MATLAB®
The Language of Technical Computing



2D Graphs

Prof. Byoungjo CHOI
Embedded Systems Engineering Dept.
Incheon National University

You will be able to

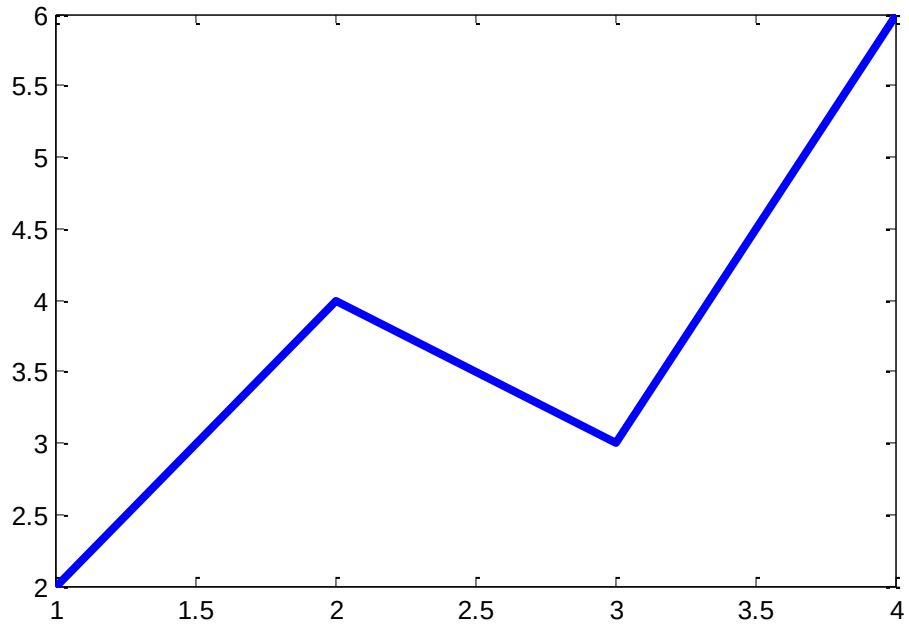
- Draw various graphs using plot,
- Set properties of plots using graphics handles,
- Visualize data using various plots including bar, stem, stairs, pie, histogram and polar plots,
- Use patch plots with colors.

Plotting a Single Vector

- plot(y)
  - The index is used as the x-axis.
  - The range of plot is determined automatically.
  - The line color starts from 'blue'.

basic_plot1.m

```
% basic plot 1
y = [ 2 4 3 6 ];
figure(1);
plot(y);
```

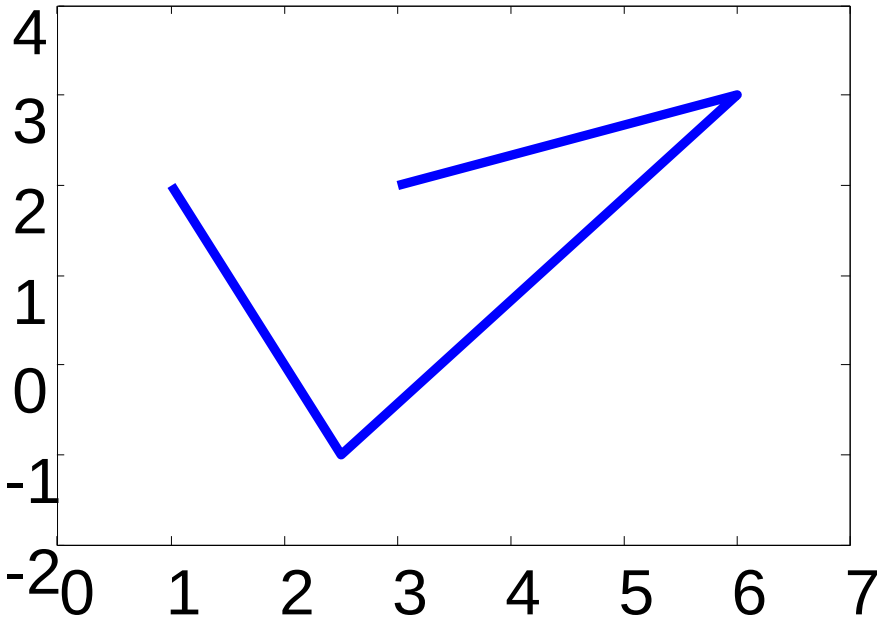


Plotting x-y Vectors

- plot(x, y)
  - x is used the corresponding x-value for y.
- axis command controls the viewing range.

basic_plot2.m

```
x = [ 1 2.5   6   3 ];
y = [ 2 -1    3   2 ];
plot(x, y);
axis([0 7 -2 +4]);
```



Multiple Plots

- `plot(x,y,u,v,t,h)`
- `plot(x,y,'r-',u,v,'b:',t,h,'k--')`
-
```
plot(x,y);
hold on;
plot(u,v);
plot(t,h);
hold off;
```

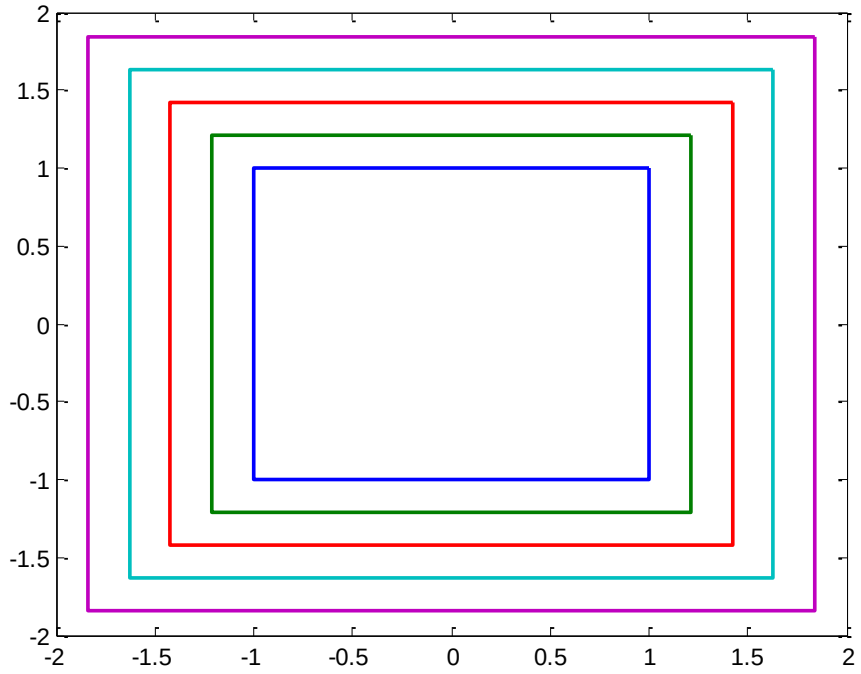
-
```
plot(X,Y);
% X - matrix
% Y - matrix
% column-wise plot
```

Plotting a Matrix

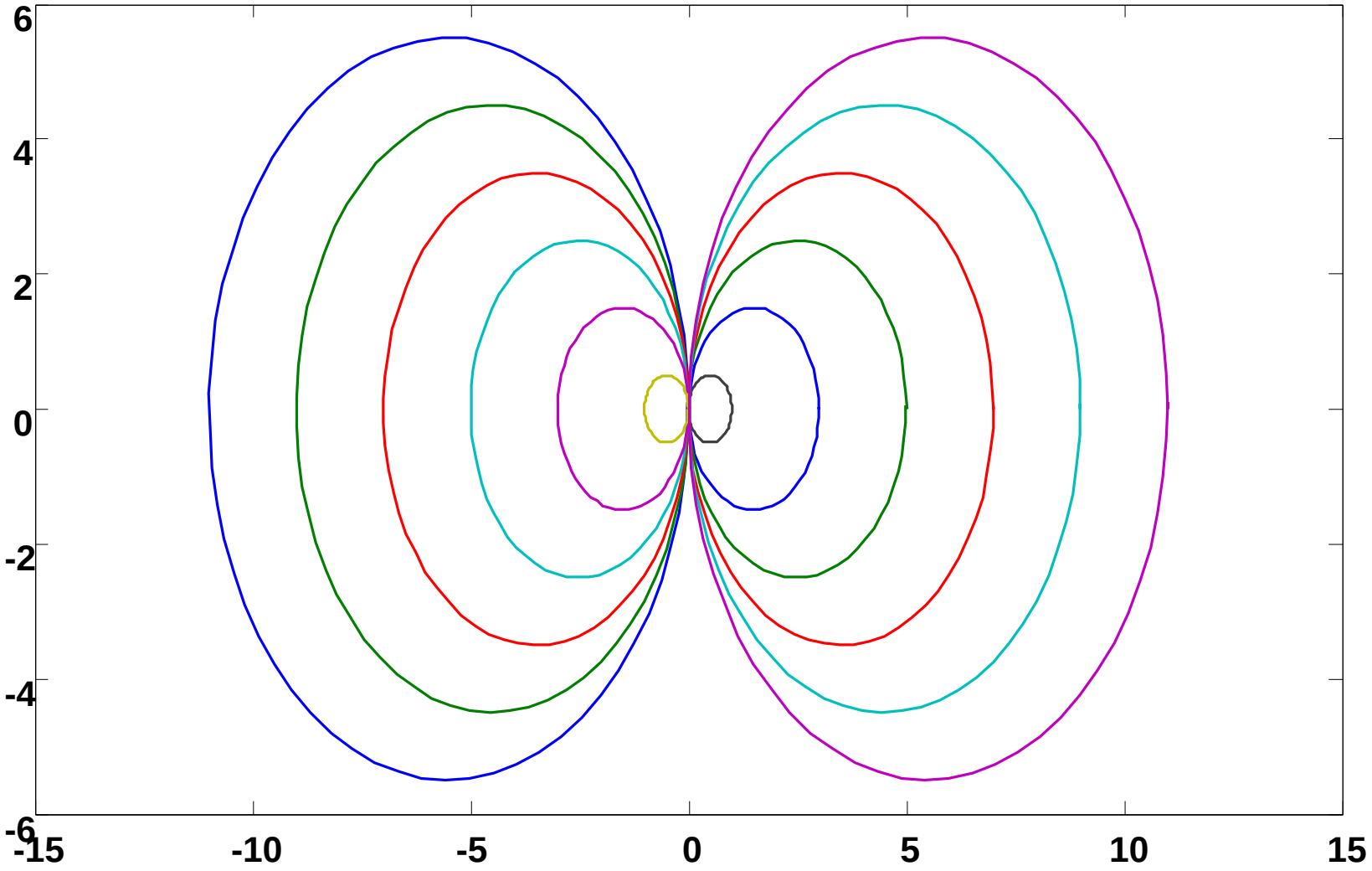
- plot(X, Y)
  - Each column of the matrices X and Y is interpreted as x, y coordinates of each graph.
  - Different colors are assigned to each graph.

squares.m

```
x = [ 1 -1 -1 1 1]';
y = [ 1 1 -1 -1 1]';
s = 1:0.21:2;
X = x * s ;
Y = y * s;
plot(X,Y,'LineWidth',2);
```



Example 5-1: Joining Circles



Script: Joining Circles

circles.m

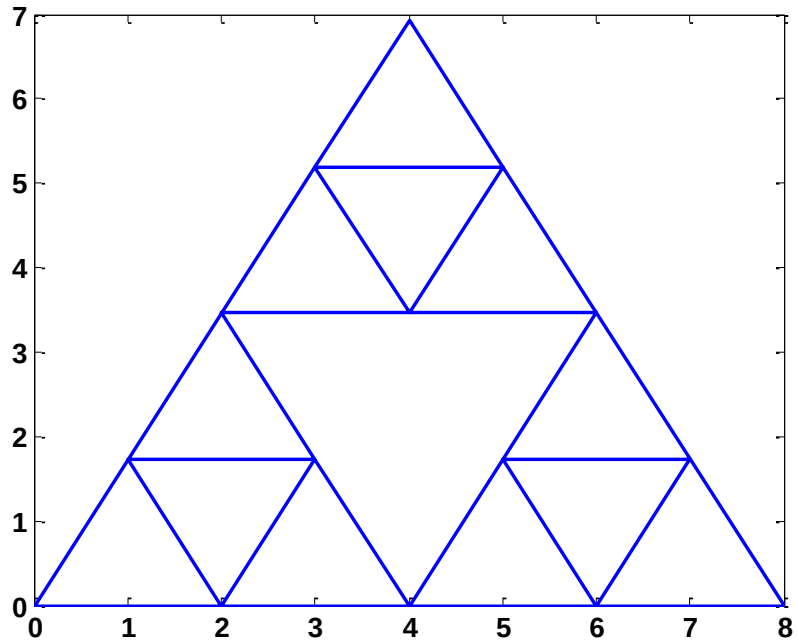
```
% Joining Circles
t = 0:0.1:2*pi+0.1; % angles
N = length(t);
x = cos(t');
y = sin(t');

c = -5.5:5.5; % center of the circles
C = repmat(c,N,1);
r = abs(c);      % radius of the circles

X = x*r + C;
Y = y*r;
plot(X,Y) % plot the circles
```

Mission 1a: Basic Plot

- Plot the following graph using basic plot command.
  - Determine the path of the plot without revisiting the point again.
  - Provide x and y.
  - Height is $\sqrt{3}$

triangle_tower.m

```
% plot the triangle tower.
x = [ ...];
y = [ ...]*sqrt(3);
plot(x, y);
```

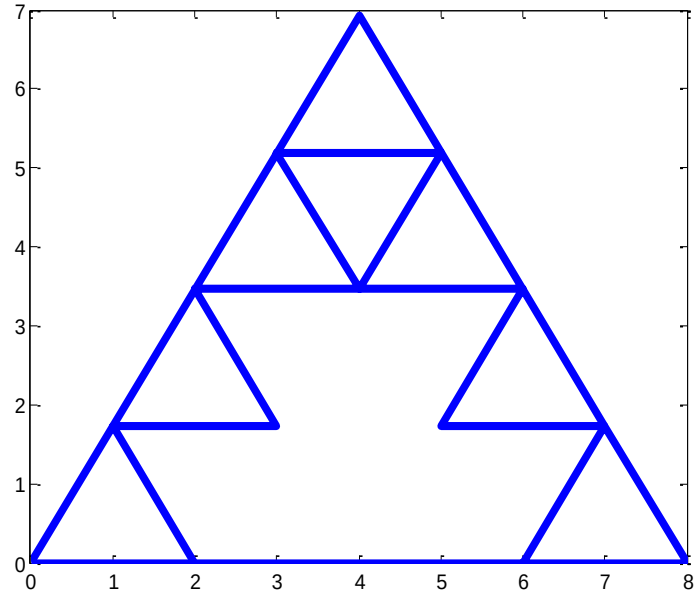


Mission 1b: Basic Plot

- Plot the following graph using basic plot command.
  - Determine the path of the plot without revisiting the point again.
  - Provide x and y.
  - Height is $\sqrt{3}$

trangle_tower.m

```
% plot the triangle tower.
x = [ ...];
y = [ ...]*sqrt(3);
plot(x, y);
```

