INTERNATIONAL JOURNAL OF RESEARCH AND INNOVATION IN APPLIED SCIENCE (IJRIAS)
ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue X October 2025

# Matrix Representation of Graph and Digraph

[1]Dr. Rajesh Kumar Saini, [2]Dr. Pradeep Kumar, [3]Prem Chand Tiwari

[1]Pandit Deendayal Upadhyaya Shekhawati University, Sikar, India

[2]Shri Shraddhanath P.G. College, Gudha Gorji, Jhunjhunu, India

[3]Shri Shraddhanath P.G. College, Gudha Gorji,Jhunjhunu, India

**DOI: https://doi.org/10.51584/IJRIAS.2025.1010000054**

**Received: 20 October 2025; Accepted: 27 October 2025; Published: 03 November 2025**

## ABSTRACT

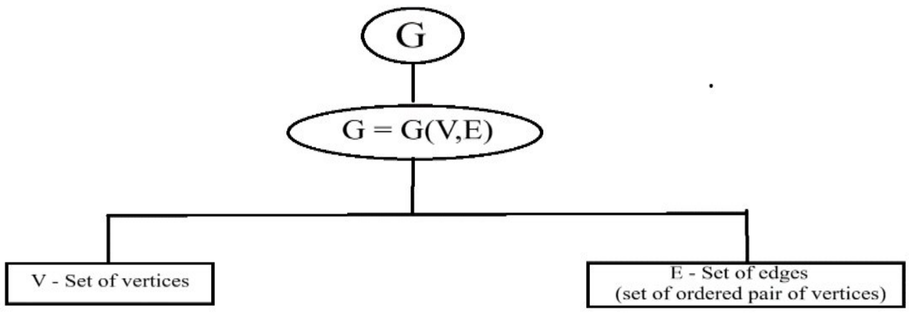
Graph theory plays a fundamental role in various fields of science and engineering, providing powerful tools for modeling and analyzing relationships among entities. One of the most effective ways to study graphs is through matrix representation. This paper explores the three primary matrix representations of graphs: the adjacency matrix**,** The adjacency matrix provides direct insight into vertex connectivity and the incidence matrix, the incidence matrix reflects the relationship between edges and vertices. and the Laplacian matrix defined as the difference between the degree matrix and the adjacency matrix, plays a central role in spectral graph theory. Matrix representations enable efficient storage, computation, and analysis of graphs using linear algebraic techniques. They form the basis for many modern algorithms in graph theory, This paper discusses the mathematical foundations, construction methods, and practical applications of these matrix forms, highlighting their essential role in both theoretical and applied graph analysis.

**Key words:** Graph Theory, Matrix Representation, Adjacency Matrix, Incidence Matrix, Laplacian Matrix, Graph Connectivity, Graph Structures, Graph Modeling.

## INTRODUCTION

Graph theory is a vital branch of discrete mathematics that focuses on the study of graphs—mathematical structures used to model pairwise relations between objects.

**Graph:** A graph G is a mathematical structure G = G(V,E) consisting of two sets one of V - vertices (nodes) and other set of E - edges that connect pairs of vertices.




One of the most effective ways to analyze and process graphs computationally is through matrix representation. By representing graphs in the form of matrices, we can leverage the powerful tools of linear algebra to perform complex computations, visualize relationships, and develop efficient algorithms for tasks such as searching, traversing, pathfinding, and clustering.

Graphs are versatile models representing objects as vertices and relationships as edges. Analyzing graph properties and dynamics often requires converting the graph into algebraic forms. Matrix representations enable this by encoding graph information into numerical matrices that can be processed using linear algebra techniques.

This research investigates three main matrix forms—the adjacency matrix, incidence matrix, and Laplacian matrix—exploring their construction, properties, and applications. We also consider the trade-offs involved in choosing a particular representation depending on graph type and analysis goals.

