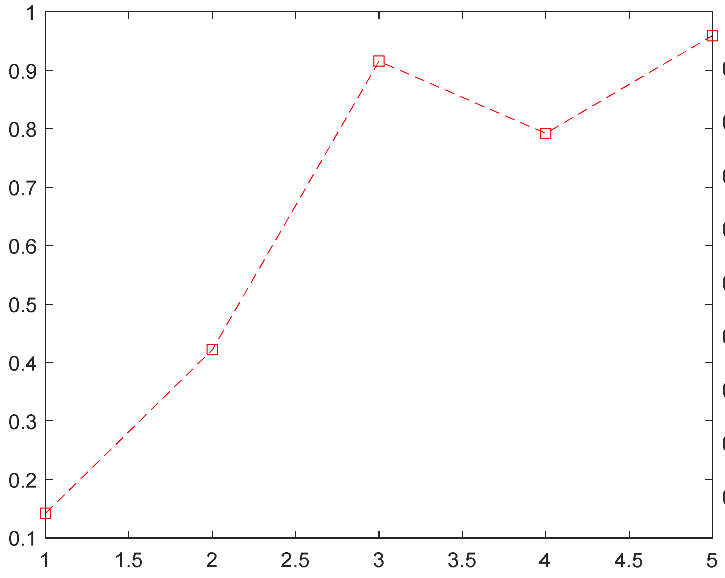
Example :   plot(1:5,rand(1,5),'rs--')

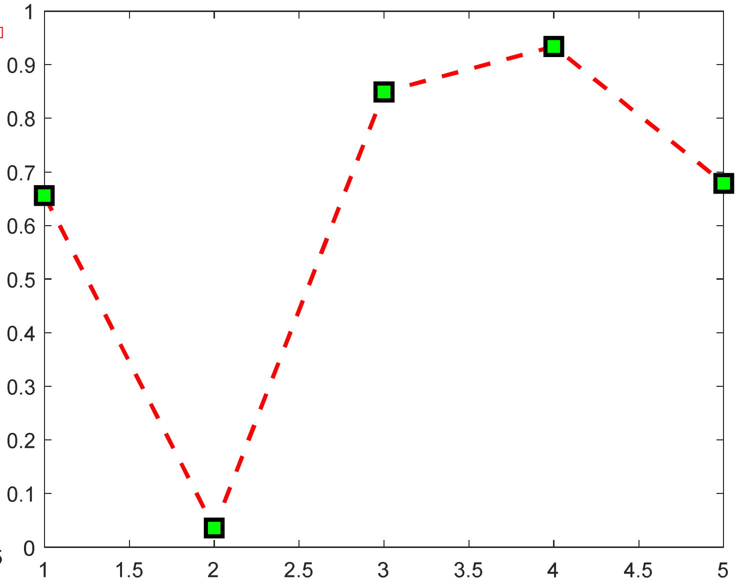


color		marker		linestyle	
b	blue	.	point	**-**	**solid**
g	green	o	circle	:	dotted
r	red	x	x-mark	-.	dashdot
c	cyan	+	plus	--	dashed
m	magenta	*	star	(none)	no line
y	yellow	s	square		
k	black	d	diamond		
w	white	v	triangle(down)		
		^	triangle (up)		
	MATLAB cycles the line color through the default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

Everything on a line can be customized

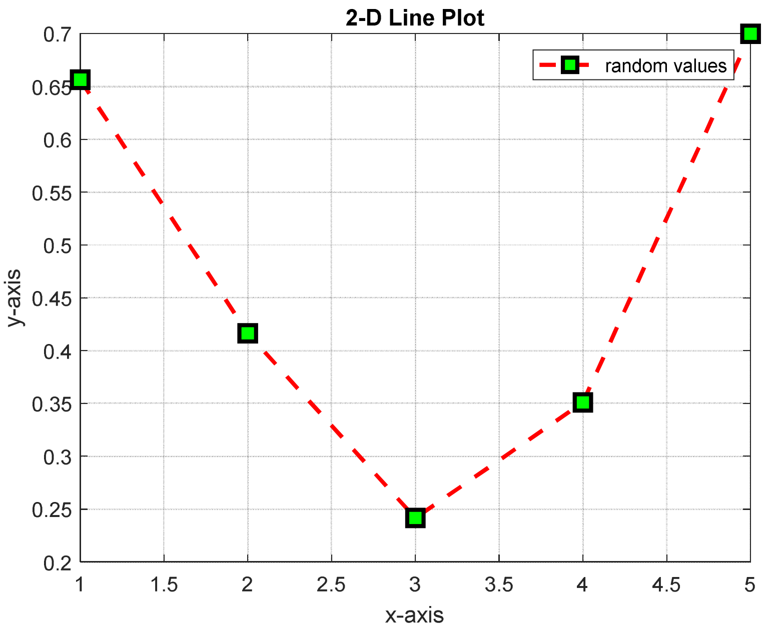
```
>>plot(1:5,rand(1,5), 'rs--' ,'LineWidth',2,'MarkerEdgeColor','k','MarkerFaceColor','g','MarkerSize',10)
```





