Homework no. 2

Let n be the system's size, ϵ - the computation error, $A \in \mathbb{R}^{n\times n}$ - a real, symmetric positive definite squared matrix, $b \in \mathbb{R}^n$ - a vector with real elements.

- Compute, when it is possible, the Cholesky decomposition for matrix A ($A = LL^T$), where L is a lower triangular matrix with positive diagonal elements ($l_{ii} > 0, \forall i$) ;
- Using this decomposition, compute the determinant of matrix A ($\det A = \det L \det L^T$) ;
- With the above computed LL^T decomposition, and using the substitution methods compute an approximative solution x_{Chol} for the system $Ax = b$;
- Verify that your computations are correct by displaying the norm:

  $$||A^{init}x_{Chol} - b||_2$$

  (this norm should be smaller than $10^{-8}, 10^{-9}$)

  A^{init} and b^{init} are the initial data, not those modified during computations. We denoted by $||\cdot||_2$ the Euclidean norm.
- Using one of the libraries mentioned on the lab's web page, compute and display the solution a LU decomposition for matrix A and the solution for system $Ax = b$;
- After computing the Cholesky decomposition $A = LL^T$ for the symmetric matrix A, compute an approximation for the inverse of this matrix, A^{-1}_{Chol}. Compute another approximation of this inverse using the library, A^{-1}_{bibl}. Display:

  $$||A^{-1}_{Chol} - A^{-1}_{bibl}||$$

  Use any matriceal norm implemented in the employed library.
- Write (and use) functions for reading vectors and matrices from keyboard, file, random initialisation and functions for displaying vectors and matrices (on display and in file).

  Write your code so it could be tested (also) on systems with $n > 100$.

- *Constraint*: In your program use only one matrix, A and a vector d that contains the diagonal elements of the initial matrix A. The LL^T decomposition will be computed and stored in lower triangular part of matrix A. By using this type of allocation, one does loses the diagonal elements of matrix A, thus the need to save these elements in vector $d, d_i = a_{ii}^{init}$.

- As input data introduce a matrix that is only symmetric. If this matrix is not positive definite, the algorithm cannot compute the Cholesky factorization. In this situation stop the computations with an appropriate error message.

Bonus 25 pt.: Compute the Cholesky decomposition for matrix A with the following storage restrictions: in your program, use for storing matrices A and L two vectors of size $n(n+1)/2$. In these vectors one stores the elements from the lower triangular part of these matrices. For matrix A the elements from the upper triangular part can be accessed using the symmetry relation. With this new type of data storage, compute the solution of the linear system $Ax = b$, x_{Chol}.

Remarks

1. The computation error ϵ, is a positive number:

$$\epsilon = 10^{-m}(\text{with } m = 5, 6, ..., 10, ...\text{at choice}).$$

The computation error will be an input for your program (read from keyboard or file) the same as data size n. One employs this number for testing the non-zero value of a variable before using it for division.

If you want to compute $s = \dfrac{1.0}{v}$, where $v \in \mathbb{R}$ is a real variable, you should not use the comparison with zero, as in the following sequence of code:

$$\text{if}(v! = 0)\ s = 1/v;$$

$$\text{else print(" division by 0")};$$

instead, you will write:

if($abs(v) > eps$) $s = 1/v$;

else print(" division by 0");

2. If we have the LL^T decomposition of symmetric and positive definite matrix A, solving the linear system $Ax = b$ is done by solving two triangular linear systems:

$$Ax = b \longleftrightarrow LL^T x = b \longleftrightarrow \begin{cases} Ly & = & b, \\ L^T x & = & y. \end{cases}$$

   First, one solves the lower triangular linear system $Ly = b$. Secondly, one solves the upper triangular system $L^T x = y$ where y is the solution obtain by solving the system $Ly = b$. The vector x obtained by solving the system $L^T x = y$ is also the solution of the initial linear system $Ax = b$.

