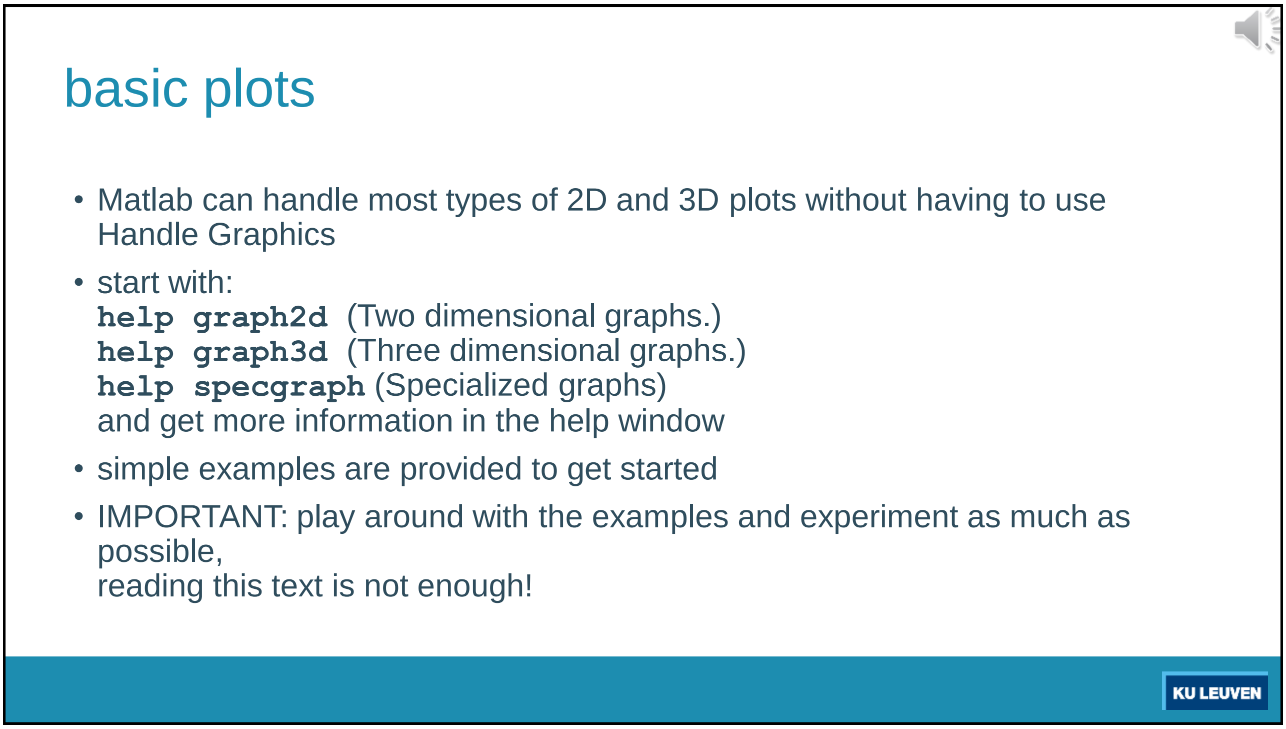
MATLAB

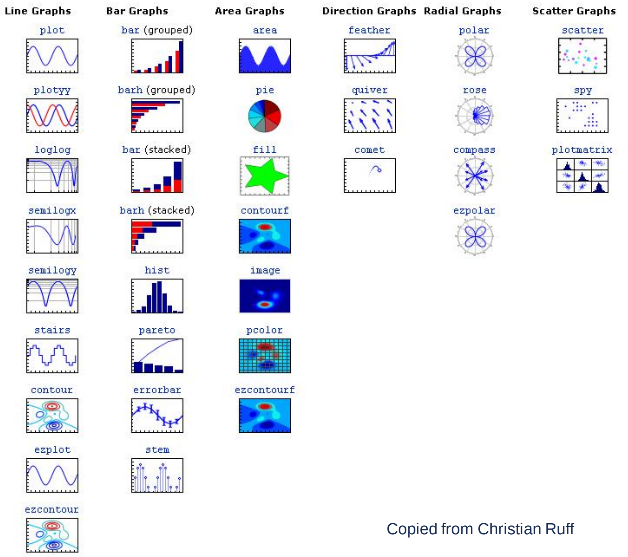
Visualization 2D



basic plots

- Matlab can handle most types of 2D and 3D plots without having to use Handle Graphics
- start with:
  `help graph2d` (Two dimensional graphs.)
  `help graph3d` (Three dimensional graphs.)
  `help specgraph` (Specialized graphs)
  and get more information in the help window
- simple examples are provided to get started
- IMPORTANT: play around with the examples and experiment as much as possible,
  reading this text is not enough!

