Homework no. 2

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

- Compute, when it is possible, an LU decomposition for matrix A ($A = LU$), where L is a lower triangular matrix and U is an upper triangular matrix with 1 on the diagonal ($u_{ii} = 1, \forall i$) ;
- Using this decomposition, compute the determinant of matrix A ($\det A = \det L \det U$) ;
- With the above computed LU decomposition, and using the substitution methods compute an approximative solution x_{LU} for the system $Ax = b$;
- Verify that your computations are correct by displaying the norm:

  $$\|A^{init} x_{LU} - b^{init}\|_2$$

  (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.
- *Constraint*: In your program use only two matrices, A and A^{init} (a copy of the initial matrix). The LU decomposition will be computed and stored in matrix A. By doing this type of allocation, one does not save the diagonal elements of matrix U. You shall take into account the fact that $u_{ii} = 1, \forall i$ when solving the upper triangular linear system $Ux = y$ (one modifies the function that implements the back substitution method).
- Using one of the libraries mentioned on the lab's web page, compute and display the solution of the system $Ax = b$ and also the matrix' A inverse, A^{-1}_{lib}. Display the following norms:

  $$||x_{LU} - x_{lib}||_2$$

  $$||x_{LU} - A^{-1}_{lib} b^{init}||_2.$$

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

Bonus 25 pt.: Compute the LU decomposition for matrix A with the following storage restrictions: in your program, use only one matrix to store matrix A. This matrix should remain unchanged, it will be used only for computing de LU decomposition. For storing the matrices L and U use two vectors of size $n(n+1)/2$, where the elements from the lower, repsectively upper part of matrices L and U will be stored. With this new type of data storage, compute the solution of the linear system $Ax = b$, x_{LU}.

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:

   if($v! = 0$) $s = 1/v$;

   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 LU decomposition of matrix A, solving the linear system $Ax = b$ is done by solving two triangular linear systems:

$$Ax = b \longleftrightarrow LUx = b \longleftrightarrow \left\{ \begin{array}{rcl} Ly & = & b, \\ Ux & = & y. \end{array} \right.$$

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

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

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

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

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

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:

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

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:

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

LU Decomposition

Assume $A \in \mathbb{R}^{n\times n}$ is a real square matrix of size n that satisfies the property:

$$\det A_k \neq 0, \forall k = 1, \ldots, n, \quad A_k = (a_{ij})_{i,j=1,\ldots,k}. \tag{6}$$

In these conditions it is possible to prove that there exists a unique lower triangular matrix $L = (l_{ij})_{i,j=1,\ldots,n}$ and a unique upper triangular matrix $U = (u_{ij})_{i,j=1,\ldots,n}$ with $u_{ii} = 1, i = 1, \ldots, n$ such that:

$$A = LU \tag{7}$$

Crout's algorithm for computing the LU decomposition

Let $A \in \mathbb{R}^{n\times n}$ be a real square matrix of size n that satisfies the above property (6). The algorithm for computing the elements of the matrices L and U has n steps. At each step, one computes simultaneously a column from matrix L and a row from matrix U.

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

One computes the elements of column p for matrix L , $l_{ip}, i = p, \ldots, n$, and the elements on row p for matrix U, $u_{pp} = 1$, $u_{pi}, i = p+1, \ldots, n$.

We know from previous steps the elements of the first $p-1$ columns from L (the elements l_{jk} with $k = 1, \ldots, p-1$) and the elements of the first $p-1$ rows from U (the elements u_{ki} with $k = 1, \ldots, p-1$).

Computing the elements from column p of matrix L : l_{ip} $i = p, \ldots, n$ $(l_{ip} = 0, i = 1, \ldots, p-1)$

$$a_{ip} = \sum_{k=1}^{n} l_{ik} u_{kp} = \sum_{k=1}^{p} l_{ik} u_{kp} = l_{ip} u_{pp} + \sum_{k=1}^{p-1} l_{ik} u_{kp}$$

$$(u_{kp} = 0, k = p+1, \ldots, n)$$

Taking into account that $u_{pp} = 1$, one can calculate column p of matrix L in the following way:

$$l_{ip} = a_{ip} - \sum_{k=1}^{p-1} l_{ik} u_{kp} \ , \ \ i = p, \ldots, n \ , \ \ l_{ip} = 0 \ i = 1, \ldots, p-1. \tag{8}$$

(u_{kp}, $k = 1, \ldots, p-1$ are elements from rows of matrix U previously computed, and l_{ik}, $k = 1, \ldots, p-1$, are values from columns of L computed in previous steps of the algorithm.)

Computing the elements of row p from matrix U: u_{pi} , $i = p+1, \ldots, n$ $(u_{pp} = 1,\ u_{pi} = 0\ ,\ i = 1, \ldots, p-1)$

$$a_{pi} = \sum_{k=1}^{n} l_{pk} u_{ki} = \sum_{k=1}^{p} l_{pk} u_{ki} = l_{pp} u_{pi} + \sum_{k=1}^{p-1} l_{pk} u_{ki}$$

$$(l_{pk} = 0, k = p+1, \ldots, n)$$

If $l_{pp} \neq 0$, one can compute u_{pi} using the formula:

$$u_{pi} = \frac{a_{pi} - \displaystyle\sum_{k=1}^{p-1} l_{pk} u_{ki}}{l_{pp}} \ , \quad i = p+1, \ldots, n \tag{9}$$

(the elements l_{pk}, $k = 1, \ldots, p-1$ are elements from columns of matrix L previously computed and u_{ki} $k = 1, \ldots, p-1$, are elements from rows of U already computed in previous steps).

If a diagonal element of L is zero, $l_{pp} = 0$, the algorithm stops, in this case, the LU decomposition cannot be computed, the matrix A, has a zero minor, $\det A_p = 0$.

Remark:

For saving the matrices L and U one can use the initial matrix A. The strictly upper triangular part of matrix A is employed to store the non-zero elements u_{ij} of matrix U with $i = 1, 2, \ldots, n$, $j = i+1, \ldots, n$ (except the diagonal elements) and the lower triangular part of matrix A for saving the elements l_{ij} of matrix L , $i = 1, \ldots, n$, $j = 1, 2, \ldots, i$. Note that we did not retain anywhere the diagonal $u_{ii} = 1$ $\forall i = 1, \ldots, n$. One must take this into account when solving the upper triangular system. The computations (8) and (9) can be performed directly in matrix A.

Example

$$A = \begin{pmatrix} 2.5 & 2 & 2 \\ 5 & 6 & 5 \\ 5 & 6 & 6.5 \end{pmatrix} = \begin{pmatrix} 2.5 & 0 & 0 \\ 5 & 2 & 0 \\ 5 & 2 & 1.5 \end{pmatrix} \begin{pmatrix} 1 & 0.8 & 0.8 \\ 0 & 1 & 0.5 \\ 0 & 0 & 1 \end{pmatrix}$$

The solution for the linear system:

$$\begin{pmatrix} 2.5 & 2 & 2 \\ 5 & 6 & 5 \\ 5 & 6 & 6.5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 2 \\ 2 \\ 2 \end{pmatrix} \quad \text{is} \quad \begin{pmatrix} 1.6 \\ -1 \\ 0 \end{pmatrix}.$$