Add a title and axis labels to the graph using the title, xlabel, and ylabel functions.

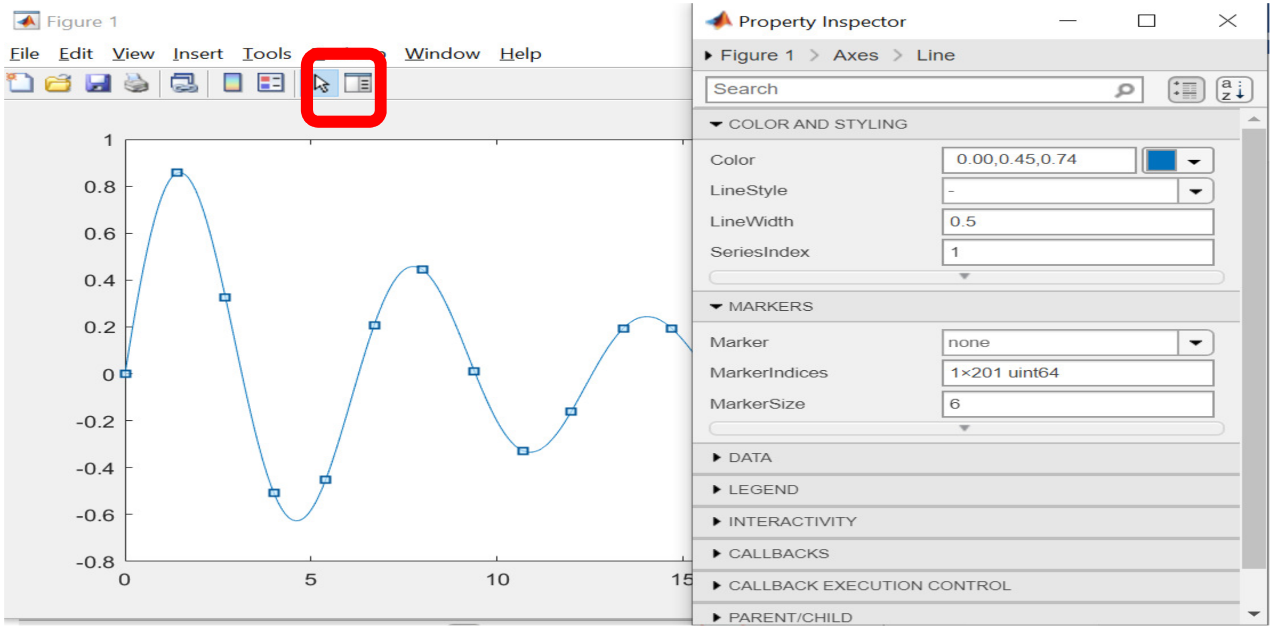
```
>>title('2-D Line Plot')

>>xlabel('x-axis')

>>ylabel('y-axis')

>> legend('random values')

>>grid on
```



Plot Tools

Plot properties can also be manipulated interactively (without having to issue commands) by clicking on the Show Plot Tools icon in the Figure Window toolbar.

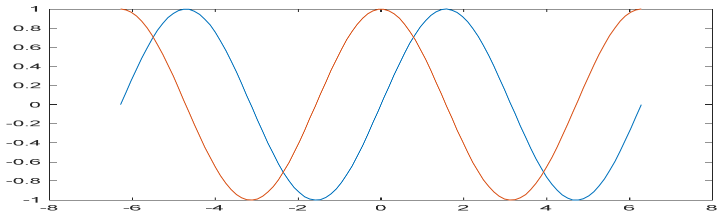
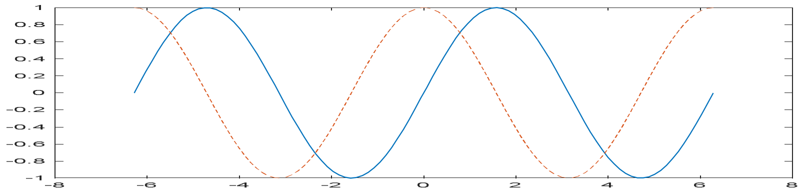
Properties such as the axis limits, gridlines, linestyle, colour and thickness, text font type and size, and legend etc... can all be adjusted be clicking on the appropriate parts of the plot.

