Study Problems 2 Applied Linear Algebra 2 Spring 2024

Problem 2.1

Consider a symmetric matrix A, i.e., $A = A^T$.

2.1.a. Consider the use of Gauss transforms to factor $A = LU$ where L is unit lower triangular and U is upper triangular. **You may assume that the factorization does not fail.** Show that $A = LDL^T$ where L is unit lower triangular and D is a matrix with nonzeros on the main diagonal. i.e., elements in positions (i, i), and zero everywhere else, by demonstrating that L and D can be computed by applying Gauss transforms appropriately to the matrix A.

2.1.b. For an arbitrary symmetric matrix the LDL^T factorization will not always exist due to the possibility of 0 in the (i, i) position of the transformed matrix that defines the i-th Gauss transform. Suppose, however, that A is a **positive definite** symmetric matrix, i.e., $x^T Ax > 0$ for any vector $x \neq 0$. Show that the diagonal element of the transformed matrix A that is used to define the vector l_i that determines the Gauss transform on step i, $M_i^{-1} = I - l_i e_i^T$, is always positive and therefore the factorization will not fail. Combine this with the existence of the LDL^T factorization to show that, in this case, the nonzero elements of D are in fact positive.

Problem 2.2

Suppose $A \in \mathbb{R}^{n\times n}$ is a nonsymmetric nonsingular diagonally dominant matrix with the following nonzero pattern (shown for $n = 6$)

$$\begin{pmatrix} * & * & * & * & * & * \\ * & * & 0 & 0 & 0 & 0 \\ * & 0 & * & 0 & 0 & 0 \\ * & 0 & 0 & * & 0 & 0 \\ * & 0 & 0 & 0 & * & 0 \\ * & 0 & 0 & 0 & 0 & * \end{pmatrix}$$

It is known that a diagonally dominant (row or column dominant) matrix has an LU factorization and that pivoting is not required for numerical reliability.

2.2.a. Describe an algorithm that solves $Ax = b$ as efficiently as possible.

2.2.b. Given that the number of operations in the algorithm is of the form $Cn^k + O(n^{k-1})$, where C is a constant independent of n and $k > 0$, what are C and k?

Problem 2.3

Let $A \in \mathbb{R}^{n\times n}$ be a nonsingular matrix, with A and A^{-1} partitioned as follows

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$$

$$A^{-1} = \begin{pmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ \tilde{A}_{21} & \tilde{A}_{22} \end{pmatrix}$$

where $A_{11} \in \mathbb{R}^{k\times k}$ and $\tilde{A}_{11} \in \mathbb{R}^{k\times k}$.

2.3.a. Assume A_{11}^{-1} and A_{22}^{-1} exist. Let $S_{11} = A_{22} - A_{21}A_{11}^{-1}A_{12}$ be the Schur complement of A with respect to A_{11} and let $S_{22} = A_{11} - A_{12}A_{22}^{-1}A_{21}$ be the Schur complement of A with respect to A_{22} Show that

$$\tilde{A}_{11} = S_{22}^{-1} \quad \text{and} \quad \tilde{A}_{22} = S_{11}^{-1}.$$

2.3.b. The assumption of the existence of A^{-1} can be turned into a consequence of the existence of the Schur complement. Show that if, $S_{11} = A_{22} - A_{21}A_{11}^{-1}A_{12}$, the Schur complement of A with respect to A_{11} exists then A is nonsingular if and only if S_{11} is nonsingular. (A similar result can be stated for S_{22}.)

Problem 2.4

Suppose an LU decomposition of a matrix $A \in \mathbb{R}^{n\times n}$ is to be computed with some form of pivoting to ensure existence. Suppose further that the matrix A is made available one row at a time.

(2.4.a) Describe an algorithm such that when the i-th row of A is received the algorithm computes the i-th row of L and the i-th row of U as well as an elementary permutation matrix P_i that ensures existence (and enhances stability).

(2.4.b) What primitives are used on each step of the algorithm and what are the dimensions of the matrices and vectors involved?

(2.4.c) Why does the pivoting strategy in the algorithm guarantee existence?

(2.4.d) What form of decomposition is computed given the pivoting strategy? (Recall, partial pivoting of rows yields $P_R A = LU$, complete pivoting yields $P_R A P_C = LU$, where P_R and P_C are permutations of rows and columns respectively. Characterize the decomposition produced by the algorithm in a similar manner.)

Problem 2.5

(Restated Golub and Van Loan 3rd Ed. p. 103 Problem P3.2.5.)

