MATH 3341: Introduction to Scientific Computing Lab

Libao Jin

University of Wyoming

February 19, 2020

Lab 04: Plotting Data

Basic Plotting

Create a `figure` window

Command	Description
`figure`	Creates a new figure window, and returns its handle.
`figure(H)`	Makes `H` the current figure, forces it to become visible, and raises it above all other figures on the screen. If Figure `H` does not exist, and `H` is an integer, a new figure is created with handle `H`.

Linear plot

Command	Description
`plot(X, Y)`	Plots vector `Y` versus vector `X`. If `X` or `Y` is a matrix, then the vector is plotted versus the rows or columns of the matrix, whichever line up. If `X` is a scalar and `Y` is a vector, disconnected line objects are created and plotted as discrete points vertically at `X`.
`plot(Y)`	Plots the columns of `Y` versus their index. If `Y` is complex, `plot(Y)` is equivalent to `plot(real(Y),imag(Y))`. In all other uses of plot, the imaginary part is ignored.
`plot(X,Y,S)`	Plots vector `Y` versus vector `X` with specified style options in `S`.

Plotting Styles

Various line types, plot symbols and colors may be obtained with `plot(X,Y,S)` where `S` is a character string made from one element from any or all the following 3 columns:

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)		
	< triangle (left)		
	> triangle (right)		
	p pentagram		

Example: plot(X,Y)

```
% Example: plot(X,Y)
X = linspace(0, 2*pi, 100);
Y = sin(X);
figure(1);
plot(X,Y);
```

Example: `plot(X,Y)`

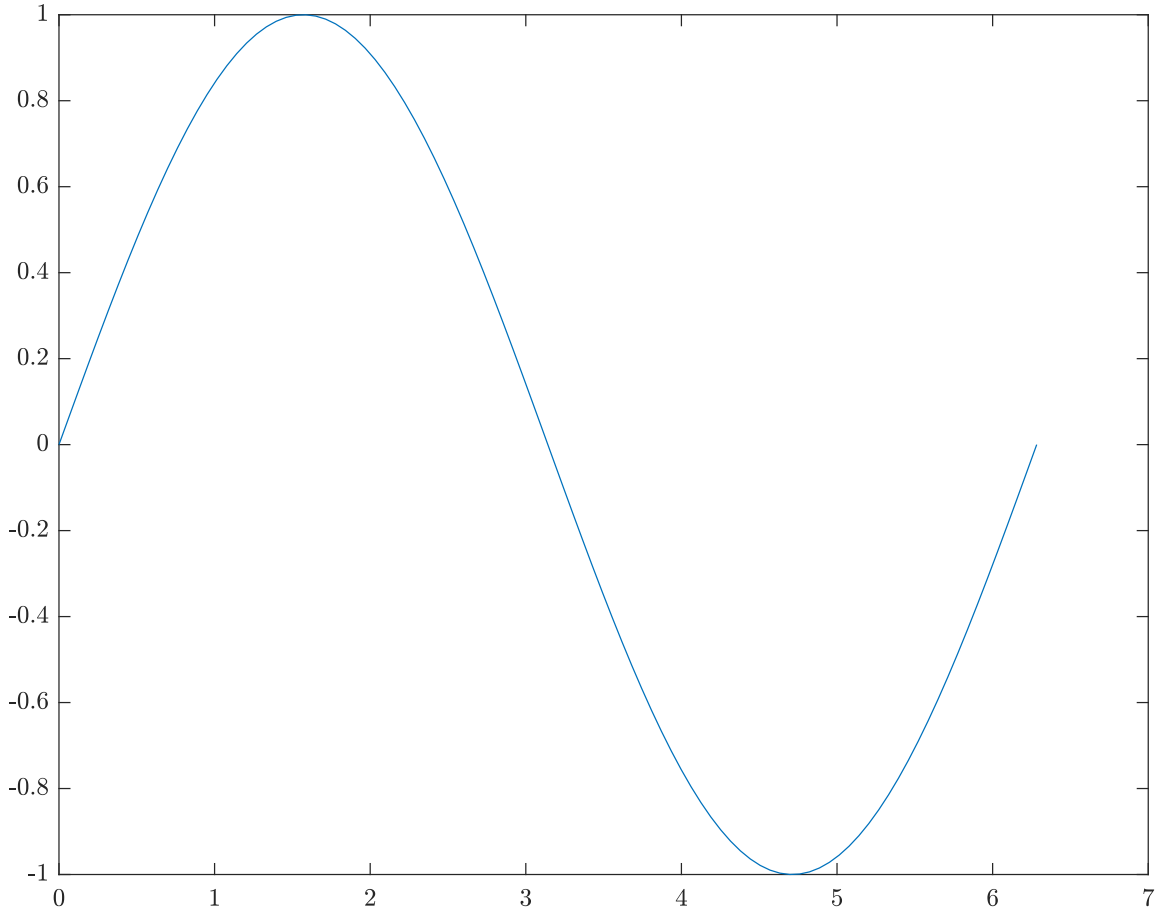

Figure 1:`plot(X,Y)`

Example: plot(Y)

```
% Example: plot(Y)
X = linspace(0, 2*pi, 100);
Y = sin(X);
figure(2);
plot(Y);
```

Example: `plot(Y)`

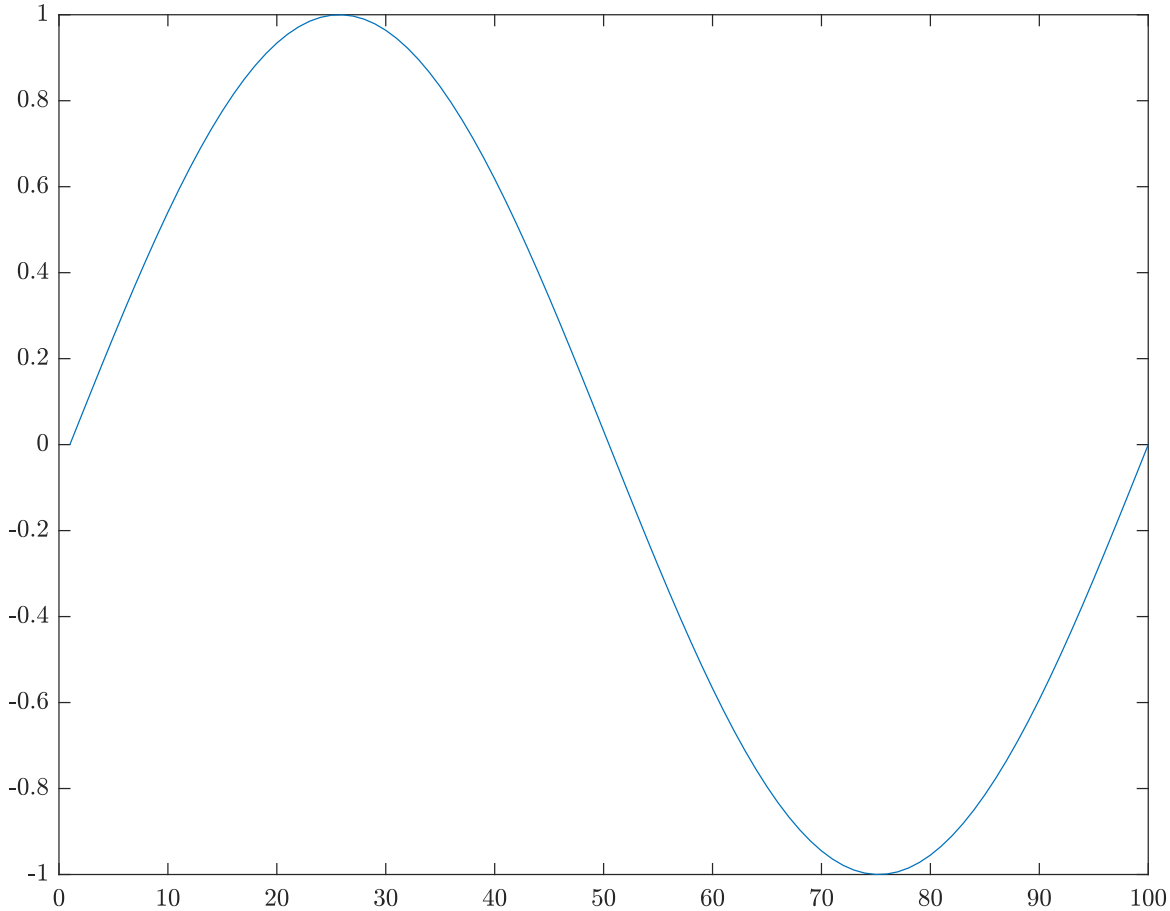

Figure 2:`plot(Y)`

Example: plot(X,Y,S)

```
% Example: plot(X,Y,S)
X = linspace(0, 2*pi, 100);
Y = sin(X);
S1 = 'go-.';  % green, circle, dashdot
S2 = 'r+:';   % red, plus, dotted
S3 = 'm*--';  % magenta, star, dashed
figure(3); plot(X,Y,S1);
figure(4); plot(X,Y,S2);
figure(5); plot(X,Y,S3);
```