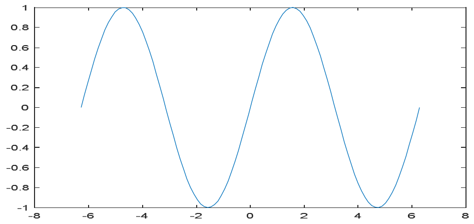
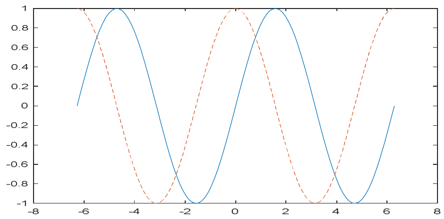


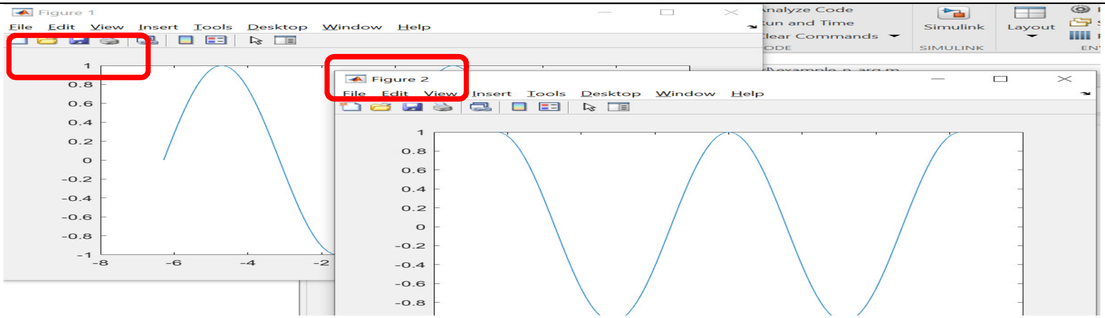
Color 0.00,0.45,0.74
LineStyle -
LineWidth 0.5
SeriesIndex 1
Marker none
MarkerIndices 1×201 uint64
MarkerSize 6

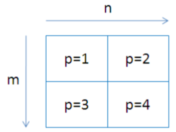
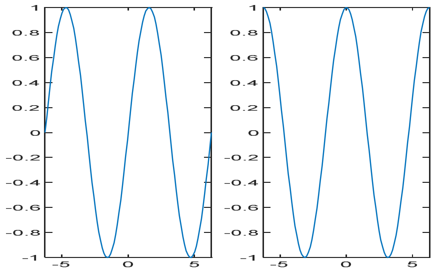
Plot Multiple data set Lines

```
x = linspace(-2*pi,2*pi);
y1 = sin(x);
y2 = cos(x);
```

`plot(x,y1,x,y2)`	
`plot(x,y1,x,y2,'--')`	

`plot(x,y1)`
`hold on`
`plot(x,y2)`
`hold off`
	 → 

plot(x,y1)
figure
plot(x,y2)
	 Figure 1
	File Edit View Insert Tools Desktop Window Help
	Figure 2
	File Edit View Insert Tools Desktop Window Help
	Analyze Code
	Run and Time
	Clear Commands
	Simulink
	Layout
	SIMULINK