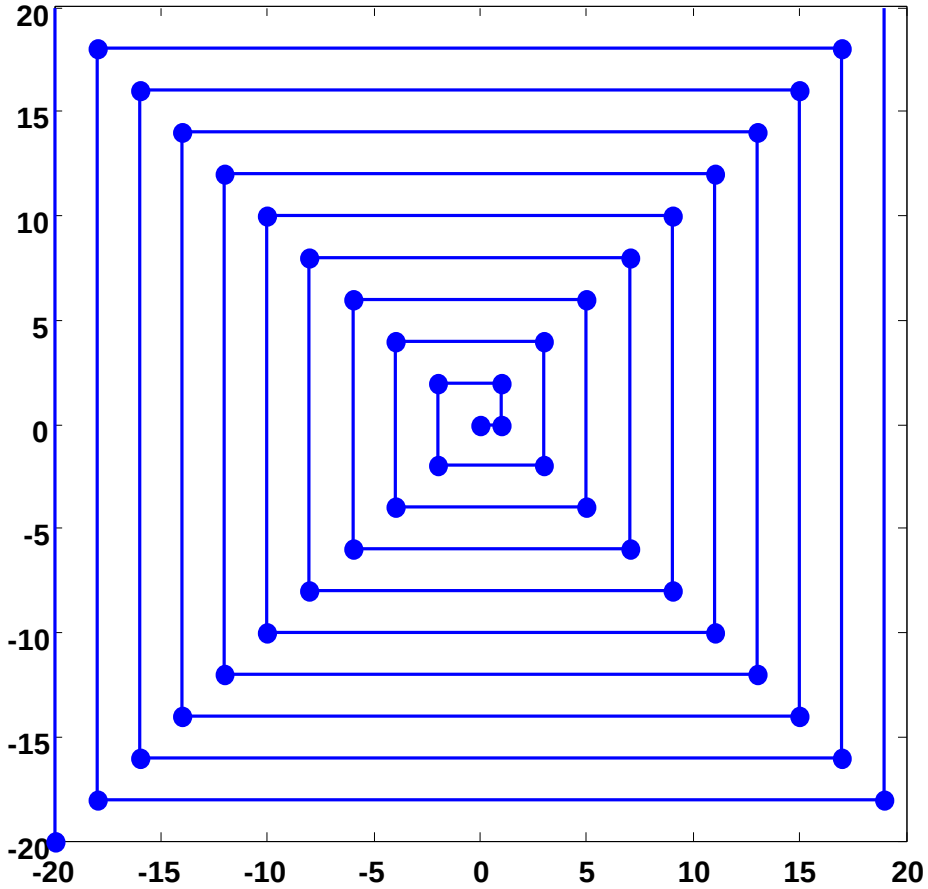


Mission 2a: Square Spiral

- Plot the following graph using basic plot command.
  - Find a rule to generate x and y coordinates of connecting points.

square_spiral.m

```
...
x = ...;
y = ...;
plot(x, y, 'b-o');
```



Mission 2a: Solution

- Sample

square_spiral.m

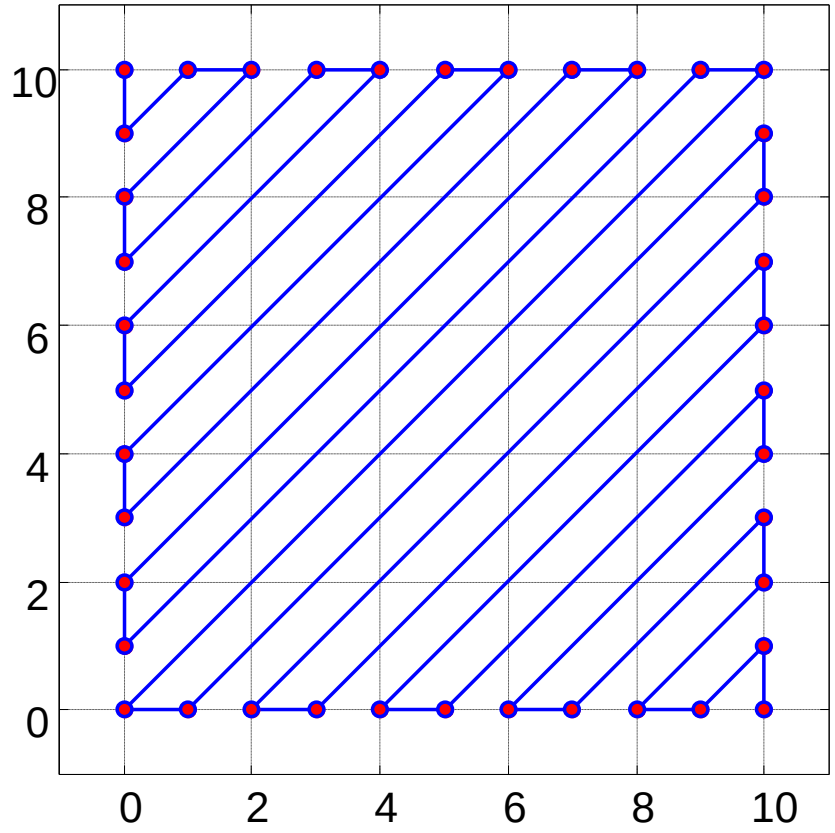
```
%% Square Spiral
x0 = 1:20;
x1 = x0 .* (-1).^(x0+1);
x2 = reshape(repmat(x1,2,1), 1,[]);
x  = [ 0 x2 ];
y0 = 2:2:20;
y1 = reshape(repmat(y0,2,1), 1, []);
n1 = 1:length(y1);
y2 = y1 .* (-1).^(n1+1);
y3 = reshape(repmat(y2,2,1), 1, []);
y  = [0 0 y3(1:end-1)];
figure(1);
plot(x,y,'b-o', 'MarkerFaceColor', 'blue');
```

Mission 2b: Snake Lines

- Plot the following graph using basic plot command.
  - Find a rule to generate
    x and y coordinates
    of connecting points.

snake_lines.m

```
...
x = ...;
y = ...;
plot(x, y, 'b-o');
```



Mission 2b: Solution

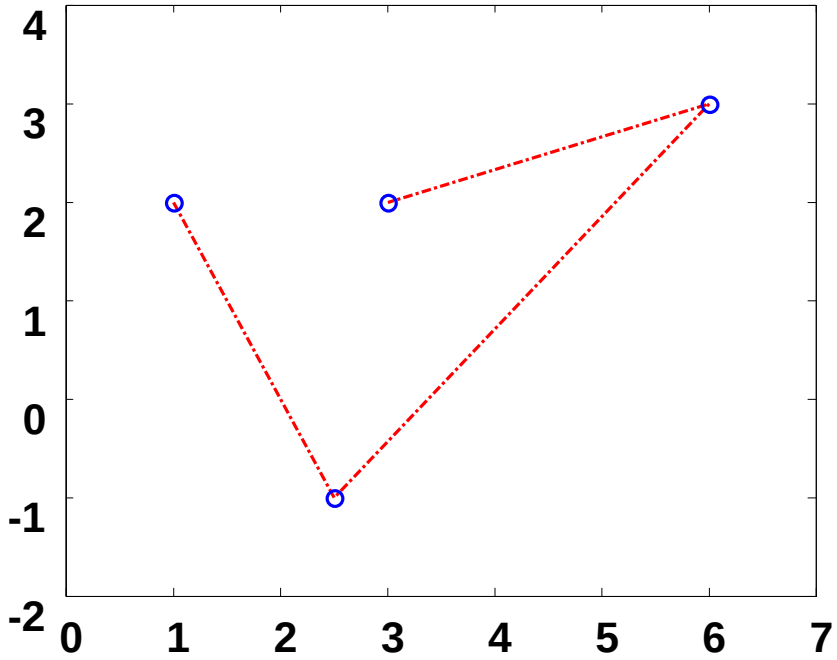
- Sample solution

snake_lines.m

Properties of Graph

- **plot(x, y,'LineSpec', ...**
  'PropertyName','PropertyValue')

```
x = [ 1 2.5   6   3 ];
y = [ 2 -1    3   2 ];
plot(x, y,'r-.o', ...
     'LineWidth', 2, ...
     'MarkerEdgeColor', 'b');
axis([0 7 -2 +4]);
```



The `plot` Command: LineSpec

- LineStyle

solid	–		dotted	:	
dashed	--		dash-dot	–.	

- LineColor

red	r		magenta	m	
green	g		yellow	y	
blue	b		black	k	
cyan	c		white	w	

- Marker

+	+	s	□	v	▽
o	○	d	◇	^	△
*	✳	p	☆	<	◁
x	×	h	✡	>	▷

Properties and Values

- 'LineWidth', 2
- 'MarkerSize', 6
- 'MarkerEdgeColor', 'b'
- 'MarkerFaceColor', 'r'
- 'LineStyle', '-'
- 'Color', [0 1 0]
- 'Marker', 'o'

Handles and Properties

```
get( h(1) )
set( h(1),'LineWidth',2 );
get( gca )
set( gca, 'FontSize', 12, ...
          'FontWeight', 'bold');
xl = get( gca, 'XLabel');
set( xl, 'FontSize', 14)
```

gca: Get Current Axis

Text Formatting

- **Font**

a{₩bfb}c	a**b**c	₩fontname{times}	abc
a{₩itb}c	a*b*c	₩fontsize{12}	abc
₩ita{₩rmb}c	*a*b*c*		

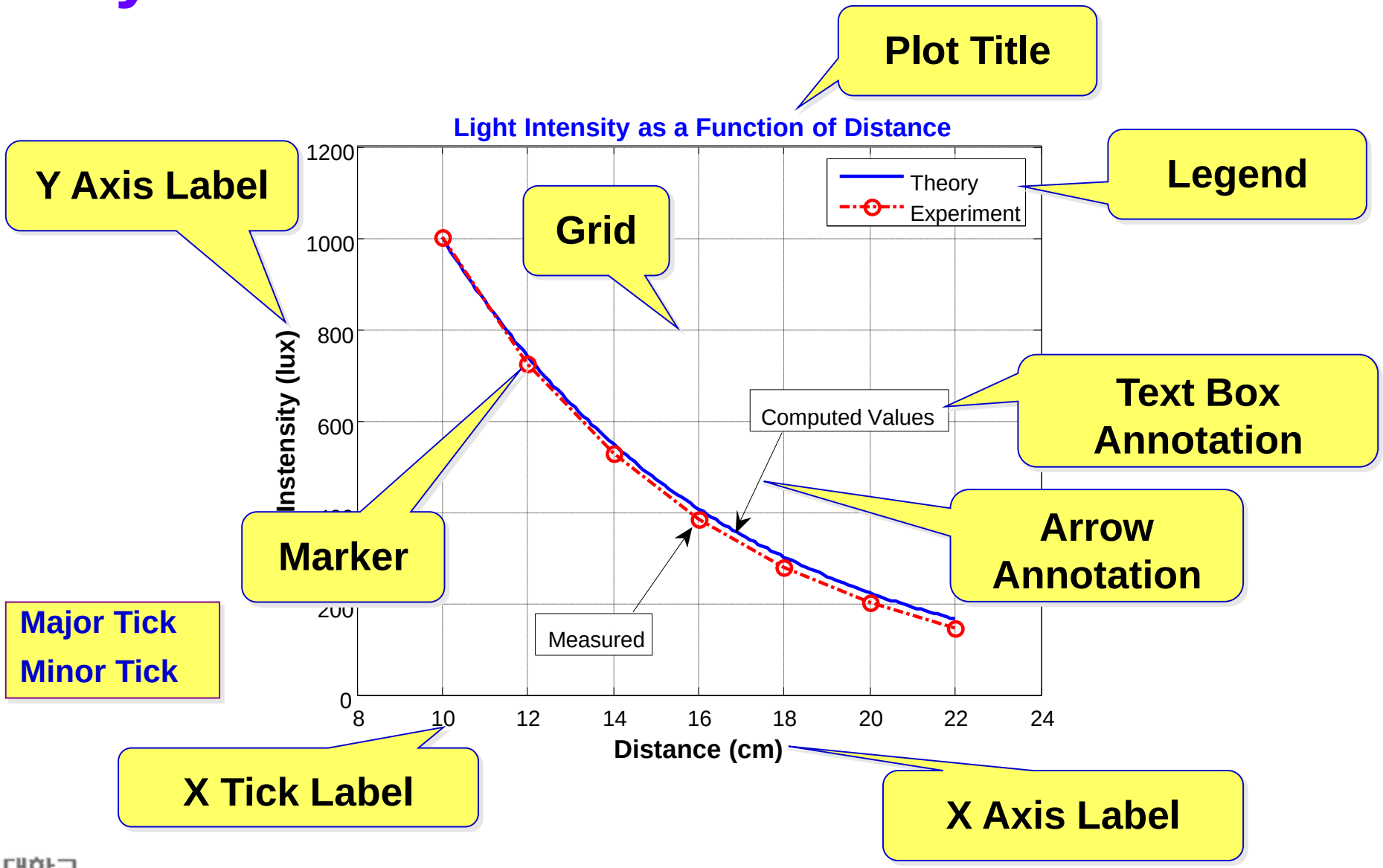
- **Subscript and Superscript**

A^{2-x}+b_i	$A^{2-x}b_i$

- **Greek Letters**

₩alpha	α	₩pi	π	₩Gamma	Γ
₩beta	β	₩sigma	σ	₩Lambda	Λ
₩gamma	γ	₩Phi	Φ	₩Omega	Ω
₩theta	θ	₩Delta	Δ	₩Sigma	Σ

