Math 4353 – Numerical Linear Algebra Spring 2020

Program 1 Due: Friday, February 7, 2020

1. Write a function subroutine, macheps.m to compute the machine epsilon to within a factor of 2. Remember, the machine epsilon, μ, is the smallest positive float such that $\mathrm{fl}(1+\mu) > 1$ (twice the unit round-off). macheps.m should have as its first line:

   ```
   function eps = macheps
   ```

2. Write an m-file, matmult.m, which computes the product $C = AB$ with $A \in \mathbb{R}^{m\text{x}n}$, $B \in \mathbb{R}^{n\text{x}p}$. The routine should take as input the arrays A and B. Output should be $C \in \mathbb{R}^{m\text{x}p}$. The first line should be:

   ```
   function C = matmult(A,B)
   ```

3. Both email me and print out and hand in the m-files macheps.m and matmult.m, and the diary file prog1run that is generated by the file NLAProg1Test.m (I will email to you).

   Notes:

   (a) My code for macheps.m should be attached with this email. In this unique case, you may use this code verbatim if you like, but with the implied agreement that you understand the purpose of each statement. Of course, you are encouraged to write your own code if you like.

   (b) For matmult you are limited to using the Matlab * operator *only on scalars* (numbers, not matrices or vectors), and you are further prohibited from using Matlab functions (dot, mmult, etc.) for matrix or vector operations. If, e.g., you want to use a dot product routine, then write your own and include it in your submission.

   (c) Since it is very simple and fast to check if the number of columns in A is equal to the number of rows in B (>> help size), your code should test for this possible input mistake and if so, give an error message (>> help error).

   (d) Always feel encouraged to play with your code. Try to break it; push its limitations (that is what my test routines will often try to do).

   (e) Remember to document your code. This means using comment lines (i) *in the help block* to describe unambiguously all input and output variables, and (ii) *in situ* to describe (when it is not obvious) what your code is doing at that step. I've attached a pdf which describes the basic function routine skeleton that I hope you will abide.