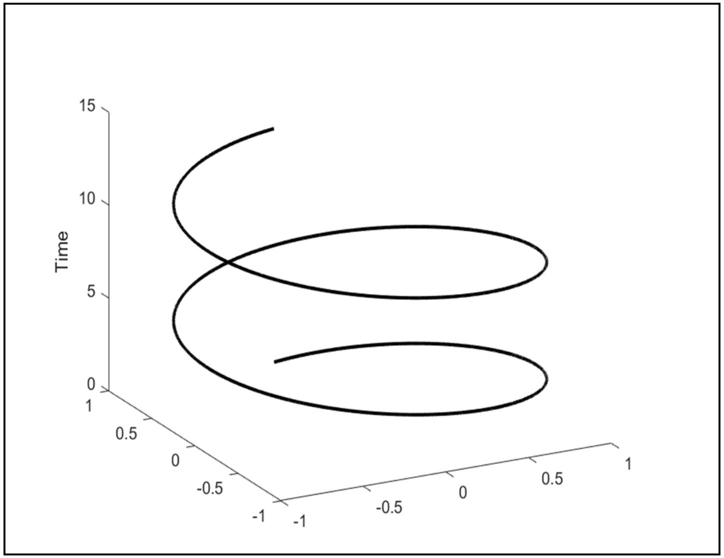
subplot(1,4,1) >> plot(x,y1) >> subplot(1,4,2) >> plot(x,y2) Subplot(m,n,p) 	

Plotting in 3 dimensions

Three-dimensional curves is just as easy as in 2D

Example :

```
> T=0:0.001:4*pi;

>> x=sin(T);

>> y=cos(T);

>> z=T;

>> plot3(x,y,z,'k','LineWidth',2)

>> zlabel('Time');
```

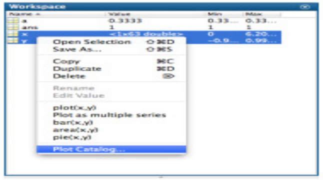


plot (using plot3) the points on a helix in 3D space.

Matrix visualization

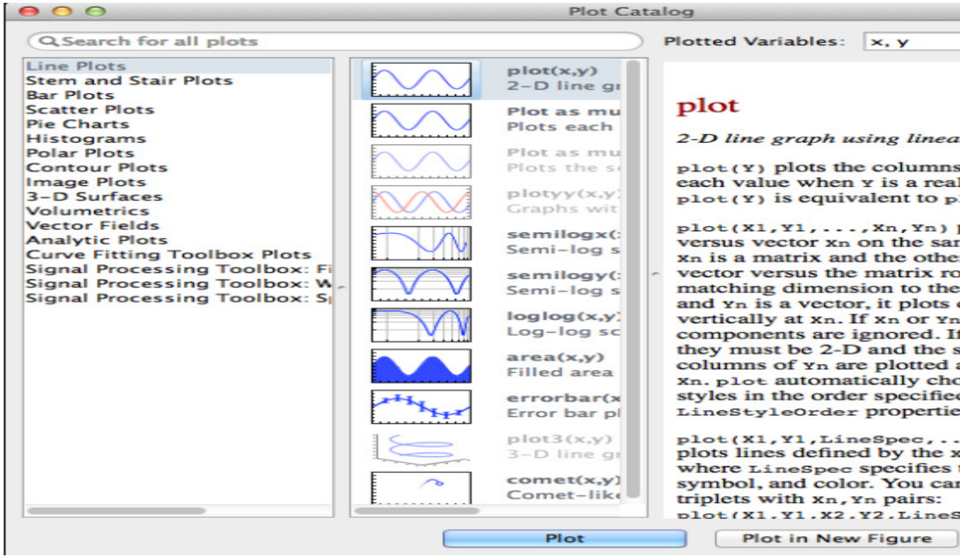
- MATLAB has many built-in plot types,
- a great way of getting a quick overview of all the different plot types is :
  - ➢ to select a variable in your Workspace Browser,
  - ➢ click on the disclosure triangle next to the plot toolbar icon
  - ➢ and select More plots...,
  - ➢ as shown in Figures bellow,

    this will launch the Plot Catalog



Name ▲	Value	Min	Max
a	0.3333	0.33...	0.33...
ans	1	1	1
x	<1x63 double>	0	6.20...
y		−0.9...	0.99...