Anatomy of Plot



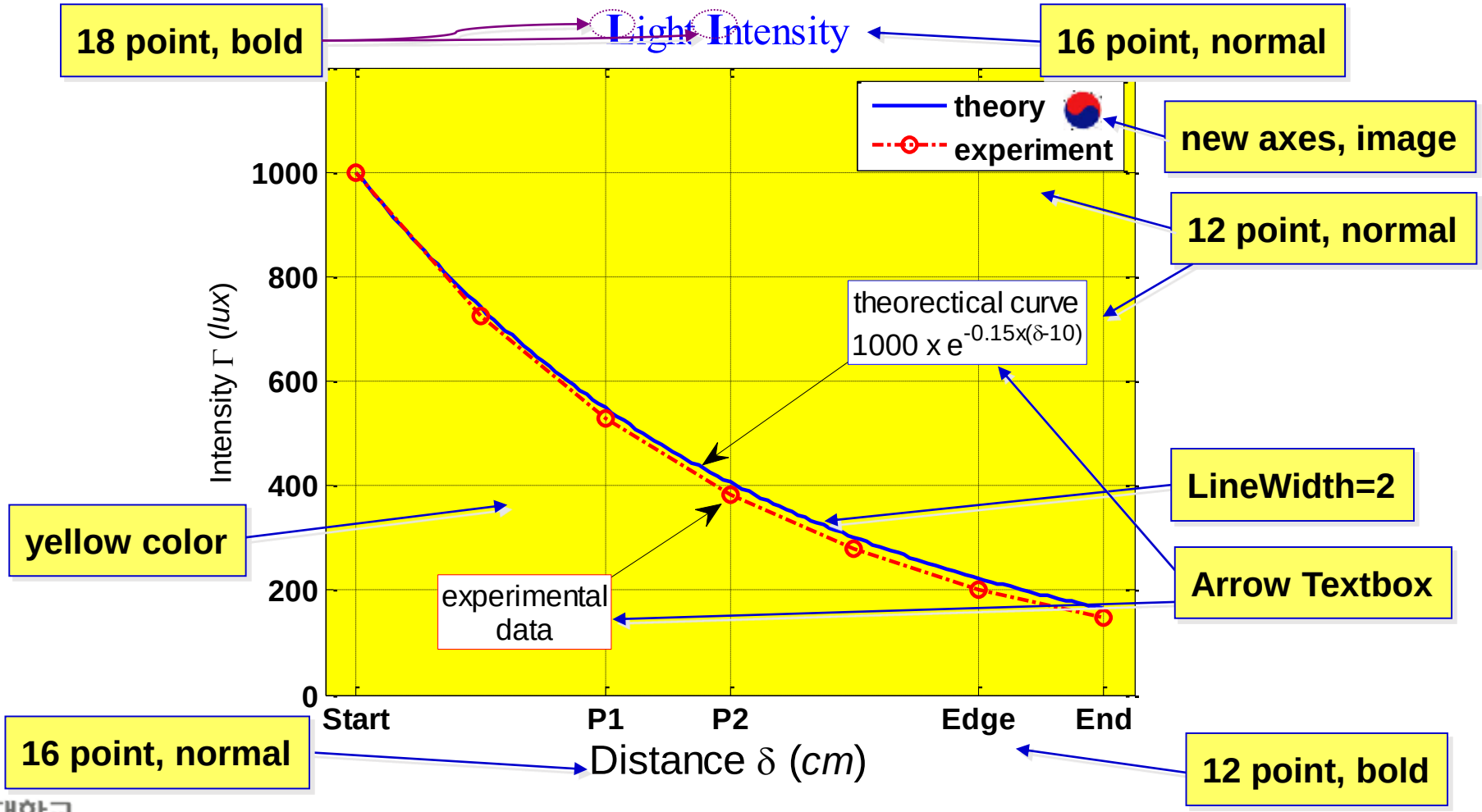
Anatomy of Plot - Script

light_intensity.m

```
% Sample Plot for Showing Various Graph Elements
d  = 10:0.1:22;  dd = 10:2:22;
y  = 1000*exp(-0.15*(d-10));
yy = 1000*exp(-0.16*(dd-10));
h  = plot(d,y,'b-',dd,yy,'r-.o');
axis([8 24 0 1200]);
set(h(1),'LineWidth',2);  set(h(2),'LineWidth',2);
xlabel('Distance (cm)');  ylabel('Intensity (lux)');
title('Light Intensity as a Function of Distance');
legend('Theory','Experiment');
grid;
```

Mission 3 – Custom Plot

- Write an m-script to plot the following graph.



$\delta = 10 \quad 12 \quad 14 \quad 16 \quad 18 \quad 20 \quad 22$

Solution 3

- Script and Screenshot

custom_plot.m

Function Plot

Optional

- fplot('expression', range, *LineSpec*)
  - fplot('x^2+4*sin(2*x)-1', [-3 3], 'r-')
  - Function expression can be your m-file.

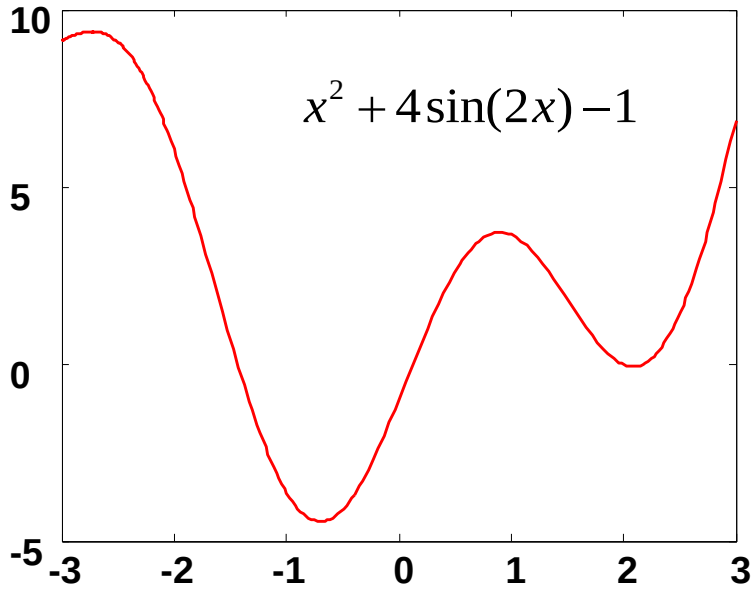
myfun.m

```
function y = myfun(x)
y = x^2+4*sin(2*x)-1;
```

fplot(@myfun, [-3 3] , 'r-')

or fplot('myfun', [-3 3], 'r-')

- ezplot('expression, *range*);
  - ezplot('x^2+4*sin(2*x)-1')
  - The default range is $-2\pi \sim 2\pi$.

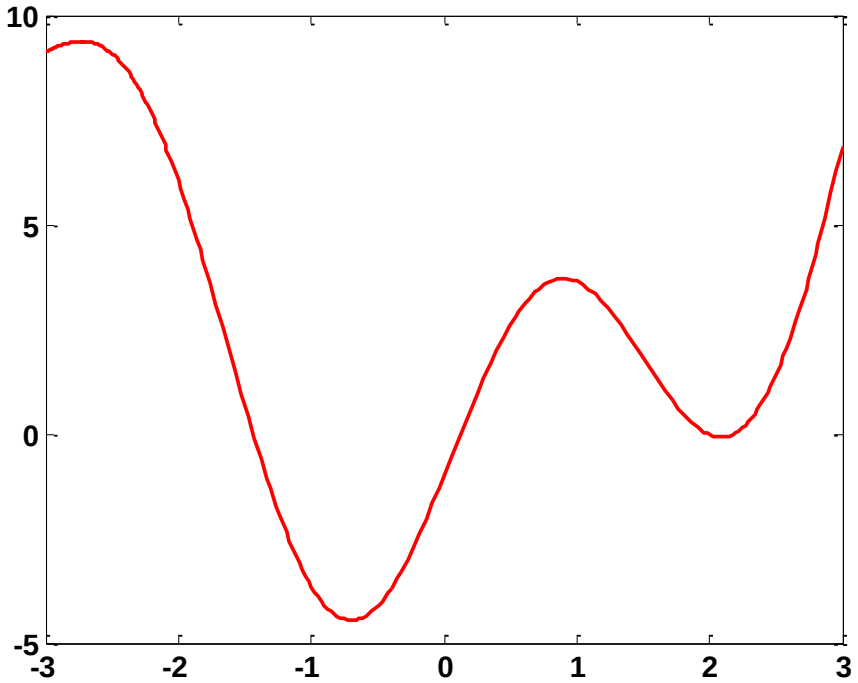


Equivalent Normal Plot

- fplot('x^2+4*sin(2*x)-1', [-3 3], 'r-')
- Need to generate x and y values
- The range of y should be determined.

fplot_eq.m

```
% Equivalent plot of fplot
%
x = -3:0.01:3;
y = x.^2+4*sin(2*x)-1;
figure(1), plot(x,y,'r-');
minY = floor(min(y));
maxY = ceil(max(y));
axis([-3 3 minY maxY]);
```



Implicit Plot

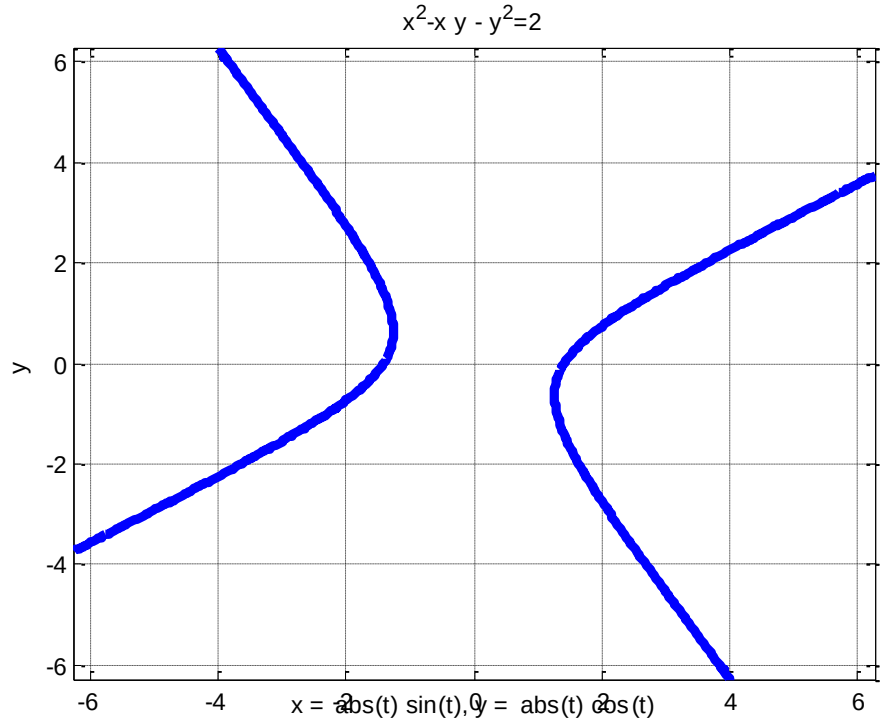
- ezplot('x^2-x*y - y^2=2')

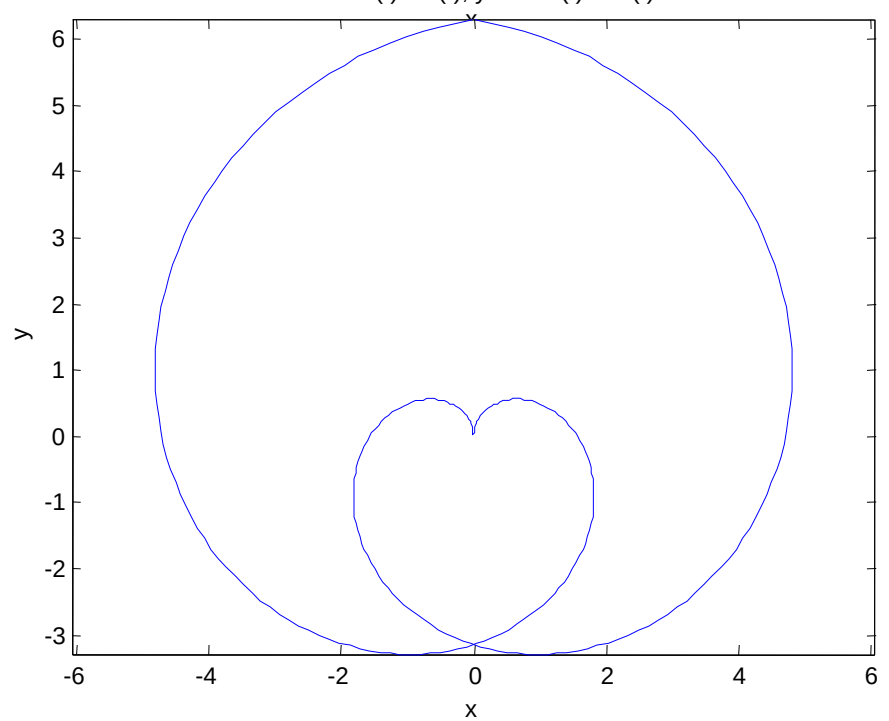
implicit_ex1.m

```
figure % new figure window
h = ezplot('x^2-x*y - y^2=2');
set(h, 'LineWidth',4)
grid;
```

- ezplot('abs(t)*sin(t)', ...
  'abs(t)*cos(t)',[-2*pi 2*pi]);

