Prof. Ch. Schwab
B. Fitzpatrick
J. Zech

Numerische Mathematik I

ETH Zürich
D-MATH

Homework Problem Sheet 4

Introduction. This problem sheet is devoted to the LU-decomposition of matrices with particular structure or properties. The basics of polynomial interpolation are also introduced.

Problem 4.1 LU-Decomposition of Band Matrices

(4.1a) Write a MATLAB function `calcLUDecBand(A,p,q)` that calculates an $\mathbf{LU}$ decomposition for band matrices $\mathbf{A}$ with right half-bandwidth p and left half-bandwidth q without pivoting ([NMI, Alg 2.42]).

Write a MATLAB function `forwardsub(A,q,b)` that solves $\mathbf{Ax} = \mathbf{b}$ by forward substitution for lower band matrices $\mathbf{A}$ with ones on the diagonal and upper half-bandwidth $p = 0$. Write another MATLAB function `backwardsub(A,p,b)` that solves $\mathbf{Ax} = \mathbf{b}$ by backward substitution for upper band matrices $\mathbf{A}$ with lower half-bandwidth $q = 0$.

In all functions, make sure you take advantage of the band structure of the matrix.

(4.1b) Test your functions on the problem $\mathbf{Ax} = \mathbf{b}$, where

$$\mathbf{A} = \begin{pmatrix} 10^{-15} & 1 & & \\ 1 & 2 & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & 2 \end{pmatrix} \in \mathbb{R}^{10\times 10} \quad \text{and} \quad \mathbf{b} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \in \mathbb{R}^{10}.$$

Calculate $\mathbf{x}$ and the residuum $\|\mathbf{r}\|_2 = \|\mathbf{b} - \mathbf{Ax}\|_2$. What can you say about the size of the residuum?

(4.1c) Write a script that measures the runtime of the $\mathbf{LU}$ decomposition from subproblem (4.1a) and compares it to the $\mathbf{LU}$ decomposition implemented in MATLAB. As an input, use $\mathbf{A}$ from subproblem (4.1b) for sizes $n = 2^j$ with $j \in \{5, 6, \dots, 12\}$ and determine the runtime as an average of ten iterations.

Plot the results in a log-log diagram. Check whether or not the dependence of the runtime on n goes along with [NMI, Tab. 2.2]!

HINT: Using the MATLAB functions `tic` and `toc`, you can measure the runtime of a code segment.

Listing 4.1: Testcalls for Problem 4.1

```
% Construct A, b
n = 10;
A = diag(2*ones(n,1)) + diag(ones(n-1,1),1) + diag(ones(n-1,1),-1);
```

```
A(1,1) = 1.0e-15;
b = ones(n,1);
% LU decomposition
result = calcLUDecBand(A,1,1);
L = eye(n) + tril(result,-1);
U = triu(result);
% solve the system, calculate the residuum
y = forwardsub(L,1,b);
x = backwardsub(U,1,y)
r = norm(b - A*x)
```

Listing 4.2: Output for Testcalls for Problem 4.1

```
>> test_call

x =

   -0.4441
    1.0000
   -0.4444
    0.8889
   -0.3333
    0.7778
   -0.2222
    0.6667
   -0.1111
    0.5556

r =

    0.1115
```

Problem 4.2 Cholesky decomposition

Let $\mathbf{0} \neq \mathbf{A} \in \mathbb{R}^{n\times n}$ be a symmetric matrix.

(4.2a) Give the definition of *positive definiteness* for the matrix $\mathbf{A}$.

Solution:

(4.2b) Show that, if $\mathbf{A}$ is positive definite, then $a_{ii} > 0$ for all $1 \leq i \leq n$.

Does the reverse implication hold as well? Justify your answer!

Solution: Since $\mathbf{x}^\top \mathbf{A}\mathbf{x} > 0$ must hold for all $\mathbf{x} \in \mathbb{R}^n \setminus \{0\}$, it also holds for the canonical vectors $\mathbf{e}_i = (0, \dots, 0, 1, 0, \dots, 0)^\top$, which have the 1 on the i^{th} position.

(4.2c) Let $\mathbf{A}$ now be positive definite as well. Define the Cholesky-decomposition of $\mathbf{A}$ and formulate a sufficient condition that the decomposition can be done.

Solution:

(4.2d) Write down an algorithm for the Cholesky-decomposition with pivoting, for which the element of the remaining submatrix with the largest absolute value is brought into the pivot position at each step.

What is the matrix-form of this pivoting Cholesky-decomposition?

Solution: Since $\mathbf{A}$ and the submatrices of all steps are SPD, the largest element is always on the diagonal (compare with [NMI, Thm. 2.35] part 3). The pivoting strategy thus only has to search the diagonal and bring the row/column of the largest element to the front.

The algorithm for the Cholesky-decomposition with pivoting:

Input: SPD Matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$.

Output: Cholesky-factor $\mathbf{R}$ and permutation matrix $\mathbf{P}$, such that $\mathbf{P}\mathbf{A}\mathbf{P}^\top = \mathbf{R}^\top\mathbf{R}$.

(4.2e) Show that a Cholesky-algorithm with full pivoting for semi-definite $\mathbf{A}$ with $r = \operatorname{rank}(\mathbf{A}) < n$ aborts after exactly r steps in exact arithmetic.

Solution:

See Lemma 1 of H. Harbrecht, M. Peters, R. Schneider: On the low-rank approximation by the pivoted Cholesky decomposition, 2010, as well as the following.

For $\mathbf{A}$ positive semi-definite, all eigenvalues satisfy $\lambda_i \geq 0$, $i = 1, \ldots, n$. For the trace of the matrix, this implies $\operatorname{tr}(\mathbf{A}) = \sum_{i=1}^{n} \lambda_i > 0$. Therefore, the existence of at least one positive diagonal entry $a > 0$ is guaranteed. Through the application of a symmetric permutation matrix, this entry can always be brought into the $(1, 1)$-position.

Problem 4.3 $\mathrm{LDL}^\top$ decomposition

From the proof on the existence of the Cholesky decomposition ([NMI, Thm. 2.36]), it follows that for specific symmetric matrices $\mathbf{A}$, there is a decomposition $\mathbf{A} = \mathbf{LDL}^\top$ where $\mathbf{L}$ is a lower triangular matrix with entries ones on the diagonal and $\mathbf{D}$ is a real-valued diagonal matrix.

(4.3a) Modify [NMI, Alg. 2.37] such that it calculates the $\mathbf{LDL}^\top$ decomposition and implement this algorithm in a MATLAB function `calcLDLDecomp(.)`. The function return value is supposed be a matrix such that the upper right half contains the corresponding entries of $\mathbf{L}^\top$ and the diagonal contains the corresponding elements of $\mathbf{D}$.

Check your algorithm on the example

$$\mathbf{M} = \begin{pmatrix} -2 & 1 & 0 \\ 1 & -2 & 1 \\ 0 & 1 & -2 \end{pmatrix}.$$

Does the $\mathbf{LDL}^\top$ decomposition exist for symmetric negative definite matrices or for indefinite matrices, i. e. does the modified algorithm compute a $\mathbf{LDL}^\top$ decomposition for those matrices? If not, give a counterexample.

Solution:

(4.3b) Using the functions `tic` and `toc` that are provided by MATLAB to measure time, determine the execution time t_n of your function `calcLDLDecomp(.)` for the input `A = gallery('moler',n)` and $n \in \{100, 200, \ldots, 1000\}$. Plot the measured times in a double logarithmic diagram and postulate a law for the execution time of the form $t_n = c \cdot n^a$.

Solution:

In the double logarithmic plot, the data points are roughly on a straight line, implying that we indeed have a law of the form $t_n = c \cdot n^a$. For a first approximation, we just take the two outmost points, i. e. $n = 100$ and $n = 1000$ with times t_1 and t_{10} and solve the system for the constants c and a:

(4.3c) Following your algorithm in subproblem (4.3a), determine the costs w_n for computing the $\mathbf{LDL}^\top$ decomposition of a $n \times n$-matrix. Therefore, assume all floating point operations cost 1 time unit. Compare the result to the postulated law in subproblem (4.3b).

Solution: Setting the costs for one elementary operation to 1 time unit, we can count the costs in the code for subproblem (4.3a). Note that there are two `for`-loops, each represented by one of the sums. We get

$$w_n = \underbrace{n+1}_{\text{outside loops}} + \sum_{j=2}^{n} \left(\sum_{i=2}^{j-1} \left(\underbrace{(i-1)}_{.*} + \underbrace{(2i-3)}_{*} + \underbrace{2}_{-\text{ and }/} \right) + \underbrace{(j-1)}_{.*} + \underbrace{(2j-3)}_{*} + \underbrace{1}_{-} \right) =$$

(4.3d) The inertia of a matrix $\mathbf{A}$ is a set of nonnegative integers (m, z, p) where m, z, and p are the number of negative, zero, and positive eigenvalues of $\mathbf{A}$, respectively.

