Matlab Tutorial - II

Based on Matlab tutorial by IPLab@SUT

What we did last week

- Basic math
- Environment
- Matrix and array operations
- Logical operations
- Strings, cells and structures

What we'll do today

- Scripts
- Functions
- Control flow
- Debugging
- I/O
- Efficiency tips

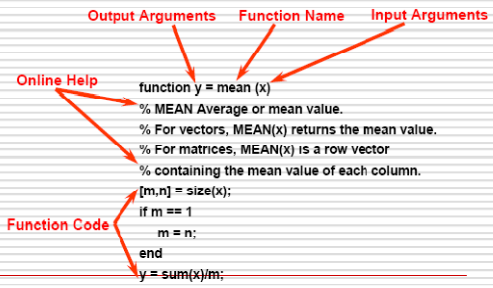
Programming/Debugging

Script and Function Files

- Script Files
  - Work as though you typed commands into MATLAB prompt
  - Variable are stored in MATLAB workspace
- Function Files
  - Let you make your own MATLAB Functions
  - All variables within a function are **local**
  - All information must be passed to functions as parameters
  - Subfunctions are supported

46

Basic Parts of a Function M-File



47

Flow Control Statements

if Statement

```
if ((attendance >= 0.90) & (grade_average >= 60))
   pass = 1;
end;
```

while Loops

```
eps = 1;
while (1+eps) > 1
    eps = eps/2;
end
eps = eps*2
```

48

Flow Control Statements for Loop

```
a = zeros(k,k)  % Preallocate matrix
for m = 1:k
    for n = 1:k
        a(m,n) = 1/(m+n -1);
    end
end
```

switch Statement

```
method = 'Bilinear';
switch lower(method)
   case {'linear','bilinear'}
      disp('Method is linear')
   case 'cubic'
      disp('Method is cubic')
otherwise
      disp('Unknown method.')
end
Method is linear
```

49

M-file Programming Features

- SubFunctions
- Varying number of input/output arguments
- ***Local*** and ***Global*** Variables
- Obtaining User Input
  - Prompting for Keyboard *Input*
  - Pausing During Execution
- Errors and Warnings
  - Displaying *error* and *warning* Messages
- Shell Escape Functions (*! Operator*)
- Optimizing MATLAB Code
  - Vectorizing loops
  - Preallocating Arrays

50

Function M-file

```
function r = ourrank(X,tol)
% rank of a matrix
s = svd(X);
if (nargin == 1)
  tol = max(size(X)) * s(1)* eps;
end
r = sum(s > tol);
```

Multiple Input Arguments use ( )

```
»r=ourrank(rand(5),.1);
```

Multiple Output Arguments, use [ ]

```
»[m std]=ourstat(1:9);
```

```
function [mean,stdev] = ourstat(x)
[m,n] = size(x);
if m == 1
  m = n;
end
mean = sum(x)/m;
stdev = sqrt(sum(x.^2)/m - mean.^2);
```

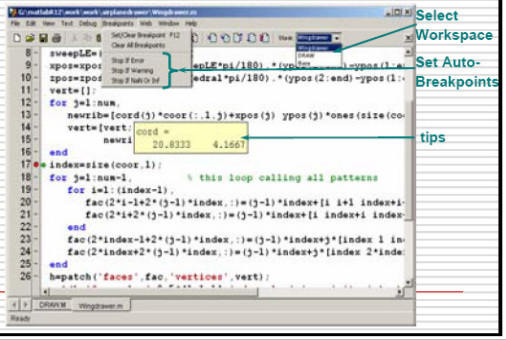
51

Editing and Debugging M-Files

- What is an M-File?
- The Editor/Debugger
- Search Path
- Debugging M-Files
  - Types of Errors (*Syntax Error* and *Runtime Error*)
  - Using ***keyboard*** and " **;** " statement
  - Setting Breakpoints
  - Stepping Through
    - Continue, Go Until Cursor, Step, Step In, Step Out
  - Examining Values
    - Selecting the Workspace
    - Viewing ***Datatips*** in the Editor/Debugger
    - Evaluating a Selection

43

Debugging



44

Input/Output

Importing and Exporting Data

- Using the Import Wizard
- Using ***Save*** and ***Load*** command

```
save fname
save fname x y z
save fname -ascii
save fname -mat
```

```
load fname
load fname x y z
load fname -ascii
load fname -mat
```