Parametric Plot

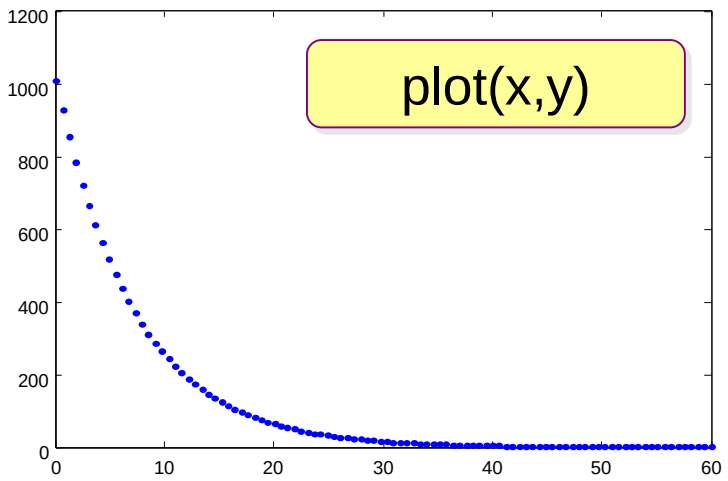


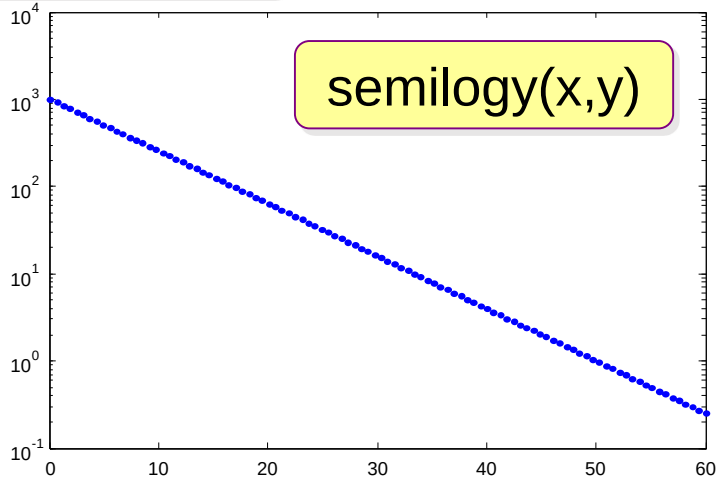


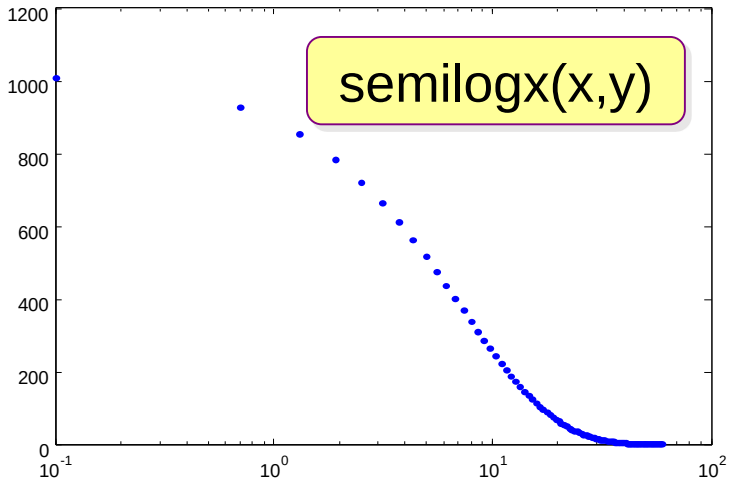
Logarithmic Axes

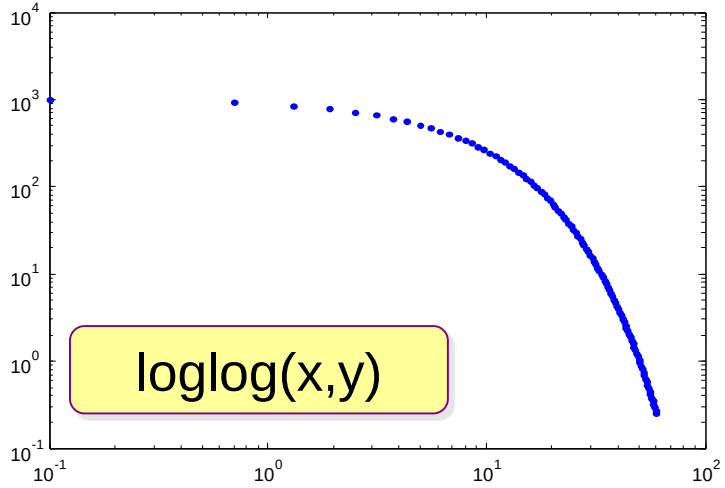
$$y = 2^{-0.2x+10}$$

```
x=linspace(0.1,60,100);  y = 2.^(-0.2*x+10);
```





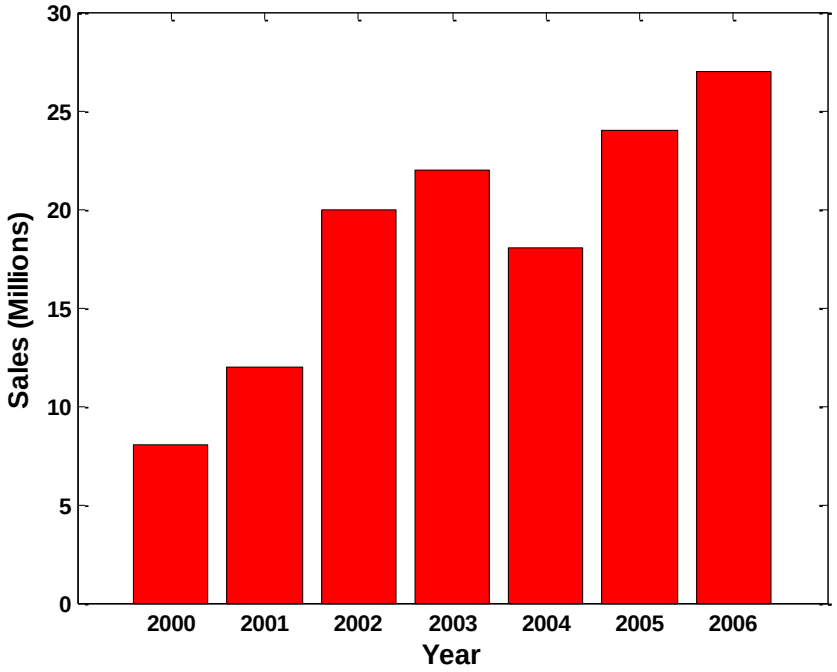




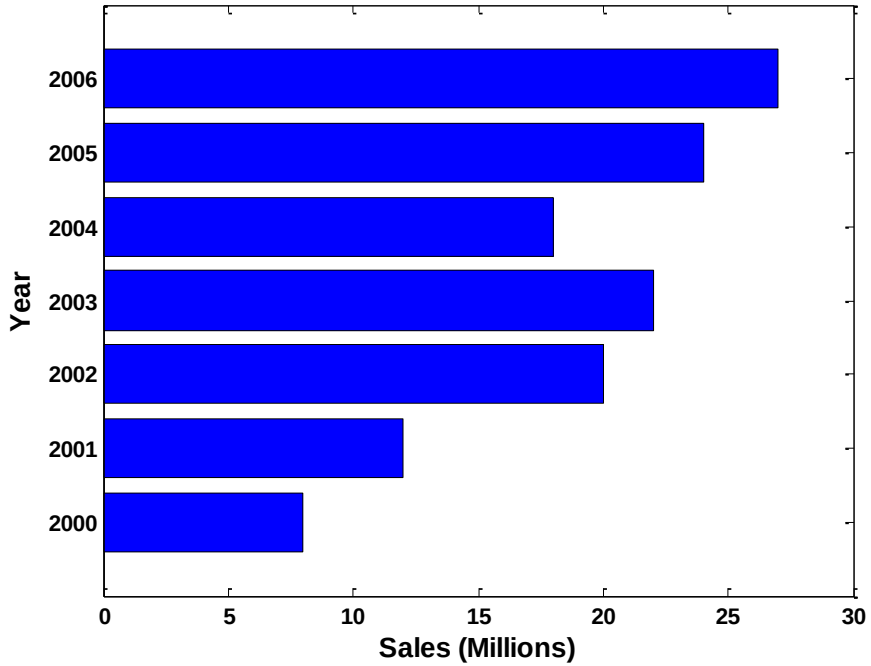
Special Graphs: Bar Charts

```
year  = 2000:2006;
sales = [8 12 20 22 18 24 27 ];
```

```
bar( year, sales,'r');
xlabel('Year');
ylabel('Sales (Millions)');
```

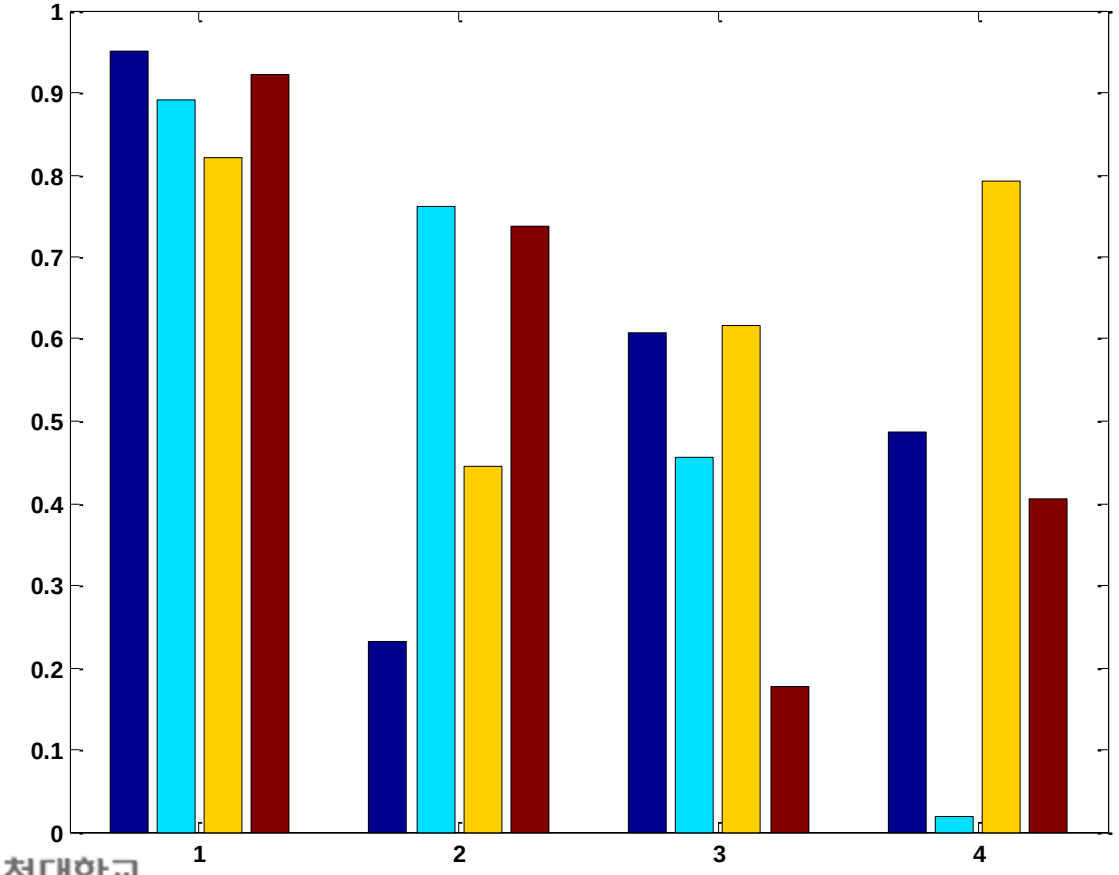



```
barh( year, sales,'b');
ylabel('Year');
xlabel('Sales (Millions)');
```



Multiple Bars

- Each column represents a bar chart.
- Different colors are assigned to different columns.

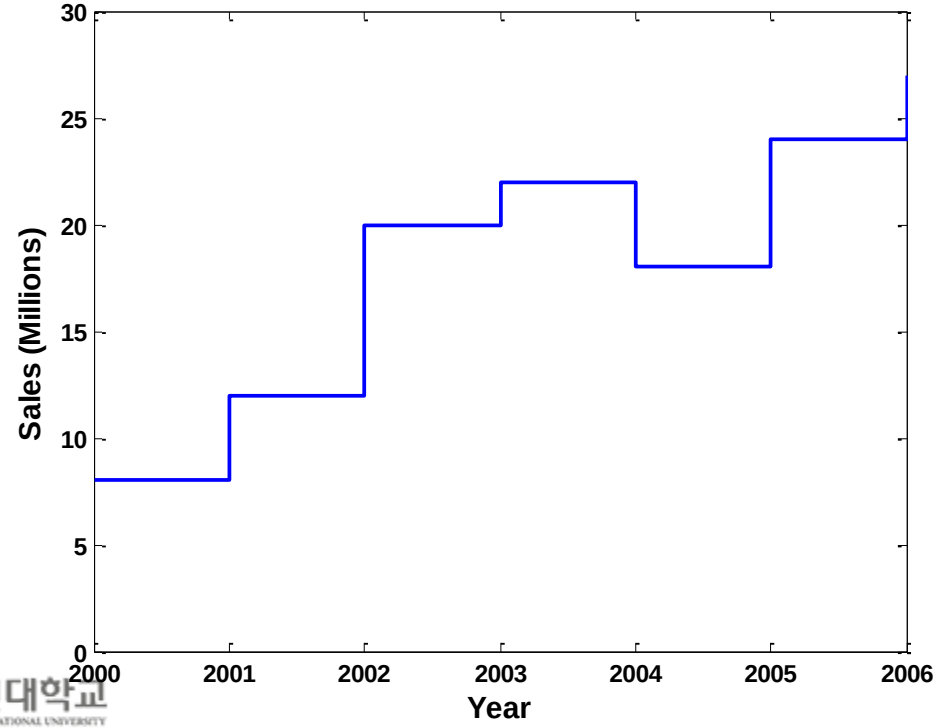



```
a=rand(4);
bar( a );
```

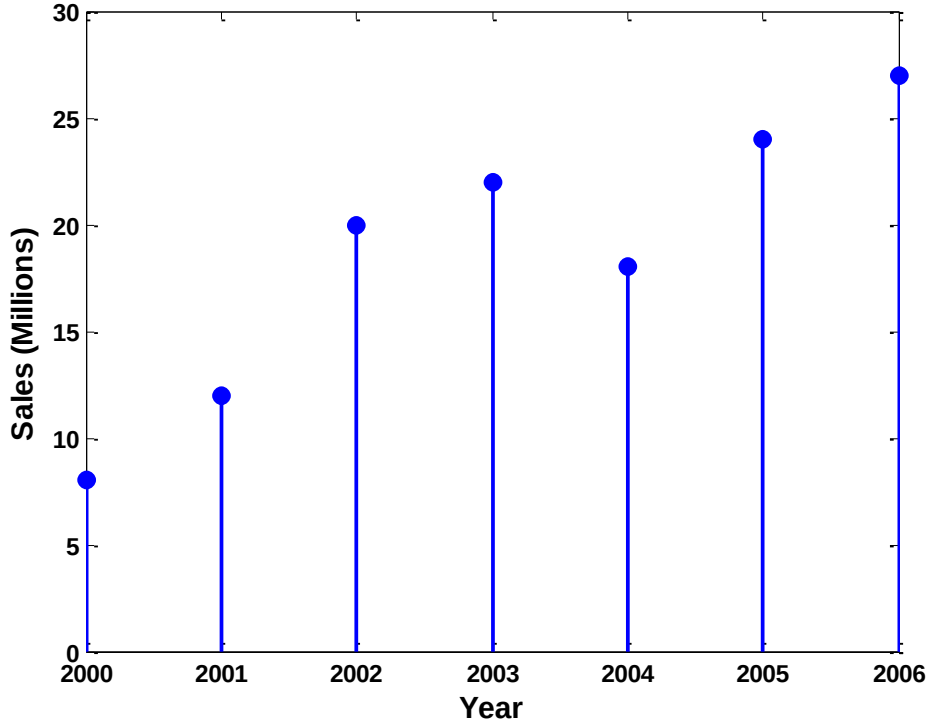
Special Graphs: Stairs & Stem

```
year  = 2000:2006;
sales = [8 12 20 22 18 24 27 ];
```

```
stairs( year, sales);
xlabel('Year');
ylabel('Sales (Millions)');
```




```
stem( year, sales,'filled');
xlabel('Year');
ylabel('Sales (Millions)');
```



Mission 4: KOSPI of the Last 3Mth.

