

## Homework #3 Due: Monday October 27

MTHBD/CMPBD 423

**Be sure to read the Homework Rules.** If you need a copy let me know. I will take off points according to those rules.

1. Solving  $Ax = b$ . You must write 3 MATLAB function files: (20 pts)

The first line of each function file is given below.

- (a) function  $x = \text{Usol}(U,b)$

This function takes an upper  $N \times N$  triangular matrix  $U$ , and an  $N$ -vector  $b$ . It returns  $x$  such that  $Ux = b$ .

- (b) function  $x = \text{Lsol}(L,b)$

This function takes a lower  $N \times N$  triangular matrix  $L$ , and an  $N$ -vector  $b$ . It returns  $x$  such that  $Lx = b$ .

- (c) function  $[L,U] = \text{myLUfact}(A)$

This function takes in a matrix ( $A$ ) and returns a lower unit triangular matrix  $L$  and upper triangular matrix  $U$  such that  $LU = A$ . The matrix  $A$  must be nonsingular and require no pivots in the factoring process. The matrix  $I$  give you will satisfy these conditions.

Download the program file `SolveThis.m` found on the class website. It defines a matrix and a  $b$ -vector. It then calls on the function files you wrote to factor the matrix and solve the equation  $Ax = b$ . It also uses MATLAB's forward slash function to solve the problem. It returns some stuff as well as the residual from both methods. The norm of your residual should be smaller than the norm of the residual using MATLAB's forward slash function.

**Note** Make sure that when you run `SolveThis.m` the only things printed to the command window are those within the `disp( )` commands. This may mean going back and putting in semi-colons to suppress output in the function files you wrote.

**What to hand in:** After you have successfully run `SolveThis.m`, clear the command window (from the edit menu), run it again, print the command window. You will return this as well as the three function files on separate sheets of paper. Total 4 pages with the successful run of `SolveThis.m` as the first page.

2. Section 4.4 numbers 14 and 26. This should take you two pages at most. If it takes you two pages, put one problem on each page. (15 pts)

3. The sensitivity of the Hilbert matrix.

(15 pts)

The Hilbert matrix is a notoriously ill-conditioned matrix. It arises in finding a polynomial of best fit in the least squares sense. It is square and symmetric. Its elements are defined by  $H_{i,j} = \frac{1}{i+j-1}$ . The command `hilb(n)` will generate this matrix. Associated with each matrix you will create an  $n$ -vector  $b = Hx$ , where  $x$  is the  $n$ -vector with all ones. This vector can be created with the command `x = ones(n,1)`. The goal is to solve  $H\tilde{x} = b$  for  $\tilde{x}$  using MATLAB linear solving command `H \ b`. Keep in mind, we know the exact solution and it is the vector of all ones  $= x$ . Fill in the table on the next page and answer the two questions afterwards. This is designed to show you how the condition number gets very large with large  $n$  and this results in large residuals and errors. The condition number is to be given with respect to the infinity norm:  $k = \text{cond}(H, \text{inf})$ . The residual and errors are to be calculated using the infinity norm as well: `norm(vector, inf)`.

**Note:** You should get a warning from MATLAB when  $n \geq 12$  stating that the matrix is either close to singular or badly scaled. It is, as noted above, ill-conditioned.

**Hand in** the completed table on the next page with answers to the two question. Do not hand in any code.

Table for problem # 3.

$n$	condition #: $k(H(n))$	residual: $\ A\tilde{x} - b\ _\infty$	error: $\ x - \tilde{x}\ _\infty$
2			
3			
4			
5			
10			
15			

- The smallest value of  $n$  for which the relative error in  $\tilde{x}$  is greater than one is:

- At this value of  $n$  the relative error is: