MATH226 Project I
LU Factorization

Erdal Sidal Dogan
MEF University
#041702023

October 23, 2019

1 Project Definition & Goal

Main goal of the project is to design and implement a computer program which will solve a given system of linear equation by using *LU Decomposition* method. LU Decomposition, often referred as LU factorization, is a method where a given matrix A is represented as product of two other matrices L and U. L is Lower Triangular matrix where the main diagonal is all 1's and there are zeros only above the main diagonal. Likewise, U is the Upper Triangular Matrix where there are not any values except zero below the main diagonal.

$$A = LU \tag{1}$$

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{1,3} \\ a_{21} & a_{22} & a_{2,3} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix} \tag{2}$$

This factorization is called the *LU factorization* of A.

We also know that a linear systems of equations are defined as;

$$Ax = b \tag{3}$$

Given the LU factorization of the matrix A, we can solve the linear system (3) in two steps: substitute (1) into (3) to obtain

$$LU\mathbf{x} = \mathbf{b},$$

and then solve the triangular systems, in order,

$$L\mathbf{y} = \mathbf{b}, \tag{4}$$
$$U\mathbf{x} = \mathbf{y}. \tag{5}$$

Equations such as (4) and (5) can be solved with methods which are called *Forward Substitution* and *Backward Substitution*. Which are computationally cheap also.

1.1 Forward Substitution

For lower triangular system $L\mathbf{x} = \mathbf{b}$

$$x_1 = b_1/l_{11}, \quad x_i = \left(b_i - \sum_{j=1}^{i-1} l_{ij}x_i \right) / \, l_{ii}, \quad i = 2, \ldots, n \tag{6}$$

1.1.1 Psuedocode

1.2 Backward Substitution

For upper triangular system $U\mathbf{x} = \mathbf{y}$

$$x_n = b_n/u_{nn}, \quad x_i = \left(b_i - \sum_{j=i+1}^{n} u_{ij} x_j \right) / \, u_{ii}, \quad i = n-1, \ldots, 1 \tag{7}$$

LU Decomposition method is computationally cheaper for solving a system comparing to *Gaussian Elimination* especially on large-dimension matrices. *Java Language* is utilized for this project.

2 Implementation & Algorithm

Not every matrix has LU Decomposition. If there are any zeros at pivot locations of a matrix, its LU factorization cannot be found without making any changes on the original matrix. In order to address this problem, we use *row interchanges.* If there are any zeros in pivot positions in a matrix, the program will automatically use row interchanging method on it. This process of interchanging rows are called *Partial Pivoting.* When we use Partial Pivoting, we use an another matrix name Permutation Matrix(P) for row interchanges. P is basically the identity matrix which its rows are interchanged.

Of course, there are matrices that has *LU Fact.* as it is. For this matrices, we use an algorithm called *Doolittle*, for others, i.e. matrices with zeros in pivot positions; first we convert them to regular matrices by applying *Partial Pivoting*, then find the LU Fact. It should be mentioned that if Partial Pivoting is used on a matrix to get LU decomposition, product of L and U will be equal to product of Permutation Matrix(P) and A matrix, instead of A matrix only;

$$PA = LU \tag{8}$$

2.1 Doolittle Algorithm

Doolittle is a simple and straightforward algorithm to compute *Lower Triangular (L)* and *Upper Triangular(U)* matrix. If anyone writes down the A, L and U matrices and starts to compute values in the lower and upper triangular matrices can see the pattern of the solution. Calculation pattern for element of L and U matrices can be generalized as;

$$u_{ij} = a_{ij} - \sum_{k=1}^{i-1} u_{kj} l_{ik} \tag{9}$$

$$l_{ij} = \frac{1}{u_{jj}} \left(a_{ij} - \sum_{k=1}^{j-1} u_{kj} l_{ik} \right) \tag{10}$$