Basic graphs: From 2-D …



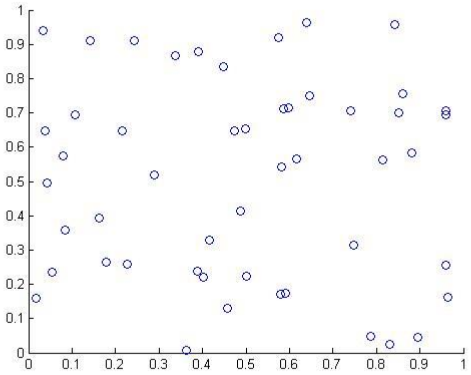
Copied from Christian Ruff

Relationship plots

- Scatter plot
  - type of display using Cartesian coordinates to display values for two variables for a set of data.
  - gives an idea of the relation between the two variables.
  - Matlab functions:
    - **`plot`**
    - **`scatter`**
- Bubble plot
  - similar to the scatter plot in which data are plotted on a two-dimensional x and y axis coordinate system. The difference is that a third data factor (z) controls the size / color of the scatter points.
  - Matlab functions:
    - **`scatter`**

Scatter plot

- either shows the relationships among the numeric values in several data series, or plots two groups of numbers as one series of XY coordinates.
- commonly used for scientific data.
- arrange the data: place x values in one row or column, and then enter corresponding y values in the adjacent rows or columns.
- *File: chart2D_scatter_01*

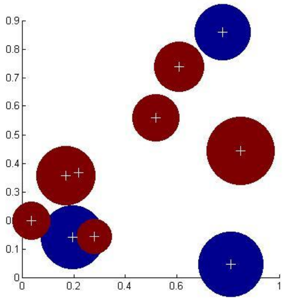


Scatter plot

```
N = 50; % Number of data points

% generate the data
xdat = rand(1,N);
ydat = rand(1,N);


% use the plot function
% specify the marker and no line style to draw only points
figure;
plot(xdat, ydat, 's');

% use the scatter function
figure;
scatter(xdat, ydat);
```

Bubble plot

- Bubble plots allow to change the size, shape, or color of each data point.
- Let the size or color of the plotted points represent an additional variable.
- **`scatter`**
- *File: chart2D_bubble_01*



KU LEUVEN

Bubble plot

```
N = 10; % Number of charges to place

xq=rand(1,N); % x positions of the charges
yq=rand(1,N); % y positions of the charges

q=100*rand(1,N)-50; % magnitude of charges (between -50 and 50)

color = 1.5+sign(q)/2; % sign(q) returns 1 or -1, so color is 1 or 2

size = abs(q)*100; % Make size of points bigger for bigger magnitude of q

scatter(xq,yq,size,color,'filled');

hold on; % add another plot on top
plot(xq, yq,'w+','MarkerSize',10) % add a cross in the center of the circles
```

KU LEUVEN

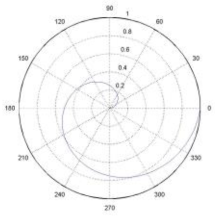
2D plots

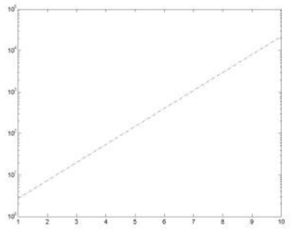
- **plot arrays of points**
  - **`Basics`**
    - **`plot`**: line-plots
      `loglog, semilogx, semilogy`: change the axis
  - **`More`**
    - **`polar`**: polar coordinates
    - **`area, fill`**: surface
    - **`stairs`**: stair plot
    - **`bar, pie`**: diagrams
    - **`contour, contourf`**: isolines
    - **`quiver`**: vector fields
    - **`gradient`**: utilities
- **plot functions, not just arrays of points**
  - **`fplot, ezplot`**

polar - semilog

- **`polar(theta,rho)`** creates a polar coordinate plot of the angle theta versus the radius rho.
- *File: plot2D_polar.m*

