12.010 Computational Methods of Scientific Programming

Matlab Lecture 2
Lecturers
Thomas A Herring
Chris Hill
Summary of Introduction to Matlab

• Looked at the basic features of Matlab:
  – Getting help
  – Variable definitions and usage
  – Math operators
  – Control statements: Syntax is available through the online help
  – M-files: Script and function types
    • Variable number of input and output arguments
Today’s Lecture

• Continue examining Matlab operations
• path and addpath commands
• Variables and constants
• IO using fopen, scanf etc.
• Formats
• Dialog boxes
Multidimensional cells and structures

• Cell arrays are similar to multidimensional arrays except that the all the cells do not need to be same
  e.g., \( a\{1,1\} = \begin{bmatrix} 1 & 2 \\ 4 & 5 \end{bmatrix} \); \( a\{1,2\} = 'Name' \); \( a\{2,1\} = 2-4i \);

• Structure arrays also exist and are accessed and created similar to C (i.e., elements are referred to by . construction)
  \( \text{patient.name} = 'John Doe' \); \( \text{patient.age} = 32 \);

• These are recent features added to Matlab and can be useful in many applications but we will not discuss further.
Program Layout

- Matlab can be run interactively; with script M-files as we have been doing; and/or function M-files
- It is possible to execute C-compiled routines called MEX files (for speed) but we will not cover this (system dependent)
- PC Matlab supports Word Notebooks but not available on Unix or Mac.
- `helpwin` on all systems invokes the help system
- `tour` and `demo` give a tour and demo of Matlab
Function M-files

• Function M-files can have multiple inputs and outputs
• The generic construction is (in an M-file whose name is that of the function.m)
  
  ```matlab
  function y = flipud(x)
  % FLIPUD Flip a matrix up/down
  % Comments about function
  .. Actual code
  ```

• Name must begin with a letter
• First line is function declaration line
• First set of contiguous comment lines are for help
• First comment (H1 line) is searched with the `lookfor` command
Function M-files 02

- Usually name is capitalized in H1 line
- Functions can invoke M-file scripts (executed in function workspace)
- M-file can contain multiple functions that are sub-functions of main function in mfile
- Functions can have zero inputs and outputs
  - *nargin* tells number of arguments passed in call
  - *nargout* tells how many outputs given
- Normally input variables are not copied to function workspace but made readable. However, if there values are changed then they are copied
Function M-files 03

- Functions can accept variable and unlimited numbers of input variables by using `varargin` as the last argument.
- Functions can have variable numbers of outputs used `varargout`.
- Use the command `global` to have variables shared between base workspace and function workspace (must be declared `global` in both places).
- Matlab lets you reach another workspace with the `evalin` function.
- You can also use `assignin` to assign values in a workspace (not recommended).
Path controls

- Matlab uses a path structure to tell it where to look for M-files.
- In simple cases, all the m-file needed are in the directory from which Matlab runs but in more complex cases this is not possible.
- The path command lists the current path.
- The addpath command adds a new directory to the path (the current directory is always searched first).
- The pwd command can be used in the addpath command e.g., addpath(pwd).
- M-files can contain multiple functions but additional functions in M-file are available only to the main function of the M-file.
- In complex systems of analysis, where functions are put in M-files should be carefully considered.
Variables and constants

- In Matlab variables are passed into functions by address unless the values are changed, in which case they are copied into the function workspace.
- Although most variables are stored as double precision in Matlab, they can be referred to as different types e.g., complex, logical.
- To create non-double precision array, the data type can be specified in the ones, zeros functions e.g. IA=zeros(20,’int8’)
- `whos` shows the type of variable
- `all`, `any`, `find` implement logical expressions in array indexing. (See `ops` for more details)
- `logical` can be used to select elements from an array
IO: fopen, scanf, printf

- `fopen` opens a file and returns a file ID number (FID):
  
  Syntax is
  
  ```matlab
  [fid,message] = fopen('filename','permissions')
  ```

- If the open is not successful, `fid` returns as -1
- [http://geoweb.mit.edu/~tah/12.010/Matlab/Lec02_01_file.m](http://geoweb.mit.edu/~tah/12.010/Matlab/Lec02_01_file.m) gives a simple example of reading and plotting a data file. Data files used here are MIT GPS data processing. Example allows a number different features in Matlab to be explored.
- This M-file also shows the use of logical and plotting functions in Matlab.
FORMATS for scan and print

• The format structure in Matlab is very similar to C (and unix programs such as awk)
• Mostly these are used for outputting values
• Basic types (see details in Matlab On-line help)
  • %f, %e, %g — floating point numbers
  • %d — integer values
  • %s, %c — String and single characters
  • \n — newline (needed often at ends of format)
  • \r — carriage return
Dialog boxes

• We can make the File selection even better in the example using a dialog box.

• The Matlab M-file
  http://geoweb.mit.edu/~tah/12.010/Matlab/Lec02_02_db.m
  shows an example of how we might do this.

• This example shows ways to get file names from a directory listing.

• At this point we try these features on Athena

• In the next two lectures, you will develop a Matlab program to manipulate data of this type.
Summary of Today’s class

- Continued examining Matlab operations
- path and addpath commands
- Variables and constants
- IO using fopen, scanf etc.
- Formats
- Dialog boxes
- Much of the lecture is spent actually using these features in the M-files that are included with the lecture.