12.010 Computational Methods of Scientific Programming

Lecturers

Thomas A Herring
Chris Hill
Summary of Today’s class

• We will look at Matlab:
  – History
  – 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
• Our approach here will be to focus on some specific problems using Matlab for analysis and for building Graphical User Interfaces (GUI) and treating graphics as objects.
MATLAB (Matrix Laboratory)

• History
  – MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects.
  – First version was released 1984.
  – Current version is version 7 (Versions come in releases; currently Release 2007a, 7.4). (commnd ver gives version)
  – Interactive system whose basic data element is an array that does not require dimensioning
  – UNIX, PC and Mac versions. Similar but differences.
MATLAB:

- All commands are executable although there is the equivalent to dimensioning. In general arrays in MATLAB are not fixed dimensions.
- Syntax is flexible but there are specific set of separators.
- Basic Structure:
  - MATLAB commands are executed in the command window called the base workspace (>> prompt).
  - MATLAB code can be put in M-files: Two types
    - Script type which simply executes the code in the M-file.
    - Function type which executes codes in a new workspace. Generally variables in the new workspace are not available in the base workspace or other workspaces.
Getting help

• Matlab has extensive help available both locally based and through the web.
• After release 13 there is a help menu in the command window.
• Help falls into two types:
  – Help on specific commands and their usage
  – Help by topic area which is useful when looking for generic capabilities of Matlab
• Matlab also comes with guides and there are third-party books such as “Mastering Matlab 5”
Basic Structure 02

- Variable types
  - Early versions of matlab had variables that are double precision, strings, cells {}, or structures.
  - After Version 6, other variable types introduced specifically single precision and integer forms can be used (saves memory space) (help datatypes)
  - Complex variables are used as needed (use *i or *j to set complex part)
  - Variables can be defined locally in current workspace or they can be global.
  - To be global must be defined that way in both base workspace and M-files
  - who and whos are used determine current workspace variables
  - Names are case sensitive, no spaces, start with letter and may contain numbers and _
  - workspace command is GUI management tool (now built into Desktop Layout).
Basic Structure 03

- I/O: File I/O is similar to C
  - fopen, fclose, fread (binary), fwrite (binary), fscanf (formatted read), fprintf (format write), fgetl (read line), fgets (read line keep new line character), scanf (string read), sprintf (string write)
  - save and load workspace.
- Math symbols: + - * / \^ (\ is left divide)
- When matrices are used the symbols are applied to the matrices.
- When symbol preceded by . Array elements are operated on pair at a time.
- ‘ means transpose array or matrix
- Lec01_01 and Lec01_02 are examples
Basic Structure 04

– Control
  • if statement (various forms)
  • for statement (looping control, various forms (similar to do)
  • while statement (similar to do while)
  • No goto statement!
  • break exists from for and while loops
  • switch case otherwise end combination
  • try catch end combination

– Termination
  • end is used to end control statements above
  • return is used in functions in the same way as Fortran.
M-files: Script and Function types

– Communication with functions and M-files
  • Script M-files:
    – Do not accept input or output arguments
    – Operate on data in workspace
    – Useful for automating a series of steps
  • Function M-files
    – Accept input arguments and return outputs
    – Internal variables are local to the function by default, but can be declared global
    – Useful for extending language
Syntax

• Flexible layout with certain characters have specific uses.
• % is the comment symbol. Everything after % is ignored
• … (3 dots) is the line continuation symbol. Must be used at a natural break in commands
• , used to separate commands, with result printed
• ; used to separate commands with result not printed
• [ ] enclose arrays and matrices, {} enclose sets (difference is multi-dimensional arrays need to be all of the same type and size)
• : is the range selector for from start:increment:end, if only one : increment is 1, if no numeric values, range for matrix elements.
Multidimensional arrays

• Matlab works naturally with 1 and 2 dimensional arrays but more than 2 dimensions can be used.
• They can be constructed a number of different ways
  – By extension: \[ a = [5 7 8 ; 0 1 9 ; 4 3 6]; \]
    \[ a(:, :, 2) = [1 0 4 ; 3 5 6; 9 8 7] \]
  – Scalar extension (Set “plane” 3 to 5)
    \[ a(:, :, 3) = 5 \]
  – Use of functions \texttt{ones}, \texttt{zeros}, \texttt{randn}
    \[ b = \texttt{zeros}(3, 3, 2) \]
  – \texttt{cat} function, \texttt{cat(ndim, arrays, … )} where \texttt{ndim} is the dimension to be concatenated in.
Multidimensional arrays 02

• `reshape` function allows redefinition of array shape e.g.,
  \( a = [1:18]; \text{reshape}(a, [3 3 2]) \)
• `squeeze` removes dimensions that are only 1 element
• `permute` allows array dimensions to be re-ordered.
• Functions that operate on elements of arrays work with multidimensional arrays but matrix type functions do not work unless a suitable 2-D array is passed
• Functions that operate on vectors use the first non-singleton index
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} = [ 1 2 ; 4 5]; 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 patient.name = ‘John Doe’; 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)
  
  ```plaintext
  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).
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
• Class Project Descriptions and groups (2-3 people) due Tuesday November 13.