32

Input/Output for Text File

•Read formatted data, reusing the format string N times.

```
»[A1…An]=textread(filename,format,N)
```

•Import and Exporting **Numeric** Data with General ASCII delimited files

```
» M = dlmread(filename,delimiter,range)
```

33

Graphics Fundamentals

35

Graphics

- Basic Plotting
  plot, title, xlabel, grid, legend, hold, axis
- Editing Plots
  Property Editor
- Mesh and Surface Plots
  meshgrid, mesh, surf, colorbar, patch, hidden
- Handle Graphics

36

2-D Plotting

Syntax:

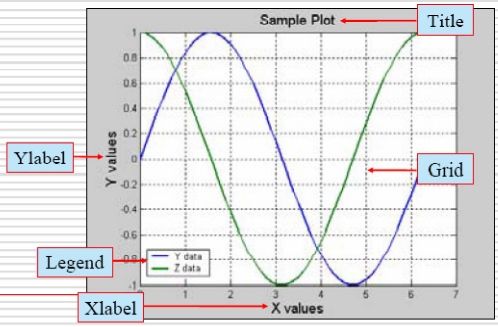
```
plot(x1, y1, 'clm1', x2, y2, 'clm2', ...)
```

Example:

```
x=[0:0.1:2*pi];
y=sin(x);
z=cos(x);
plot(x,y,x,z,'linewidth',2)
title('Sample Plot','fontsize',14);
xlabel('X values','fontsize',14);
ylabel('Y values','fontsize',14);
legend('Y data','Z data')
grid on
```

37

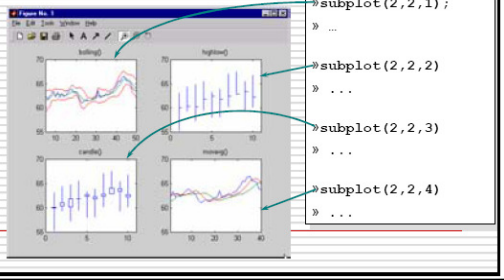
Sample Plot



38

Subplots

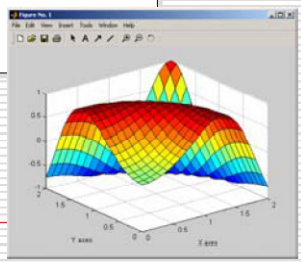
Syntax: `subplot(rows,cols,index)`



39

Surface Plot Example

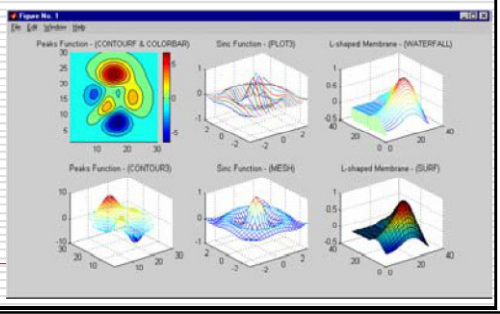
```
x = 0:0.1:2;
y = 0:0.1:2;
[xx, yy] = meshgrid(x,y);
zz=sin(xx.^2+yy.^2);
surf(xx,yy,zz)
xlabel('X axes')
ylabel('Y axes')
```



40

3-D Surface Plotting

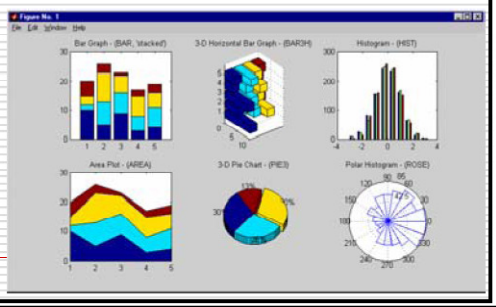
contourf-colorbar-plot3-waterfall-contour3-mesh-surf



41

Specialized Plotting Routines

bar-bar3h-hist-area-pie3-rose



42

Tips and tricks

Efficiency

- Whenever possible, use matlab inbuilt functions
  - sum, .*, repmat..
- Pre-allocate space
- Vectorize code
- Profile your code
  - profile on;<your_code>; profile report

See research.microsoft.com/~minka/software/matlab.html for more tricks