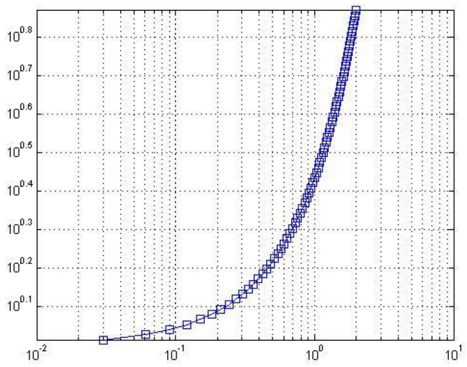
- **`semilogx`** and **`semilogy`** plot data as logarithmic scales for the x- and y-axis, respectively. logarithmic
- File: *plot2D_semilog.m*





loglog

- `loglog` plots data on a log-log scale
- File: *plot2D_loglog.m*

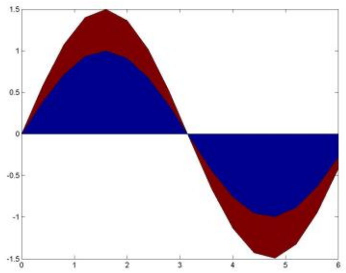


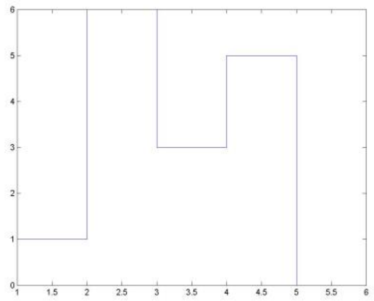
KU LEUVEN

area - stairs

- `area(Y)` Area fill of a two-dimensional plot.
- File: *plot2D_area.m*

- `stairs(Y)` draws a stairstep plot of the elements of Y.
- File: *plot2D_stairs.m*



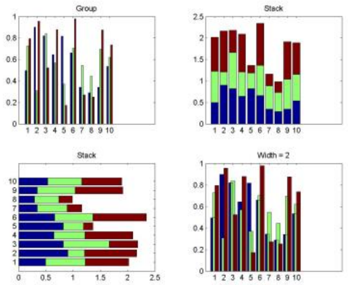


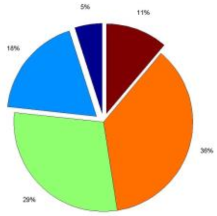
KU LEUVEN

bar - pie

- A **`bar`** chart displays the values in a vector or matrix as horizontal or vertical bars.
- File: *plot2D_bar.m*

- **`pie(X)`** draws a pie chart using the data in X. **`pie(X,explode)`** offsets a slice from the pie.
- File: *plot2D_pie.m*

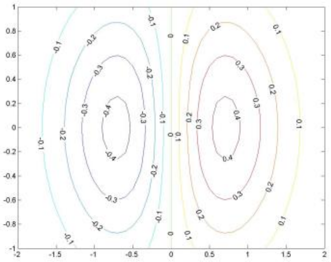


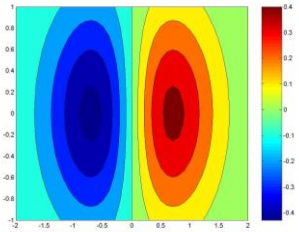


contour - contourf

- **`contour`** displays 2-D isolines generated from values given by a matrix Z.
- File: *plot2D_contour.m*

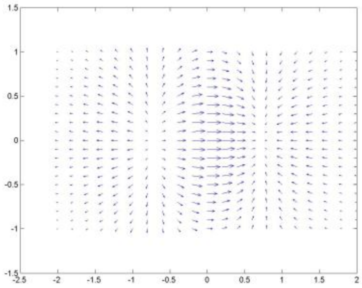
- **`contourf`** displays isolines and fills the areas between the isolines using constant colors. File: *plot2D_contourf.m*





quiver

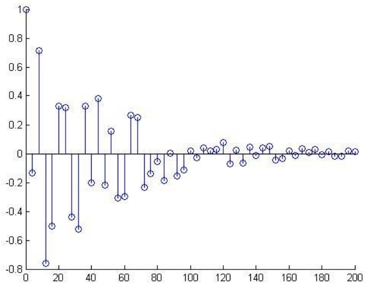
- **`quiver:`** displays velocity vectors as arrows with components (U,V) at the points (X,Y).
- File: *plot2D_quiver.m*