(a) Accessing the *Plot Catalog*



plot

2-D line graph using linea

plot(Y) plots the columns
each value when Y is a real
plot(Y) is equivalent to p

plot(X1,Y1,...,Xn,Yn)
versus vector Xn on the sar
Xn is a matrix and the othe
vector versus the matrix ro
matching dimension to the
and Yn is a vector, it plots
vertically at Xn. If Xn or Yn
components are ignored. If
they must be 2-D and the s
columns of Yn are plotted a
Xn. plot automatically cho
styles in the order specified
LineStyleOrder propertie

plot(X1,Y1,LineSpec,...
plots lines defined by the x
where LineSpec specifies t
symbol, and color. You car
triplets with Xn,Yn pairs:
plot(X1,Y1,X2,Y2,LineS

(b) The *Plot Catalog*

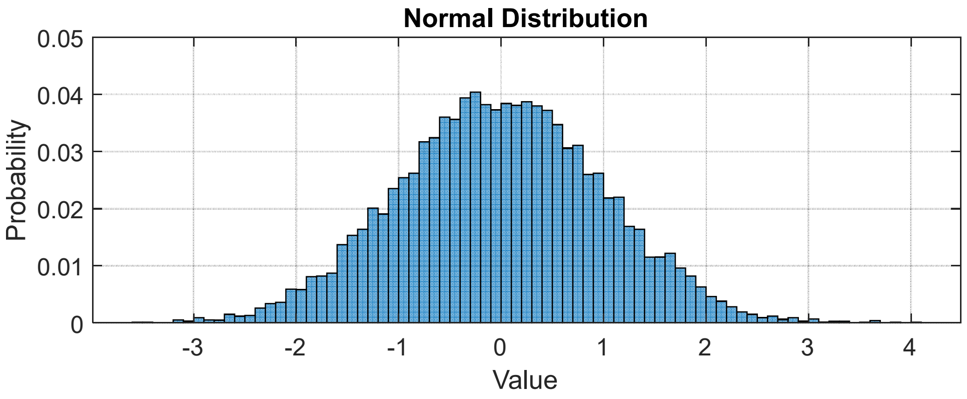
Exercice

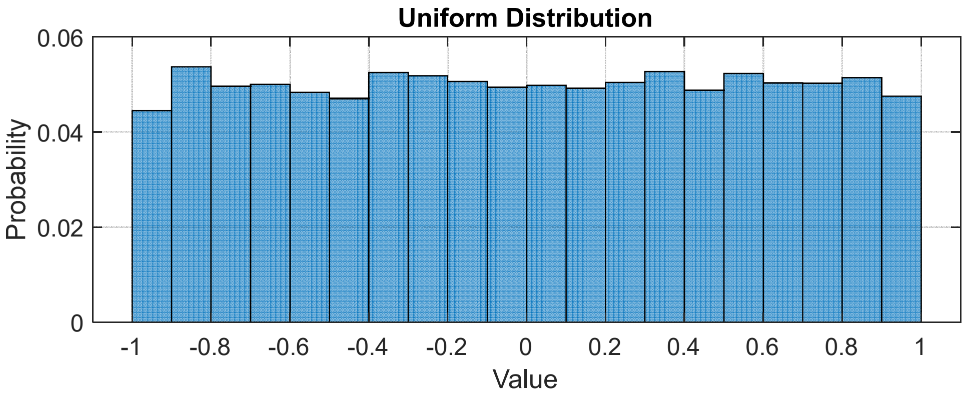
Plot normal distribution and uniform distribution using MATLAB, by following these steps:

1. Generate random samples from each distribution.

2. Plot the histograms of the generated samples to visualize the distributions.

```
% Define parameters
sample_size = 10000; % Number of samples
mu = 0; % Mean of the normal distribution
sigma = 1; % Standard deviation of the normal distribution
lower_bound = -1; % Lower bound of the uniform distribution
upper_bound = 1; % Upper bound of the uniform distribution

% Generate random samples
normal_samples = mu + sigma * randn(sample_size, 1); % Normal distribution
uniform_samples = lower_bound + (upper_bound - lower_bound) * rand(sample_size, 1);
% Uniform distribution

% Plot histograms
figure;
subplot(2, 1, 1);
histogram(normal_samples, 'Normalization', 'probability', 'BinWidth', 0.1);
title('Normal Distribution');
xlabel('Value');
ylabel('Probability');
grid on;

subplot(2, 1, 2);
histogram(uniform_samples, 'Normalization', 'probability', 'BinWidth', 0.1);
title('Uniform Distribution');
xlabel('Value');
ylabel('Probability');
grid on;
```


Normal Distribution
Probability
0.05
0.04
0.03
0.02
0.01
0
-3
-2
-1
0
1
2
3
4
Value


Uniform Distribution
Probability
0.06
0.04
0.02
0
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
Value