- Draw a plot of the KOSPI for the past 3 months. Use sensible labels for x-axis and y-axis as in the graph below.
  - Get the raw data from KOSCOM site.

    http://stock.koscom.co.kr/kse_sise/kse_daily_jisu.jsp

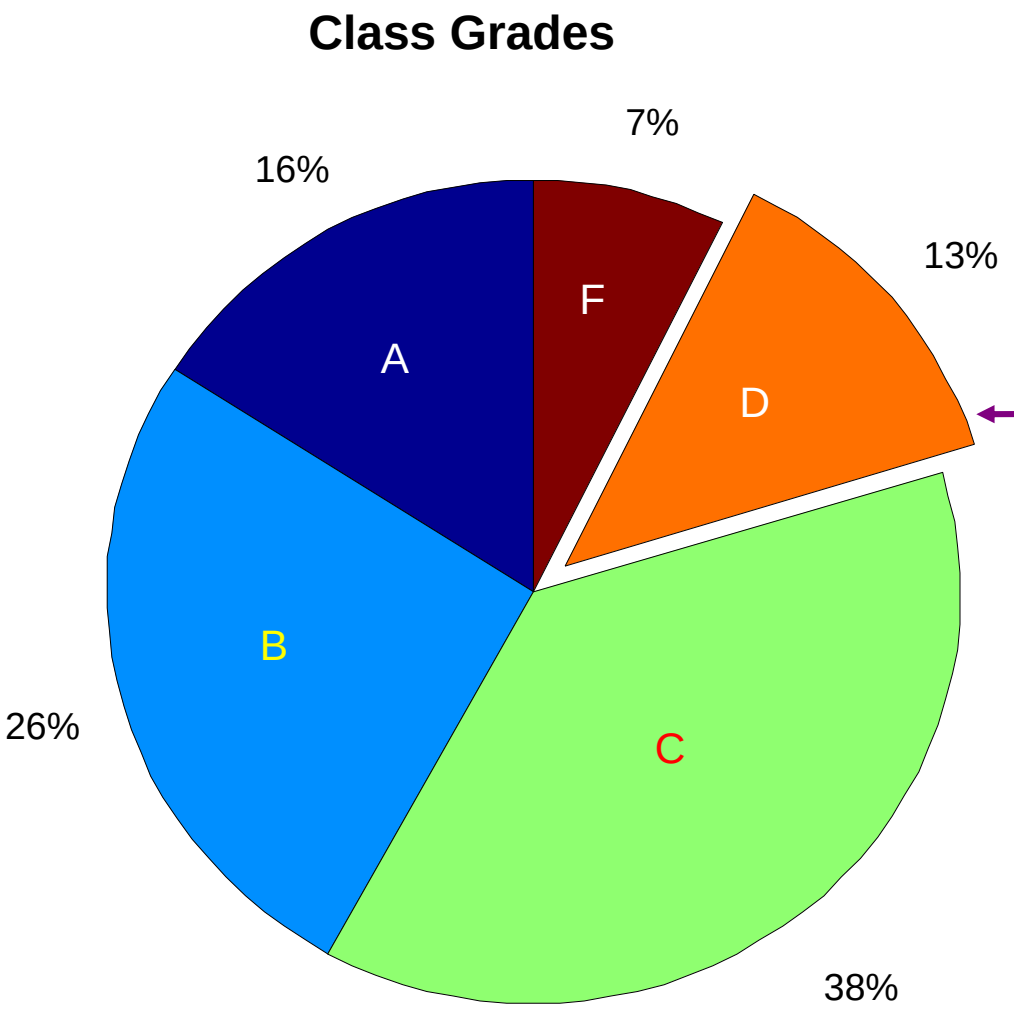
    Visit the site and select the data and paste it in Excel.
  - Read in B column of the Excel file into MATLAB and plot it.

Solution 4

- Script and Screenshot

kospi_3month.m

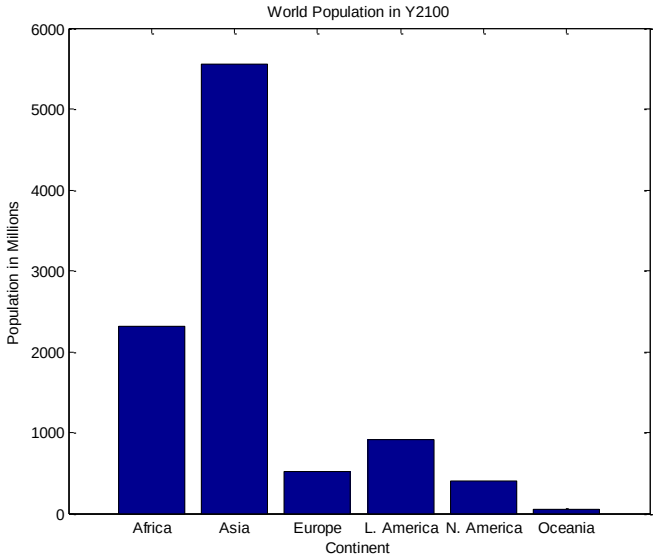
Pie Chart

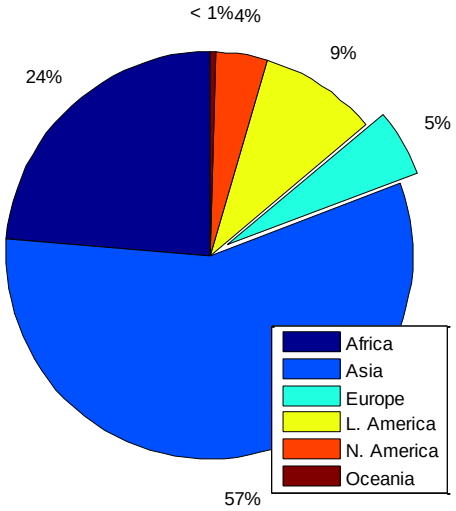



```
grades = [ 11 18 26 9 5];
pie(grades,[ 0 0 0 1 0]);
title('Class Grades');
```

Mission 5: World Pop. in Y2100

- Represent the predicted world population in year 2100 in a bar chart and a pie chart.
  - Get the raw data from Wikipedia.
  - http://en.wikipedia.org/wiki/World_population
  - Draw a bar char and a pie char based on the data.
  - Use sensible legend or tick label.





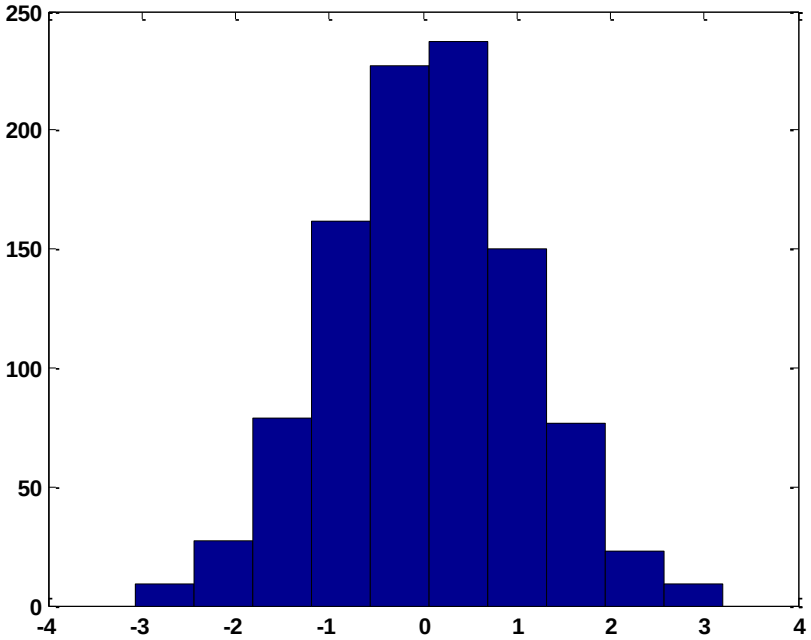
Solution 5

- Script and Screenshot

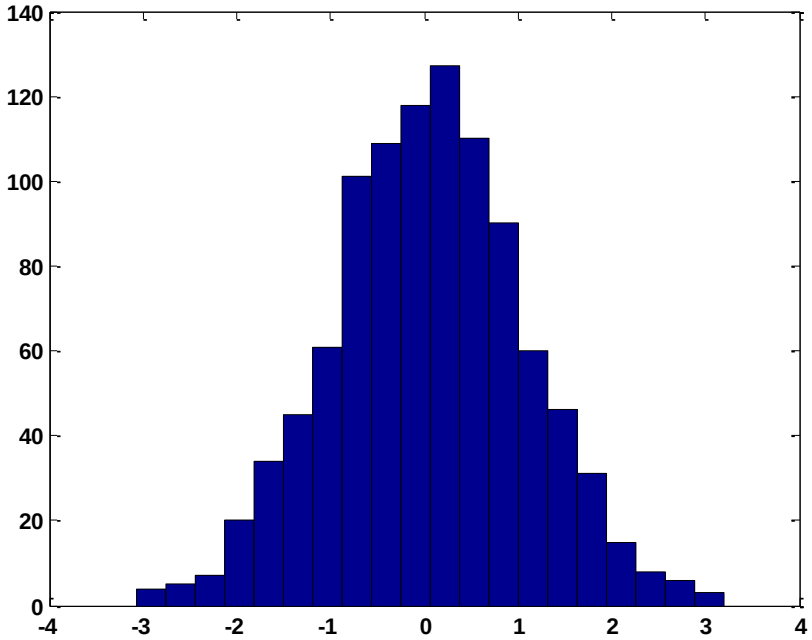
world_population.m

Histogram (1/2)

- Number of occurrences in each bin.
- The default number of bins is 10.



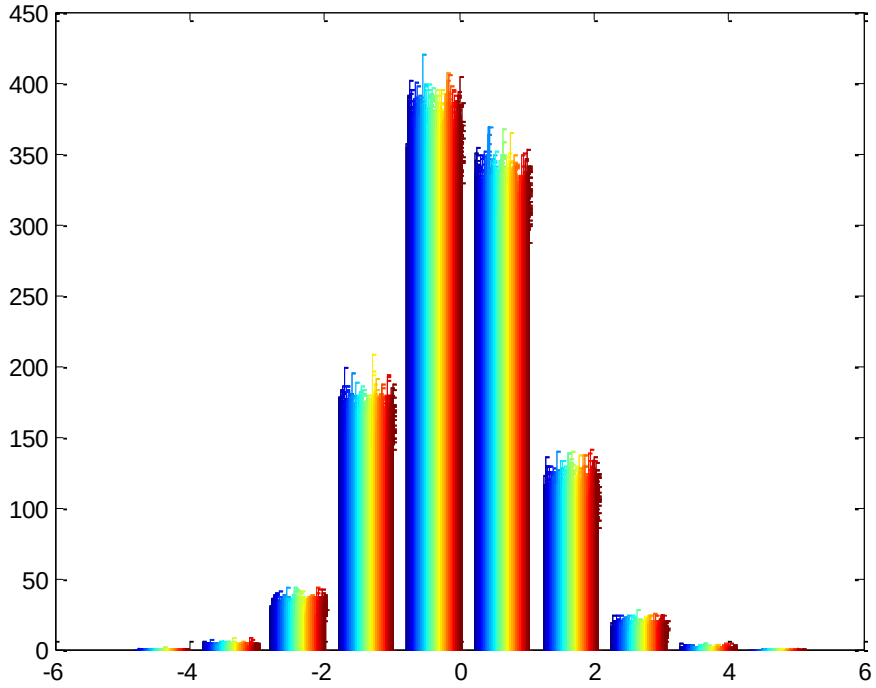

```
s=randn(1,1000);
hist(s)
```



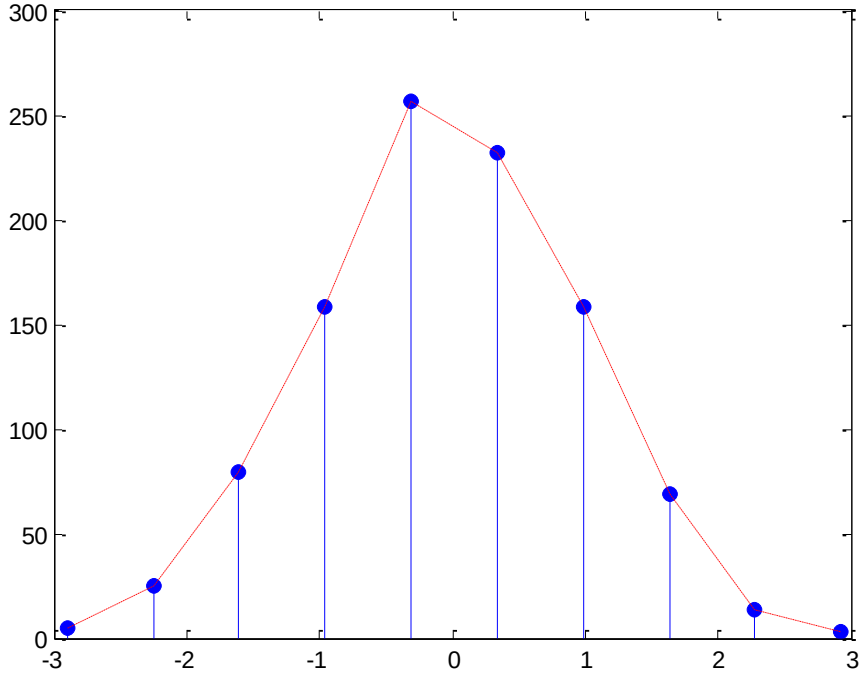

```
s=randn(1,1000);
hist(s, 20)
```

Histogram (2/2)

- Histogram of a Matrix
- Use of histogram data




```
s=randn(1000);
hist(s)
```




```
s=randn(1,1000);
[n x ] = hist(s);
stem(x,n,'filled'); hold on;
plot(x,n,'r-.');
```

