Homework 1 – Math 104B, Summer 2011

Due on Thursday, August 11th, 2011

Section 6.1: 6a and 6d.

Section 6.2: 8b and 8c.

Section 6.3: 8a, 8e, 8f.

Programming problems:

1. Write a program that implements the LU factorization (without permutations) seen in class, $A = LU$. The input should be a real square matrix A and the outputs should be three:

   (a) A flag variable: $f = 1$, if the matrix has LU factorization, $f = 0$, if it doesn't.

   (b) L a lower triangular matrix with ones in the diagonal.

   (c) U an upper triangular matrix.

   If $f = 0$, the contents of the output variables L and U do not matter.

2. Write a program that implements either backward o forward substitution to solve the system $Ax = b$ when A is invertible and lower or upper triangular. That is, the input should be two:

   (a) The independent term vector b.

   (b) The coefficient matrix A, either lower or upper triangular, in both cases invertible.

   (c) A flag variable: $f = 1$ if the matrix is upper triangular, $f = 0$ if it is lower triangular.

   The output should be the solution x.

3. Implement the algorithm for Gaussian Elimination seen in class. Write a subroutine that solves a linear system $Ax = b$ when A is square. The inputs should be two:

   (a) The coefficient matrix A.

(b) The independent term vector b.

The outputs should be three:

(a) A flag variable: $f = 1$ if the matrix is invertible, that is if the solution is unique, $f = 0$ if it is not. If $f = 0$ the contents of the other two variables do not matter.

(b) The solution x.

(c) A matrix whose upper triangular part, including the diagonal, is the modified upper triangular matrix, and the lower triangular part, not including the diagonal, has the multipliers.

4. In order to test the last program, consider the $n \times n$ matrix with entries

$$A_{i,j} = \begin{cases} 1 & \text{if } i = j \\ \frac{1}{(i+j)^2} & \text{otherwise} \end{cases}$$

For $n = 10, 20, 30, \ldots, 90, 100$, pick the right hand side b so that the solution to $Ax = b$ is the vector $x = [1, 2, \ldots, n]^T$ (do this in your program, before calling your subroutine). Then solve the system of equations for the ten values of n and compute the relative error in the computed solution $\widehat{x}$:

$$e = \frac{\|\widehat{x} - x\|_2}{\|x\|_2}.$$

where

$$\|x\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}.$$

Notes: The output should be a table with the values of the relative error e for each value of n. Do not print the solution or the matrix.