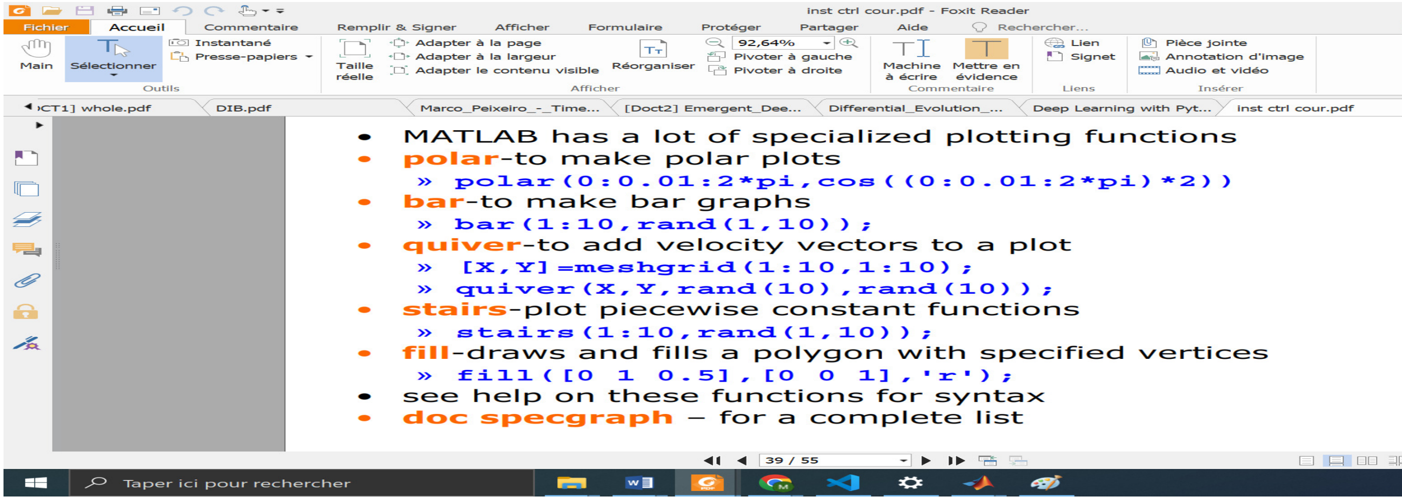
1) Dans les deux distributions , on va créer des nombres aléatoirement
2)
   a. Dans la loi uniforme, ces nombres sont compris entre 0 et 1
   b. Dans la loi normale ces nombres …..

3) Dans la loi normale, le nombre de valeurs proche de la moyenne est grand , et autant on s’éloigne de la moyenne, l’effectif diminue
4) Dans la loi uniforme, l’effectif des valeurs est presque égale : les valeurs sont dispersées uniformément dans l’intervalle borne inf, borne sup

'probability'	The height of each bar is the relative number of observations (number of observations in bin / total number of observations), and the sum of the bar heights is less than or equal to 1.

MATLAB offers lots of different plots.

the following 2D functions in MATLAB: loglog, semilogx, semilogy, plotyy, polar, fplot, fill, area, bar, barh, hist, pie, errorbar, scatter.



- MATLAB has a lot of specialized plotting functions
- **polar**-to make polar plots
  ```
  » polar(0:0.01:2*pi,cos((0:0.01:2*pi)*2))
  ```
- **bar**-to make bar graphs
  ```
  » bar(1:10,rand(1,10));
  ```
- **quiver**-to add velocity vectors to a plot
  ```
  » [X,Y]=meshgrid(1:10,1:10);
  » quiver(X,Y,rand(10),rand(10));
  ```
- **stairs**-plot piecewise constant functions
  ```
  » stairs(1:10,rand(1,10));
  ```
- **fill**-draws and fills a polygon with specified vertices
  ```
  » fill([0 1 0.5],[0 0 1],'r');
  ```
- see help on these functions for syntax
- **doc specgraph** – for a complete list

Ya aussi visualiser matrices et surf……

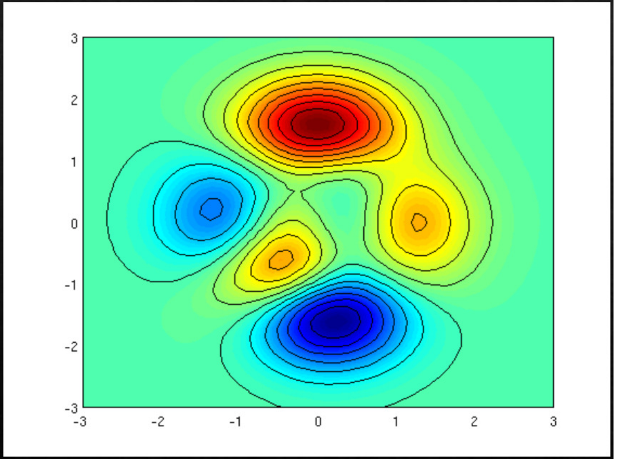
Meshgrid et autres very well++++etc ,   dans ++++ alot of exercices .pdf

