Visualisation with MATLAB

MATH3203

Dr P. M. Burrage

Some Useful Notes

- To print figures for inclusion in M/Word (and keep preview capability):
  - print –depsc –tiff –r300 myfig
- Basic plot command
  - Can specify up to 3 fns in one command
- Note that MATLAB does interpolation automatically, plus automatic axes, colours, line style.

Data Acquisition – Source of Data

- Output from MATLAB program
- Imported to MATLAB from another application program
- Load test.dat
  - E.g. 20x2 array called ‘test’
- fread or fscanf

2D Data Visualisation

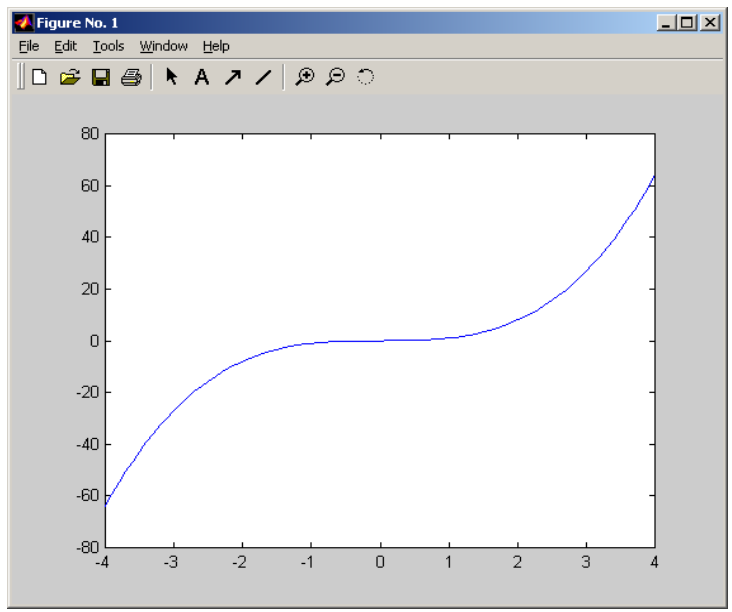
- Automatic interpolation
- Automatic axes, colour, line style
- Plot $y = x^3, \quad x \in [-4,4], \quad \Delta x = 0.1$

```
x=-4:0.1:4;
y=(x).^3;
plot(x,y,'b')
```

2D Data Visualisation

- Automatic interpolation
- Automatic axes, colour, line style
- Plot $y = x^3, \quad x \in [-4,4], \quad \Delta x = 0.1$

```
x=-4:0.1:4;
y=(x).^3;
plot(x,y,'b')
```



Colours

- b blue
- r red
- y yellow
- c cyan
- g green
- k black
- Etc.

Line Style

- plot(x,y,'—b')

:	Dots
-.	Dashed
-	Solid
--	Dashed
*	Asterisk
o	Circles
'+'	Plus signs
.	Point
'square'	Box
'diamond'	Diamonds

