SVKM'S NMIMS Deemed to be UNIVERSITY | NAVI MUMBAI

MATLAB
Unit 4-Lecture 12

BTech (CSBS) -Semester VII

26 August 2022, 09:35AM

*few figures/content have been prepared/referred from internet source/books

Basic plotting

- Overview,
- axis labels, and annotations,
- creating simple plots,
- specifying line styles and colours
- adding titles,
- multiple data sets in one plot,

Overlay plot

Method 1: Using the `plot` command to generate overlay plots

```
plot(x1,y1, x2,y2,':', x3,y3,'o')
```

Method 2: Using the `hold` command to generate overlay plots

```
% - Script file to generate an overlay plot with the hold  command -
x = linspace(0,2*pi,100);          % Generate vector x
y1 = sin(x);                       % Calculate y1
plot(x,y1)                         % Plot (x,y1) with solid line
hold on                            % Invoke hold for overlay plots
y2 = x; plot(x,y2,'--')            % Plot (x,y2) with dashed line
y3 = x - (x.^3)/6 + (x.^5)/120;    % Calculate y3
plot(x,y3,'o')                     % Plot (x,y3) as pts. marked by 'o'
axis([0 5 -1 5])                   % Zoom in with new axis limits
hold off                           % Clear hold command
```

Overlay plot

Method 3: Using the `line` command to generate overlay plots

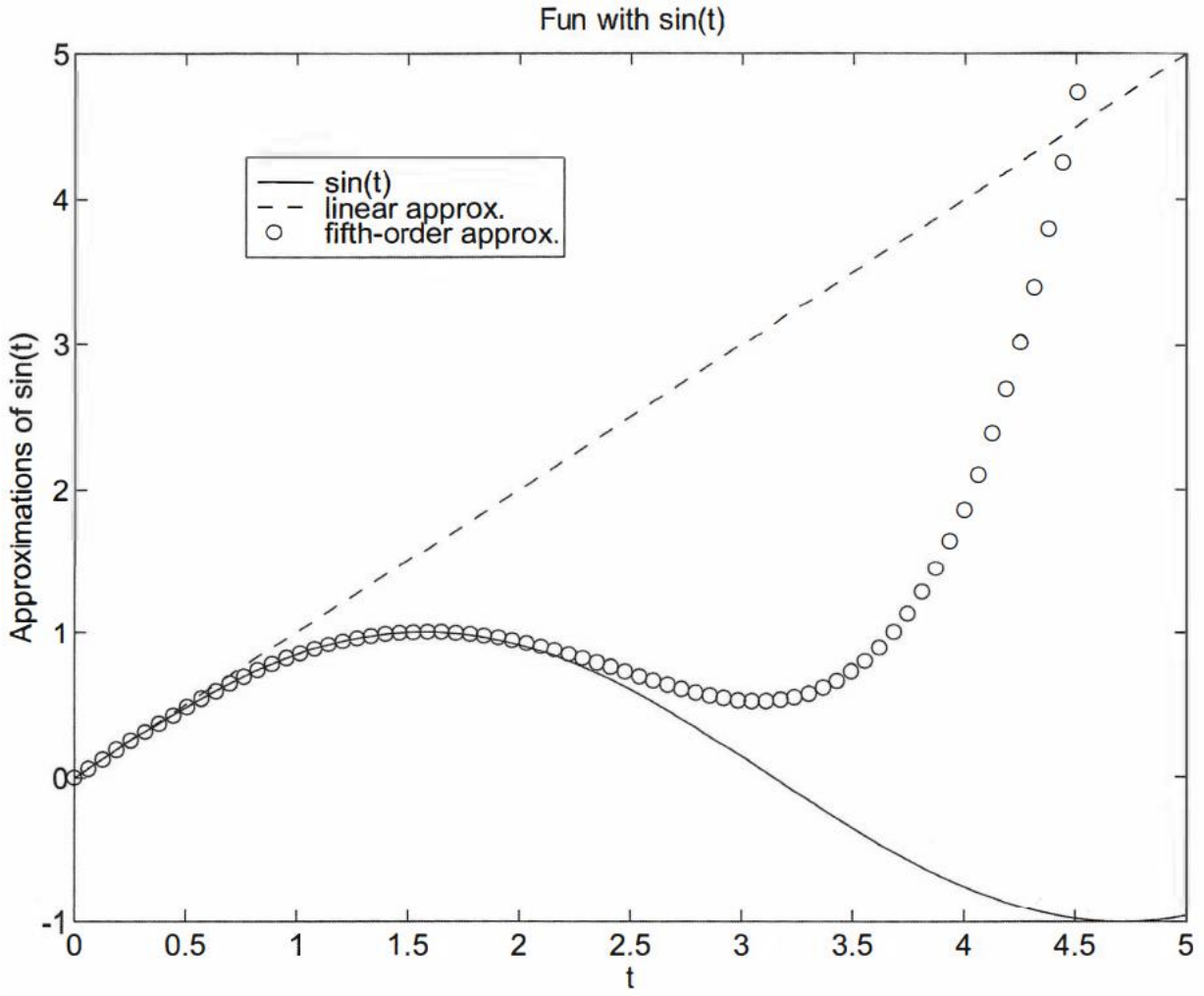
```
% -- Script file to generate an overlay plot with the line command --
% -----------------
% First, generate some data
t = linspace(0,2*pi,100);       % Generate vector t
y1 = sin(t);                    % Calculate y1, y2, y3
y2 = t;
y3 = t - (t.^3)/6 + (t.^5)/120;

% Now, plot the three lines
plot(t,y1)                      % Plot (t,y1) with (default) solid line
line(t,y2,'linestyle','--')     % Add line (t,y2) with dashed line and
line(t,y3,'marker','o',...      % Add line (t,y3) plotted with circles--
        'linestyle', 'none')    % but no line
% Adjust the axes
axis([0 5 -1 5])                % Zoom in with new axis limits

% Dress up the graph
xlabel('t')                     % Put x-label
ylabel('Approximations of sin(t)')
                                % Put y-label
title('Fun with sin(t)')        % Put title

legend('sin(t)','linear approx.','fifth-order approx.')
                                % add legend
```

Overlay plot



Specialized 2D plot

`area`	creates a filled area plot,
`bar`	creates a bar graph,
`barh`	creates a horizontal bar graph,
`comet`	makes an animated 2-D plot,
`compass`	creates arrow graph for complex numbers,
`contour`	makes contour plots,
`contourf`	makes filled contour plots,
`errorbar`	plots a graph and puts error bars,
`feather`	makes a feather plot,
`fill`	draws filled polygons of specified color,
`fplot`	plots a function of a single variable,

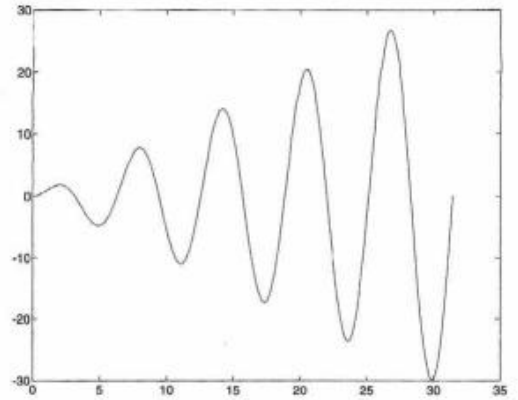
Specialized 2D plot

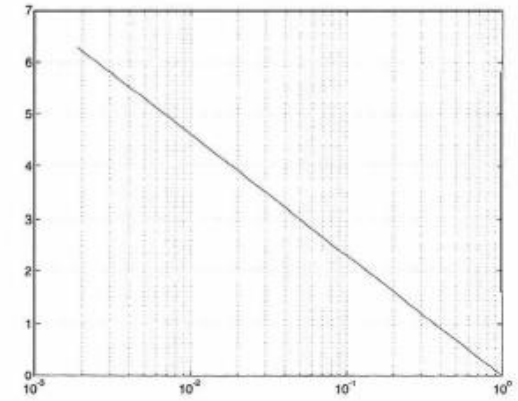
`fplot`	plots a function of a single variable,
`hist`	makes histograms,
`loglog`	creates plot with log scale on both the x-axis and the y-axis,
`pareto`	makes pareto plots,
`pcolor`	makes pseudocolor plot of a matrix,
`pie`	creates a pie chart,
`plotyy`	makes a double y-axis plot,
`plotmatrix`	makes a scatter plot of a matrix,
`polar`	plots curves in polar coordinates,
`quiver`	plots vector fields,
`rose`	makes angled histograms,
`scatter`	creates a scatter plot,

Specialized 2D plot

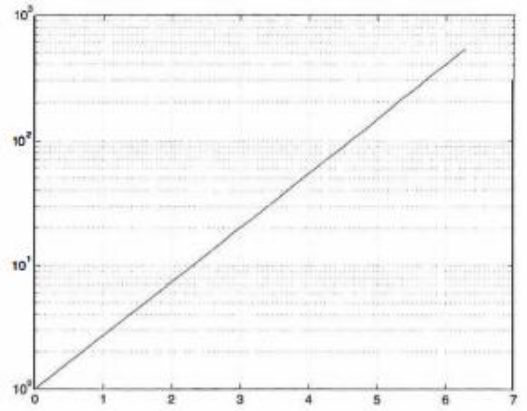
`semilogx`	makes semilog plot with log scale on the x-axis,
`semilogy`	makes semilog plot with log scale on the y-axis,
`stairs`	plots a stair graph, and
`stem`	plots a stem graph.

