Computing for Math and Stats

Lecture 13.

Remember, Remember the 5th of November

- Matlab also has anonymous functions
- Nothing to do with the Anonymous group
- These are simple functions defined on the fly
- Usually for one time use or for passing as arguments
- Handy little tool

Defining Anonymous functions

- They are introduced with the @ (at sign)
  - @(x) sin(x)/x
  - @(x, y) sin(x)+sin(y)
- These are not matrices
- These are handles to functions
- They contain all is needed to run the function
- They can be assigned to variables like any other quantity

Function handles

- All functions have function handles
  - Try @sin
- This is different from the function call
  - Typing sin means invoke sin with no arguments and return the result
  - Typing @sin means return the handle to the function
- Any function that accepts functions as arguments accepts handles

Function name is not handle

- The function name is just a string
- Matlab can evaluate the function that corresponds to the string using feval
  - feval('sin',pi/2)
- The preferred method is handles
  - feval(@sin,pi/2)
  - Newest version of Matlab lists the string version as about to be retired.
- See powplot.m and anonplay.m

Global vs Local Variables

- Any variable used inside a function is a local variable by default
  - In other words, accessible only inside the function
  - If two functions use two variables with the same name, each one uses their own copy
- If we want a variable inside a function to be accessible from the command window and scripts, we declare it global
- The jargon is “the scope of variable XXX is”

Variables inside Anonymous Functions

- Variables inside anonymous functions are evaluated at the time of function creation
  - This means that subsequent changes to the variable will have no effect
- Rule: Anonymous functions should be used for simple things only (unless you really know the internals of Matlab)

Things to try at home

- Using fplot, plot a polynomial. The polynomial should be defined as an anonymous function. The anonymous function should be the first argument to fplot.
- Create a function named mydot that accepts as arguments two vectors. It returns their dot product.
- Create a function named matbyvec that accepts a matrix and a (column) vector as arguments. It returns the product of the matrix times the vector. Use a double for loop rather than use the built in matrix-vector multiplication.
- Create a function named mymmult that accepts two matrices as arguments and returns their product. Use a triple for loop rather than use the built-in matrix-matrix multiplication.