CS 417/517 Computational Methods and Software

Spring 2004

HW 2

Assigned: Thurs Jan 29, 2004; Due: Thurs Feb 5, 2004

Homework problems will be posted in the course directory: `www.cs.odu.edu/~pothen/Courses/CS417`. If any corrections are found to be necessary, they will be posted in this directory as well. You must show your work to receive credit for your answers. In problems where you are asked to give reasons, an answer without a stated reason will receive no credit.

1. Answer the following questions as true or false, *giving a reason for your answer*.

   (a) Floating point addition is commutative: the order in which two numbers are written when they are added makes no difference to the result; i.e., $a + b = b + a$.

   (b) Floating point addition is associative: the order in which three numbers are added, makes no difference to the result; i.e., $(a + b) + c = a + (b + c)$.

   (c) If two real numbers can be exactly represented as floating point numbers, then their exact sum can always be represented as a floating point number without any rounding errors.

   (d) An upper triangular system of equations always has a solution, which can be computed by back-substitution.

   (e) Solving an upper triangular system of equations involving n equations and n unknowns costs $n^2/2 + O(n)$ arithmetic operations.

2. Read Algorithm 2.1 in the text book for forward substitution that solves a lower triangular system of equations $L\underline{y} = \underline{b}$. Study the Matlab M-file `forwardsubs.m` provided in the course directory to see how it is implemented. Now read Algorithm 2.2 for back-substitution in the book that solves an upper triangular system of equations $U\underline{x} = \underline{b}$, and write a Matlab program implementing it as an M-file.

   You should turn in a hard copy of your M-file and also email the M-file as a plain ascii file to me at pothen@cs.odu.edu.

3. Use the Matlab program you wrote in the previous problem to solve systems of equations with random upper triangular matrices of order 100, 200, 500, and 1000. Use the following commands to generate the upper triangular matrix and the right-hand side vector. (The sentences in parentheses are comments that explain the Matlab commands.)

```
n= 100; (change this to the values you need in future steps.)
A = sprand(n, n, 0.1); (creates a sparse random matrix of order n.)
A = A + eye(n);  (adds one to each diagonal element.)
U = triu(A);  (extracts the upper triangular part of A.)
b = sum(U, 2); (adds the rows  of U together to create b.)
e= ones(n,1); (creates a column vector of n ones.)
spy(U);    (shows a picture of the matrix U.)
```

Solve the system of equations $U\underline{x} = \underline{b}$ using your M-file. If the solution is computed in the vector `x`, then you can plot the difference between it and the vector of all ones by the Matlab command `plot(x-e)`. Print the plot for each value of n, and submit it. Also, report the time taken by your program for back-substitution for each value of n, using the command `cputime`.

P.S. If you are not able to get your program for back-substitution working, then you can do this problem with the forward-substitution code that I have provided in `forwardsubs.m`. But now create a lower triangular matrix, and make other changes as needed.