Example: `plot(X,Y,S)`

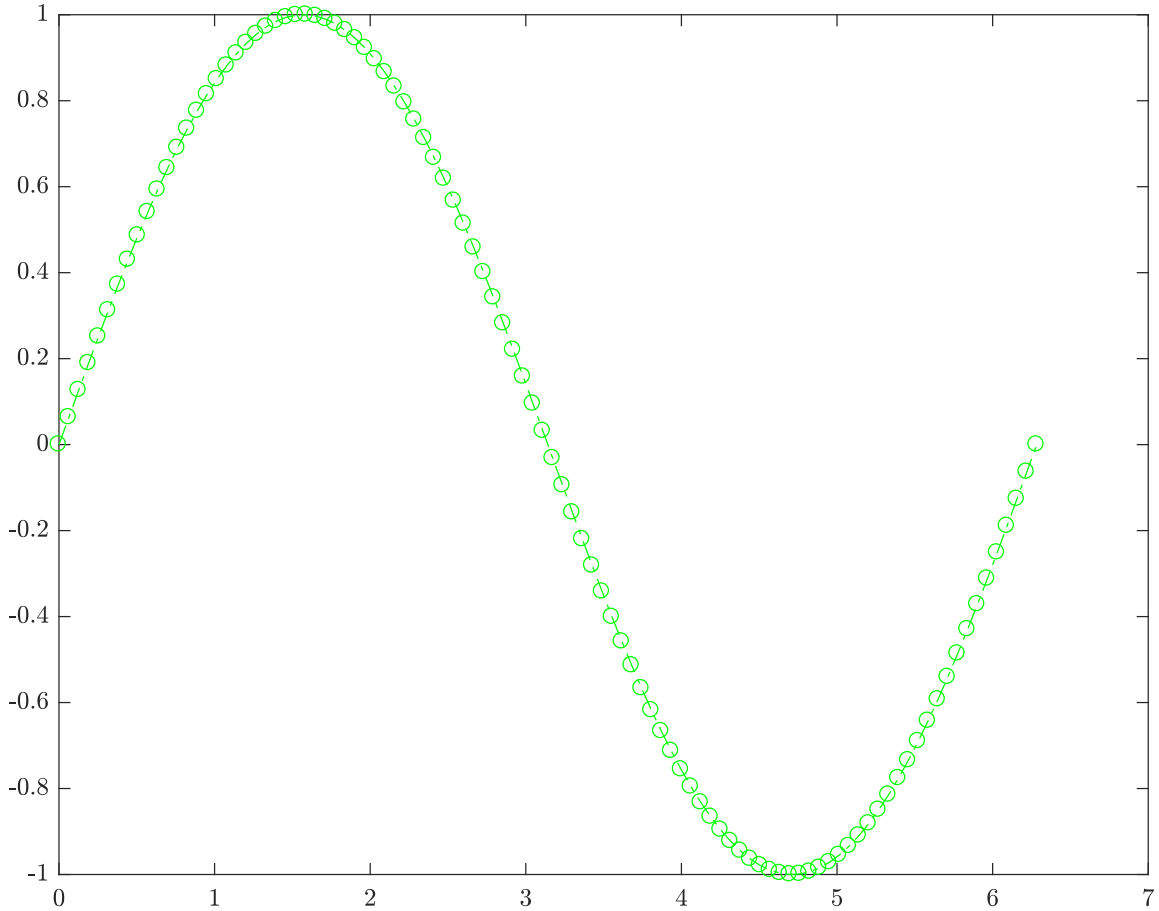

Figure 3:`plot(X,Y,'go-')`

Example: `plot(X,Y,S)`

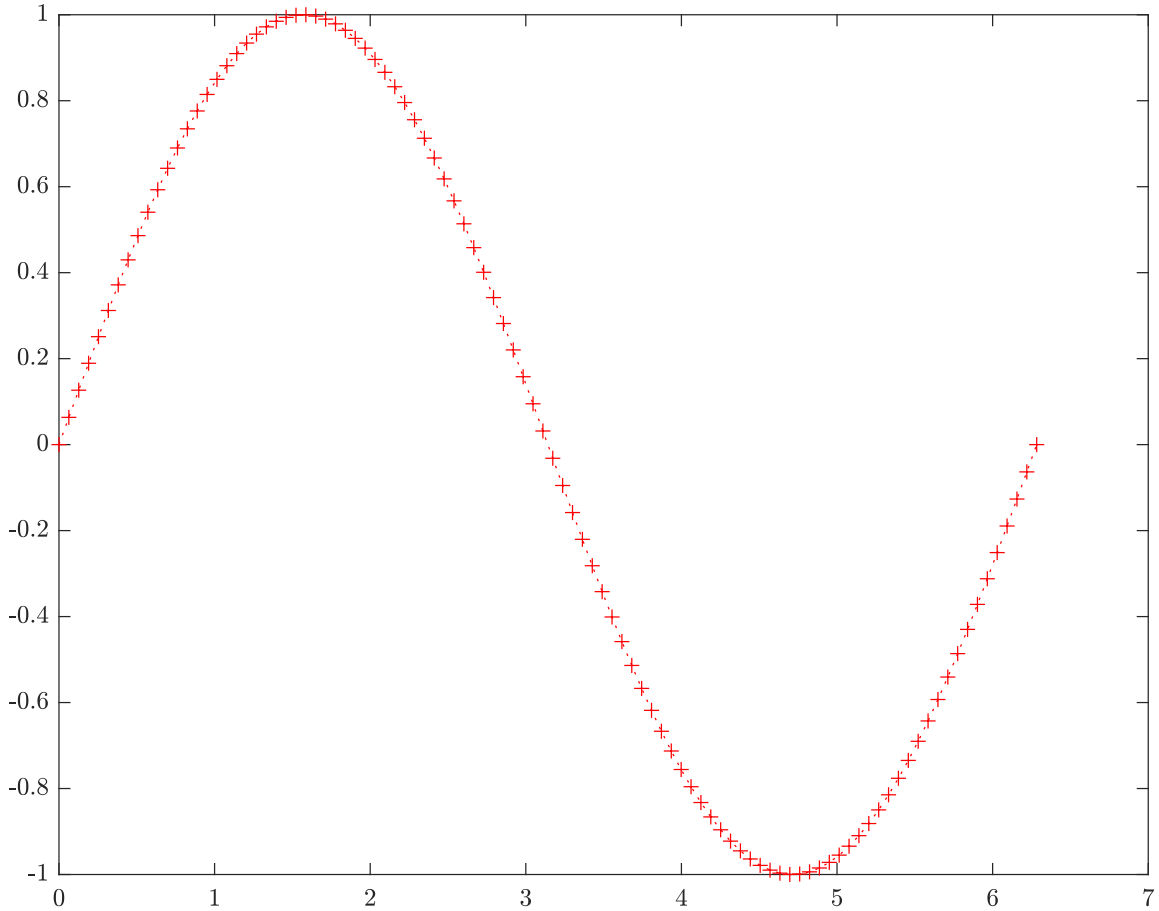

Figure 4:`plot(X,Y,'r+:')`

Example: `plot(X,Y,S)`

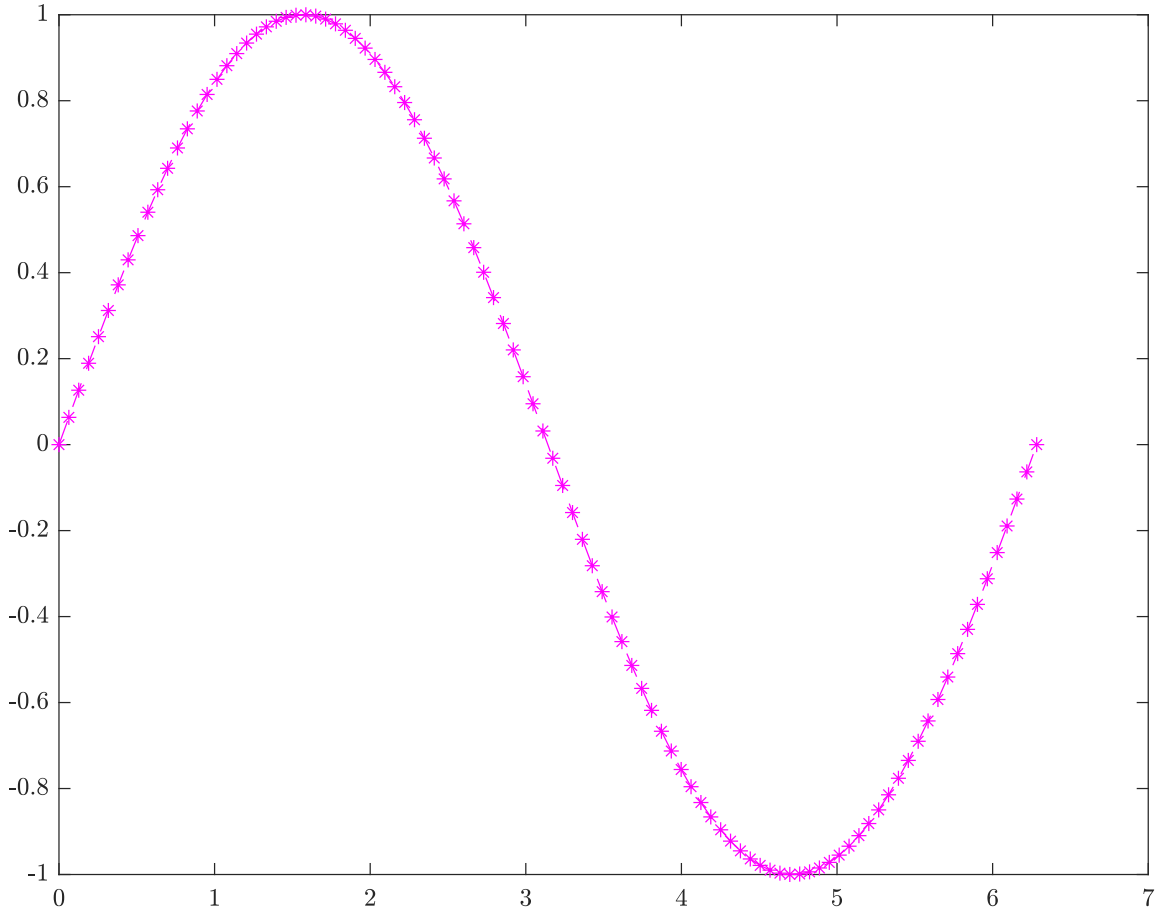

Figure 5:`plot(X,Y,'m*--')`

Multiple Plots in a Single Figure

- `plot(X1,Y1,S1,X2,Y2,S2,...)`: Combines the plots defined by the (X,Y,S) triples, where the X's and Y's are vectors or matrices and the S's are strings.
- `hold on`: holds the current plot and all axis properties, including the current color and linestyle, so that subsequent graphing commands add to the existing graph without resetting the color and linestyle.
