Numerical Analysis 1 – Class 5

Thursday, February 8th, 2018

Subjects covered

- The SVD and dimensionality reduction
- Applications of the SVD: Image compression, Latent sematic indexing
- Solving a system of linear equations: Gaussian elimination.
- Matrix decompositions and linear systems: LU, Cholesky, QR.

Reading

- N. Kutz, Chapter 2.1 (Direct solvers for linear systems).
- C. Moler, Chapter 2 (available online for free here: http://www.mathworks.com/moler/lu.pdf).
- "A Singularly Valuable Decomposition -- The SVD of a Matrix", D. Kalman (available on Blackboard).
- "The Fundamental Theorem of Linear Algebra", G. Strang (available on Blackboard).
- “Using Linear Algebra for Intelligent Information Retrieval”, M. Berry, et al. (available on Blackboard).
- "Mathematicians of Gaussian Elimination", J. Grcar (available on Blackboard).

Problems

Most of the following problems require you to write a program. As always, please make sure you also write a test program which validates your program's implementation. You will be graded on both your program as well as on your test. E-mail your answers to our TA: Mohamed Elbehiry, elbehiry.m@husky.neu.edu.

1. One way to solve the linear system $A x = b$ involves performing a QR decomposition on *A*, moving *Q* to the RHS and then doing backsubstitution on the system

$$R x = Q^T b$$

   Please write a program which implements this algorithm. Feel free to use the implementation of QR called MyQR which I have placed on Blackboard, or use the native Matlab implementation. To test your program you can simply verify that the x you compute satisfies the original problem statement $A x = b$ for a set of random matrices *A* and vectors *b*.

2. Many problems of engineering interest can be reduced to the problem of solving a tridiagonal matrix equation $A x = b$, where A is of form

$$A = \begin{pmatrix} a_1 & c_1 & 0 & 0 & 0 \\ b_1 & a_2 & c_2 & 0 & 0 \\ 0 & b_2 & a_3 & c_3 & 0 \\ 0 & 0 & b_3 & a_4 & c_4 \\ 0 & 0 & 0 & b_4 & a_5 \end{pmatrix}$$

That is, the matrix elements are non-zero only on the main diagonal and the two adjacent sub-diagonals. Solving this equation may proceed using the so-called "Thomas algorithm", which is similar to Gaussian elimination. A good description of the algorithm is contained in a tutorial by W. Lee (available on Blackboard). Other descriptions are available on the web.

Your assignment: write a Matlab program to solve an arbitrary tridiagonal system $Ax = b$ using the Thomas algorithm. (Don't worry about pivoting – assume inputs which don't need pivoting or other pre-processing.) Your program should accept the matrix A and vector b as input, and return the vector x. To test your program you can simply verify that the x you compute satisfies the original problem statement $A x = b$. Use any method you can think of to generate random tridiagonal matrices.

Note: There are several implementations of the Thomas algorithm on the web. Feel free to read about them and use them for ideas, but do not directly copy the online implementations. Please write your own code.