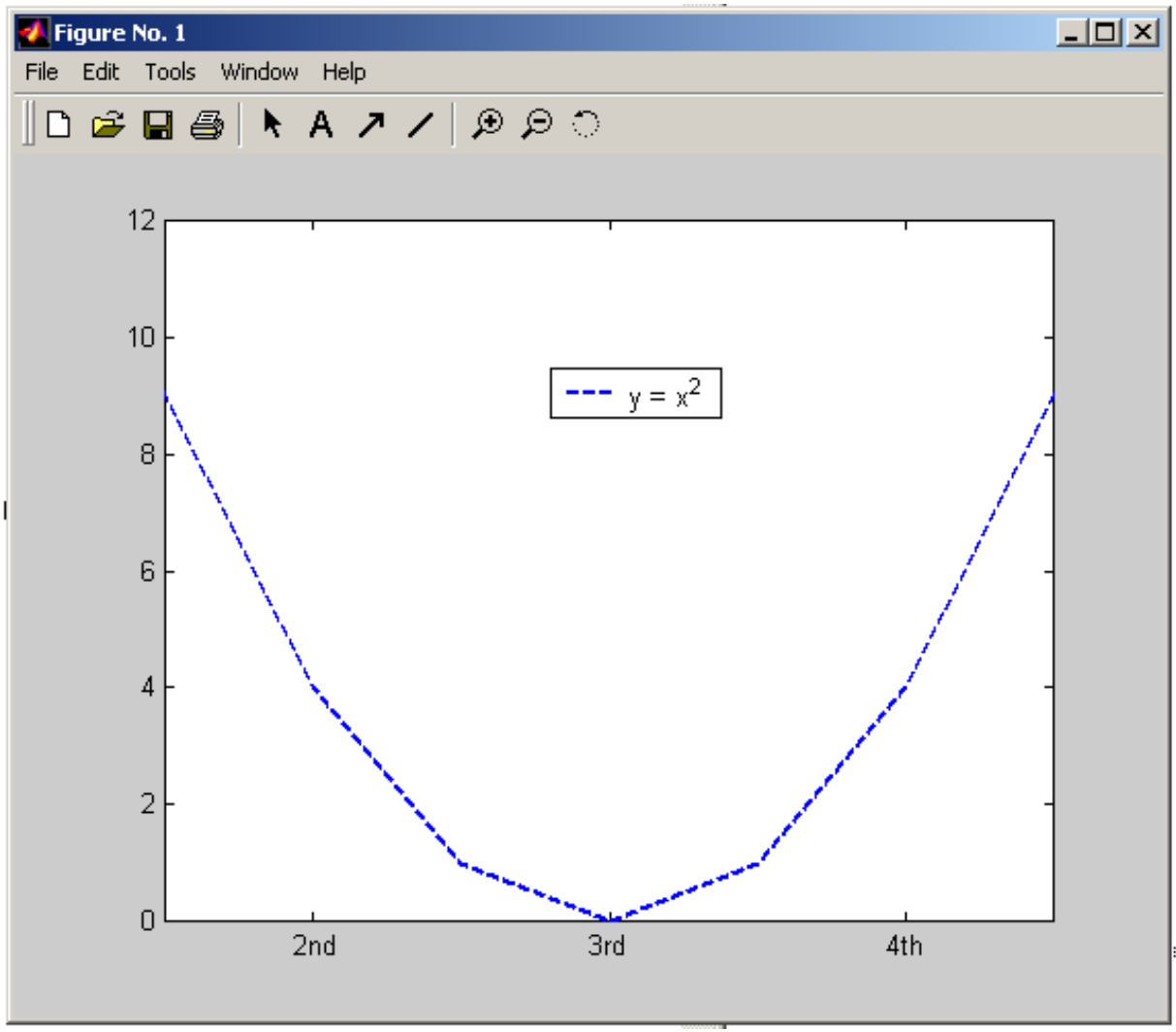
Labelling & Graph Styles

- Label the x-axis: xlabel(‘x’)
- Label the y-axis: ylabel(‘y’)
- Provide a title: title(‘Graph of y=x^3’);
- Plot a grid: grid
- For graph style, can specify
  - Line thickness
  - Graph tick marks
  - Axis limits
  - Legend

Example

```
x = -4 : 1.0 : 4;
y = x .^ 2;
fig1 = plot(x,y,'b');
set(fig1,'LineWidth',2);
set(fig1,'LineStyle','- -');
set(gca,'Xtick',-4 : 2 : 4);
set(gca,'XTickLabel',{'1st','2nd','3rd','4th','5th'});
axis([-3 3 0 12])
legend(fig1,'y = x ^ 2')
```

- And the result:



Now try Exercise 1!

- Axes control
  - Automatic
  - Can override: axis([xmin xmax ymin ymax])
- Other axis commands
  - axis('square') – square dimensions
  - axis('equal') – equal unit spacing on x and y axes; ensures correct aspect ratio
  - axis('tight') – forces x,y axis limits to max data values
  - axis('off') – hides the axes, tickmarks, labels
  - axis('on') – reinstates

