School of Mathematics and Statistics
MAST30028 Numerical Methods & Scientific Computing
Week 1

1 Starting MATLAB

Start up the application (double click on the Toolbar at the bottom of the screen or use the Applications Shortcuts).

The Desktop opens with a default layout of various windows, comprising the Development Environment. The default layout includes:

- the Command Window. You enter commands to MATLAB at the command prompt `>>`. You can use tab-completion, and use the up-arrows. Drag-n-drop also works. Can use as a 'calculator'
- the Current Directory (where you see files that you create or run)
- the Workspace (where you see the currently defined variables and their values) and Variable Editor
- and the Editor (if you are editing a file)

You can customize many parts of the Desktop to suit your own style of working e.g. undock the editor.

The only place you can save to is your Users area of the C: drive (Windows) and the default Current Directory is `C:\ ... \MATLAB`. You can save your work there but this can be wiped clean overnight, so it's a a good idea to save your work.

There are several options:

- save to a USB drive.
- save to your University Webmail account. Use a browser to connect to your student email account and save your work as Attachments. Limited to 20Mb but this is plenty for source code, text files etc.

You will be running some files that I provide, so you should first add these files/folders to your Current Directory (in case you want to edit them) and then add them to the Search Path so that MATLAB can find them. The easiest way to do this is to do `Home -> Set Path`, highlight the Current Directory, then press `Add with Subfolders` (this includes all files/subfolders in the path).

The files you will need are kept on the Lab Server (Lab-Materials L:).

Drag and drop the folder `Week1` from `L: \MAST30028` to `C:\...\MATLAB` and include it in the path, as described above. Now MATLAB knows how to find the files in Week1.

You may have to do this at the start of every lab.

2 Getting Help

Here are 3 useful ways to find help: help, the function browser, and Search in the doc/Help system.

a. help: gives brief help from the command line

   e.g. `help atan`

   Brief description of command, then syntax /options, then examples of usage

   `help` also works on directories of related commands

   e.g. `help elfun`

b. At the command prompt, the symbol f_x denotes the function browser. Click on this to browse for a function, arranged in a tree-like structure of directories, the same tree structure as in the doc/Help system.

c. most importantly, the powerful Help Browser: either hit the F1 key, or the `Home->Help` menu or `Home->?`, or...

`doc`

Once in the help system, the Home symbol brings up documentation for all Mathworks products that the University owns a site license for. We'll only be using MATLAB.

The Documentation is organized in a tree structure, with Language Fundamentals, Mathematics, Graphics etc. as the top branches, then split into finer branches etc.

On any level you can Search Documentation in the top bar and refine by Product or category.

Note that the Help system is available to you during the lab exam, so get familiar with it!

For beginners, I suggest the Getting Started tab (just under MATLAB) which includes a number of tutorials for beginners. You should look at all of these in weeks 1-2.

Finally, the Examples tab has video or M-file demonstrations for many topics. I am setting some of these (esp. the Getting Started videos) as pre-reading.

The internet has about 10,000 MATLAB tutorials. See MATLAB resources on the LMS for links to some of them. Cleve Moler (founder of MATLAB) has a book online for free. See the LMS.

3 MATLAB as a calculator

Finish Part 1 of the `lab1.m` demo. At this stage, you should know the commands:

```
+ - * / ^ = ;
clc clear 1i exp sin cos atan abs log
disp help doc who whos save load
```


Exercise Set 1

a. Compute the following on the command line: $2/3, 2^3, 2^{1/3}, \log_2 3, \cos(3)\sin(2), \sin(2+3i), \sqrt{-i}$

b. What is wrong with the following MATLAB commands?


```
x=2; y =sin(2x) - tan x
x=2; y =e^x
2and1=3
```


4 Scripts and functions; program flow control

As soon as you need to string more than a few MATLAB commands together, it's time to stop working at the command line and collect these commands together in a program. In MATLAB, programs are stored in *M-files*, so called because they have the extension `.m`.

There are 2 types of M-file:

a. scripts, which are just collections of MATLAB commands (like a shell script)

b. functions, which have (optional) input and output arguments. These act like procedures, functions or subroutines in other languages.