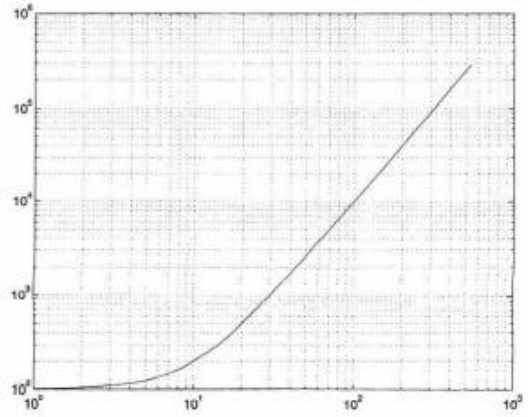
Specialized 2D plot

Function	Example Script	Output
fplot	$f(t) = t\sin t, \ \ 0 \le t \le 10\pi$ `fplot('x.*sin(x)',[0 10*pi])` Note that the function to be plotted must be written as a function of x.	

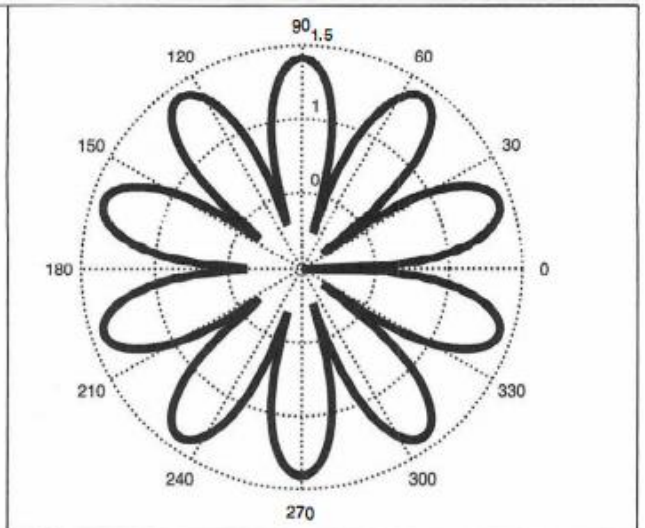
semilogx	$x = e^{-t}, \ \ y = t, \ 0 \le t \le 2\pi$
	`t = linspace(0,2*pi,200);`
	`x = exp(-t); y = t;`
	`semilogx(x,y), grid`
		

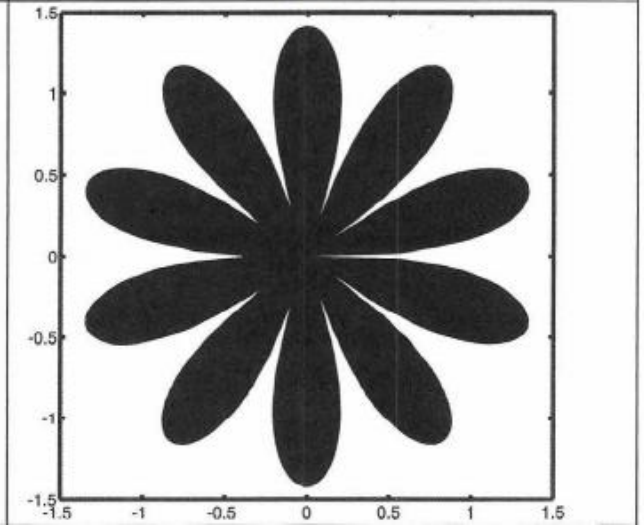
Specialized 2D plot

semilogy	$x = t,\ y = e^t,\ \ 0 \le t \le 2\pi$
	`t = linspace(0,2*pi,200);`
	`semilogy(t,exp(t))`
	`grid`
		

loglog	$x = e^t,\ y = 100 + e^{2t},\ \ 0 \le t \le 2\pi$
	`t = linspace(0,2*pi,200);`
	`x = exp(t);`
	`y = 100 + exp(2*t);`
	`loglog(x,y), grid`
		