- `hold off`: returns to the default mode whereby `plot` commands erase the previous plots and reset all axis properties before drawing new plots.

Example: plot(X1,Y1,S1,X2,Y2,S2,...)

```
% Example: plot(X1,Y1,S1,X2,Y2,S2,...)
X = linspace(0, 2*pi, 100);
Y1 = sin(X);
Y2 = cos(X);
Y3 = sin(2 * X);
S1 = 'go-.';
S2 = 'r+:';
S3 = 'm*--';
figure(6); plot(X,Y1,S1,X,Y2,S2,X,Y3,S3);
```

Example: `plot(X1,Y1,S1,X2,Y2,S2,...)`

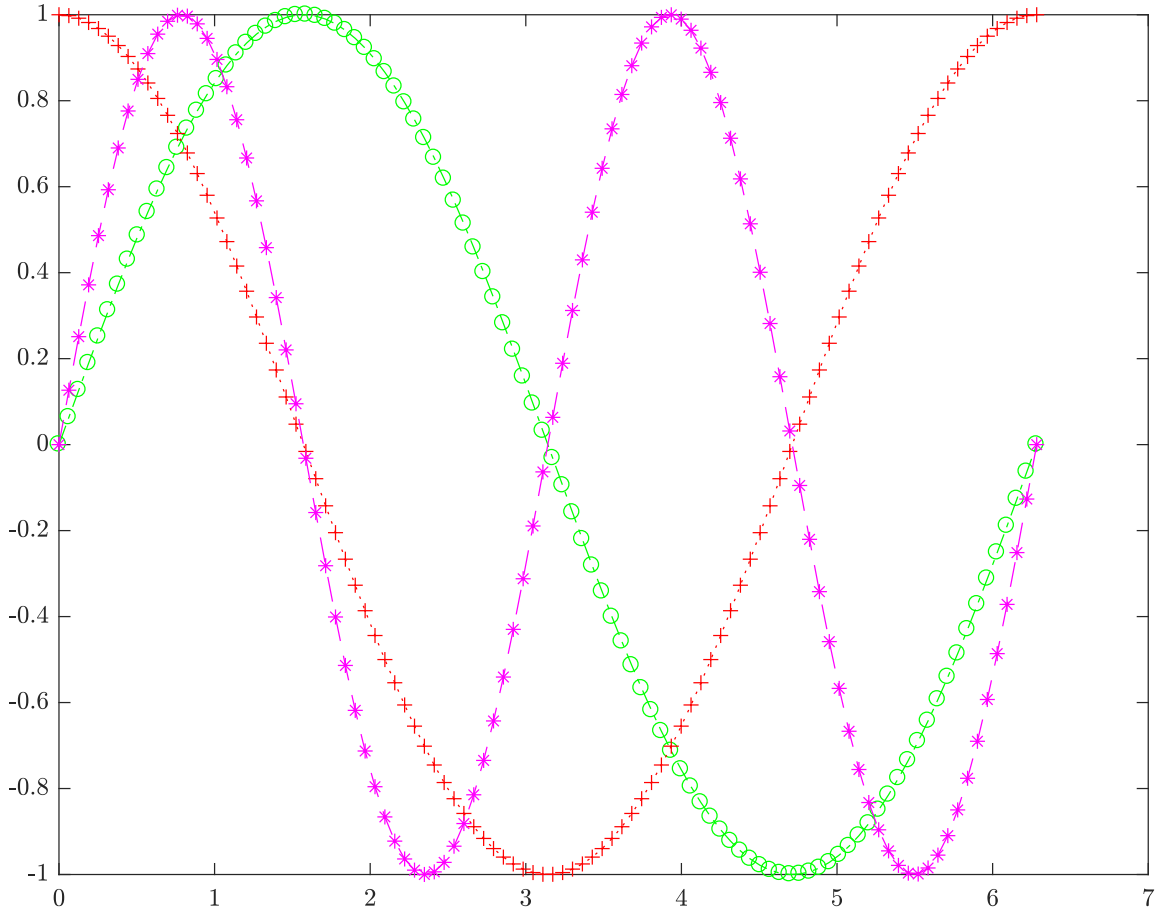

Figure 6:`plot(X,Y1,S1,X,Y2,S2,X,Y3,S3)`

Example: hold on

```
% Example: hold on
X = linspace(0, 2*pi, 100);
Y1 = sin(X);
Y2 = cos(X);
Y3 = sin(2 * X);
S1 = 'go-.';
S2 = 'r+:';
S3 = 'm*--';
figure(7);
hold on;
plot(X,Y1,S1);
plot(X,Y2,S2);
plot(X,Y3,S3);
```

