Khulna University Studies Volume 19(2): 141-153 : 2022

REVIEW ARTICLE SCIENCE & ENGINEERING

MATRIX FACTORIZATION, DECOMPOSITION AND SPLITTING METHODS AND ITS APPLICATIONS IN PHYSICAL PROBLEMS

Md. Azmol Huda*, Mohammad Wahiduzzaman and Munnujahan Ara

Mathematics Discipline, Khulna University, Khulna 9208, Bangladesh

KUS: 21/37:30112021

Manuscript submitted: November 30, 2021 Accepted: June 29, 2022

Abstract

Matrix factorization is the process that transforms a matrix into the product of some constituent matrices. This is comparable to factoring a number into the product of several numbers. Matrix splitting methods are similar to matrix factorization process which transforms a matrix into the sum of some basis matrices. In this short review article, we address the different types of matrix factorization and matrix splitting methods as well as their applications in the physical problems rather than exhibiting their computational procedure. Some matrix structural facts are shown to exhibit the fundamental pattern of different matrix decompositions.

Keywords: Matrix decomposition, matrix factorization, matrix splitting, LU decomposition, QR decomposition, singular value decomposition

Introduction

There are numerous matrix algorithms in linear algebra which are dependent on a variety of criteria to solve the problem. In computer programming storage capacity is crucial issue to preserving the efficiency of computational purposes. The amount of arithmetic operations required to complete the algorithm is also a key consideration (Faragó 2001 & Gilbert 2016). The algorithm complexity of a matrix can be reduced by converting it into the product or sum of upper triangular matrices, lower triangular matrices, diagonal matrices, permutation matrices, symmetric matrices, or banded matrices. Most of these structured matrices are used for matrix factorization and matrix splitting (Howard 2015; Hsu 2011).

Scientists and engineers have to handle a large system of linear equations when working on their physical ground. For example, in the solution of finite difference method of a partial differential equation there may arise a system of linear equations with large number of zeros and a banded structure matrix with non-zero entries. In a steady-state concentration problem, when a system of reactors, reservoirs or any type of tank with chemical where the concentration vary from each one tank to other may arise to a system of linear equations with large number of entries (Ke 2018; Lei-Hong 2014; Morse 1993). Numerical solution of this large system of linear equations in computational platform requires huge amount of memory. The naive Gaussian elimination method is not suitable to handle this type of problem perfectly as it is prohibitively expensive (Piziak 1999). The matrix iterative solution method is suitable for this large system of equations especially, for sparse system (Pedrag 2001; Varga 1960; Yifken 2021; Yuan 2018; Zbigniew 1998). The key purpose of this review work is to discuss to some matrix factorization and matrix splitting methods in a one shed for better understanding and be able to distinguish among them as well as in which domain of linear algebra they can be applied.

The paper is organized as follows: Introductory discussion is given in section 1. An overview of different matrix decompositions based on the solving the system of linear equations is given section 2. The matrix factorization methods based on eigenvalue and eigenvector are discussed in section 3. In section 4, various matrix splitting strategies for solving matrix iterative systems are presented. In section 5, the discussion part discusses some of the applications of matrix factorization and matrix splitting methods. Section 6 contains the conclusion.

*Corresponding author:<azmol@math.ku.ac.bd>
DOI: https://doi.org/10.53808/KUS.2022.19.02.2137-se

Decomposition based on solving the system of linear equations:
In this section, we describe the different matrix decomposition methods that are used to solving the system of linear equation and to finding the inverse of the matrix (Gilbert 2017; Howard 2015).

QR Decomposition

QR and LU decompositions are the two extensively used factorization methods in linear algebra among many others. The QR algorithm is credited to Francis and Kublanovskaya that was discovered independently in the late 1950s (Hsu 2011). The main idea of QR decomposition is to factoring a matrix $A \in R^{m\times n}$ of order $m\times n$, as a product of an orthogonal matrix and an invertible upper triangular matrix as

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

Where $Q \in R^{m\times n}$ is an orthogonal matrix and $R \in R^{m\times n}$ is an upper triangular matrix of order $m\times n$. The general form of QR decomposition can be written as

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} = \begin{bmatrix} q_{11} & q_{12} & \cdots & q_{1n} \\ q_{21} & q_{22} & \cdots & q_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ q_{m1} & q_{m2} & \cdots & q_{mn} \end{bmatrix} \cdot \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ 0 & r_{22} & \cdots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & r_{mn} \end{bmatrix} \tag{2}$$