Soukout 7or

```
% The code for function (falling.m) . File name should be the same as function name.
function height = falling(t)
   global GRAVITY
   height = 1/2*GRAVITY*t.^2;     % The height of a freely falling object
end

% The code for main program (main.m).  Should be in the same path with the function code.
global GRAVITY
GRAVITY = 32;
y = falling((0:.2:5)')
```

contour, pcolor

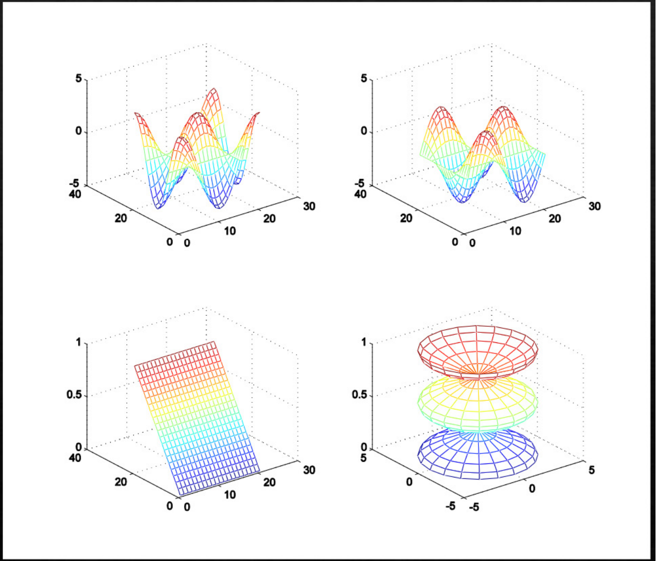
```
% Obtain data from evaluating peaks function
[x,y,z] = peaks;
% Create pseudocolor plot
pcolor(x,y,z)
% Smooth the colors
shading interp
% Hold the current graph
hold on
% Add the contour graph to the pcolor graph
contour(x,y,z,15,'k')    % 15-level, black line
% Return to default
hold off
```



Subfigures and layout

- **subplot, mesh**

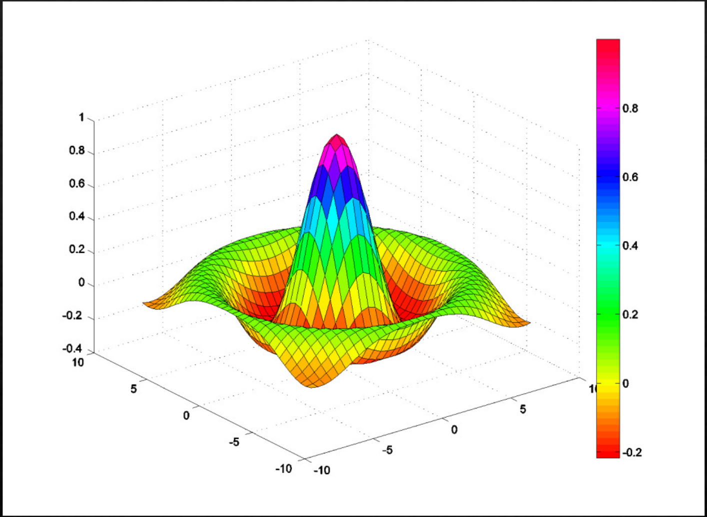
```
t = 0:pi/10:2*pi;
% cylinder with a self-defined profile
[X,Y,Z] = cylinder(4*cos(t));
subplot(2,2,1);   % left-up
mesh(X)
subplot(2,2,2);   % right-up
mesh(Y)
subplot(2,2,3);   % left-down
mesh(Z)
subplot(2,2,4);   % right-down
mesh(X,Y,Z)
```

