A Very Brief Intro to MATLAB [1]

(Keep as a reference)

A few general principles

- Type commands at the prompt and press Enter.
- Unless declared otherwise, variables are row vectors (1 by n arrays).
- The command `syms x` declares `x` to be a symbolic variable.
- MATLAB is case sensitive, i.e. `X` $\not\equiv$ `x`.
- The command `clear` will clear all variables. Always clear before starting a new computation. The command `clear` will not clear the screen.
- Ending a command with a semicolon "`;`" suppresses the output.
- Enter all commands exactly as given in the assignments.
- `ans` indicates the output from the preceeding command. Two useful commands are `pretty(ans)` and `simple(ans)`.
- MATLAB does both symbolic and numerical calculations.
- When you make a mistake, you do not have to retype the whole command. Use $\uparrow$ and $\downarrow$ to return to a line, correct the errors and re-press Enter. (Sometimes you also need to `clear`.)
- Access `Help` by clicking Help $\rightarrow$ Matlab Help, or by typing `helpdesk` or `helpwin`.
- Text may be added to your work after the symbol `%`.
- Save, print and exit by clicking the File icon.
- Many advanced procedures may be accomplished using Toolboxes. See the reference manual for details.
- MATLAB may be used as a programming language.
- There is an O.U. MATLAB web page at: `www.math.ohiou.edu/~matlab`, which contains assignments, reference materials and a sample solution to an actual assignment.
- For a more details on MATLAB and how to use it we suggest: *A Guide to MatLab for Beginners and Experienced Users*, by B. Hunt, R. Lipsman, and J. Rosenberg, Cambridge Univeristy Press, New York, 2001.

Some basic commands using a symbolic variable (try them)**.**

- `syms x` ..... This makes a symbolic variable.
- `f = x*sin(x)` .....This makes `f` a symbolic function.
- `f1 = diff(f)` ..... `f1` - is the derivative of `f`.
- `f2 = diff(f,2)` ..... `f2` - This is the second derivative of `f`.
- `F = int(f)` ..... `F` - This is the antiderivative of `f`.
- `int(f,0,pi)` .....This is a definite integral.
- `limit(log(cos(x))/x^2,0)` ..... MATLAB uses L'Hopital's rule to find limits.
- `limit(log(x)^2/x,inf)` ..... Also for ∞ / ∞.
- `ezplot(f)` .....Plot a graph using the default interval.
- `ezplot(f,0,4*pi)` ..... Plot a graph for specified interval.
- `polyn = x^5 - x^4 - 7*x^3 + x^2 + 6*x`
  `factor(polyn)`
  `solve(polyn)` ..... This solves the equation "`polyn = 0`"
- `expr = cos(x)^5 + sin(x)^4 + 2*cos(x)^2 - 2*sin(x)^2 - cos(2*x)`
  `simple(expr)`
- `ode = 'Dx = -a*x'`
  `dsolve(ode,'x(0)=3')` .....This solves the initial value problem at x(0)=3.

Some basic commands using arrays.

- `x = -2:.1:2;` ..... Makes `t` a vector with entries from `-2` to `2` in `.1` increments.
  `f = inline('x.^3 - 2*x','x')` ..... Defines a function $f(x) = x^3 - 2x$.
  `y = f(x)` .....This evaluates `f(x)` for each entry of `x`.
  `plot(x,y)` ..... This plots the pairs of points `(x(k), y(k))` for `k = 1, 2,...`
- `x = -2:.05:2; y = x;`
  `Z = sin(x'*y); mesh(Z)` ..... ' means transpose.
  A figure window will appear with a graph. Click on Tools and select Rotate 3D. Point the cursor at the graph and "click and drag" to rotate the graph.
- `A = [1 2 3; 4 5 6; 7 8 10], C = [1 2; 3 4; 5 6]`
  `A*C` ..... multiplies the matrices.
  `b = [1 2 3]', A\b` .....solves `Ax = b` by Gaussian elimination.