CS 322: Problem Set 3

Due: Thursday, July 11, 2002 (In Lecture)

The policies for this homework assignment are as follows:

- You may work with at most one other person on all of the questions. Consult the course website for the Academic Integrity rules.
- Problem sets will be weighted equally in determining your final grade. The number of questions or total number of points on a given problem set is irrelevant.
- Writing will be graded on content, as well as on grammar, spelling, punctuation, etc.
- Mathematical exercises will be graded on the overall set-up of the problem as well as correctness.
- Points will be deducted on the Matlab questions for poorly commented code and inefficient code/redundant computations.
- For the Matlab questions, you should hand-in the appropriate plot(s) and the script/function(s) necessary to generate the plot(s).
- Each team of up to two students should submit the assignment jointly. Submit the assignment as follows. At the top of the first page, write your name, the course number, the problem set number, your e-mail address, your student ID number, and the date. This information should be repeated for the second partner (if applicable).

1. (10 points) **Fun with Floating Point Numbers**

   (a) Suppose two points (x_0, y_0) and (x_1, y_1) are on a straight line with $y_1 \neq y_0$. Two formulas are available to find the x-intercept of the line:

   $$x = \frac{x_0 y_1 - x_1 y_0}{y_1 - y_0}$$

   and

   $$x = x_0 - \frac{(x_1 - x_0)y_0}{y_1 - y_0}.$$

   Show that both formulas are algebraically correct. Use the data $(x_0, y_0) = (1.31, 3.24)$ and $(x_1, y_1) = (1.93, 4.76)$ and three-digit rounding arithmetic to compute the x-intercept both ways. Which method is better and why?

   (b) What is the nearest floating point number to 125 on a base-5 computer with 6-bit mantissas? (Note: The nearest floating point number can be to the left or the right of 125.) Show your work.

   (c) If $A = \begin{pmatrix} 10^{-16} & 0 \\ 0 & 1 \end{pmatrix}$, why is it bad to solve the linear system $Ax = b$ for some b of size 2×1 using Gaussian Elimination with pivoting? Assume that the computer has machine precision equal to 10^{-16}. Show your work and explain your answer. Hint: Material found in Section 6.4 of your textbook will prove helpful on this part of the question.

2. **(10 points) Matrix Computations**

   (a) A matrix M is tridiagonal if $m_{ij} = 0$ whenever $|i-j| > 1$. Write a Matlab function `C = Prod(A,B)` that computes the product of an $n \times n$ upper triangular matrix A and an $n \times n$ tridiagonal matrix B. Your solution should be efficient (no superfluous floating point arithmetic) and vectorized. Write a script that generates an upper triangular matrix A of size 10×10 and a tridiagonal matrix B of size 10×10 and computes $C = AB$ using `Prod`. Any matrices A and B that fit the above specification will do. Demonstrate that your script and function work as intended by running the diary in Matlab. To do this, type "diary filename" at the Matlab prompt. Then run your script. After the script has finished running, you should type "diary off." Turn in the diary file in addition to the script and function files. Note: You will want to leave off the semicolons in your script so that the output will appear.

(b) Do problem P5.2.13 on page 187 of your textbook.

3. (10 points) Which Algorithm is More Efficient?

As we have seen in lecture, there are many ways to solve $n \times n$ linear systems of equations. This problem investigates two different ways of solving $Ax = b$ for the case of symmetric matrices, i.e., for matrices A with the property that $A = A^T$. Your task will be to determine which method below is more efficient in solving linear systems of the above type.

The first method is based upon the LU factorization of a matrix. This method is as follows:

Method 1: Solving $Ax = b$ using an LU Factorization

(a) Factor A into $A = LU$, where L is unit lower triangular and U is upper triangular. This is also known as Gaussian Elimination.

(b) Solve $Ly = b$ (forward substitution).

(c) Solve $Ux = y$ (backward substitution).

The second method is based upon the LDL^T factorization of a symmetric matrix. This method is as follows:

Method 2: Using an LDL^T Factorization

(a) Factor A into $A = LDL^T$, where L is unit lower triangular and D is diagonal.

(b) Solve $Ly = b$ (forward substitution).

(c) Solve $Dz = y$.

(d) Solve $L^T x = z$ (backward substitution).

Note that the LDL^T factorization of a symmetric matrix A exists whenever the LU factorization of the matrix exists.

Determine the number of flops (accurate to the leading term) for each of the methods above. Which method is more efficient for the solution of linear systems involving symmetric matrices? Show all of your work; it is not enough to simply state the number of flops for each step from the references.

In order to determine the number of flops required for the LU factorization, consult the code for GE.m on page 225 of your textbook. Similarly, consult the codes for LTriSol.m on page 211 and UTriSol.m on page 212 in order to determine the number of flops for forward and backward substitution, respectively. In order to determine the number of flops necessary to compute the LDL^T factorization of A, consult Algorithm 4.1.2 on page 139 of the handout. Finally, you will have to determine the number of flops to solve $Dz = y$ without referring to any code.