MATLAB II

Computational Physics

MATLAB
(you need to know)
Lecture 2
Outline

• Using MATLAB's Plot function
• File Input and Output
  • Output to Files
  • Input from Files using Import Data Wizard
• MATLAB Functions
Plotting with MATLAB

- `plot(x,y,'clm')`
  - 2D plot of one array against another
  - `x` is the array with `x` array
  - `y` is the array with `y` array
  - `clm` is string to give desired properties of the plot:
    - `c` = color symbols: `c m y r g b w k`
    - `l` = linestyle symbols:
      - `-` -- : -. omit symbol=no line
    - `m` = marker symbols to indicate style of points:
      - `+ o * x s d ^ v > < p h`
      - `none=no symbol`
Labelling the Plot

• xlabel('name of x axis') - prints a label along the x-axis
• ylabel('name of y axis') - prints a label along the y-axis
• title('title for plot') - writes a title across the top of the graph
• axis([xmin xmax ymin ymax]) - sets limits for plot with MATLAB vector shown
%define arrays
x = 0:10;
y = x.*x;

% make the plot with
% dotted blue line and
% squares for points
plot(x,y,'b:s');

% label the plot
xlabel('x');
ylabel('y');
title('Example');
Interactive Edit of Plot

- You may edit many plot features and save final result for your report.

- Features:
  - select feature to edit - dialog box appears
  - add text
  - draw lines and arrows
  - output final result (see File menu)
Exporting a Plot Image

1. Select 'Export' under File Menu in Figure Window.
2. Select 'JPEG image' as the file type.
3. Name your file (e.g., 'fall_with_function.jpg') and click 'Save'.
Multiple Lines/Multiple Figures

- **hold on** - maintains lines in current plot and allows additional plots to be added
- **hold off** - stops this; next call to plot will make a new figure
- **close** - closes the current figure window
- **figure(i)** - allows multiple figures to be prepared by same program. This command sets up the ith figure window for plotting.
Multiple Plot Example

% create arrays
x = 0:10;
y = x.*x;
z = 5*x;

% plot arrays
plot(x,y,'b:s');

% hold graph to add
% plot of z array
hold on
plot(x,z,'k');
hold off
Program Output

- MATLAB outputs variables and arrays by default in its own format.
  - `;` at end of line inhibits this.
- Use MATLAB function `fprintf` to write formatted output to screen.
- `fprintf` may be used with `fopen` and `fclose` to write an output file.
% define an array - ; means don't print
x = 0:1:10;

% print out x using MATLAB default
x

% now write x to file my_file.dat

% open the file my_file.dat for writing
fid = fopen('my_file.dat', 'w')

% now write each element of x
for i=1:11
    fprintf(fid, '%f \n', x(i)) % to file
    fprintf( '%f \n', x(i))   % to screen
end

% finally close the file
fclose(fid)

\textit{fopen} "opens" a file for reading or writing. It returns \textit{fid}, which is the file identifier used to reference the file throughout the program.

\textit{fprintf} is MATLAB function like C printf and fprintf.

Must "close" the file at end of script with \textit{fclose}.
About fprintf

- Format follows "C" standard
  `fprintf( file_id, 'format-string', variables)`

- Format String
  - Characters (note % does NOT mean comment)
    - `%f` - floating point number
    - `%d` - integer
  - Number format: let `l` be number of spaces and `p` be number of decimals then specify
    float format: `%l.f`
  - Special character for "new line": `\n`

- Example:
  `fprintf( '%5.2f', x)  % prints value of x as xx.xx`
MATLAB for General Plotting using *load* command

- MATLAB can be used to plot data saved in files.
  - Steps:
    - Use *load* to read array from file
    - Create vectors from columns
    - Plot vectors

```matlab
% my_data.dat contains
% array with 3 cols and % N rows ... e.g.
%    x1  y1  z1
%    x2  y2  z2
%    ... ... ...
%    xN  yN  zN

load my_data.dat

% vector from first col
x = my_data(:,1);
% vector from second col
y = my_data(:,2);

plot(x,y);
```
Importing data using the Import Data Wizard

- Data may be imported from files using GUI
  
  - Select Import Data...
  
  - Enter File Name
  
  - Select column separator
  
  - Approve import
MATLAB Functions

• Sometimes you find that same set of commands is repeated many times in your program.

• Can define a function in MATLAB to replace these lines with a single command

• Advantages
  • Simpler program - easier to debug
  • Function code only has to be checked in function - not several times within single program
  • Functions can be reused.
Function Example

% define time array
tm = 0:10;  % times

% define constants
g = -9.8;  % grav. acc.
v0 = 0.0;  % initial v
x0 = 1000.0;  % initial x

% calc. positions for times tm
x = falling_ball(tm, g, v0, x0);
y = falling_ball(tm, -3, v0, x0);

hold off
plot(tm,x);
hold
plot(tm,y,'r');

function z = falling_ball(t,g,v,x)
% falling ball equation
% t is time array
% g is grav. acc.
% v is initial velocity
% x is initial position

% z is vector with positions
for k=1:length(t)
z(k) = x + v*t(k) + 0.5*g*t(k)*t(k);
end

FILE: falling_ball.m
Code is executed when called by
fall_with_function script