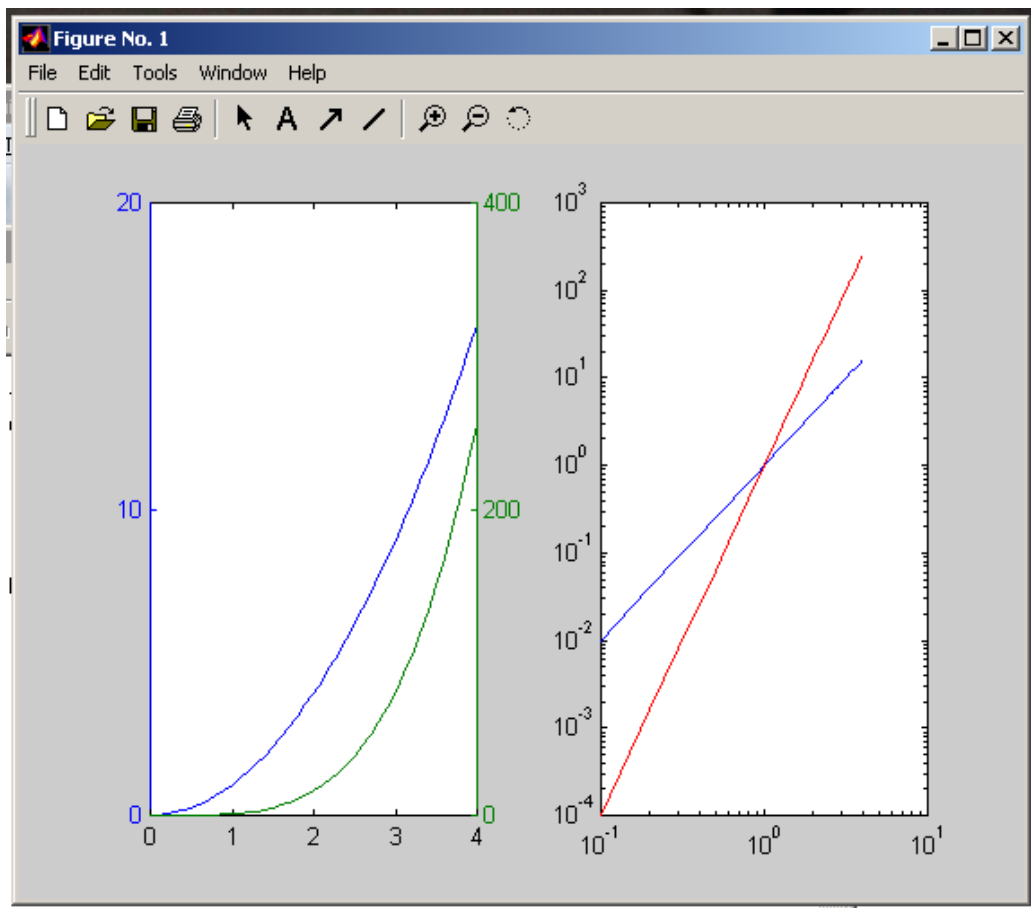
- Can include tex commands in labelling
  - E.g. \lambda
- Note positioning of “legend”
- Very long titles can be split over more than one line:
  - title({‘1st part of title’,’2nd part of title’})

Now try Exercise 2!

- plotyy
  - Useful for plotting two functions with widely-differing abscissae and function values on same graph
- loglog
  - Logarithmic scale (e.g. for interpreting slope of graph)
- subplot(m,n,p) = pth subplot within m x n array

- Example

```
x = 0 : 0.1 : 4;  y = x .^ 2;  y2 = x .^ 4;
subplot(1,2,1); plotyy(x,y,x,y2);
subplot(1,2,2); loglog(x,y,'b',x,y2,'r')
```



Now try Exercise 3!

Further Graph Annotations

- Sideways text: sidetext('text to be printed sideways')
- Positioned text: text(a,b,'text to go at this position')
- Can include special characters (e.g. \sigma, \lambda) in text
- Line thickness: plot(x,y,'b','LineThickness',2)
- Can specify font and fontsize, angle of text etc.
  - text(xpos,ypos,'angled bold text goes here', 'Rotation',30,'FontWeight','bold','FontName','Courier')

- Can add arrows to figures, within text
  - text(xpos,ypos,’message \rightarrow’,’HorizontalAlignment’,’right’)
  - \downarrow with VerticalAlignment, plus arrow position of ‘bottom’
- num2str is useful for converting digits to string values
  - E.g. for printing the coordinates at an intersection
  - num2str(xval1)

Now try Exercise 4!

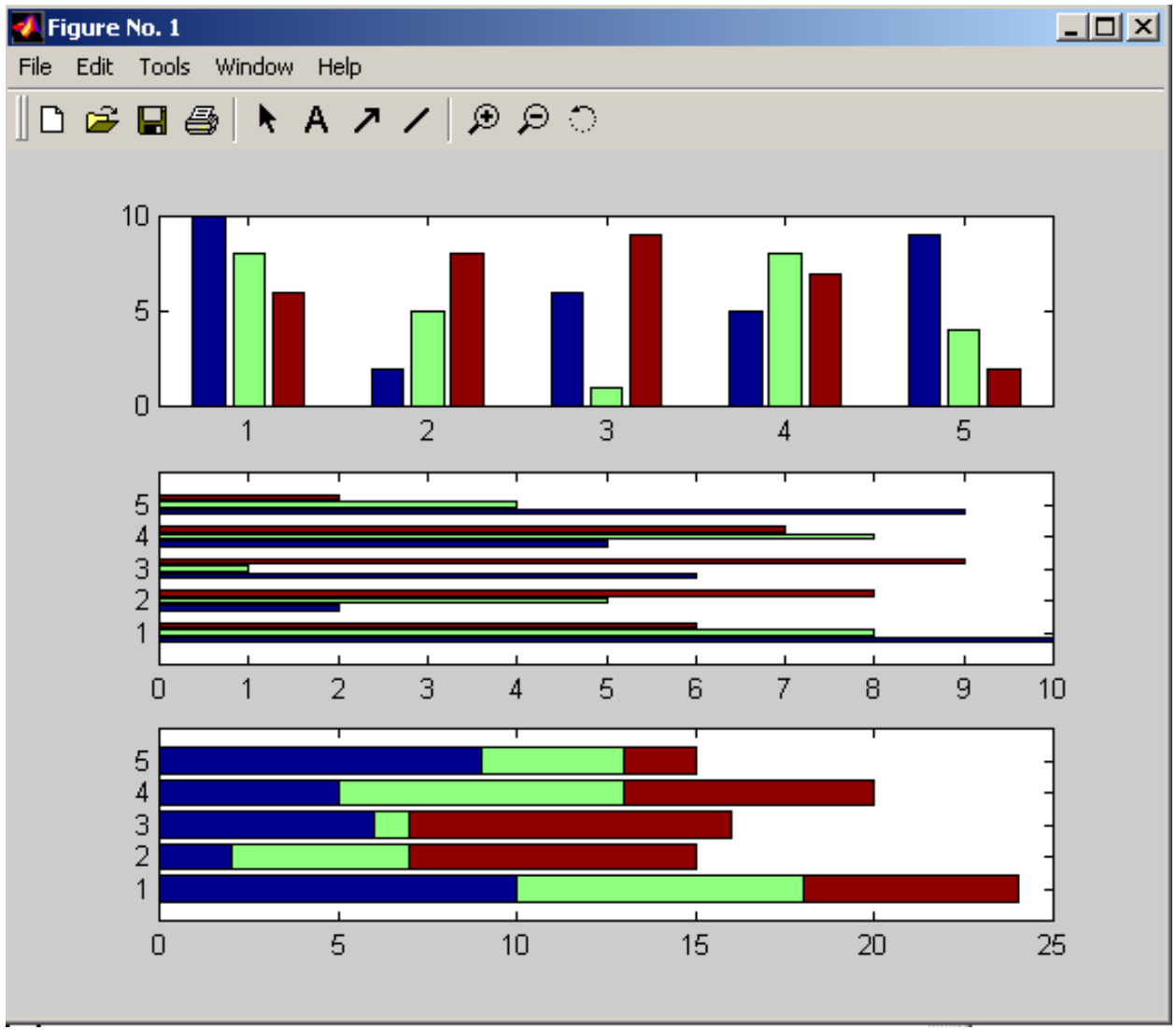
2D Specialised Graphs

- Barcharts
  - bar(y) where y is an m x n array
  - bar(y,width) (default width 0.8; experiment with 0.5 or 1.5)
  - barh(y) – horizontal bars
  - bar3(y) and bar3h(y) for 3D effect
  - bar(y,'stack') stacks the row entries of y
  - bar(y,'group') is the default

- Example

```
y = [10 8 6; 2 5 8; 6 1 9; 5 8 7; 9 4 2];
subplot(3,1,1); bar(y,'group')
subplot(3,1,2); barh(y,'group')
subplot(3,1,3); bar(y,'stack')
```