Example: `hold on`

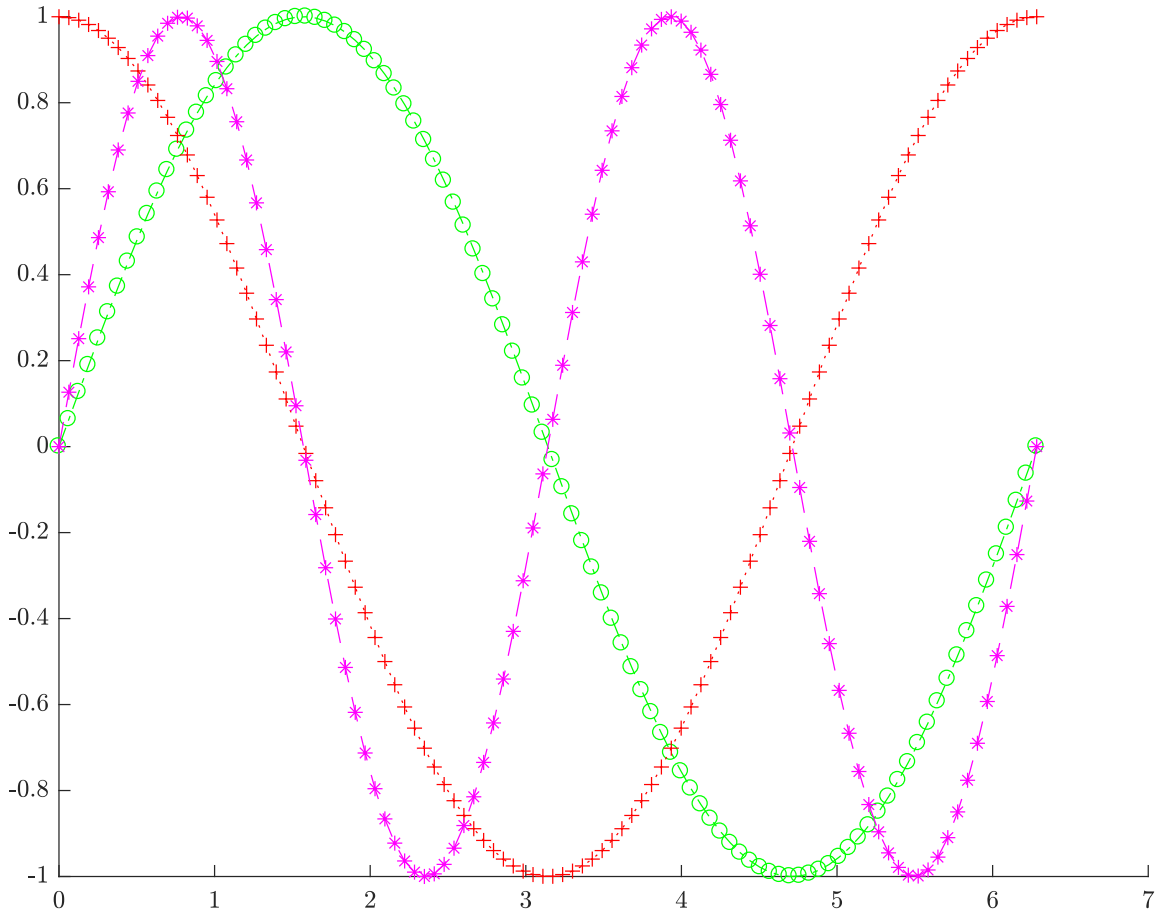

Figure 7:`hold on`

Add More Elements to the Plot

- `grid`: Grid lines.
- `xlabel`: X-axis label.
- `ylabel`: Y-axis label.
- `title`: Graph title.
- `legend`: Display legend.
- `axis`: Control axis scaling and appearance.

Example: title, grid, xlabel, ylabel, legend

```
% Example: title, grid, xlabel, ylabel, legend
X = linspace(0, 2*pi, 100);
Y1 = sin(X); Y2 = cos(X); Y3 = sin(2 * X);
S1 = 'go-.'; S2 = 'r+:'; S3 = 'm*--';
figure(8); hold on;
plot(X,Y1,S1); plot(X,Y2,S2); plot(X,Y3,S3);
title('Trig functions');
grid on;  % grid minor;
xlabel('$x$', 'interpreter', 'latex');
ylabel('$y$', 'interpreter', 'latex');
lgd = legend('$\sin(x)$', '$\cos(x)$', '$\sin(2x)$',...
             'Location', 'best');
lgd.Interpreter = 'latex';
axis([0, 2*pi, -1, 1]);
```

Example: `title`, `grid`, `xlabel`, `ylabel`, `legend`

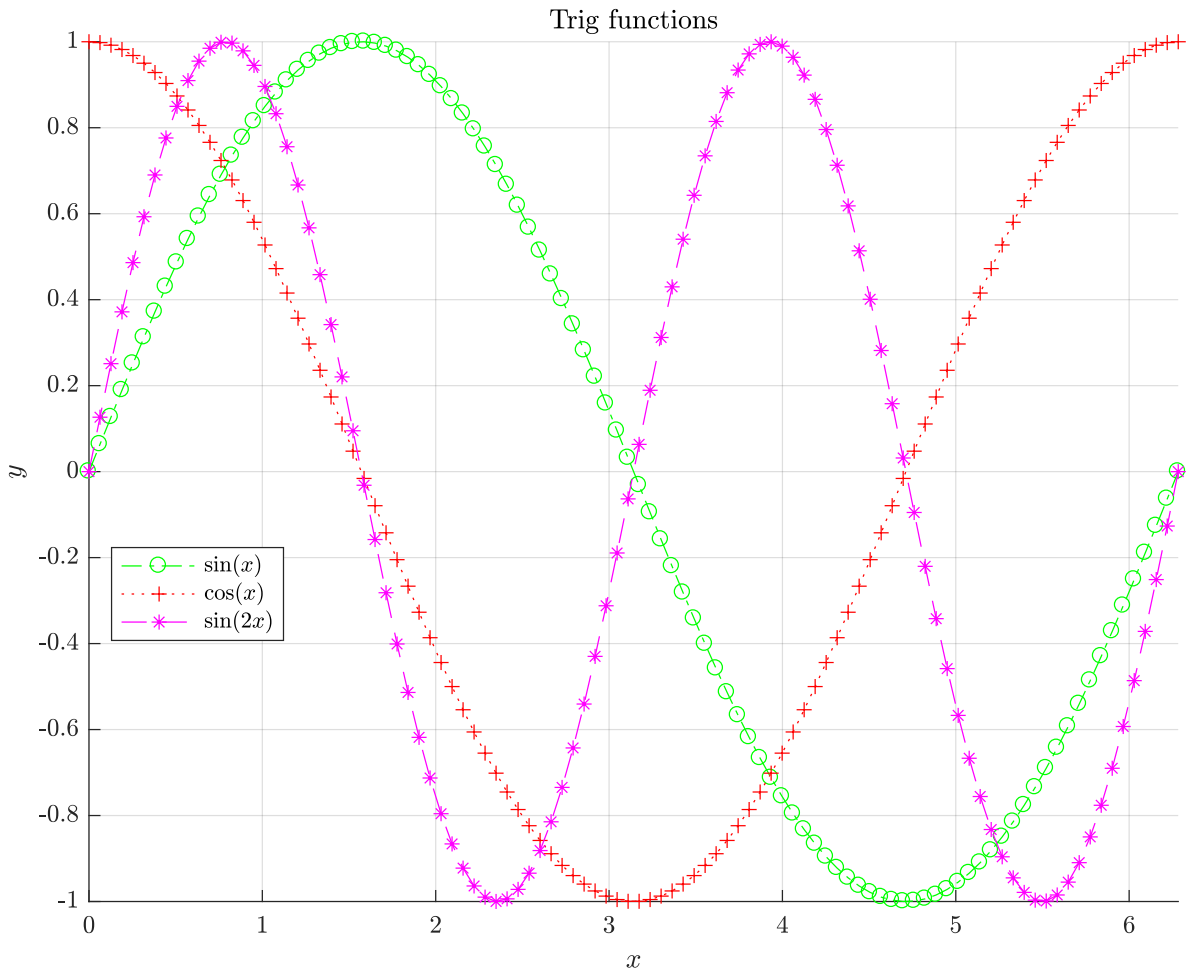

Figure 8: `title`, `grid`, `xlabel`, `ylabel`, `legend`

Advanced Plotting

Modifying Properties after Plotting

- `gcf`: Get handle to current figure.
- `gca`: Get handle to current axis.
- `get`: Get object properties.
- `set`: Set object properties.

Example: gcf, gca, get, set

```
% Example: gcf, gca, get, set
X = linspace(0, 2*pi, 100); Y = sin(X);
figure(9);
plot(X, Y);
axis([0, 2*pi, -1, 1]);
set(get(gca, 'Title'), 'String', 'sin(x)');
set(get(gca,'Children'), 'LineWidth', 1.0,...
        'LineStyle', ':',...
        'Marker', 'd',...
        'MarkerSize', 4,...
        'MarkerEdgeColor', 'y',...
        'MarkerFaceColor', 'r');
set(gca, 'XTick', [0, pi / 2, pi, 3 * pi / 2, 2 * pi]);
set(gca, 'XTickLabel', {'0', '$\pi/2$', '$\pi$',...
          '$3 \pi / 2$', '$2\pi$'});
```