Prove Sylvester's Law of Inertia which states that if $\mathbf{A} \in \mathbb{R}^{n\times n}$ is symmetric and $\mathbf{X} \in \mathbb{R}^{n\times n}$ is nonsingular, then $\mathbf{A}$ and $\mathbf{X}^T\mathbf{A}\mathbf{X}$ have the same inertia.

HINT: For a symmetric matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ the k^{th} largest eigenvalue of $\mathbf{A}$ is given by

$$\lambda_k(\mathbf{A}) = \max_{\dim(S)=k} \quad \min_{0\neq \mathbf{y}\in S} \frac{\mathbf{y}^T\mathbf{A}\mathbf{y}}{\mathbf{y}^T\mathbf{y}}$$

Solution: Suppose for some k we have that $\lambda_k(\mathbf{A}) > 0$ and define the subspace $S_0 \subseteq \mathbb{R}^n$ by

$$S_0 = \text{span}\{\mathbf{X}^{-1}q_1, \ldots, \mathbf{X}^{-1}q_k\}, \quad q_i \neq 0$$

where $\mathbf{A}q_i = \lambda_i(\mathbf{A})q_i$ and $i = 1, \ldots, k$.

we have that

$$\lambda_k(\mathbf{X}^T\mathbf{A}\mathbf{X}) \geq \min_{y\in S_0}\left\{\frac{\mathbf{y}^T(\mathbf{X}^T\mathbf{A}\mathbf{X})\mathbf{y}}{\mathbf{y}^T(\mathbf{X}^T\mathbf{X})\mathbf{y}}\frac{\mathbf{y}^T(\mathbf{X}^T\mathbf{X})\mathbf{y}}{\mathbf{y}^T\mathbf{y}}\right\} \geq \lambda_k(\mathbf{A})\sigma_n(\mathbf{X})^2.$$

(4.3e) Suppose $\mathbf{A}$ has been reduced to some tridiagonal matrix $\mathbf{T}$ that has the same eigenvalues as $\mathbf{A}$ through the application of some eigenvalue preserving transformation. We can find the inertia of $\mathbf{A}$ by calculating the inertia of $\mathbf{T}$ instead. This leads to a performance enhancement as operations such as Gaussian elimination, forward substitution, and back substitution are more efficient for banded matrices such as the tridiagonal $\mathbf{T}$.

Write an efficient algorithm `inertia.m` which takes as input the matrix T below, applies Sylvester's Law of Inertia, and outputs the inertia (m, z, p) where m, z, and p are as described above.

$$\mathbf{T} = \begin{pmatrix} -2 & -1 & 0 & 0 \\ -1 & 0 & -2 & 0 \\ 0 & -2 & -15 & -8 \\ 0 & 0 & -8 & 9 \end{pmatrix}$$

Problem 4.4 Schur Complement

The so-called Schur complement plays a central role in many algorithms of numerical linear algebra. It is defined as follows. Suppose $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$ are respectively $p \times p$-, $p \times q$-, $q \times p$- and $q \times q$-matrices, and that $\mathbf{A}$ is invertible. Then the Schur complement of the block $\mathbf{A}$ of the matrix

$$\mathbf{M} := \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix}$$

is the $q \times q$-matrix $\mathbf{S} = \mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B}$. In this problem assume that $\mathbf{M} \in \mathbb{R}^{(p+q)\times(p+q)}$ is symmetric positive definite.

(4.4a) Let $\mathbf{S} \in \mathbb{R}^{q\times q}$ be symmetric and positive definite, and $\mathbf{b} \in \mathbb{R}^q$. Show that the vector $\mathbf{x}^* := \mathbf{S}^{-1}\mathbf{b}$ is the unique minimizer of the function

$$f : \begin{cases} \mathbb{R}^p & \to \quad \mathbb{R} \\ \mathbf{x} & \to \quad \frac{1}{2}\mathbf{x}^\top \mathbf{S}\mathbf{x} - \mathbf{b}^\top \mathbf{x} \end{cases} . \tag{4.4.1}$$

HINT: Find an equivalent expression for $f(\mathbf{x}) - f(\mathbf{x}^*)$ that is guaranteed to be positive for $\mathbf{x} \neq \mathbf{x}^*$. To that end remember what it means that $\mathbf{S}$ is positive definite (SPD).