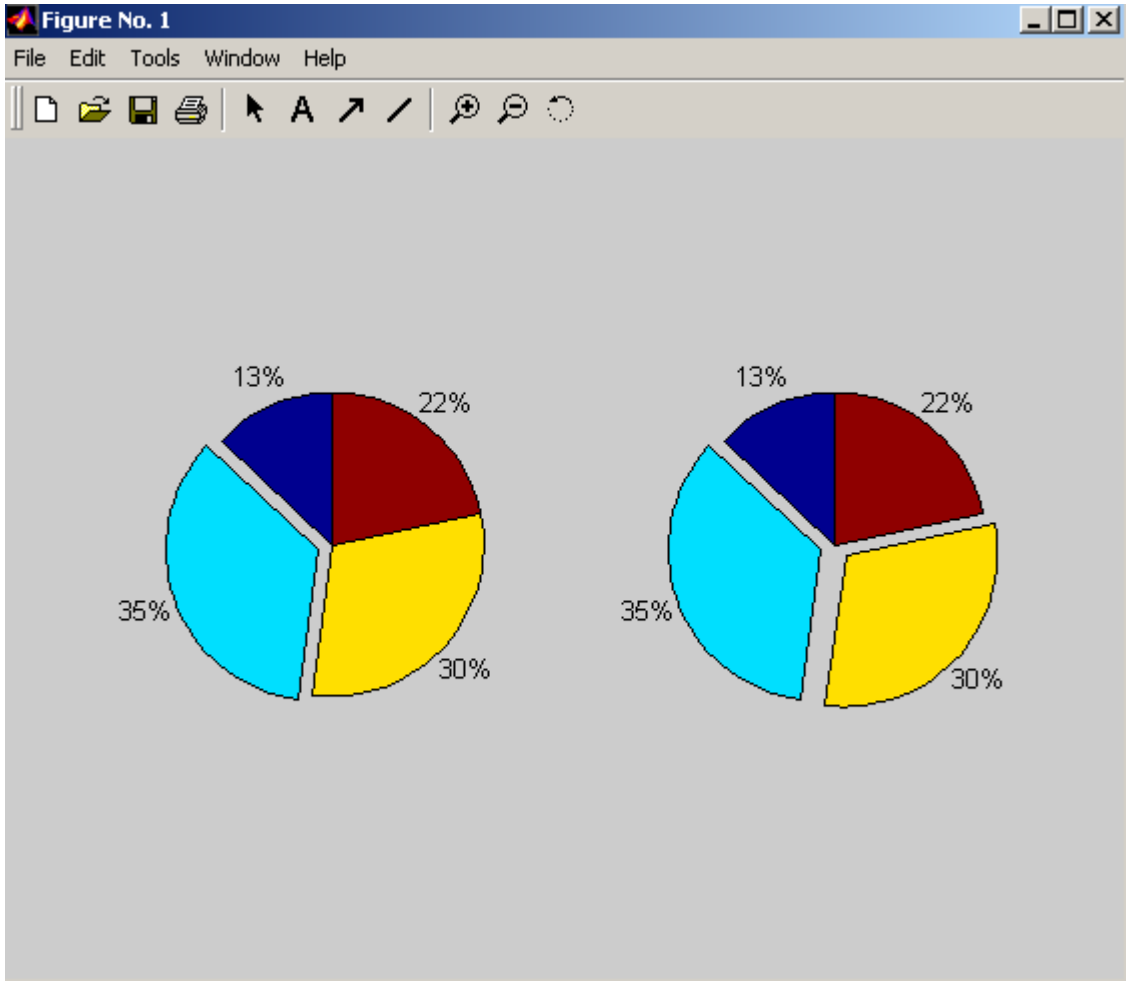
- which plots as:



- Pie charts
  - pie(y)
- Explode pie chart segment
  - explode = zeros(size(y)) for 1D array
  - explode = zeros(size(sum(y))) for 2D array
  - In 1D case:
    - [c,offset] = max(y) (or min, for example)
    - explode(offset) = 1;
    - pie(y,explode);