Finish Part 2 of the `lab1.m` demo. At this stage, you should know the commands:

```
function  %  sqrt log2
for end :  if else elseif
> >= < <= == ~= && || ~
true false isreal while
```

Exercise Set 2

a. Modify `newtonFuncForLoop` so it returns 2 outputs: `root5` and the *residual* `root5^2-5` i.e. how much we failed in attempting to find a number x so that $x^2 - 5 = 0$. Give your variables meaningful names; update your help file. Test your new function `newtonForTwoOutputs`.

b. Modify `newtonFuncForLoop` so it uses 2 inputs: x_0 and the number a we are trying to find the square root of. Newton's algorithm for the square root of a is

$$x_{n+1} = \frac{x_n}{2} + \frac{a}{2x_n}$$

Update your help file; test your new function `newtonForTwoInputs`.

c. Modify `newtonFuncForLoop` to accept as an input the number of iterations. What would be a good condition on the input to check?

d. Modify `newtonFuncForLoop` to run until $|x_n^2 - 5| \leq 10^{-10}$. Try to output the number of iterations it took. This will require a different kind of loop.

Hint: In MATLAB 123.456×10^{78} can be entered as `123.456e78`.

e. Use the debugger to find out what is wrong with the code `FibBad` in `Week1`. It is supposed to compute the nth term of the Fibonacci sequence, defined by

$$x_n = x_{n-1} + x_{n-2} \qquad x_1 = 1, x_0 = 0$$

Notice that many (not all) `for` loops can (and should) be avoided by using the vectorization properties of MATLAB functions (which we'll come to next).

5 1D arrays (vectors) in MATLAB

Finish Part 3 of the `lab1.m` demo. At this stage, you should know the commands:


```
zeros ones linspace rand randn
.* ./ .^
length sum prod cumsum max min
sort mean var
```


Exercise Set 3

a. Look up the MATLAB function `diff`. Write your own version, only for the case of a vector input.

b. Represent the function $f(x) = x^2 e^{-x^2}, |x| \leq 2$ as an array. Plot this function.

c. Use your function `newtonFuncWhileLoop` from Exercise 2d. Run it on a range of initial guesses in $(0, 2.5)$. Plot the number of iterations taken as a function of the initial guess.

Here I mean you to explicitly loop though a range of initial guesses, not vectorize `newtonFuncWhileLoop`.

6 Output in MATLAB

Try these after Lecture 2, on output in MATLAB. At this stage, you should know the commands:

`disp format fprintf sprintf`

Exercise Set 4

a. Predict what the following output commands will produce, then check.

(a) `fprintf('%2d',123)`

(b) `fprintf('%2d\n',123)`

(c) `fprintf('%8d\n',123)`

(d) `fprintf('%8d\n',123.456)`

(e) `fprintf('%8f\n',123.456)`

(f) `fprintf('%10.2f\n',123.456)`

(g) `fprintf('%10.8e\n',123.456)`

(h) `fprintf('%8d\n',1:4)`

b. Modify your `newtonFuncForLoop` to print a table of the iterates . Use `fprintf`, once before the loop to generate the column headings and once inside the loop to generate each row of the table.

The output should look like:

```
 n     iterate    |xn-x_nm1|      residual
1     2.25000000   2.500000e-01   6.250000e-02
2     2.23611111   1.388889e-02   1.929012e-04
3     2.23606798   4.313320e-05   1.860471e-09
4     2.23606798   4.160139e-10   8.881784e-16
```


More exercises

a. Design a function that computes $n!$ where n is an integer. You may use either recursion (if you know what that is) or iteration (a loop).

Hint: use a recurrence relation between $n!$ and $(n-1)!$

b. Given a set of data $\{x_i\}$, the following statistics are commonly used.

i. the sample mean

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

ii. the sample variance

$$\mathrm{Var}(x) = \frac{1}{n-1}\sum_{i=1}^{n} (x_i - \bar{x})^2$$

iii. the kth trimmed mean

$$\bar{x}_k = \frac{x_{(k+1)} + x_{(k+2)} \cdots + x_{(n-k-1)} + x_{(n-k)}}{n-2k}$$

where $x_{(1)} \le x_{(2)} \le \cdots \le x_{(n)}$ are the data points after ordering. It is the mean after discarding the k smallest and k largest values.

Write a MATLAB function that takes a vector of data and an integer k as input and returns the sample mean, sample variance and trimmed mean as outputs. You may use the MATLAB functions `length, sort, sum`.

Write a driver function to test your function, and output the results using `fprintf`.