3. In order to compute the norm $\|A^{init} x_{Chol} - b^{init}\|_2$ we use the following formulae:

$$A = (a_{ij}) \in \mathbb{R}^{n \times n} \ , \ x \in \mathbb{R}^n \ , \ Ax = y \in \mathbb{R}^n \ , \ y = (y_i)_{i=1}^n$$

$$y_i = \sum_{j=1}^{n} a_{ij} x_j \ , \quad i = 1, 2, \ldots, n$$

$$z = (z_i)_{i=1}^n \in \mathbb{R}^n \quad , \quad \|z\|_2 = \sqrt{\sum_{i=1}^{n} z_i^2}$$

   Take care when computing the vector Ax_{Chol} - matrix A will be modified after computing the Cholesky decomposition : it contains in the lower triangular part the elements of matrix L, and the elements of the initial matrix A are in the strictly upper triangular part of matrix A and the diagonal elements are in vector d.

Substitution methods

Consider the linear system:

$$Ax = b \tag{1}$$

where the matrix A is triangular. In order to find the unique solution of the linear system (1), the matrix A must be non-singular ($\det A \neq 0$). The determinant of upper triangular matrices has the following formula:

$$\det A = a_{11}a_{22}\cdots a_{nn}$$

Consequently, we assume:

$$\det A \neq 0 \Longleftrightarrow a_{ii} \neq 0 \quad \forall i = 1, 2, \ldots, n$$

Consider the linear system (1) with lower triangular matrix:

$$\begin{array}{lllllllllll}
a_{11}x_1 & & & & & & & & & = & b_1 \\
a_{21}x_1 & + & a_{22}x_2 & & & & & & & = & b_2 \\
\vdots & & & & & & & & & & \\
a_{i1}x_1 & + & a_{i2}x_2 & + & \cdots & + & a_{ii}x_i & & & = & b_i \\
\vdots & & & & & & & & & & \\
a_{n1}x_1 & + & a_{n2}x_2 & + & \cdots & + & a_{ni}x_i & + \ \cdots & + \ a_{nn}x_n & = & b_n
\end{array}$$

The unknown variables x_1, x_2,...,x_n will be computed sequentially, using the system's equations starting with the first and ending with the last. Using the first equation, we compute the value of x_1:

$$x_1 = \frac{b_1}{a_{11}} \tag{2}$$

From the second equation , using the above computed value of x_1, we obtain:

$$x_2 = \frac{b_2 - a_{21}x_1}{a_{22}}$$

By employing values x_j previously computed and system's equation i we get:

$$x_i = \frac{b_i - a_{i1}x_1 - \cdots - a_{ii-1}x_{i-1}}{a_{ii}}$$

Last equation yields the value of x_n:

$$x_n = \frac{b_n - a_{n1}x_1 - -a_{n2}x_2 - \cdots - a_{nn-1}x_{n-1}}{a_{nn}}$$

The above described method is named *forward substitution algorithm* for solving linear systems of equations with upper triangular matrices:

$$x_i = \frac{\left(b_i - \sum\limits_{j=1}^{i-1} a_{ij}x_j\right)}{a_{ii}} \quad , \quad i = 1, 2, \ldots, n \tag{3}$$

Next, we consider the linear system (1) with upper triangular matrix:

$$\begin{array}{ccccccccccc}
a_{11}x_1 & + & \cdots & + & a_{1i}x_i & + & \cdots & + & a_{1n-1}x_{n-1} & + & a_{1n}x_n & = & b_1 \\
 & & \ddots & & & & & & & & & & \\
 & & & & a_{ii}x_i & + & \cdots & + & a_{in-1}x_{n-1} & + & a_{in}x_n & = & b_i \\
 & & & & & & \ddots & & & & & & \\
 & & & & & & & & a_{n-1n-1}x_{n-1} & + & a_{n-1n}x_n & = & b_{n-1} \\
 & & & & & & & & & & a_{nn}x_n & = & b_n
\end{array}$$

The unknown variables x_1, x_2,...,x_n will be computed sequentially, using system's equations starting with the last and ending with the first. Using the last equation, we compute the value of x_n:

$$x_n = \frac{b_n}{a_{nn}} \tag{4}$$

From equation number $(n-1)$, using the above computed value of x_n, we obtain:

$$x_{n-1} = \frac{b_{n-1} - a_{n-1n}x_n}{a_{n-1n-1}}$$

By employing values x_j previously computed and system's equation i we get:

$$x_i = \frac{b_i - a_{ii+1}x_{i+1} - \cdots - a_{in}x_n}{a_{ii}}$$

First equation yields the value of x_1:

$$x_1 = \frac{b_1 - a_{12}x_2 - \cdots - a_{1n}x_n}{a_{11}}$$

The above described method is named *back substitution algorithm* for solving linear systems of equations with upper triangular matrices:

$$x_i = \frac{\left(b_i - \sum\limits_{j=i+1}^{n} a_{ij}x_j\right)}{a_{ii}} \quad , \quad i = n, n-1, \ldots, 2, 1 \tag{5}$$

Cholesky's decomposition (LL^T)

Let $A \in \mathbb{R}^{n\times n}$ be a real, symmetric ($A = A^T$), positive definite square matrix of size n.

A symmetric matrix is a matrix that is equal to it's transpose. The elements from the lower triangular part are equal to the elements from the upper triangular part of the matrix (the information is duplicated).

A matrix is positive definite if it satisfies the following relation:

$$(Ax, x)_{\mathbf{R^n}} > 0 \quad , \quad \forall x \in \mathbf{R}^n \ , \ x \neq 0 \tag{6}$$

A positive definite matrix is non-singular ($\det A \neq 0$).

For a symmetric, positive definite matrix A one wants to find a decomposition of the following form:

$$A = LL^T$$

where $L \in \mathbf{R}^{n\times n}$ is a lower triangular matrix and L^T is it's transpose (upper triangular matrix).

Algorithm for Computing the LL^T Decomposition (Cholesky's method)

Let A be a real, symmetric, positive definite squared matrix. The Cholesky's method for computing the lower triangular matrix L has n steps. At each step one computes the elements of a column from matrix L.

Step p $(p = 1, 2, \ldots, n)$

In this step one computes the elements of column p of matrix L, $l_{ip}, i = p, \ldots, n$ ($l_{ip} = 0, i = 1, \ldots, p-1$). First one computes the diagonal element, l_{pp} and then the other elements of column p, $l_{ip}, i = p+1, \ldots, n$.

We know from previous steps the elements of the first $p-1$ columns from L (the elements l_{ij} with $j = 1, \ldots, p-1\ , \ \forall i$)

Computation of the diagonal element l_{pp} :
Using the relationţia $A = LL^T$:

$$\begin{aligned} a_{pp} &= (LL^T)_{pp} = \sum_{j=1}^{n} l_{pj} l_{jp}^T = (l_{pj} = 0, j = p+1, \ldots, n \,,\, l_{jp}^T = l_{pj}) = \\ &= \sum_{j=1}^{p-1} l_{pj}^2 + l_{pp}^2 \end{aligned}$$

Thus:

$$a_{pp} = \sum_{j=1}^{p-1} l_{pj}^2 + l_{pp}^2$$

In the above relation the only unknown element is l_{pp} because the coefficients $l_{pj}, j = 1, \ldots, p-1$ are placed on the first $p-1$ columns of matrix L that were compute in previous steps of the algorithm.

We deduce the following formula:

$$l_{pp} = \pm \sqrt{a_{pp} - \sum_{j=1}^{p-1} l_{pj}^2} \tag{7}$$

If in the above formula the argument of the square root function is negative the algorithm stops, the Cholseky decomposition cannot be computed. This situation can occur when matrix A is not positive definite.

Computation of l_{ip} , $i = p+1, \ldots, n$
We use again the relation $A = LL^T$:

$$\begin{aligned} a_{ip} &= (LL^T)_{ip} = \sum_{j=1}^{n} l_{ij} l_{jp}^T = (l_{jp}^T = l_{pj}, l_{pj} = 0, j = p+1, \ldots, n) = \\ &= \sum_{j=1}^{p-1} l_{ij} l_{pj} + l_{ip} l_{pp} \end{aligned}$$

If $l_{pp} \neq 0$ (for symmetric, positive definite matrices A this is always true)one can compute the coefficients from the p column of matrix L in the following way:

$$l_{ip} = \left(a_{ip} - \sum_{j=1}^{p-1} l_{ij} l_{pj} \right) \Big/ l_{pp} \,,\; i = p+1, \ldots, n \tag{8}$$

(the elementa l_{ij} and l_{pj} $j = 1, \ldots, p-1$ are already computed in the previous $p-1$ steps and l_{pp} was calculated above, at the beginning of step$p$)

If the matrix is not positive definite and the situation $l_{pp} = 0$ occurs, the algorithm stops, the LL^T decomposition cannot be computed. For positive definite matrices $l_{pp} \neq 0, \forall p$.

Remark:

For storing the lower triangular part of matrix L one can use the lower triangular part of the initial matrix A:

$$l_{ij} = a_{ij} \quad , \quad i = 1, 2, \ldots, n \ , \ j = 1, 2, \ldots, i.$$

Note that in this way the diagonale elements of the initial matrix A, a_{ii}, are lost. This is the reason why, we use vector d to store this elements:

$$d_i = a_{ii} \quad , \quad i = 1, 2, \ldots, n$$

The computations (7) and (8) can be performed directly in matrix A.

With this type of storage, one must take care how to apply the substitution methods for solving the two triangular linear systems:

$$Ly = b \quad \text{and} \quad L^T x = y.$$

The computation of the vector $A^{init} x_{Chol}$ must be adapted to the new storage for matrix A.

Computing the approximation of the inverse of a matrix

If one knows a numerical algorithm for solving linear systems of equations (in our case, we shall use the Cholseky decomposition), the columns of the iverse matrix can be approximated by solving n linear systems.

Column j of the inverse matrix A^{-1} is approximated by computing the solution of the linear system:

$$\begin{aligned}&Ax = e_j \ , \ j = 1, 2, \ldots, n,\\ &e_j = (0, \ldots, 1, 0 \ldots, 0)^T \quad , 1 \text{ is on position } j \text{ in vector } e_j\end{aligned}$$

The algorithm for computing $A^{-1}_{Chol}$is the following:

- compute the Cholesky factorization of matrix A, $A = LL^T$;
- for $j = 1, \ldots, n$
    1. $b = e_j$;
    2. solve the lower triangular system $Ly = b$, one obtains the solution y^*
    3. solve the upper triangular system $L^T x = y^*$, one obtains the solution x^*
    4. save x^* in column j of matrix A^{-1}_{Chol}

The above described method , is in fact a procedure for solving the matrix equation:

$$AX = I_n \ , \ X \in \mathbb{R}^{n\times n} \ , \ I_n = \text{unity matrix.}$$

Example

$$A = \begin{pmatrix} 2.25 & 3 & 3 \\ 3 & 9.0625 & 13 \\ 3 & 13 & 24 \end{pmatrix} = \begin{pmatrix} 1.5 & 0 & 0 \\ 2 & 2.25 & 0 \\ 2 & 4 & 2 \end{pmatrix} \begin{pmatrix} 1.5 & 2 & 2 \\ 0 & 2.25 & 4 \\ 0 & 0 & 2 \end{pmatrix}$$

For $b = \begin{pmatrix} 9 \\ 35.0625 \\ 61 \end{pmatrix}$ the system's solution is $x^* = \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix}$

$$A^{-1} = \begin{pmatrix} \frac{776}{729} & -\frac{176}{243} & \frac{7}{27} \\ -\frac{176}{243} & \frac{80}{81} & -\frac{4}{9} \\ \frac{7}{27} & -\frac{4}{9} & \frac{1}{4} \end{pmatrix} \approx \begin{pmatrix} 1.0645 & -0.7243 & 0.2593 \\ -0.7243 & 0.9877 & -0.4444 \\ 0.2593 & -0.4444 & 0.2500 \end{pmatrix}$$