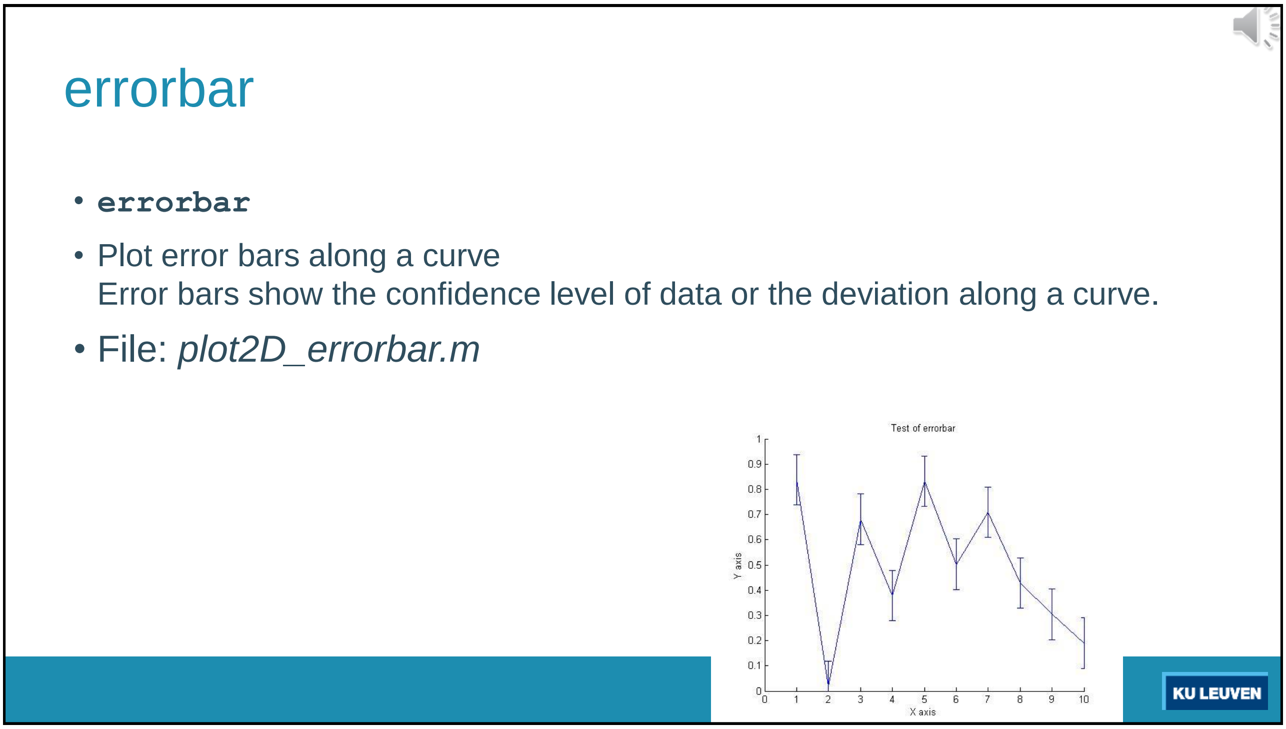
KU LEUVEN

stem

- For discrete-time signals, use the command `stem` which plots each point with a small open circle and a straight line.
- plot y[k] versus k:
  `stem(k,y)`
- use
  `stem(k,y,'filled')`
  to get circles that are filled
- File: *plot2D_stem.m*



