Numerical Methods Fall 2006

Excercise #1 – working with Matlab

The first two exercise sessions are not graded. See those exercises as brain twisters which make you comfortable with Matlab. Do as much as you can. A rather complete tutorial can be found at www.ph.tn.tudelft.nl/~csp/matlab/matlablong.pdf or on Blackboard. A shorter one can be found at http://www.mines.utah.edu/gg_computer_seminar/matlab/matlab.html

1. Preparations
   1.1. Start Matlab using 'matlab' or 'matlab –nojvm'.
   1.2. Change the working directory 'NMDA'. If it does not exist yet, create it.
   1.3. Create a subdirectory 'Ex1' and change into it. Create a new directory for every exercise.

2. Variable assignment and vectors
   2.1. Create a vector (1,2,3,6,5,7) and assign it to the variable v.
   2.2. Change the 4^{th} and 5^{th} element of v to 5 and 6, respectively. Can you do it with only one command?
   2.3. Create the vector 'seven' which holds all the multiples of 7 between 1 and 1000. How many elements does this vector have? (Tip: look up the function "length")
   2.4. Extract from the vector 'seven' all the even multiples of 7. (Tip: you find them at the positions 1,3,5… in the vector. You can address the last element of a vector with 'end')
   2.5. Assign two vectors (5,0,0) and (2,2,0) to the variables v1 and v2. Use the well-known formula $\cos(\alpha)=\mathbf{v1}*\mathbf{v2}/(|\mathbf{v1}|*|\mathbf{v2}|)$ to calculate the angle α between those vectors (Tip: the function 'norm' gives the Euclidean norm of a vector. Think about your result.

3. Function m-files
   3.1. Open an editor, e.g. nedit or the editor of the Matlab Java environment.
   3.2. Create a file alpha.m. Define in it the function alpha from 2.5 using 'function … end'. Think about the arguments and the return value. Remember to put a comment in the line after the 'function', which is shown when you type 'help alpha'. Also remember to put the file in the current working directory or in the load path. Make the function give back the angle in degrees.

4. Sieve of Eratosthenes: The sieve of Eratosthenes is an ancient method to find prime numbers. It works as follows:
   - set N to the square root of the highest number you want to are interested in.
   - make a vector of the numbers 2 to N^2.
   - Keep the first number in the list as a prime number and remove all multiples of this number.
   Repeat this step until you reach the number N.
   Comments:
   There are several ways top implement the last step. You could make a list of prime numbers to which you add a newly found number, or you could just 'cross out' non-prime numbers from your original list.
   The list of steps before is an example of how to tackle a programming problem: First translate your problem into a list of steps which can be carried out by the computer. Then think about how to implement them in your chosen programming language.