An example of QR decomposition is

$$\begin{bmatrix} 3 & 4 & 1 \\ 2 & 1 & 5 \\ 1 & 7 & 6 \end{bmatrix} = \begin{bmatrix} -0.8018 & -0.0851 & -0.5915 \\ -0.5345 & -0.3405 & 0.7735 \\ -0.2673 & 0.9364 & 0.2275 \end{bmatrix} \begin{bmatrix} -3.7417 & -5.6125 & -5.0780 \\ 0 & 5.8737 & 3.8307 \\ 0 & 0 & 4.6412 \end{bmatrix}$$

LU Decomposition (Crout's method or triangularization method)

Alan Turing, an English mathematician, is credited for the *LU* decomposition. This mostly used factorization method is applied to solve the system of linear equation and finding the inverse of a matrix as an alternative to the Gauss-Jordan elimination method. The *LU* decomposition factorizes the coefficient matrix into a product of upper and lower triangular matrices, namely,

$$A = LU \tag{3}$$

where A is a square matrix of order $n\times n$, L is the lower triangular and U is the upper triangular matrix of order $n\times n$. The general form of LU decomposition can be expressed as

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & \cdots & 0 \\ l_{21} & l_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{bmatrix} \cdot \begin{bmatrix} 1 & u_{12} & \cdots & u_{1n} \\ 0 & 1 & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \tag{4}$$

The key feature of LU decomposition is that all the diagonal entries of the upper triangular matrix of U are 1. Another name of LU decomposition method is Crout's method or triangularization method (Gilbert 2017; Howard 2015; Hsu 2011).

Khulna University Studies, Volume 19(2): 141-153: 2022
DOI: https://doi.org/10.53808/KUS.2022.19.02.2137-se

A typical example of LU decomposition is

$$\begin{bmatrix} 3 & 4 & 1 \\ 2 & 1 & 5 \\ 1 & 7 & 6 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0.6667 & -0.2941 & 1 \\ 0.3333 & 1 & 0 \end{bmatrix} \begin{bmatrix} 3 & 4 & 1 \\ 0 & 5.6667 & 5.667 \\ 0 & 0 & 6 \end{bmatrix}$$

LDU *Decomposition (Doolittile method)*

In generally, the LU decomposition approach is not unique. For example, the Eq. (4) can be written as

$$A = LU = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ l_{21}/l_{11} & 1 & 0 & \cdots & 0 \\ l_{31}/l_{11} & l_{32}/l_{22} & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ l_{n1}/l_{11} & l_{n2}/l_{22} & l_{n2}/l_{33} & \cdots & 1 \end{bmatrix} \cdot \begin{bmatrix} l_{11} & 0 & \cdots & 0 \\ 0 & l_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & l_{nn} \end{bmatrix} \cdot \begin{bmatrix} 1 & u_{12} & \cdots & u_{1n} \\ 0 & 1 & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \tag{5}$$

Here, the diagonal entries are shifted from the left to the right factor, and all the diagonal entries in the lower triangular matrix has been transformed in 1. The factorization (5) is known as LDU decomposition or Doolittle method of the matrix A where L is a lower triangular matrix, in which all the diagonal entries are 1, D is a diagonal matrix, and U is an upper triangular matrix in which also all the entries on the main diagonal are 1. An example of LDU decomposition is

$$\begin{bmatrix} 3 & 4 & 1 \\ 2 & 1 & 5 \\ 1 & 7 & 6 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0.3333 & 1 & 0 \\ 0.6667 & 1.1176 & 1 \end{bmatrix} \begin{bmatrix} 3 & 0 & 0 \\ 0 & 5.6667 & 0 \\ 0 & 0 & -7.4118 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

PLU *Decomposition*

The presence of the LU decomposition of a matrix is not always assured. However, by introducing a permutation matrix Q and conducting the row interchange before performing the LU decomposition, it is possible to factorize the coefficient matrix A into the product of lower and upper triangular matrix. Multiplying the matrix A by a permutation matrix Q, the LU decomposition becomes

$$\begin{aligned} &QA = LU \\ &\text{or } A = Q^{-1}LU = PLU, \ \text{ where } P = Q^{-1} \end{aligned} \tag{6}$$

The process of factoring a matrix A in this way is called the PLU decomposition and a typical example is given by

$$\begin{bmatrix} 3 & 4 & 1 \\ 2 & 1 & 5 \\ 1 & 7 & 6 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0.3333 & 1 & 0 \\ 0.6667 & 1.1176 & 1 \end{bmatrix} \begin{bmatrix} 3 & 0 & 0 \\ 0 & 5.6667 & 0 \\ 0 & 0 & -7.4118 \end{bmatrix}$$

Choleski's Decomposition
The Choleski decomposition was devised for simply real symmetric matrices. This factorization procedures splits the matrix A into

$$A = LL^T \tag{7}$$

where L is the lower triangular matrix and L^T is the transpose of L. The general form of Choleski's decomposition can be written as

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & \cdots & 0 \\ l_{21} & l_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{bmatrix} \cdot \begin{bmatrix} l_{11} & l_{21} & \cdots & l_{n1} \\ 0 & l_{22} & \cdots & l_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & l_{nn} \end{bmatrix} \tag{8}$$

A numerical example of Choleski decomposition is

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 1 & 3 & 6 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Decomposition based on eigenvalue and eigenvector
Factorization by Diagonalization or Eigenvalue Decomposition
Before going to the discussion of main topics in this section, we pose the diagonalization problem. Two matrices A and B are said to be similar if

$$B = P^{-1}AP \tag{9}$$

If the matrix B is in simple form, e.g. diagonal matrix D such that $D = P^{-1}AP$, then we can ascertain many properties such as rank, nullity, determinant of A directly from the diagonal matrix D. For instance, if the diagonal entries of D are eigenvalues, then the product of these eigenvalues will be the determinant of A (Gilbert 2016).

Theorem 1: If A is an $n \times n$ square matrix and if A has n linearly independent eigenvectors, then A is diagonalizable i.e. $D = P^{-1}AP$ where P is the invertible matrix whose columns are the eigenvectors of A.
From the theorem 1, it can be concluded that

$$A = PDP^{-1} \tag{10}$$

Thus the matrix A can be factored with the aid of eigenvectors and an example is given below:

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 1 & 3 & 6 \end{bmatrix} = \begin{bmatrix} -0.5438 & -0.8165 & 0.1938 \\ 0.7812 & -0.4082 & 0.4722 \\ -0.3065 & 0.4082 & 0.8599 \end{bmatrix} \begin{bmatrix} 0.1270 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 7.8730 \end{bmatrix} \begin{bmatrix} -0.5438 & -0.8165 & 0.1938 \\ 0.7812 & -0.4082 & 0.4722 \\ -0.3065 & 0.4082 & 0.8599 \end{bmatrix}^{-1}$$

Factorizing by Orthogonal Diagonalization

In the earlier section, the matrix A was any square matrix and the columns of the matrix P were the eigenvectors. There was no concern about the orthogonality criteria of those eigenvectors. But, if we consider any special class of matrices e.g. the symmetric matrix then the column vectors of the matrix P will be orthogonal. Therefore, the matrix A can be factored as $A = PDP^{-1}$ where the columns of P are orthogonal. Since P is orthogonal, $P^{-1} = P^T$ and consequently

$$A = PDP^T \tag{11}$$

An example of this factorization is

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 1 & 3 & 6 \end{bmatrix} = \begin{bmatrix} -0.5438 & -0.8165 & 0.1938 \\ 0.7812 & -0.4082 & 0.4722 \\ -0.3065 & 0.4082 & 0.8599 \end{bmatrix} \begin{bmatrix} 0.1270 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 7.8730 \end{bmatrix} \begin{bmatrix} -0.5438 & -0.8165 & 0.1938 \\ 0.7812 & -0.4082 & 0.4722 \\ -0.3065 & 0.4082 & 0.8599 \end{bmatrix}^T$$

Spectral Decomposition

The spectral decomposition is an alternative form of orthogonal diagonalization. If A is symmetric matrix, then it can be orthogonally diagonalizable by the column matrix $P = [v_1, v_2, \cdots, v_n]$, where $v_1, v_2, \cdots, v_n$ are orthogonal eigenvectors. The orthogonal diagonalization $A = PDP^T$ can be written in an alternative form as ....

$$A = PDP^T = [v_1, v_2, \cdots, v_n] \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix} \begin{bmatrix} v_1^T \\ v_2^T \\ \vdots \\ v_n^T \end{bmatrix} = [\lambda_1 v_1, \lambda_2 v_2, \cdots, \lambda_n v_n] \begin{bmatrix} v_1^T \\ v_2^T \\ \vdots \\ v_n^T \end{bmatrix}$$

Now multiplying, the matrix A reduces to

$$A = \lambda_1 v_1 v_1^T + \lambda_2 v_2 v_2^T + \cdots + \lambda_n v_n v_n^T \tag{12}$$

This linear factorization is called the spectral decomposition of the matrix A and an numerical example is

$$\begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix} = 1.3820 \begin{bmatrix} -0.8507 \\ 0.5257 \end{bmatrix} \begin{bmatrix} -0.8507 & 0.5257 \end{bmatrix} + 3.6180 \begin{bmatrix} 0.5257 \\ -0.8507 \end{bmatrix} \begin{bmatrix} 0.5257 & -0.8507 \end{bmatrix}$$

Schur's Decomposition

In section 3.2, the orthogonal decomposition was $A = PDP^T$, where D is a diagonal matrix whose diagonal entries are the eigenvalues of the matrix A. Issai Schur, a German mathematician (Howard 2015 & Hsu 2011), proposed a matrix decomposition where the diagonal matrix D is replaced by an upper triangular matrix S like as

$$A = PSP^T \tag{13}$$

where

$$S = \begin{bmatrix} \lambda_1 & * & * & \cdots & * \\ 0 & \lambda_2 & * & \cdots & * \\ 0 & 0 & \lambda_3 & \cdots & * \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \lambda_n \end{bmatrix}$$

The Schur decomposition of the matrix $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 1 & 3 & 6 \end{bmatrix}$ is $\begin{bmatrix} 10.6978 & -0.4767 & 2.6436 \\ 0 & 2.7586 & 0.8857 \\ 0 & 0 & -3.4564 \end{bmatrix}$.

Hessenberg Decomposition

The Hessenberg decomposition

$$A = PHP^T \tag{14}$$

is similar to the Schur's decomposition where the upper triangular matrix S is replaced by the Hessenberg matrix H of the following form:

$$H = \begin{bmatrix} * & * & \cdots & * & * & * \\ * & * & \cdots & * & * & * \\ 0 & * & \cdots & * & * & * \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & * & * & * \\ 0 & 0 & \cdots & 0 & * & * \end{bmatrix}$$

The Hessenberg form of the matrix $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 1 & 3 & 6 \end{bmatrix}$ is $\begin{bmatrix} 3 & -4.0249 & -0.8944 \\ -2.2361 & 6.8 & -4.6 \\ 0 & -6.6 & 0.2 \end{bmatrix}$.

Singular Value Decomposition

It is observed that all the decompositions based on the eigenvalues and eigenvectors discussed in the earlier sections resulted from either square matrix or especial type symmetric matrix; no rectangular matrix was considered for decomposition. Singular value decomposition overcomes the barrier of decomposing a rectangular matrix of order $m \times n$. Herein, before going to the main topics, we first outline the most important definition; singular values of a matrix.

Singular values of an $m \times n$ matrix:

Let A be the matrix of order $m \times n$, then the matrix $A^T A$ is a symmetric matrix. Since $A^T A$ is symmetric matrix, it is orthogonally diagonalizable. Let $\lambda_1, \lambda_2, \ldots, \lambda_n$ be the eigenvalues of the symmetric matrix $A^T A$, then the square root of eigenvalues

$$\sigma_i = \sqrt{\lambda_i},\ 1 \le i \le n$$

are called the singular values of the matrix A.

Suppose A be a square matrix of order $m \times n$ and rank r. Then the factorization

$$A = U \sum V^T \tag{15}$$

where U is an $m \times m$ matrix, V is an $n \times n$ orthogonal matrix and $\sum$ is an $m \times n$ matrix where all the diagonal entries are the singular values of the matrix A. The decomposition (15) is called the singular value decomposition of matrix A. The general form of singular value decomposition can be expressed as

$$A = U \sum V^T = \begin{bmatrix} u_1 & u_2 & \cdots & u_k \mid u_{k+1} & \cdots & u_m \end{bmatrix} \left[\begin{array}{cccc|c} \sigma_1 & 0 & \cdots & 0 & \\ 0 & \sigma_2 & \cdots & 0 & 0_{k\times(n-k)} \\ \vdots & \vdots & \ddots & \vdots & \\ 0 & 0 & \cdots & \sigma_k & \\ \hline \multicolumn{4}{c|}{0_{(m-k)\times k}} & 0_{(m-k)\times(n-k)} \end{array}\right] \left[\begin{array}{c} v_1^T \\ v_2^T \\ \vdots \\ v_k^T \\ \hline \vdots \\ v_n^T \end{array}\right] \tag{16}$$

An example of singular value decomposition is

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} = \begin{bmatrix} -0.2298 & 0.8835 & 0.4082 \\ -0.5247 & 0.2408 & -0.8165 \\ -0.8196 & -0.4019 & 0.4082 \end{bmatrix} \begin{bmatrix} 9.5255 & 0 \\ 0 & 0.5143 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} -0.6196 & -0.7849 \\ -0.7849 & 0.6196 \end{bmatrix}^T$$

Reduced Singular Value Decomposition

In singular value decomposition, it is notable that the matrix $\sum$ contains a few numbers of zero rows and zero columns which is superfluous and does not carry any information. These extra rows and columns can be removed by block multiplication from the expression and finally zero blocks drop out, just keeping the mathematical expression as

$$A = \begin{bmatrix} u_1 & u_2 & \cdots & u_k \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_k \end{bmatrix} \begin{bmatrix} v_1^T \\ v_2^T \\ \vdots \\ v_k^T \end{bmatrix} \tag{17}$$

The reduced singular value decomposition of the above matrix is

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \approx \begin{bmatrix} -0.2298 & 0.8835 & 0.4082 \\ -0.5247 & 0.2408 & -0.8165 \end{bmatrix} \begin{bmatrix} 9.5255 & 0 \\ 0 & 0.5143 \end{bmatrix} \begin{bmatrix} -0.6196 & -0.7849 \\ -0.7849 & 0.6196 \end{bmatrix}^T$$

The factorization (17) is called the reduced singular value decomposition which is used for dimension reduction in multivariate data analysis.

Matrix Splitting

In this section, we give a brief idea of matrix splitting procedures. Different matrix iteration solution methods e.g., Jacobi iteration method, Gauss-Seidel iteration method, successive over relaxation method will not discuss herein as all of these methods are available in the most text books (Gilbert 2016; Howard 2015; Hsu 2011). Here, we only discuss the basic principle of matrix splitting methods. Moreover, where and how matrix splitting methods can be applied in the iteration solution of system of equations will be outlined (Gilbert 2016; Howard 2015) & (Zbigniew 1998, 2001).

Suppose, we have given a system of linear equations

$$Ax = k \tag{18}$$

where $A \in R^{n\times n}$ is real non-singular square matrix and $k \in R^n$ is a column vector. The key idea of the matrix splitting methods is to divide the matrix A into two or more parts like as

$$A = M - N \tag{19}$$

where M and N matrices are in simpler formation.

Regular Splitting

The matrix M is a non-singular matrix with a simple structure such as diagonal or upper triangular or lower triangular matrix and so on. If the matrix M is so, then the solution of the system of equations can be written as

$$Ax = Mx - Nx = k \qquad or \qquad Mx = Nx + k \tag{20}$$

Now suppose $x^0 \in R^n$, therefore the iteration algorithm process becomes

$$Mx^{(m+1)} = Nx^{(m)} + k, \ m = 0,1,2,\ldots \tag{21}$$

Equivalently,

$$x^{(m+1)} = (M^{-1}N)x^{(m)} + M^{-1}k, \ m = 0,1,2,\ldots \tag{22}$$

If the matrix A has a regular splitting, then the matrix $M^{-1}N$ has non-negative entries (Howard 2015). The Eq. (22), shows that at every stage of the iteration, the system of linear equations are being solved with matrix M . We outline the regular splitting method providing a concrete example in below:

Example 1: Suppose the system of equations is given by

$$6x + 2y + z = 8; \quad 2x + 4y + 2z = 6; \quad 3x + 2y + 5z = 4$$

The matrix form of this system is

$$\begin{bmatrix} 6 & 2 & 1 \\ 2 & 4 & 2 \\ 3 & 2 & 5 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 8 \\ 6 \\ 4 \end{bmatrix}$$

For the regular splitting case, choose

$$M = \begin{bmatrix} 6 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 5 \end{bmatrix}, \ N = \begin{bmatrix} 0 & -2 & -1 \\ -2 & 0 & -2 \\ -3 & -2 & 0 \end{bmatrix}$$

The iterated solution will be

$$x^{m+1} = \begin{bmatrix} 1/6 & 0 & 0 \\ 0 & 1/4 & 0 \\ 0 & 0 & 1/5 \end{bmatrix}.\begin{bmatrix} 0 & -2 & -1 \\ -2 & 0 & -2 \\ -3 & -2 & 0 \end{bmatrix} x^m + \begin{bmatrix} 1/6 & 0 & 0 \\ 0 & 1/4 & 0 \\ 0 & 0 & 1/5 \end{bmatrix}.\begin{bmatrix} 8 \\ 6 \\ 4 \end{bmatrix}; \ m = 0,1,2,\ldots \tag{23}$$