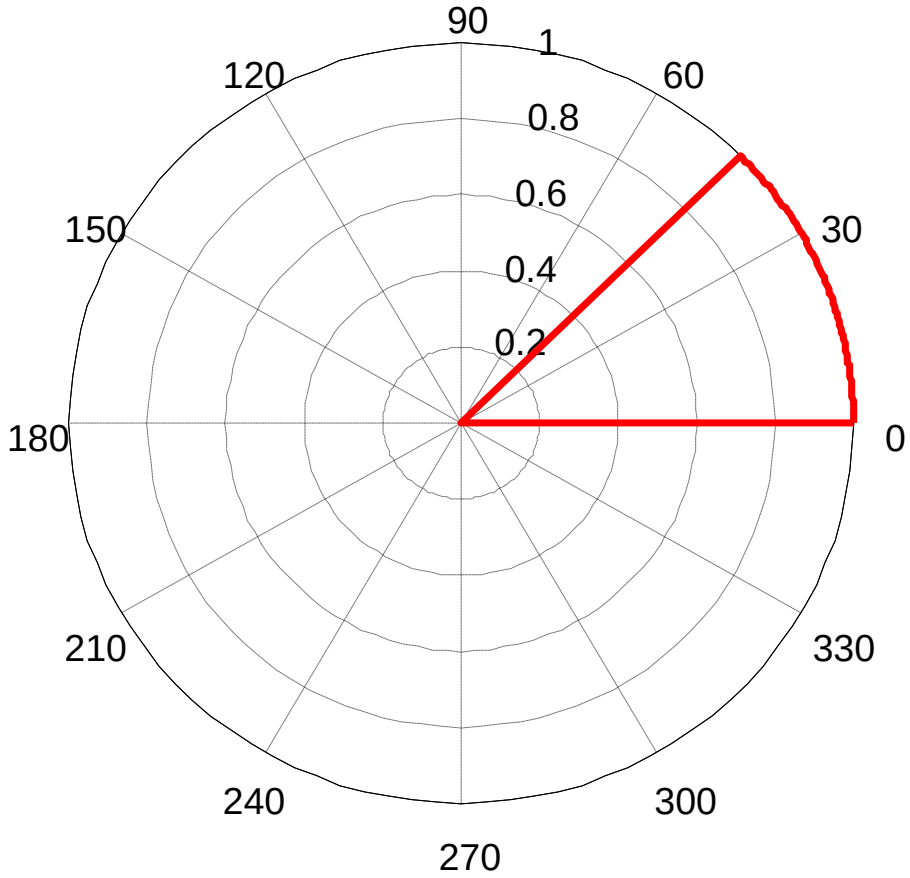
Polar Plot

- **polar(theta, radius, 'line spec')**

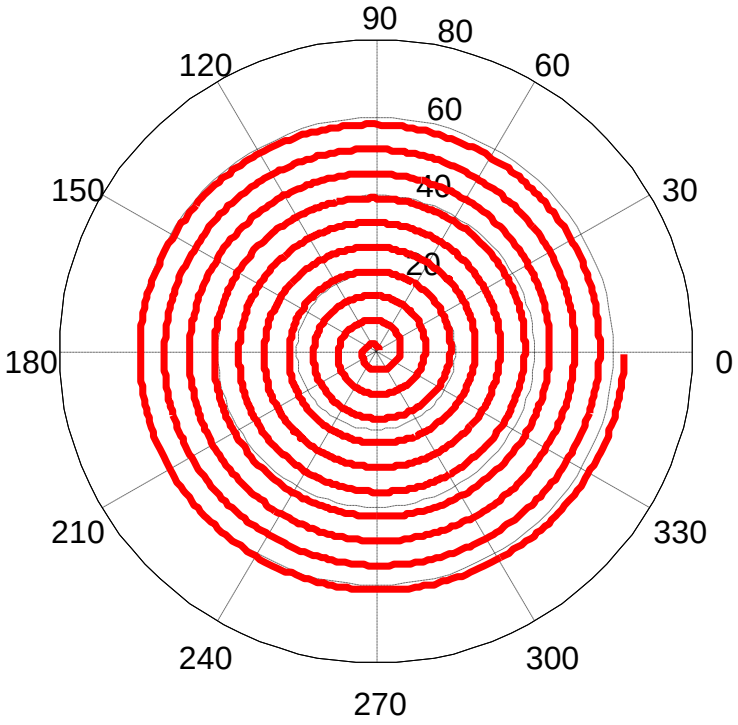
  θ r

polar_pie.m

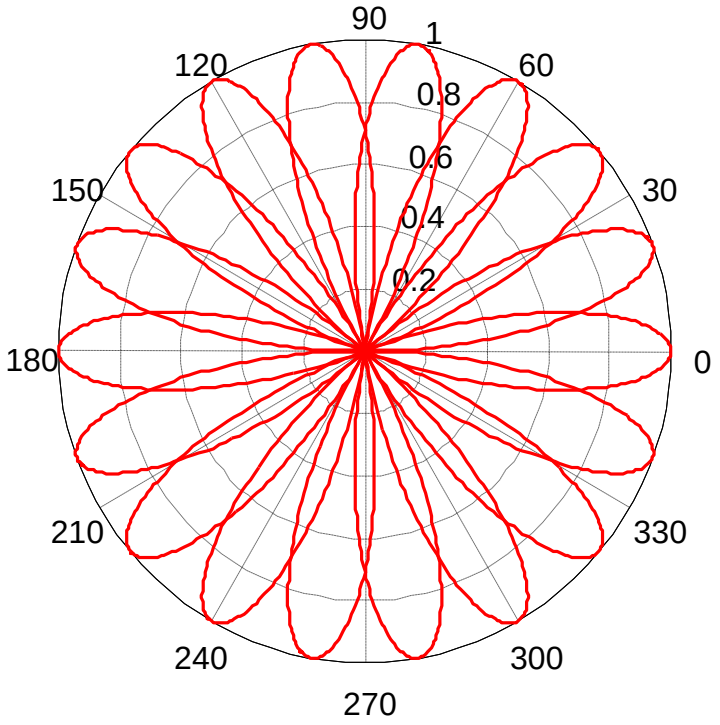
```
t  = 0:0.01:pi/4;
th = [t 0 0 ];
r  = [ones(1,length(t)) 0 1 ];
h  = polar( th, r, 'r-');
set(h,'LineWidth',3);
```



More Polar Plots



$$\theta = 0 \sim 20\pi$$
$$r = \theta$$



$$\theta = 0 \sim 4\pi$$
$$r = \sin(4.5\theta)$$

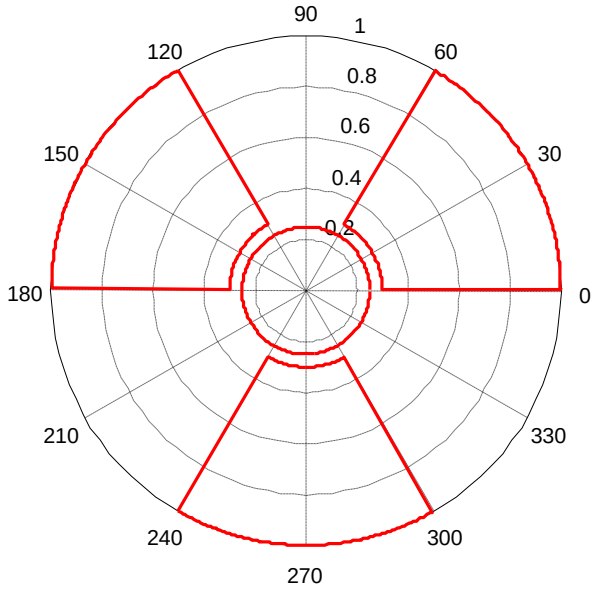
polar_spiral.m

```
th = 0:0.01:20*pi;
h = polar(th,th,'r-');
set(h, 'LineWidth',3);
```

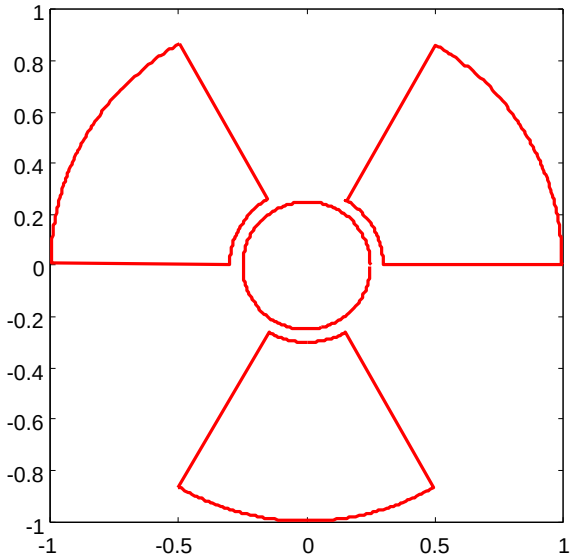
polar_petal.m

```
th = 0:0.01:4*pi;
r  = sin(4.5*th);
h = polar(th,r,'r-');
set(h, 'LineWidth',2);
```

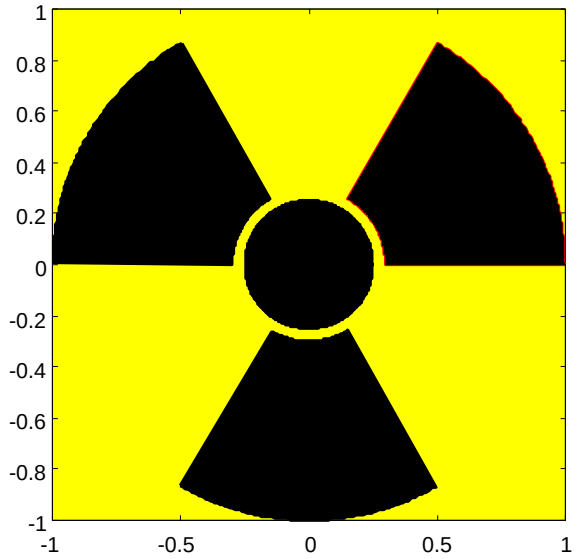
Polar / Plot / Patch



Polar



Plot



Patch

Script for Polar / Plot / Patch

- polar(θ , r) ←→plot($r\cos\theta$, $r\sin\theta$)
- patch($r\cos\theta$, $r\sin\theta$, Color)

polar_patch_demo.m

```
% Polar, Plot and Patch demo
%% a wing
t1 = 0:0.01:pi/3; % counter-clockwise
tr = t1(end:-1:1); % reverse direction
nt = length(t1);
th = [t1 tr 0];
r1 = [ones(1,nt) 0.3*ones(1,nt) 1];
%% a circle
t2 = 0:0.01:2*pi;
r2 = ones(1,length(t2));

%% Polar
figure(1);
h1  = polar( th, r1, 'r-'); hold on;
h2  = polar( th+2/3*pi, r1, 'r-');
h3  = polar( th-2/3*pi, r1, 'r-');
h0  = polar( t2, 0.25*r2, 'r-');
set([h0 h1 h2 h3], 'LineWidth', 3);
hold off;
```

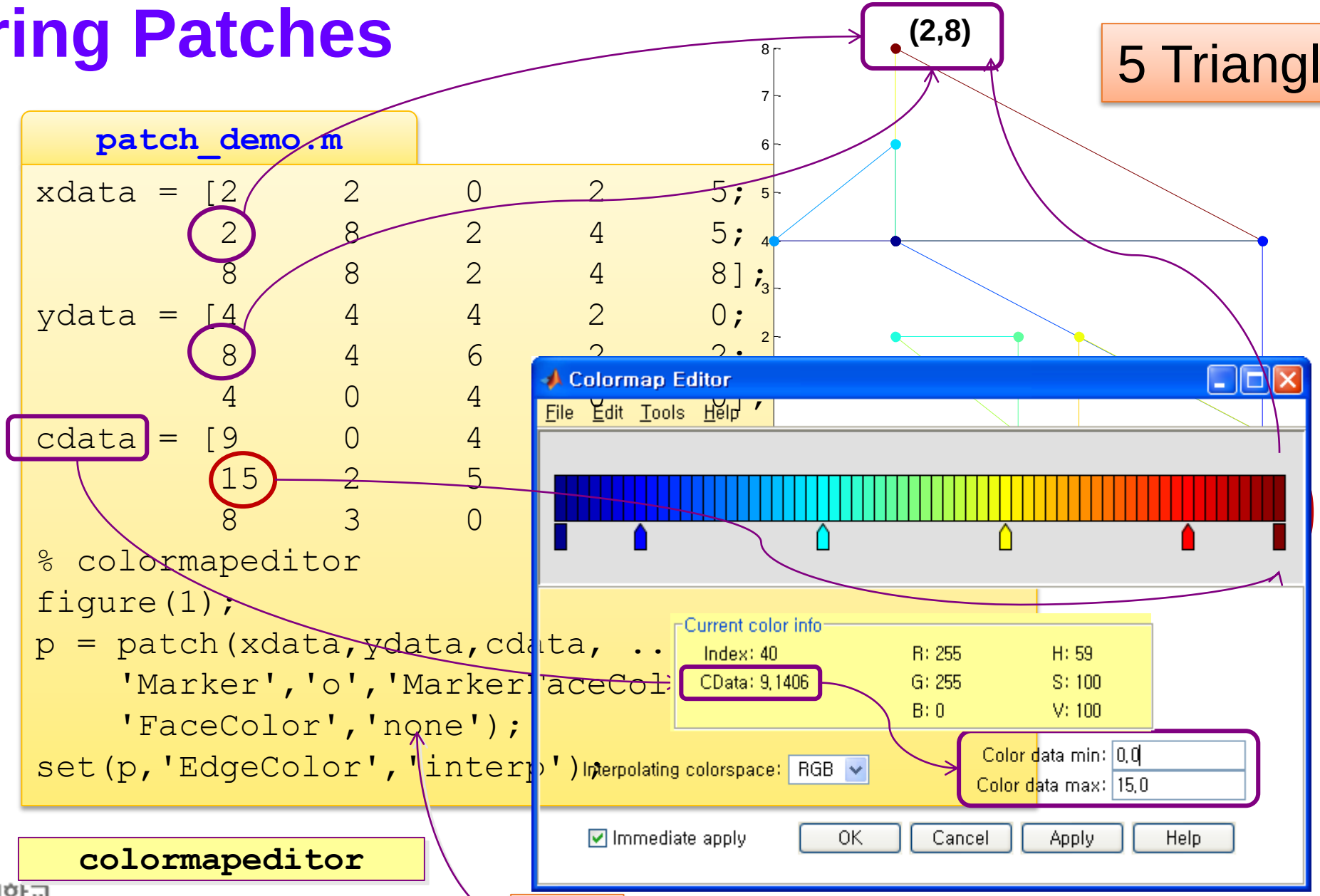
```
%% Plot
figure(2);
h1  = plot( r1.*cos(th), r1.*sin(th), 'r-'); hold on;
h2  = plot( r1.*cos(th+2/3*pi), r1.*sin(th+2/3*pi), 'r-');
h3  = plot( r1.*cos(th-2/3*pi), r1.*sin(th-2/3*pi), 'r-');
h0  = plot( 0.25*r2.*cos(t2), 0.25*r2.*sin(t2), 'r-');
set([h0 h1 h2 h3], 'LineWidth', 3);
axis square;
hold off;

%% Patch - filled polygons
figure(3);
h1  = patch( r1.*cos(th), r1.*sin(th), 'k'); hold on;
h2  = patch( r1.*cos(th+2/3*pi), r1.*sin(th+2/3*pi), 'k');
h3  = patch( r1.*cos(th-2/3*pi), r1.*sin(th-2/3*pi), 'k');
h0  = patch( 0.25*r2.*cos(t2), 0.25*r2.*sin(t2), 'k');
set(gca,'Color','yellow');
axis square;
hold off;
```