Specialized 2D plot

polar	$r^2 = 2\sin 5t, \quad 0 \le t \le 2\pi$
	`t = linspace(0,2*pi,200);`
	`r = sqrt(abs(2*sin(5*t)));`
	`polar(t,r)`
		

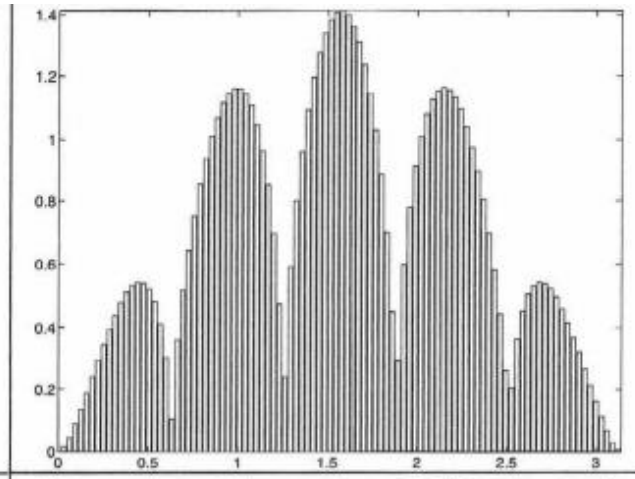
fill	$r^2 = 2\sin 5t, \quad 0 \le t \le 2\pi$
	$x = r\cos t, \quad y = r\sin t$
	`t = linspace(0,2*pi,200);`
	`r = sqrt(abs(2*sin(5*t)));`
	`x = r.*cos(t);`
	`y = r.*sin(t);`
	`fill(x,y,'k'),`
	`axis('square')`
		

Specialized 2D plot

bar

$$r^2 = 2\sin 5t, \quad 0 \le t \le 2\pi$$
$$y = r\sin t$$

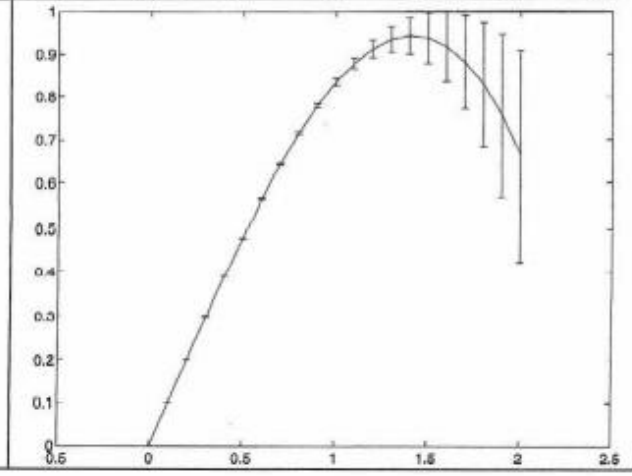
```
t = linspace(0,2*pi,200);
r = sqrt(abs(2*sin(5*t)));
y = r.*sin(t);
bar(t,y)
axis([0 pi 0 inf]);
```



errorbar

$$f_{\text{approx}} = x - \frac{x^3}{3!}, \quad 0 \le x \le 2$$
$$error = f_{\text{approx}} - \sin x$$

```
x = 0:.1:2;
aprx2 = x - x.^3/6;
er = aprx2 - sin(x);
errorbar(x,aprx2,er)
```

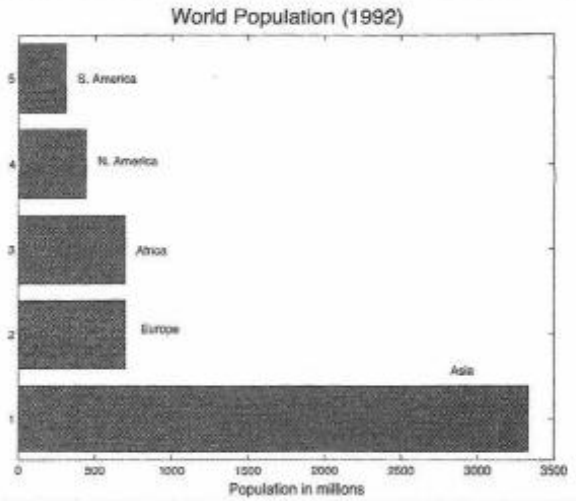


Specialized 2D plot

barh

World population by continents.