Define the elementary matrix $N_k^{-1} = I - y_k e_k^T \in \mathbb{R}^{n \times n}$, where $1 \leq k \leq n$ is an integer, $y_k \in \mathbb{R}^n$ and $e_k \in \mathbb{R}^n$ is the k-th standard basis vector. N_k^{-1} is a Gauss-Jordan transform if it is defined by requiring $N_k^{-1} v = e_k \, \nu_k$ for a particular given vector $v \in \mathbb{R}^n$ whose elements are denoted $\nu_j = e_j^T v$. For example, if $n = 6$ and $k = 3$ then

$$N_3^{-1} = \begin{pmatrix} 1 & 0 & * & 0 & 0 & 0 \\ 0 & 1 & * & 0 & 0 & 0 \\ 0 & 0 & * & 0 & 0 & 0 \\ 0 & 0 & * & 1 & 0 & 0 \\ 0 & 0 & * & 0 & 1 & 0 \\ 0 & 0 & * & 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} \nu_1 \\ \nu_2 \\ \nu_3 \\ \nu_4 \\ \nu_5 \\ \nu_6 \end{pmatrix}$$

where $*$ indicates a value that must be determined.

(2.5.a) Determine how to choose y_k and define N_k^{-1} given a vector $v \in \mathbb{R}^n$, i.e., determine the values of the elements of y_k in terms of the values of the elements of v so that $N_k^{-1} v = e_k \, \nu_k$.

(2.5.b) Determine when N_k^{-1} exists and is nonsingular.

(2.5.c) Show how a series of N_k^{-1} can be used to transform a nonsingular matrix $A \in \mathbb{R}^{n \times n}$ into a nonsingular diagonal matrix $D \in \mathbb{R}^{n \times n}$, i.e., all of the off-diagonal elements of D are 0 and all of the diagonal elements are nonzero. You may assume that A is such that all of the N_k^{-1} exist.

(2.5.d) Does the factorization that this transformation induces have any structure other than that in D?

Problem 2.6

It is known that if partial or complete pivoting is used to compute $PA = LU$ or $PAQ = LU$ of a nonsingular matrix then the elements of L are less than 1 in magnitude, i.e., $|\lambda_{ij}| \leq 1$. Now suppose $A \in \mathbb{R}^{n\times n}$ is a symmetric positive definite matrix, i.e., $A = A^T$ and $x \neq 0 \rightarrow x^T Ax > 0$. It is known that A has a factorization $A = LL^T$ where L is lower triangular with positive elements on the main diagonal (the Cholesky factorization). Does this imply that $|\lambda_{ij}| \leq 1$? If so prove it and if not give an $n \times n$ symmetric positive definite matrix with $n > 3$ that is a counterexample and justify that it is indeed a counterexample.

Problem 2.7

Suppose $PAQ = LU$ is computed via Gaussian elimination with complete pivoting. Show that there is no element in $e_i^T U$, i.e., row i of U, whose magnitude is larger than $|\mu_{ii}| = |e_i^T U e_i|$, i.e., the magnitude of the (i, i) diagonal element of U.

Problem 2.8

Consider $S \in \mathbb{R}^{n\times n}$ whose nonzero elements have the following pattern for $n = 8$:

$$S = \begin{pmatrix} 1 & 0 & 0 & 0 & \mu_1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & \mu_2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & \mu_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha & \beta & 0 & 0 & 0 \\ 0 & 0 & 0 & \gamma & \delta & 0 & 0 & 0 \\ 0 & 0 & 0 & \delta_1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \delta_2 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \delta_3 & 0 & 0 & 0 & 1 \end{pmatrix}$$

The pattern generalizes to any n easily. Assume that for any n, S is a nonsingular matrix.

2.8.a We have considered several basic transformations (Gauss transforms, Gauss-Jordan transforms, elementary permutations, Householder reflectors) that can be used to compute factorizations efficiently. Assume that S is diagonally dominant (both row-wise and column-wise).

Using whatever combination of these transformations you think appropriate, describe an algorithm to compute stably a factorization of S for any n that can be used to solve $Sx = b$. **Your algorithm should be designed to require as few computations as possible.** Your solution must include a description of how you exploit the structure of the matrix and its factors.

2.8.b Assume that you have the factorization of S defined by your algorithm from Part (2.8.a), describe an algortihm to solve $Sx = b$. **Your algorithm should be designed to require as few computations as possible.** Your solution must include a description of how you exploit the structure of the matrix and its factors.

2.8.c Determine the order of computational complexity, i.e., give k in $O(n^k)$, when your factorization algorithm is applied to a matrix of any dimension n.

2.8.d Determine the order of computational complexity, i.e., give k in $O(n^k)$, when your algorithm to solve $Sx = b$ given the factorization is applied to a matrix of any dimension n.