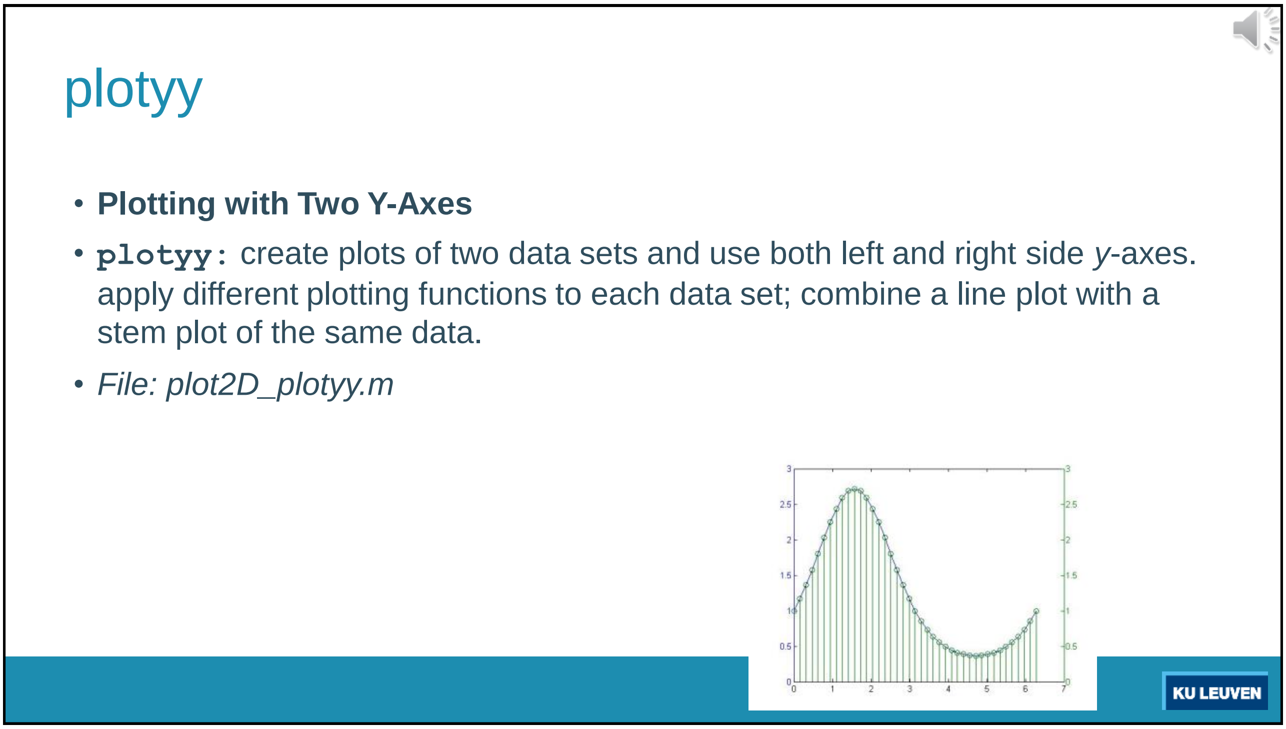
KU LEUVEN



errorbar

- **`errorbar`**
- Plot error bars along a curve
  Error bars show the confidence level of data or the deviation along a curve.
- File: *plot2D_errorbar.m*

Test of errorbar
Y axis
X axis
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
1
2
3
4
5
6
7
8
9
10
KU LEUVEN



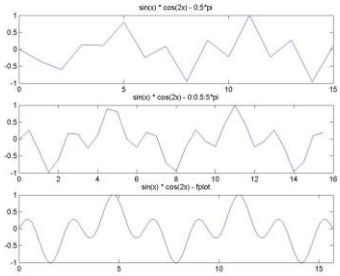
plotyy

- **Plotting with Two Y-Axes**
- **`plotyy:`** create plots of two data sets and use both left and right side *y*-axes. apply different plotting functions to each data set; combine a line plot with a stem plot of the same data.
- *File: plot2D_plotyy.m*

0
0.5
1
1.5
2
2.5
3
0
1
2
3
4
5
6
7
0
0.5
1
1.5
2
2.5
3
KU LEUVEN

fplot

- **`fplot`** plots a **function** defined by a m-function or function handles.
  The function must be of the form y = f(x), where x is a vector whose range specifies the limits
- `fplot` adaptively
  determines
  the sampling rate
- File: *plot2D_fplot.m*


sin(x) * cos(2x) - 0:5*pi
sin(x) * cos(2x) - 0:0.5:5*pi
sin(x) * cos(2x) - fplot