

CS513, Spring 18

Prof. Ron

Assignment #4

Due April 5, 2018

General: you are not allowed in this assignment to use the `Matlab` command `A\b` for finding a solution of a linear system. Also, the assignment will require you to solve triangular systems of equations. When you write a code for that, you will get 3% extra credit, if you shorten that code by using the `Matlab` notation `A(:,j)` for the j th column of the matrix A .

(1) In this question, you are asked to develop an algorithm for solving a *symmetric* linear system.

(a) You are given a linear system $Ax = b$, A symmetric and non-singular. Prove that Gauss elimination without pivoting preserves the symmetry of A in the following sense: after i elimination steps, the $(n - i) \times (n - i)$ lower right portion of the matrix is still symmetric. (Hint: do a small example to see what happens.)

(b) Use (a) in order to modify the Gauss elimination method for LU factorization for a symmetric A . Your new algorithm should obtain the correct L and U without modifying the elements of A below the diagonal. Write a `Matlab` code for that algorithm. Your code will be used in Q.2. Submit a well-documented printout of your code and explain (on a separate sheet), how you avoided processing the lower portion of the matrix.

(c) Find the complexity of your algorithm (remember that the complexity of Gauss elimination for general matrices is $n^3/3$).

(d) Test your observation in (c): run your algorithm on 2 – 3 examples of symmetric matrices of different sizes (see Q.2), and run on those matrices the Gauss elimination algorithm from Q.2. Check the `tic-toc` count for each run (run `tic-toc` about 100 times and average in order to get a reliable reading) and compare. Compare your experiment here with your analysis in (c).

(2) In this question you compare three algorithms for solving linear systems of equations. One of the main points of the question is to experiment with the notions of *conditioning* and *stability*, and to be able, eventually, to distinguish between the two. A good answer should not only consist of a successful code, but should demonstrate your ability to digest the notions of stability and conditioning via the numerical experiments.

The three algorithms are

(a) LU factorization using Gauß elimination (without pivoting) and then solving the so-obtained triangular systems.

(b) *LU* factorization using `Matlab`'s `lu` routine.

(c) *LU* using *full* pivoting: full pivoting means that you bring to the pivot location the *largest* entry (in absolute value) among all the entries that are available. You do that by permuting rows *and* columns.

Next we generate various different matrices A . Some choices are given below

$$(a_{ij}) = \left(\frac{1}{i+j-1}\right)$$

$$(a_{ij}) = (i^{j-1})$$

To these choices add two of your own, and also you need to generate at least four random matrices (use `Matlab`'s routine `rand`), and also some symmetric random matrices (an easy way to generate a symmetric matrix is to take a square matrix B and to define $A := B + B'$.) Furthermore, you may play with the *order* of each matrix by specifying different orders.

In order to associate A with b and x , choose first x (take convenient vectors like $x = (1, \dots, 1)$ or $x = (1, 2, \dots)$). Then compute b simply by multiplying Ax . Then “forget” your x (save the original x for error analysis).

Now you are ready to go: apply to each one of the (many) pairs (A, b) each of the three algorithms and solve for x . Save then $A, b, x_1, x_2, x_3, e_1, e_2, e_3$, where x_i 's are the three (different?) solutions you obtained for the system, and e_i 's are the corresponding errors. (If you incorporate correctly your algorithm from Q.1 into the experiment here, you will get 3% extra credit).

The issues now are the conditioning of each system $Ax = b$, the stability of the algorithm that solves the system, and the complexity of the algorithm. From all the examples you ran, choose 5 which in your opinion represent the issues in the best manner. Turn in your well-documented codes, the output (for these 5 cases you've chosen), and for each one of them discuss briefly the issue.

What did you learn from this assignment about the conditioning issue and the stability issue when A is a random matrix?

Keep your code handy. Our TA may ask you to send him the code in order to try it himself.

(3) This question deals with deflation of eigenvectors. You are given the matrix

$$A = \begin{pmatrix} 309 & 228 & -240 \\ 60 & -117 & 510 \\ 12 & 6 & 298 \end{pmatrix} / 49,$$

and are told that the vector $v = [6 \ 3 \ 2]'$ is an eigenvector of A .

- (i) Deflate v from A .
- (ii) Find (say, directly) the eigenvectors of the 2X2 deflated matrix.
- (iii) Use (ii) in order to find an eigenbasis of A to \mathbb{R}^3 (i.e., a basis for \mathbb{R}^3 made of eigenvectors of A).
- (iv) Diagonalize A , i.e., write it in the form PDP^{-1} .