Jacobi Iteration Method

The Jacobi iteration method splits the coefficient matrix A of a linear system of equations into the summation of a lower triangular matrix L , an upper triangular matrix U and a diagonal matrix D in the following way,

$$(L+D+U)x=k \tag{24}$$

This splitting form is expressed in an iteration form $x^{m+1}=Cx^m+d$ by the following way

$$\begin{aligned} &Dx=(-L-U)x+b \\ &\text{or } x=D^{-1}(-L-U)x+D^{-1}b \end{aligned} \tag{25}$$

which implies that

$$x^{m+1}=Cx^m+d;\ \ m=0,1,2,\dots \tag{26}$$

Now consider the example 1, the Jacobi iteration method splits the coefficient matrix A as

$$L=\begin{bmatrix}0&0&0\\2&0&0\\3&2&0\end{bmatrix},\ U=\begin{bmatrix}0&2&1\\0&0&2\\0&0&0\end{bmatrix}\text{ and } D=\begin{bmatrix}6&0&0\\0&4&0\\0&0&5\end{bmatrix}$$

and the final iteration form goes to

$$x^{m+1}=\begin{bmatrix}1/6&0&0\\0&1/4&0\\0&0&1/5\end{bmatrix}.\left(\begin{bmatrix}0&0&0\\-2&0&0\\-3&-2&0\end{bmatrix}-\begin{bmatrix}0&2&1\\0&0&2\\0&0&0\end{bmatrix}\right)x^m+\begin{bmatrix}1/6&0&0\\0&1/4&0\\0&0&1/5\end{bmatrix}.\begin{bmatrix}8\\6\\4\end{bmatrix};\ m=0,1,2,\dots$$

Gauss-Seidel Iteration Method

The Gauss-Seidel iteration method follows the same splitting procedure of Jacobi iteration method but differs only forming the final iteration step $x^{m+1}=Cx^m+d$ as the following way [5]:

$$\begin{aligned} (D+L)x&=-Ux+b \\ x&=(D+L)^{-1}(-U)x+(D+L)^{-1}b \\ x^{m+1}&=Cx^m+d;\ \ m=0,1,2,\dots \end{aligned} \tag{27}$$

and the final step of iteration of example 1 goes to

$$x^{m+1}=\begin{bmatrix}1/6&0&0\\-1/12&1/4&0\\-1/15&-1/10&1/5\end{bmatrix}.\begin{bmatrix}0&2&1\\-2&0&2\\-3&-2&0\end{bmatrix}x^m+\begin{bmatrix}1/6&0&0\\-1/12&1/4&0\\-1/15&-1/10&1/5\end{bmatrix}.\begin{bmatrix}8\\6\\4\end{bmatrix};\ m=0,1,2,\dots$$

Successive Over Relaxation Method

The successive over relaxation method also follows the same splitting principle of Gauss-Seidel method with an extra constant number ω, which is called relaxation parameter and the splitting equation becomes (Hsu, 2011)

$$\omega(D+L)x=-\omega Ux+\omega k \tag{28}$$

Now adding both sides of Eq. (28) by $(1-\omega)Dx$, we get

$$(D-\omega L)x=((1-\omega)D-\omega U)x+\omega k$$

Solving this for x, we obtain

$$x=(D-\omega L)^{-1}((1-\omega)D+\omega U)x+\omega(D-\omega L)^{-1}k \tag{29}$$

Table1. Efficiency comparison of different decomposition method

Cost efficiency of matrix A of order $n\times n$, when n is very large	
Algorithm	Cost in Flops
Gauss-Jordan elimination method	$\approx(2/3)n^3$
QR decomposition method	$\approx n^2$
LU decomposition method	$\approx(2/3)n^3$
LDU decomposition method	$\approx n^2$

Discussion

Matrix factorization and splitting methods decomposes a complex form matrix into simpler constituent parts. In computer, to implement any matrix algorithm efficiently decomposition or splitting is essential. In this section, we outline some applications of QR, LU, LDU, PLU and Choleski's factorization method (Gilbert 2016; Howard 2015). Particularly all of these methods are applied to solve the system of linear equations. The naive Gaussian elimination method is suitable for small size of matrix but not appropriate if the system of equation is very large i.e. the coefficient matrix is very bulky. In any computational software packages, many simple algorithm arise computation complexities and the number of arithmetic operations and round of error increases in an exponential order. The factorization of a matrix into the lower triangular or upper triangular or diagonal form reduces the number of operations notably. The cost efficiency of the large system of equations by using different algorithms is given in the Table 1. Also in Table 2, the factorizations based on the solving system of linear equations are shown (Gilbert 2016; Howard 2015).

Table 2. Factorization based on the solving of system of linear equations

Factorization	Applicable to	Characteristics
$A=QR$	$m\times n$ matrix A	Q is an orthogonal matrix of order $m\times m$ R is an upper triangular matrix of order $m\times n$
$A=LU$	Square matrix A	L is a lower triangular matrix U is an upper triangular matrix with 1's on the diagonal
$A=LDU$	Square matrix A	L is a lower triangular matrix with 1's on the diagonal D is a diagonal matrix U is an upper triangular matrix with 1's on the diagonal
$A=PLU$	Square matrix A	$P=Q^{-1}$, where Q is an permutation matrix L is a lower triangular matrix U is an upper triangular matrix with 1's on the diagonal
Cholesky's $A=LL^T$	Symmetric matrix A	L is a lower triangular matrix L^T is the transpose of the lower triangular matrix L

Table 3. Factorization based on eigenvalues and eigenvectors

Decomposition	Applicable to	Characteristics
Eigen decomposition $A = PDP^{-1}$	Square matrix A	P is a matrix whose column vectors are eigenvectors of A D is a diagonal matrix whose diagonal entries are eigenvalues of A
Orthogonal decomposition $A = PDP^T$	Symmetric matrix A	P is a matrix whose column vectors are orthogonal eigenvectors of A D is a diagonal matrix whose diagonal entries are eigenvalues of A
Spectral decomposition $A = \lambda_1 v_1 v_1^T + \cdots + \lambda_n v_n v_n^T$	Symmetric matrix A	λ_i are the eigenvalues of A v_i are the eigenvectors of A and v_i^T are the transpose of v_i
Schur decomposition $A = PSP^T$	A is not symmetric but square and has real eigenvalues	P is a matrix whose column vectors are eigenvectors of A S is the upper triangular matrix which is in Schur form and eigenvalues of A along the main diagonal
Hessenberg decomposition $A = PHP^T$	A is not symmetric but square	P is the matrix same as Schur decomposition H is the triangular matrix of Hessenberg form
Singular value decomposition $A = U \sum V^T$	A is of order $m \times n$, need not to be square or symmetric	$U = [u_1, \cdots, u_m]$ is an orthonormal basis of the column space A $\sum$ is a diagonal matrix whose non-zero diagonal entries are singular values of A $V = [v_1, \cdots, v_n]$ orthogonally diagonalizes $A^T A$

Factorization based on eigenvectors also plays an important role in the application of identifying different conics, and optimization by using quadratic forms. Now a day in the technological environment, efficient transmission and data storage of massive amounts of digital information has become a serious issue. For example, a black-and-white image could be scanned and it preserves the data as a rectangular array of pixels (points) which is saved as matrix A by giving a numerical value to each pixel based on its gray level. A lot of extra information may be added to process this. The reduced singular value decompositions can be used to compress this visual data in order to reduce storage space requirements and speed up electron transmission. The decompositions based on eigenvalues and eigenvectors are given in Table 3.

Matrix iteration methods, Jacobi, Gauss-Seidel, and successive over relaxation methods are used when the system of equations is particularly large and sensitive in the coefficient matrix (Gilbert 2016; Howard 2015). All of these methods divide the coefficient matrix into diagonal, lower diagonal, and upper diagonal matrices that are carefully chosen. Table 4 shows the techniques for splitting.

Table 4. Matrix splitting for iteration solution

Regular splitting	$A = M - N$	$x^{(m+1)} = (M^{-1}N)x^{(m)} + M^{-1}k,\ m = 0,1,2,\ldots$
Jacobi splitting	$A = L + D + U$	$x^{m+1} = D^{-1}(-L-U)x^{m} + D^{-1}b;\ m = 0,1,2,\ldots$
Gaus-Seidel	$A = L + D + U$	$x^{m+1} = (D+L)^{-1}(-U)x^{m} + (D+L)^{-1}b;\ m = 0,1,2,\ldots$
Successive over relaxation	$A = L + D + U$ $\omega(D+L)x = -\omega Ux + \omega k$	$x = (D-\omega L)^{-1}((1-\omega)D + \omega U)x + \omega(D-\omega L)^{-1}k$

Conclusion

The theoretical utility of matrix decomposition has long been recognized, and it has become the backbone of numerical linear algebra. Matrix decompositions, matrix splitting, and the canonical forms are the most important ideas in matrices theory. All of these converted forms of original matrix simplify the computing platform, allowing a wide range of problems to be solved, including systems of linear equations, dimension reduction, and noise removing form digital image among others. This short overview article explains the logic behind matrix decompositions and matrix splitting methods. The applications of these matrix transformations as well as their characterizations are also outlined shortly in tables.