Solution: We want to show that $f(\mathbf{x}) - f(\mathbf{x}^*) > 0$ for all $\mathbf{x} \neq \mathbf{x}^* := \mathbf{S}^{-1}\mathbf{b}$.

$$f(\mathbf{x}) - f(\mathbf{x}^*) = \frac{1}{2}\mathbf{x}^\top \mathbf{S}\mathbf{x} - \mathbf{b}^\top \mathbf{x} - f(\mathbf{x}^*) = \frac{1}{2}\mathbf{x}^\top \mathbf{S}\mathbf{x} - (\mathbf{x}^*)^\top \mathbf{S}\mathbf{x} - f(\mathbf{x}^*) =$$

(4.4b) Prove that

$$\mathbf{y}^T\mathbf{S}\mathbf{y} = \min_{\mathbf{x}\in\mathbb{R}^p} \begin{pmatrix}\mathbf{x}\\ \mathbf{y}\end{pmatrix}^T \mathbf{M} \begin{pmatrix}\mathbf{x}\\ \mathbf{y}\end{pmatrix}, \quad \mathbf{y} \in \mathbb{R}^q\ .$$

HINT: The expression, of which we take the minimum, is structurally close to f from (4.4.1). Hence, the result of (4.4a) can be used.

Solution: Define

$$f(\mathbf{x}) = \begin{pmatrix}\mathbf{x}\\ \mathbf{y}\end{pmatrix}^T \mathbf{M} \begin{pmatrix}\mathbf{x}\\ \mathbf{y}\end{pmatrix} = \mathbf{x}^T\mathbf{A}\mathbf{x} + \mathbf{y}^T\mathbf{C}\mathbf{x} + \mathbf{x}^T\mathbf{B}\mathbf{y} + \mathbf{y}^T\mathbf{D}\mathbf{y}.$$

Then $\nabla f(\mathbf{x}) = 2\mathbf{x}^T\mathbf{A} + \mathbf{y}^T\mathbf{C} + \mathbf{y}^T\mathbf{B}^T$, and $\nabla f(\mathbf{x}_0) = 0$ and $\mathbf{C} = \mathbf{B}^T$ imply that $\mathbf{x}_0 = -\mathbf{A}^{-1}\mathbf{B}\mathbf{y}$. By evaluating f at $\mathbf{x}_0$, we conclude that

(4.4c) Prove that $\mathbf{S}$ is symmetric positive definite.

Solution: From the definition of $\mathbf{S}$ one has $\mathbf{S}^T = \mathbf{D}^T - \mathbf{B}^T\mathbf{A}^{-T}\mathbf{C}^T$. Since the matrix $\mathbf{M}$ is symmetric by assumption, $\mathbf{D}^T = \mathbf{D}$, $\mathbf{C}^T = \mathbf{B}$, $\mathbf{B}^T = \mathbf{C}$ and $\mathbf{A}^T = \mathbf{A}$.

(4.4d) Prove that

$$\kappa_2(\mathbf{S}) \le \kappa_2(\mathbf{M}).$$

Solution: Since $\mathbf{M}$ and $\mathbf{S}$ are positive definite, the result in **??** can be applied to $\|\mathbf{M}\|_2$ and $\|\mathbf{S}\|_2$. Note that, due to subproblem (4.4b)

$$\|\mathbf{S}\|_2 = \sup_{\mathbf{y}\neq 0} \frac{\mathbf{y}^T\mathbf{S}\mathbf{y}}{\mathbf{y}^T\mathbf{y}} \le \sup_{\mathbf{y}\neq 0}\sup_{\mathbf{x}} \frac{\begin{pmatrix}\mathbf{x}\\ \mathbf{y}\end{pmatrix}^T \mathbf{M} \begin{pmatrix}\mathbf{x}\\ \mathbf{y}\end{pmatrix}}{\begin{pmatrix}\mathbf{x}\\ \mathbf{y}\end{pmatrix}^T \begin{pmatrix}\mathbf{x}\\ \mathbf{y}\end{pmatrix}} \le \sup_{\begin{pmatrix}\mathbf{x}\\ \mathbf{y}\end{pmatrix}\neq 0} \frac{\begin{pmatrix}\mathbf{x}\\ \mathbf{y}\end{pmatrix}^T \mathbf{M} \begin{pmatrix}\mathbf{x}\\ \mathbf{y}\end{pmatrix}}{\begin{pmatrix}\mathbf{x}\\ \mathbf{y}\end{pmatrix}^T \begin{pmatrix}\mathbf{x}\\ \mathbf{y}\end{pmatrix}} = \|\mathbf{M}\|_2.$$

Now writing the 2-condition number of $\mathbf{M}$ as

Published on March 16, 2016.

To be submitted on April 5, 2016.

MATLAB: Submit all file in the online system. Include the files that generate the plots. Label all your plots. Include commands to run your functions. Comment on your results.

References

[NMI] Lecture Notes for the course "Numerische Mathematik I".

Last modified on March 18, 2016