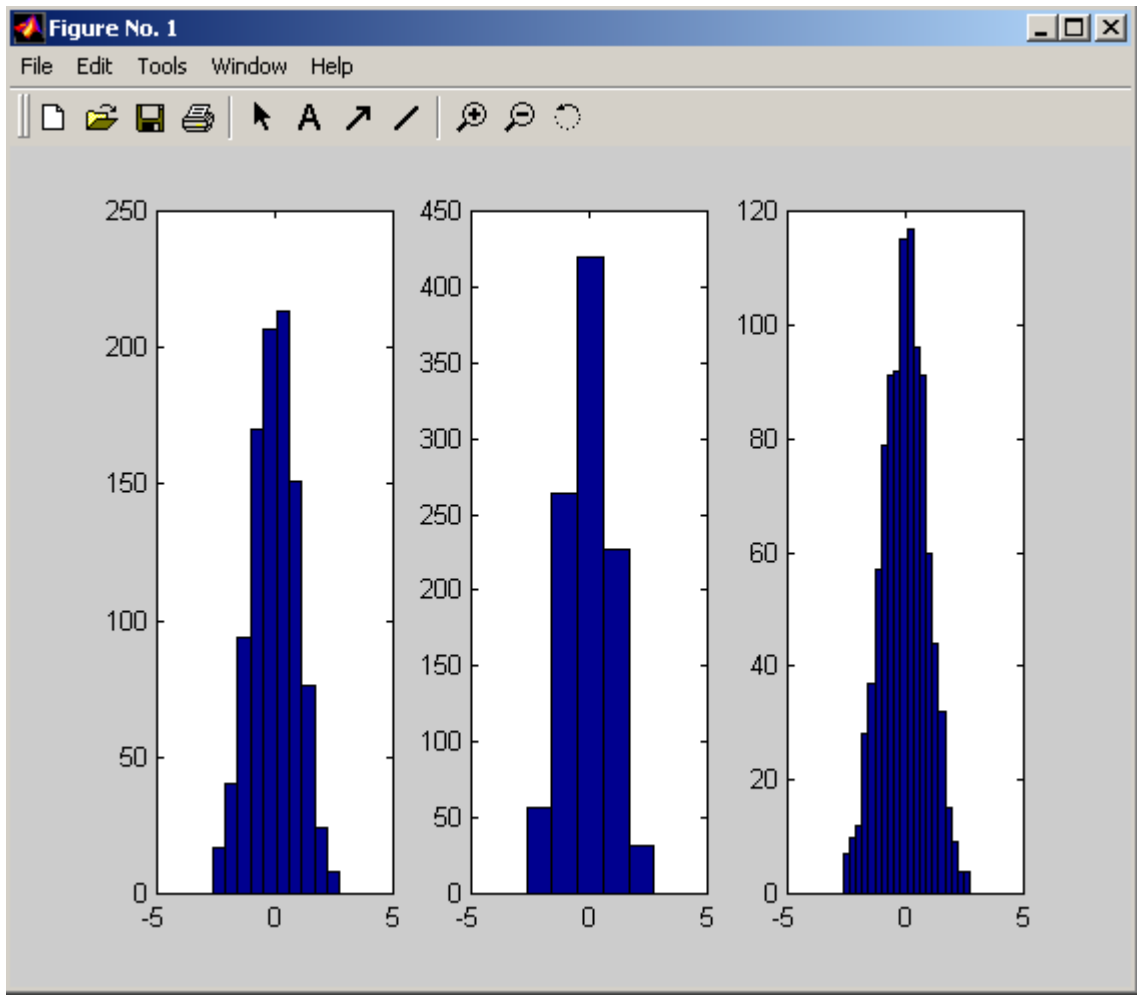
- Example
  - y = [3 8 7 5];
  - explode = zeros(size(y)); (= 0 0 0 0)
  - [c,offset] = max(y); (c = 8, offset = 2)
  - explode(offset) = 1;
  - pie(y,explode);
- Can explode 2nd and 3rd sectors in pie:
  - explode = [0 0.5 0.25 0]
  - pie(y,explode)

- Pie charts:



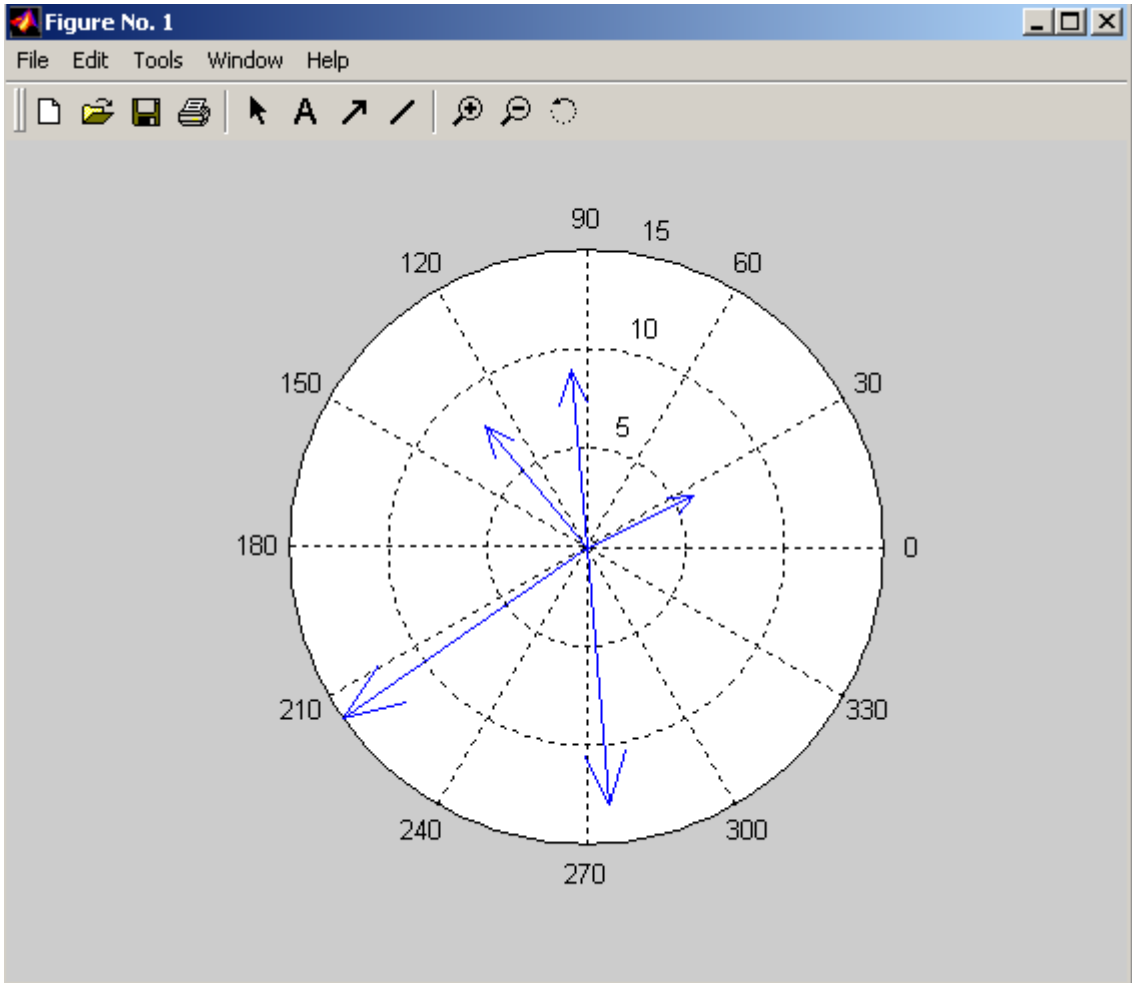
- Note: pie3 for 3D effect

- Histograms
  - E.g. y = randn(1000,1); hist(y)
  - 10 bins by default
  - hist(y,5)
  - hist(y,20)



- Compass plots
  - E.g. for plotting wind direction/strength on a circular grid
  - Work in radians:
  - xv = pi/180 * [215 275 130 25 95];
  - yv = [15 13 8 6 9] (corresponding wind strengths)
  - Convert from polar to cartesian coordinates
  - [x,y] = pol2cart(xv,yv);
  - compass(x,y)

- With the resulting plot:



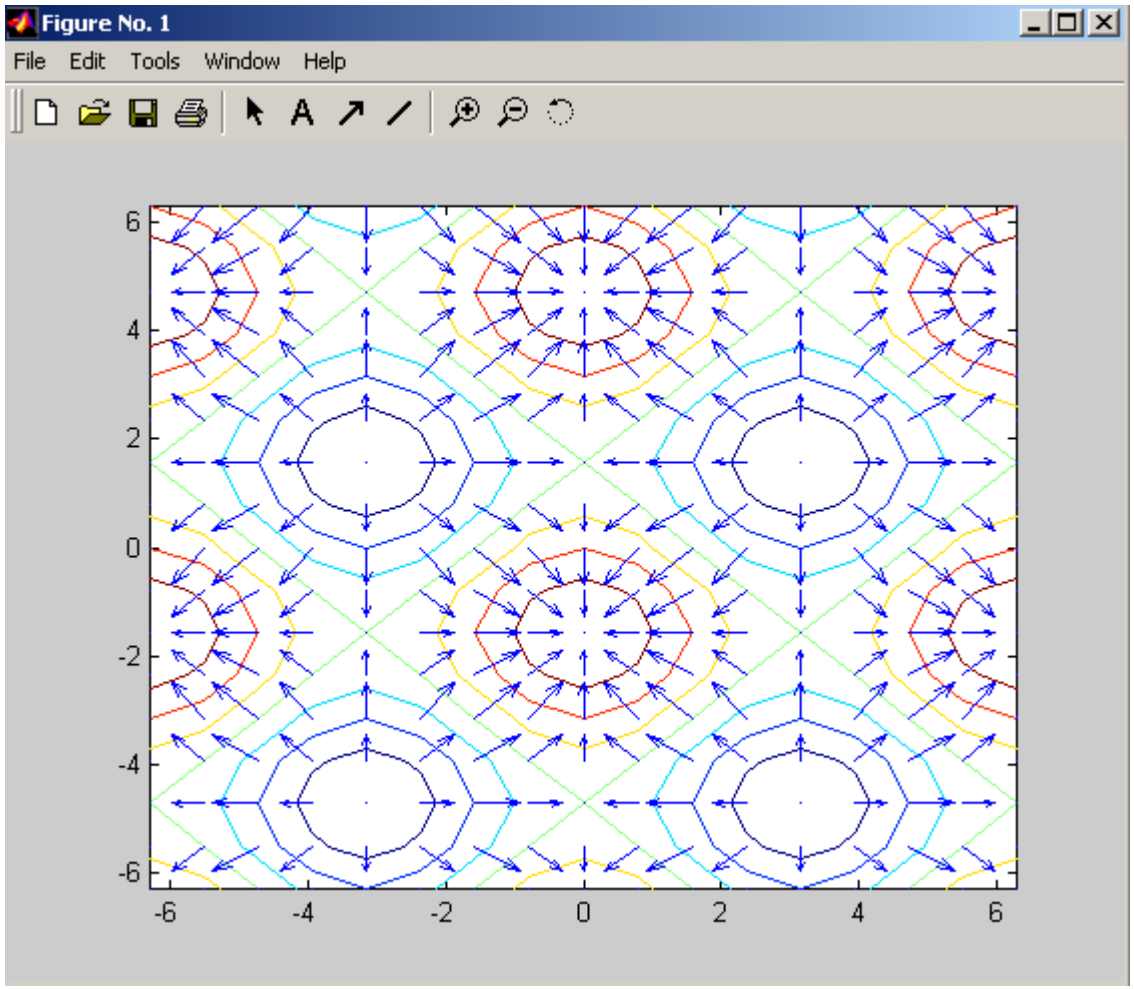
- Meshgrid
  - Need matrix input for surface, mesh and contour
  - Convert x and y vectors to a grid of values
  - [X,Y] = meshgrid(x,y);
  - Rows of X are copies of vector x
  - Columns of Y are copies of vector y

- Feather plots
  - For displaying complex numbers, where real and imaginary parts are separated
- 2D Quiver plots
  - Shows vectors at given points
  - Plot a contour graph; hold on
  - Compute gradient at each point
  - Superimpose gradient vectors on original plot

- Example

```
[x,y] = meshgrid(-2*pi : pi/4 : 2*pi);
z = cos(x) – sin(y);
contour(x,y,z);
hold on;
[dx,dy] = gradient(z, pi/4, pi/4);
quiver(x,y,dx,dy)
```

- With figure:



Now try Exercises 5 and 6!
Then try Exercise 7!