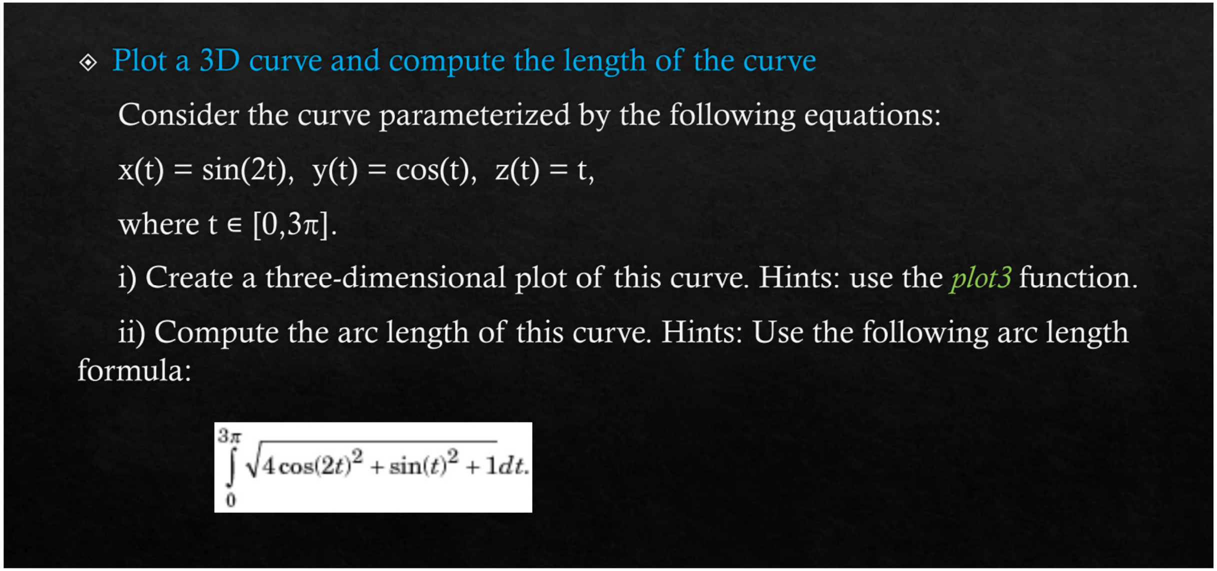


Color Surface Plot

- **surfc**

```
[X,Y] = meshgrid(-8:.5:8);
R = sqrt(X.^2 + Y.^2) + eps;
Z = sin(R)./R;   % sinc function
surfc(X,Y, Z)
colormap hsv     % color map
colorbar    % show color scaling
view([1 1 1])  % view angle
```





◈ Plot a 3D curve and compute the length of the curve

Consider the curve parameterized by the following equations:

x(t) = sin(2t), y(t) = cos(t), z(t) = t,

where t ∈ [0,3π].

i) Create a three-dimensional plot of this curve. Hints: use the *plot3* function.

ii) Compute the arc length of this curve. Hints: Use the following arc length formula:

$$\int_0^{3\pi} \sqrt{4\cos(2t)^2 + \sin(t)^2 + 1}\,dt.$$

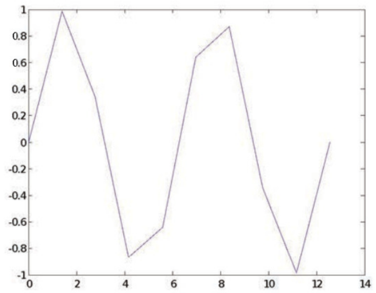
Exercice idea pr obtenir schéma ci-après

- Write a function with the following declaration:
  `function plotSin(f1)`

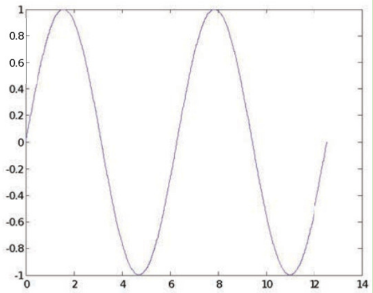
- In the function, plot a sine wave with frequency f1, on the interval [0,2π]: $\sin(f_1 x)$

- To get good sampling, use 16 points per period.

10 x values:



1000 x values:



Exercice sur « inst ctrl cour.pdf »

Plot the learning trajectory

- In helloWorld.m, open a new figure (use `figure`)
- Plot knowledge trajectory using `tVec` and `knowledgeVec`
- When plotting, convert `tVec` to days by using `secPerDay`
- Zoom in on the plot to verify that `halfTime` was calculated correctly