Acknowledgements

The authors would like to thanks to all the faculty members of mathematics discipline, Khulna University, Khulna.

Conflict of Interests

The author declares no conflict of interest.

References

Faragó I., Tarvainen P. (2001), Qualitative analysis of matrix splitting methods, Computers & Mathematics with Applications, Vol. 42, Issues 8–9, pp. 1055-1067.

Gilbert Strang (2016), Introduction to Linear Algebra, 5th edition, Wellesley-Cambridge Press, New York, ISBN: 978-09802327-7-6.

Howard Anton (2015), Chris Rorres, Elementary Linear Algebra: Applications Version, 10th edition, Wiley & Sons, New York.

Hsu D., Kakade S. M. and Zhang T. (2011), "Robust Matrix Decomposition With Sparse Corruptions," in IEEE Transactions on Information Theory, 57(11), pp. 7221-7234, doi: 10.1109/TIT.2011.2158250.

https://en.wikipedia.org/wiki/Matrix_decomposition.

https://en.wikipedia.org/wiki/Matrix_splitting.

https://machinelearningmastery.com/introduction-to-matrix-decompositions-for-machine-learning/.

Ke, YF., Ma, CF. & Zhang, H. (2018), The Modulus-based Matrix Splitting Iteration Methods for Second-Order Conde Linear Complementarity Problems, Numer. Algorithm 79, pp. 1283-1303.https://doi.org/10.1007/s11075-018-0484-4.

Lei-Hong Z., Wei Hong Y. (2014), An Efficient Matrix Splitting Method for the Second-Order Cone Complementarity Problem, SIAM J. Optim., 24(3), pp. 1178-1205.

Morse, A.S., (1993), A gain matrix decomposition and some of its applications, Systems & Control Letters, 21(1), pp. 1-10.

Piziak, R., Odell, P. L. (1999). Full Rank Factorization of Matrices, Mathematics Magazine. 72 (3): pp. 193-200. doi:10.2307/2690882. JSTOR 2690882.

Predrag S., Dragan D. (2001), A New Type of Matrix Splitting and its Applications, Acta Mathematica Academiae Scientiarum Hungaricae, 92(1), pp. 121-135. Doi: 10.1023/A: 1013712329516.

Stewart G.W. (1993), On the early history of singular value decomposition, SIAM Review, 35(4), pp. 551-566.

Uhlmann, J.K. (2018), A Generalized Matrix Inverse that is Consistent with Respect to Diagonal Transformations, SIAM Journal on Matrix Analysis and Applications, 239 (2), pp.781–800, doi:10.1137/17M113890X

Uhlmann, J.K. (2018), A Rank-Preserving Generalized Matrix Inverse for Consistency with Respect to Similarity, IEEE Control Systems Letters, arXiv:1804.07334, doi:10.1109/LCSYS.2018.2854240, ISSN 2475-1456

Khulna University Studies, Volume 19(2): 141-153: 2022
DOI: https://doi.org/10.53808/KUS.2022.19.02.2137-se

Varga, Richard S. (1960), Factorization and Normalized Iterative Methods, In Langer, Rudolph E. (ed.). Boundary Problems in Differential Equations. Madison: University of Wisconsin Press. pp. 121–142.

Varga, Richard S. (1962), Matrix Iterative Analysis, New Jersey: Prentice-Hall, LCCN 62-21277.

Yifen Ke, (2021), The Matrix Splitting Iteration Method for Nonlinear Complementarity Problems Associated with Second-Order Cone. Bulletin of the Iranian Mathematical Society, 47(1), pp. 31-53.

Yuan, G., Zheng, W. S., Shen, L., & Ghanem, B. (2018). A Generalized Matrix Splitting Algorithm. arXiv preprint arXiv:1806.03165.

Zbigniew I.W. (1998), Matrix Splitting Principles, Novi Sad J. Math., 28(3), pp. 197-209.

Zbigniew I.W. (2001), Matrix Splitting Principles, International Journal of Mathematics and Mathematical Sciences, 28(5), pp. 251-284.