Example: gcf, gca, get, set

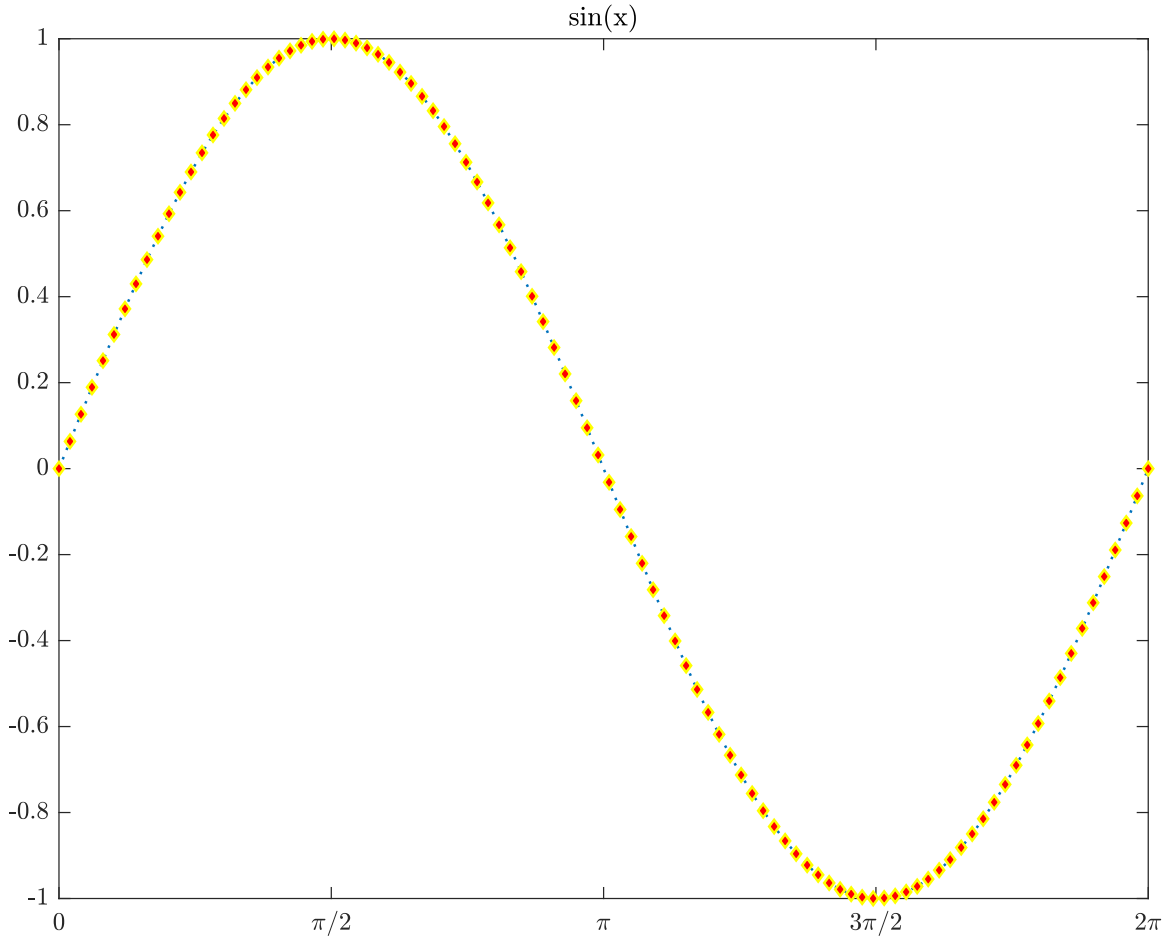

Figure 9:Example: gcf, gca, get, set

Create Axes in Tiled Positions: `subplot`

Run `help subplot` in the Command Window:

`subplot(m,n,p)`, or `subplot(mnp)`, breaks the Figure window into an `m`-by-`n` matrix of small axes, selects the p-th axes for the current plot, and returns the axes handle. The axes are counted along the top row of the Figure window, then the second row, etc.

Example: subplot

```
% Example: subplot
X = linspace(0, 2*pi, 100);
Y1 = sin(X); Y2 = cos(X);
Y3 = sin(2 * X); Y4 = cos(2 * X);
figure(10);
subplot(2,2,1); plot(X,Y1,'gd-'); title('sin(x)');
subplot(2,2,2); plot(X,Y2,'ro:'); title('cos(x)');
subplot(2,2,3); plot(X,Y3,'ch-.'); title('sin(2x)');
subplot(2,2,4); plot(X,Y4,'b<--'); title('cos(2x)');
```

Example: subplot

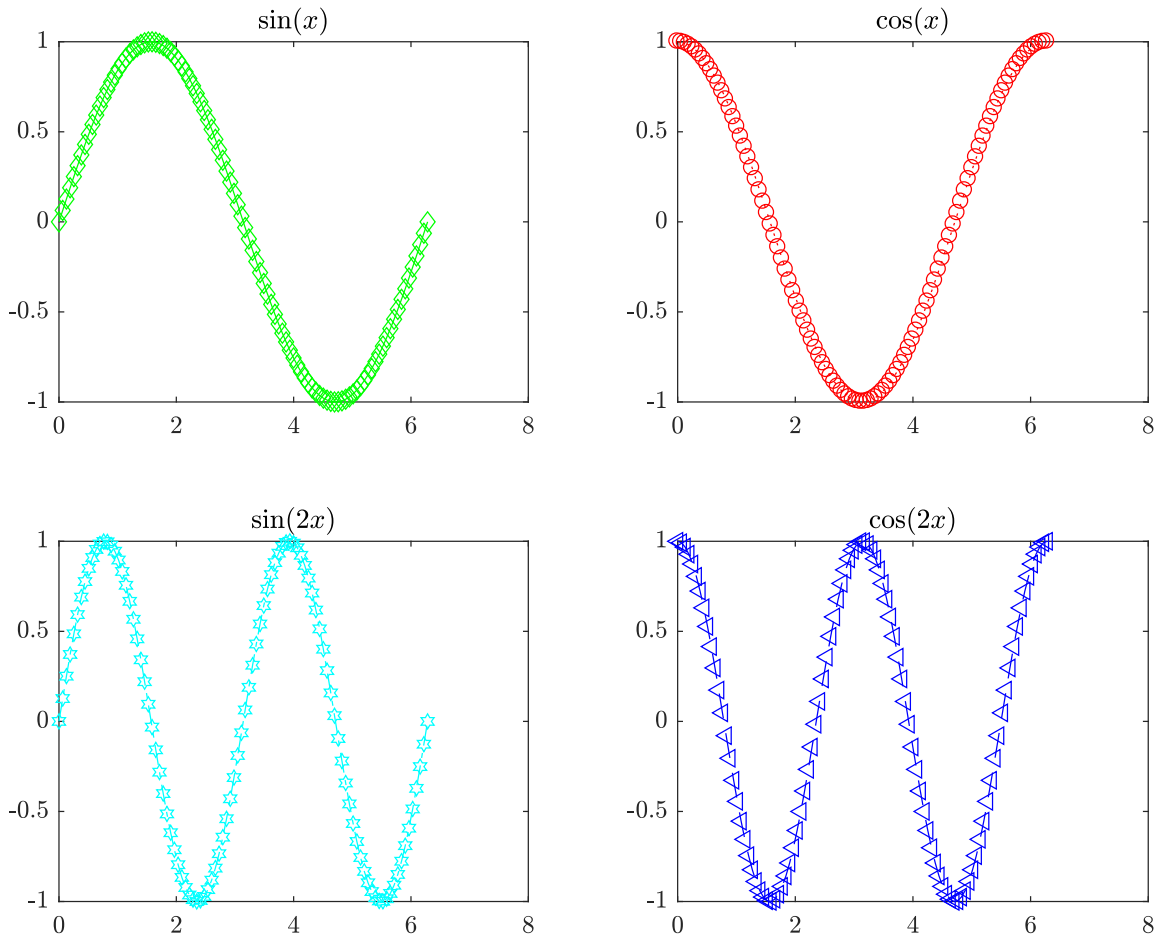

Figure 10:subplot

plotyy, semilogy, semilogx, loglog

- `plotyy`: Graphs with y tick labels on the left and right.
  - `plotyy(X1,Y1,X2,Y2,FUN1,FUN2)` uses `FUN1(X1,Y1)` to plot the data for the left axes and `FUN2(X2,Y2)` to plot the data for the right axes.
- `semilogy`: semilogy Semi-log scale plot, same as `plot`, except a logarithmic (base 10) scale is used for the Y-axis
- `semilogx`: semilogx Semi-log scale plot, same as `plot`, except a logarithmic (base 10) scale is used for the X-axis
- `loglog`: Log-log scale plot, same as `plot`, except logarithmic scales are used for both the X- and Y- axes.