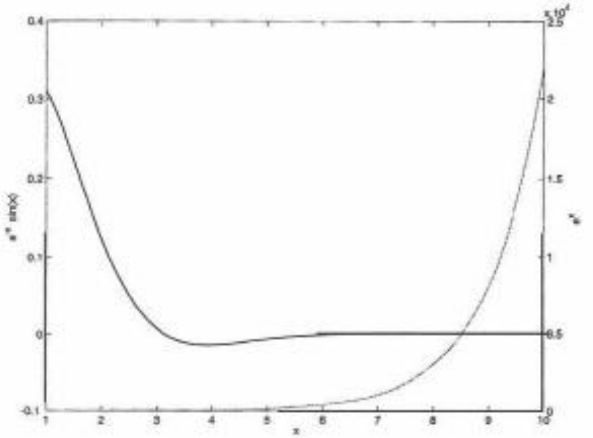
```
cont = char('Asia','Europe','Africa',...
     'N. America','S. America');
pop = [3332;696;694;437;307];
barh(pop)
for i=1:5,
      gtext(cont(i,:));
end
xlabel('Population in millions')
Title('World Population (1992)',
      'fontsize',18)
```



plotyy

$$y_1 = e^{-x}\sin x,\ 0 \le t \le 10$$
$$y_2 = e^{x}$$

```
x = 1:.1:10;
y1 = exp(-x).*sin(x);
y2 = exp(x);
Ax = plotyy(x,y1,x,y2);
hy1 = get(Ax(1),'ylabel');
hy2 = get(Ax(2),'ylabel');
set(hy1,'string','e^-x sin(x)');
set(hy2,'string','e^x ');
```

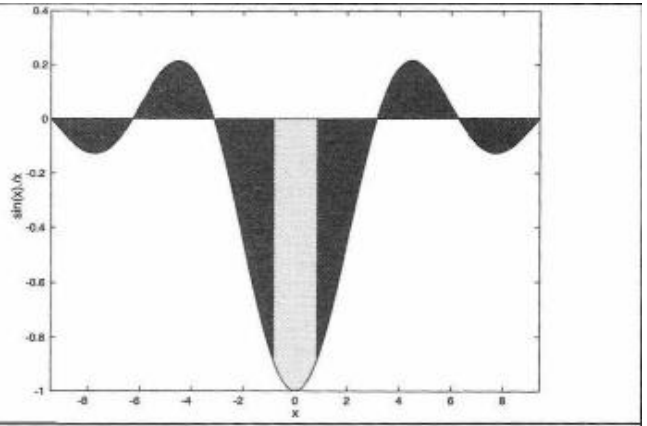


Specialized 2D plot

area

$$y = \frac{\sin(x)}{x}, \quad -3\pi \le x \le 3\pi$$

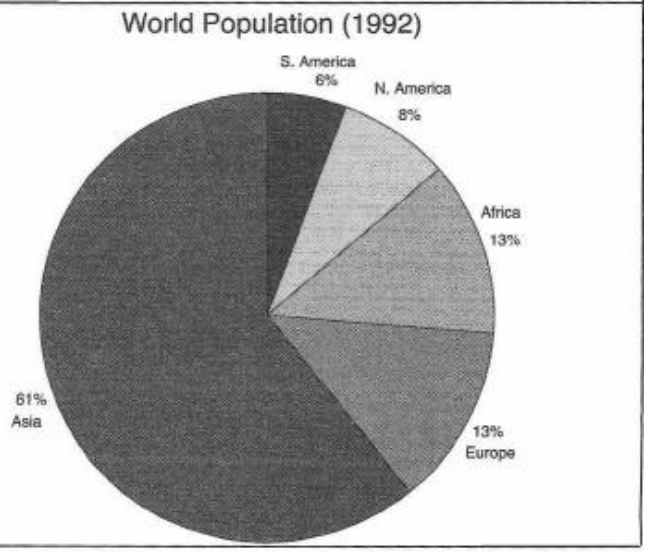
```
x = linspace(-3*pi,3*pi,100);
y = -sin(x)./x;
area(x,y)
xlabel('x'), ylabel('sin(x)./x')
hold on
x1 = x(46:55); y1 = y(46:55);
area(x1,y1,'facecolor','y')
```



pie

World population by continents.

```
cont = char('Asia','Europe','Africa',...
     'N. America','S. America');
pop = [3332;696;694;437;307];
pie(pop)
for i=1:5,
      gtext(cont(i,:));
end
Title('World Population (1992)',...
     'fontsize',18)
```



Using subplot to multiple plot

If you want to make a few plots and place the plots side by side (not overlay), use the **subplot** command to design your layout. The subplot command requires three integer arguments:

```
subplot(m,n,p)
```