NUMERICAL LINEAR ALGEBRA, 2011

HOMEWORK ASSIGNMENT 2

Well performed this homework assignment gives 1 credit point

To be handed in by September 19 at the latest

Exercise 2 a. Solve question Q2.18 in the text book. (**0.5 point**)

Exercise 2 b. Solve question Q2.20 in the text book. (**0.5 point**)

COMPUTER EXERCISE 2

To be handed in by September 19at the latest

a) Consider Algorithm 2.3 in the text book for solving a system of linear equations by Gaussian elimination **without pivoting**. Add backward substitution to the algorithm.
Interchange the two last loops on j and k and check, by implementing in MATLAB, that you get the same solution.
Hand in the two versions as m-files.
Also hand in solutions to the system with matrix and right-hand-side:
A=delsq(numgrid('S',7)), b=ones(25,1)
obtained by the two variants. This system arises when discretizing a certain partial differential equation problem.

b) Implement Algorithm 2.4 in MATLAB, **without pivoting**, and add a similar implementation of the back-substitution. Verify that the cpu-time for solving a linear system with this algorithm roughly is $O(n^3)$ for an $n \times n$ system. Use the MATLAB command **cputime** and for instance random matrices of size $n = 200,\ 400,\ 800,\ 1600$.

c) Compare your implementation in b) with MATLAB:s backslash (\). Examine the difference in efficiency between the two algorithms for solving $n \times n$ systems. Take as large n as your computer masters.

d) So far we have not studied the effect of ill-conditioning and the need for pivoting. We will study two test-cases for these aspects.
· The so called Hilbert matrix is a wellknown test matrix for ill-conditioning. You get it by the function **hilb** in MATLAB. Compute the condition number of the matrix by **cond**.
Test the Hilbert matrix of size $n = 10$ and a random right-hand-side. Compare the solutions obtained by the algorithm backslash (\) and your algorithm from b) (without pivoting). Draw conclusions regarding ill-conditioning and the reliability of the computed results. Do you need to pivot in this case?
· Test the matrix in the file **test-matrix.mat** on the course web-page and a random right-hand side. Compare the solutions obtained by the algorithm backslash (\) and your algorithm from b) (without pivoting). Draw conclusions regarding the need for pivoting in order to get a stable algorithm. Is this matrix ill-conditioned?
Note about grading: This exercise is graded according to how well you have made your implementations and discussions about the results.