Example: plotyy

```
% Example: plotyy
x = 0:0.1:10;
y1 = 200 * exp(-0.05 * x) .* sin(x);
y2 = 0.8 * exp(-0.5 * x) .* sin(10 * x);
figure(11)
[hAx, hLine1, hLine2] = plotyy(x,y1,x,y2,'plot','stem');
set(hLine1, 'LineStyle', '--');
set(hLine2, 'LineStyle', ':');
grid minor;
xlabel('Time ($\mu$s)')
ylabel(hAx(1), 'Slow Decay')
ylabel(hAx(2), 'Fast Decay')
title('Multiple Decay Rates')
```

Example: plotyy

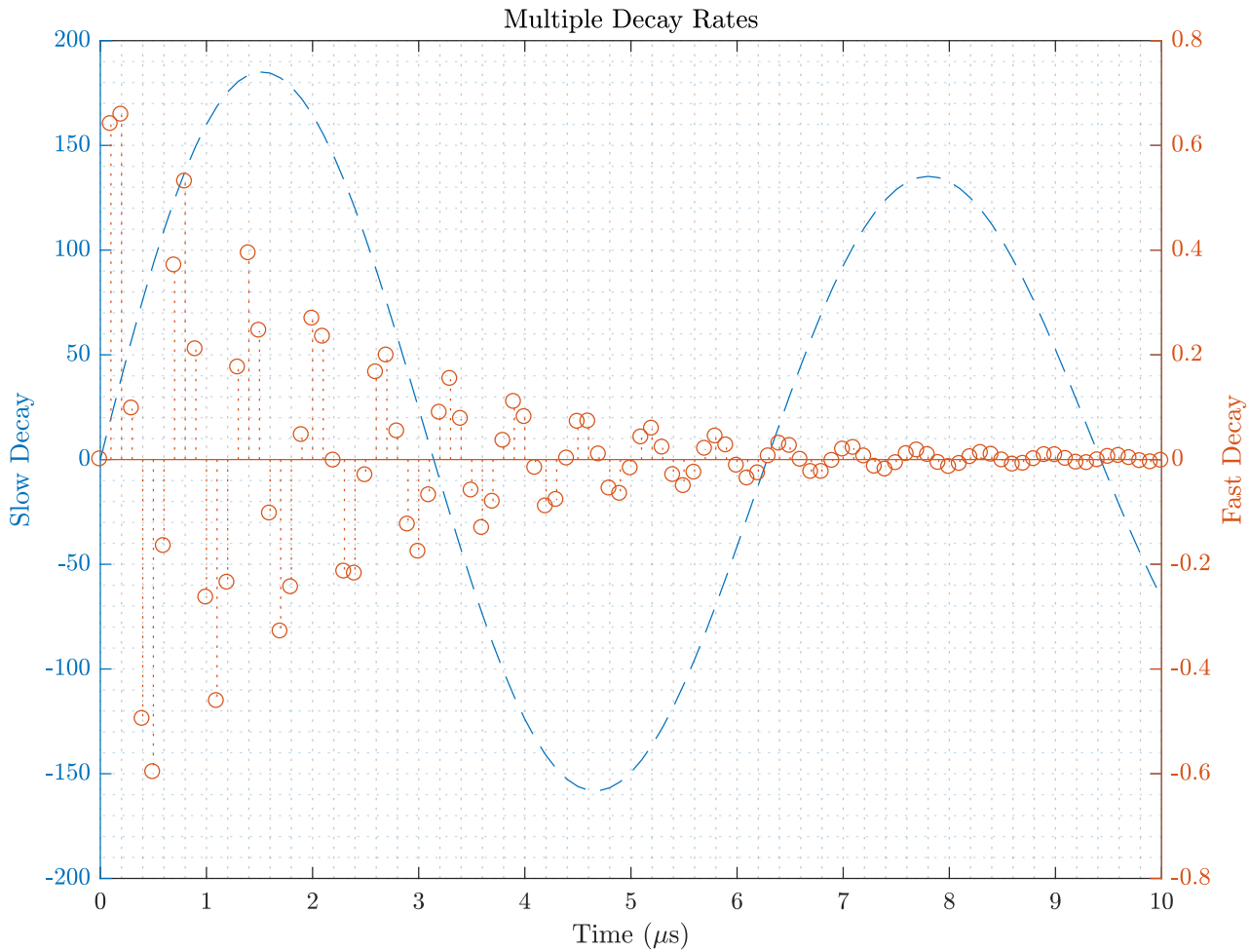

Figure 11:plotyy

Saving Figures

- `saveas`: Save Figure or Simulink block diagram in desired output format.
- `print`: Print or save a figure or model.
- `num2str`: Convert numbers to character representation.
- `strcat`: Concatenate text.
- `mkdir`: Make new directory

Example: print

```
% Example: print
mkdir figures
prefix = './figures/figure_';
for i = 1:11
    name = strcat(prefix, num2str(i));
    fig = figure(i);
    set(fig, 'PaperPositionMode', 'auto');
    pos = get(fig, 'PaperPosition');
    set(fig, 'PaperSize', [pos(3) pos(4)]);
    print(fig, '-dpdf', name);
end
```

Summary

- `figure`
- `hold`
- `plot`, `semilogy`, `plotyy`
- `subplot`
- `title`, `xlabel`, `ylabel`, `legend`, `axis`, `grid`
- `gcf`, `gca`, `get`, `set`
- `saveas`, `print`
- `strcat`, `num2str`