Coloring Patches

patch_demo.m

```
xdata = [2     2     0     2     5;
         2     8     2     4     5;
         8     8     2     4     8];
ydata = [4     4     4     2     0;
         8     4     6     2     2;
         4     0     4     0     0];
cdata = [9     0     4     0     0;
        15     2     5     2     2;
         8     3     0     3     3];
% colormapeditor
figure(1);
p = patch(xdata,ydata,cdata, ..
    'Marker','o','MarkerFaceCol
    'FaceColor','none');
set(p,'EdgeColor','interp');
```

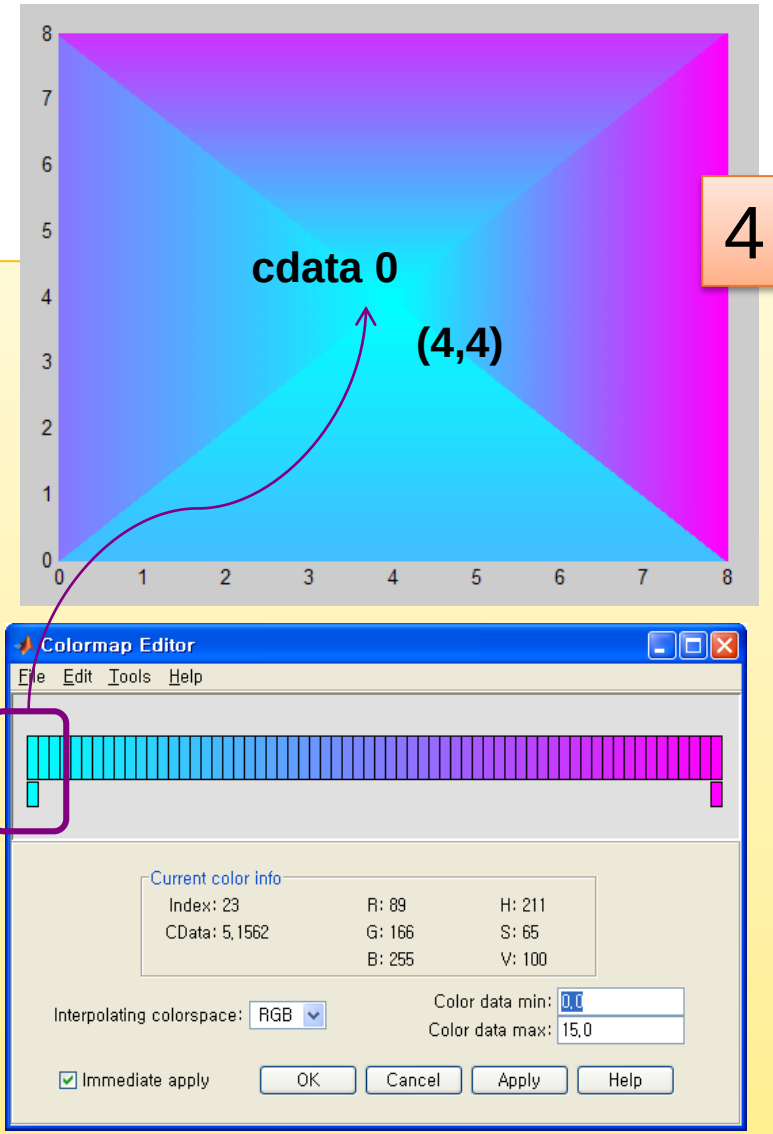
colormapeditor



Patch Pyramid

patch_pyramid.m

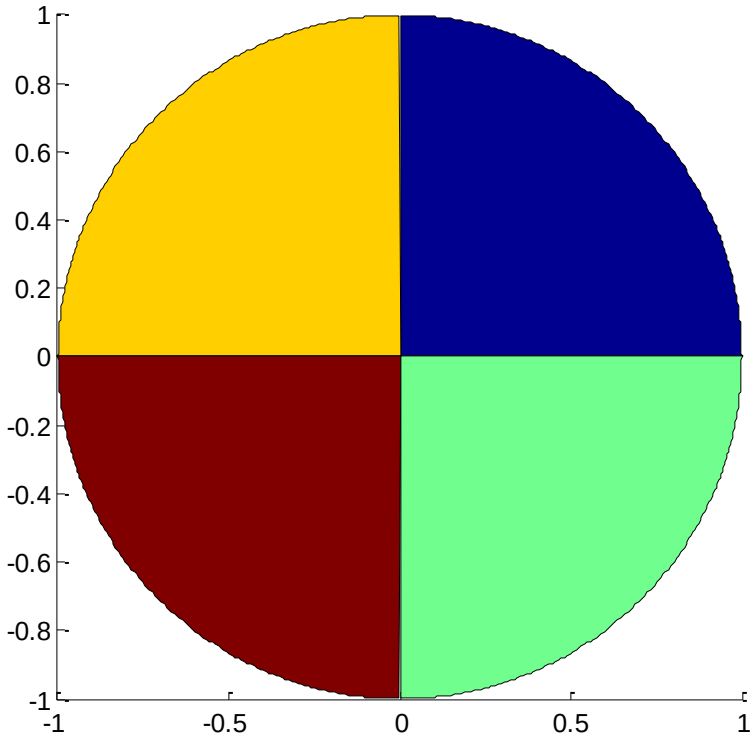
```
xdata = [0     0     0     8;
         8     0     8     8;
         4     4     4     4];
ydata = [0     0     8     8;
         0     8     8     0;
         4     4     4     4];
cdata = [4     8    12    15;
         4     8    12    15;
         0     0     0     0];
% Select the current colomap
colormap('Cool');
figure(1);
p = patch(xdata,ydata,cdata,'FaceColor','interp');
set(p,'EdgeColor','interp');
```



4 Triangles

Mission 6a: Color Ball

- Write a script for plotting the color ball below using patch.



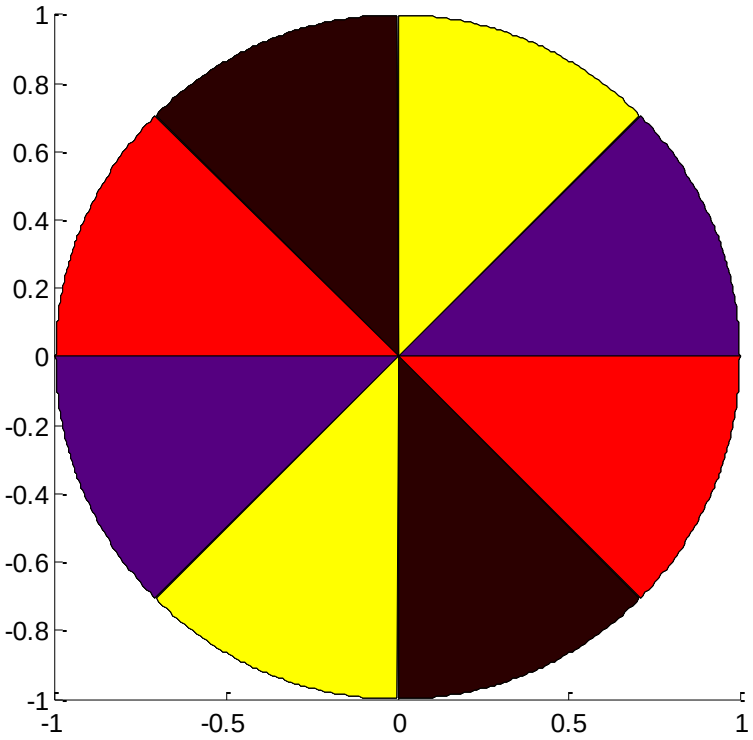
Solution 6a

- Script and Screenshot

patch_color_ball.m

Mission 6b: Juggling Ball

- Write a script for plotting the color juggling ball below using patch.



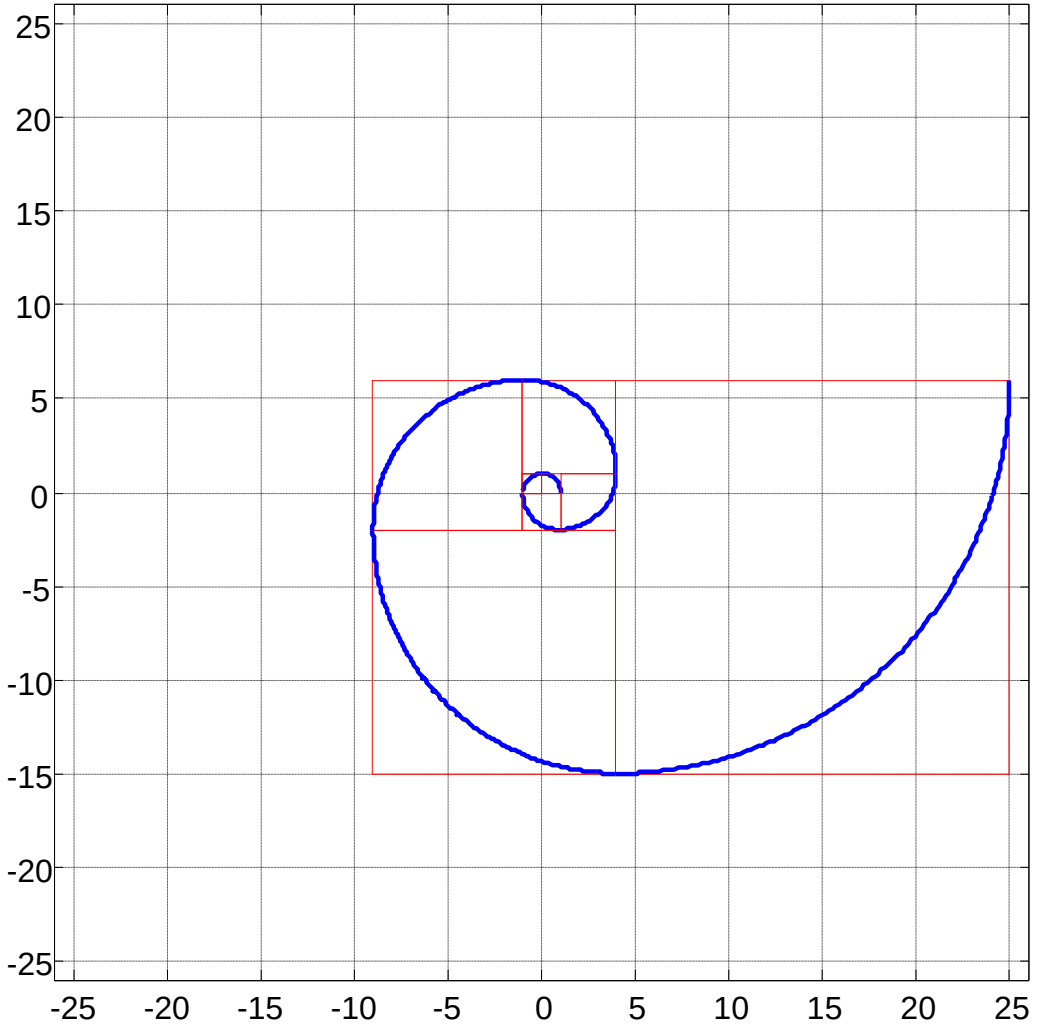
Solution 6b

- Script and Screenshot

patch_jcolor_ball.m

Mission 7: Fibonacci Plot

- Write a script for generating a Fibonacci plot.
  - Refer to Wikipedia for Fibonacci number.
  - Number of arcs N may be arbitrary.
  - Bonus for arbitrary starting direction and rotation control.

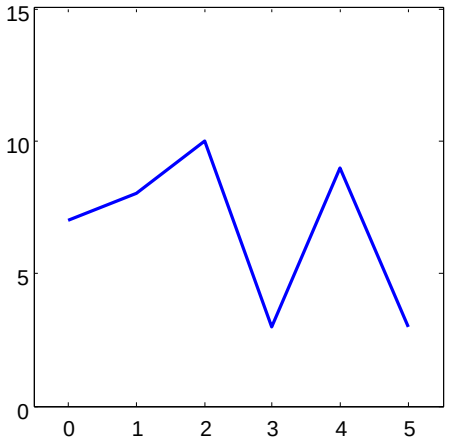


Solution 7

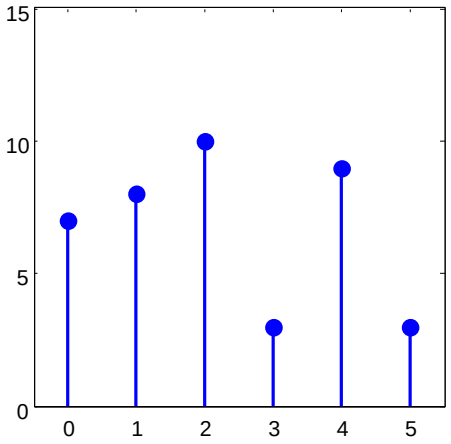
- Script and Screenshot

fibonacci_arcs.m

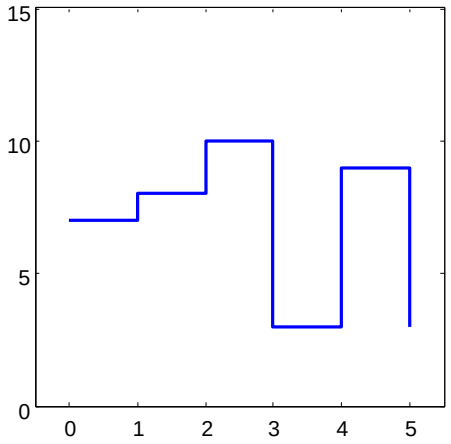
Subplot - Example



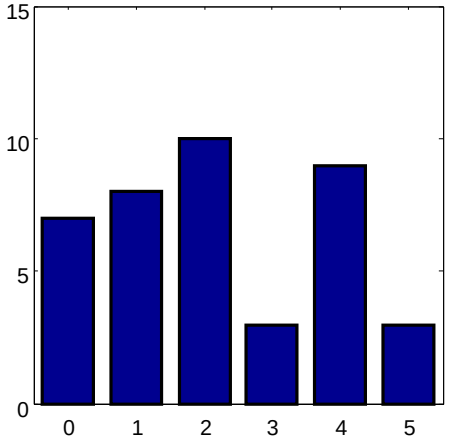
plot



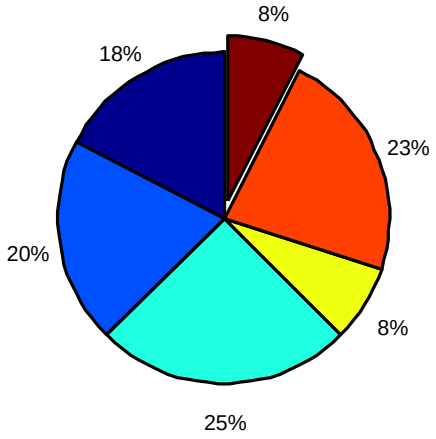
stem



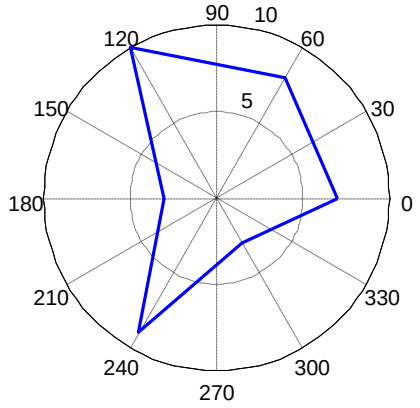
stairs



bar



pie



polar

Subplot - Script

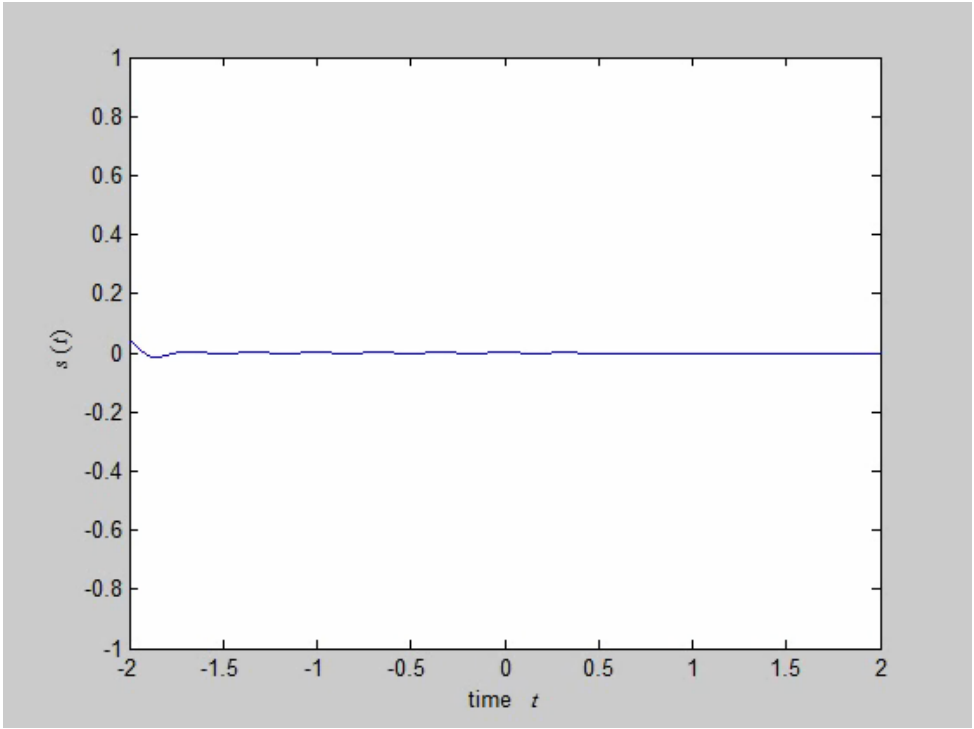
sbuplots.m

```
% Subplot Demo
y = floor(10*rand(1,6))+3;
x = [1:length(y)] - 1;

subplot(2,3,1); plot(x,y,'b-'); axis([-0.5 5.5 0 15]);
subplot(2,3,2); stem(x,y,'filled'); axis([-0.5 5.5 0 15]);
subplot(2,3,3); stairs(x,y,'b-'); axis([-0.5 5.5 0 15]);
subplot(2,3,4); bar(x,y,0.75); axis([-0.5 5.5 0 15]);
subplot(2,3,5); pie(y,[0 0 0 0 0 1]);
subplot(2,3,6); polar([x x(1)]*2*pi/6,[y y(1)],'b-');
```

Mission 8: Wave-packet

- Plot the following moving graph using the following equation.

$$s(t) = \cos\left(6\pi(t-d)\right)\ \exp\left(-\pi(t-d)^2\right), d = -3 \sim 3$$



for

pause

Play

Solution 8

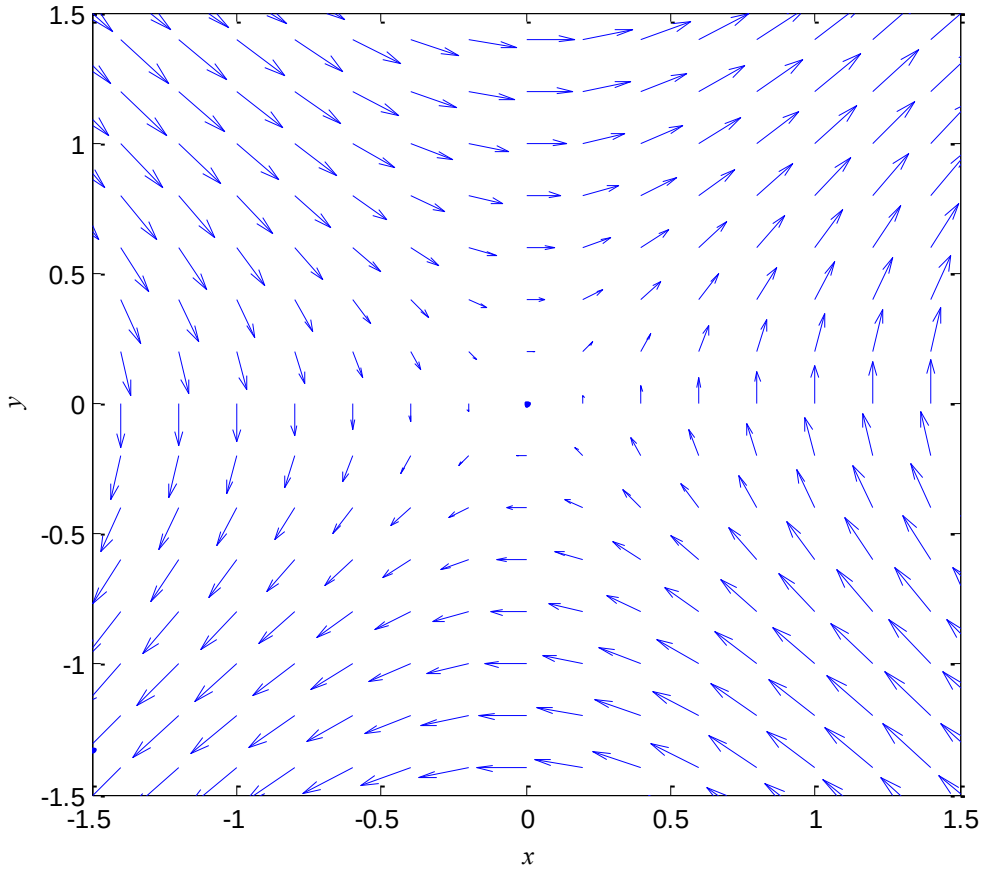
- Script and Screenshot

wavepacket.m

Mission 9: Vector Field

- Plot a vector field corresponding to the following differential equation.

$$\frac{dy}{dx} = \frac{\sin(x)}{\sin(y)}$$



meshgrid

quiver

Solution 9

- Script and Screenshot

vectorfield.m

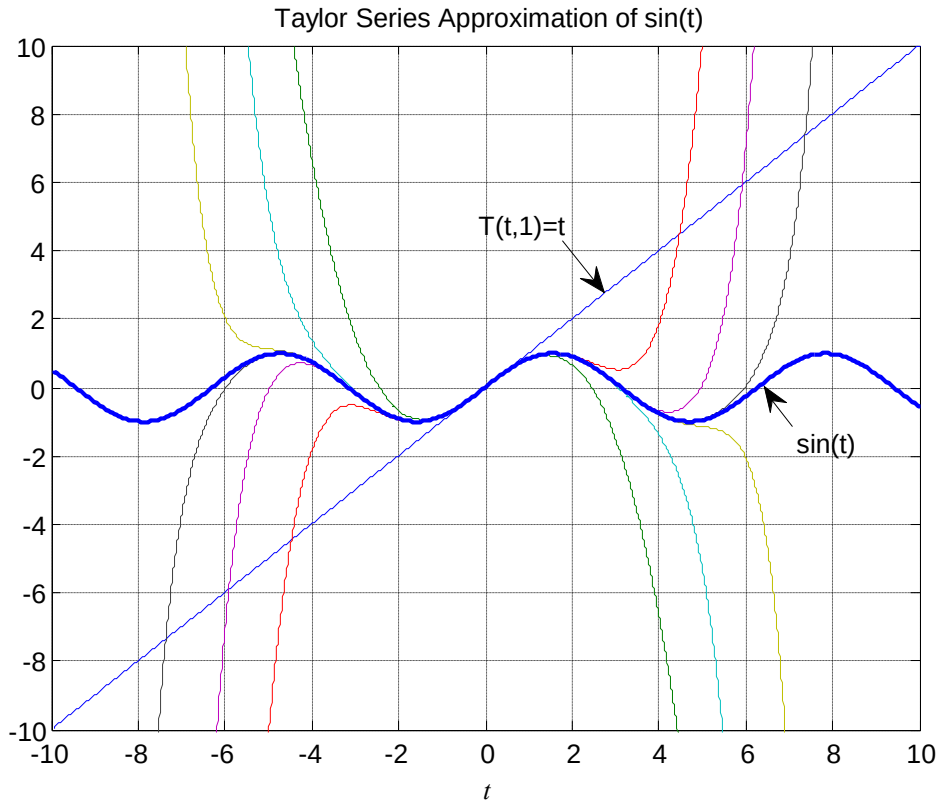
Mission 10 – Taylor Series

- The Taylor series expansions of $\sin(t)$ may be written as;

$$\sin(t) \cong T(t, N)$$

$$= \sum_{n=1}^{N} (-1)^{n-1} \frac{t^{2n-1}}{(2n-1)!}$$

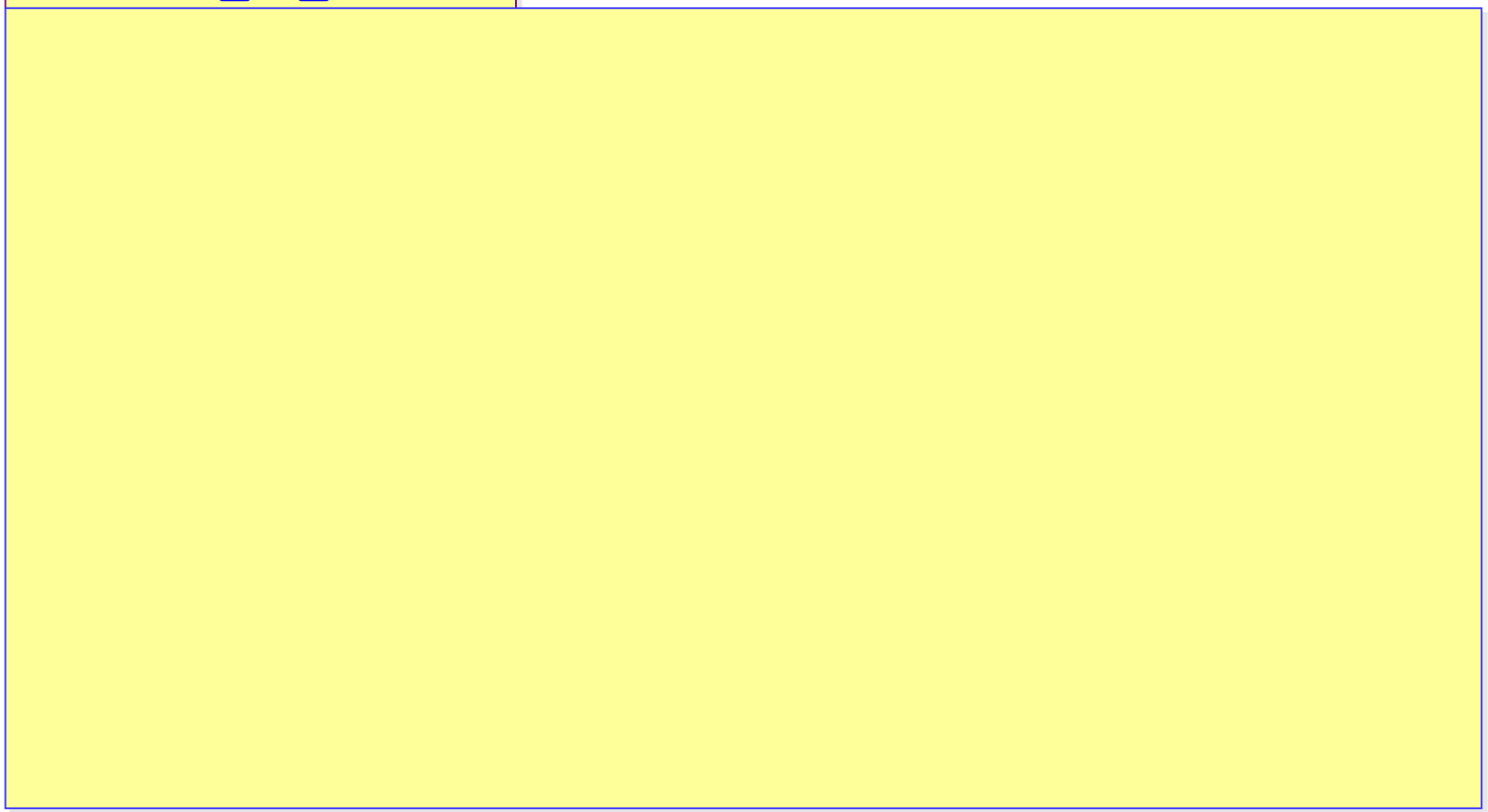
  - Plot $T(t, N)$ along with $\sin(t)$ for $N = 1 \sim 7$.
  - Add annotations and a title, as shown on the right graph.



Solution 10

- Script and Screenshot

Taylor_of_sin.m



Summary

- Recognize the following commands?

```
h = plot(x,y,'r-o');
set( h, 'LineWidth',2)
hold on
figure
figure(1)
set(gca, 'Color', 'yellow')
grid
axis square
title('KOSPI');
xlabel('{\it t}');
barh(X)
stem(t,y,'filled')
stairs(x,y)
polar(theta,rho)
box off
pie(data, [0 0 1 0])
patch(x,y,'k');
colormap('Cool')
legend('Theory','Data');
ezplot('cos(x)')
reshape(M,1,[])
repmat(N,2,1)
[s xbin] = hist(X)
```