Graphs are ubiquitous structures in mathematics and computer science, used to model relationships and interactions between objects. A graph is formally defined as a set of vertices (nodes) connected by edges (links)**,** and it provides a natural and flexible way to represent systems such as communication networks, transportation systems, social media platforms, biological processes, and the internet.

As the complexity and scale of these systems grow, so does the need for efficient, scalable, and systematic methods to store, analyze, and manipulate graphs. One of the most powerful and widely adopted approaches for doing so is through matrix representations. By encoding graphs as matrices, we can apply a wide range of mathematical and algorithmic tools from linear algebra**,** numerical computation**,** and data science to perform advanced operations that would be difficult or inefficient in purely structural or list-based forms.

**Background and Related Work :**

Matrix-based graph representations date back to early graph theory studies. The adjacency matrix is the most intuitive and widely used form, while incidence matrices are essential in combinatorial optimization. The Laplacian matrix plays a pivotal role in spectral graph theory, which links graph structure to eigenvalue spectra. recent work has extended matrix representations into domains such as graph signal processing, graph neural networks, and large-scale network analysis, highlighting the continued relevance and adaptability of these tools.

There are several matrix representations of graphs, each offering unique insights and advantages:

**Matrix Representations of Graphs:**

Matrix representations not only simplify the storage and manipulation of graphs but also enable the application of well-established linear algebra techniques. They form the basis of many modern advancements, including graph neural networks**,** spectral clustering**,** and recommendation systems**.**

This paper explores the construction, properties, and applications of different matrix representations of graphs. By understanding these representations, we can gain deeper insights into the structure and dynamics of complex networks, and apply this knowledge to real-world problems across various disciplines.

There are several types of matrix representations used in graph theory, each suited to specific types of graphs and analytical goals:

These matrix forms not only aid in theoretical analysis but also underpin many practical algorithms. For example, Google's PageRank algorithm for ranking web pages is based on a modified adjacency matrix. Similarly, graph neural networks (GNNs), a rapidly growing field in machine learning, rely on matrix-based representations to learn patterns from graph-structured data.

Furthermore, matrix representations support scalable computation on large graphs using modern hardware and software. Sparse matrix libraries, parallel processing, and GPU acceleration can be leveraged to efficiently process massive networks, such as those found in social media analytics, recommendation systems, and bioinformatics.

Despite their utility, each matrix representation has trade-offs in terms of memory usage, computational efficiency, and applicability. Choosing the right representation often depends on the nature of the graph (directed vs. undirected, weighted vs. unweighted, sparse vs. dense) and the problem being solved.

This paper aims to provide a comprehensive study of the matrix representations of graphs, exploring their construction, mathematical properties, advantages, and limitations. Additionally, we examine their applications in both classical graph algorithms and modern computational fields. Through this exploration, we highlight how these algebraic representations bridge the gap between discrete structures and continuous mathematics, providing a unified framework for graph analysis and manipulation.

**Adjacency Matrix**: This is a square matrix used to represent the connections between vertices. Each entry indicates whether a pair of vertices is connected by an edge. For weighted graphs, the entries represent edge weights. This representation is especially useful for implementing graph algorithms and checking for the existence of specific edges.

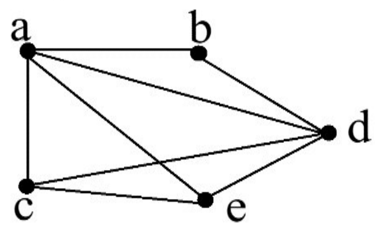
$A(G) = \left[\mathbf{a_{ij}}\right]_{\mathbf{n\times n}}$ = 1, if the number of edges directed from ith vertex to jth vertex

= 0, otherwise.

The Adjacency Matrix is one of the most straightforward and commonly used representations. It allows for rapid access to edge information and is especially effective in dense graphs or when performing repeated connectivity checks. However, for sparse graphs, its storage requirement can be large.

Since the graph is undirected, A[i][j]=A[j][i].

For a simple graph, diagonal entries A[i][i]=0 because loops are not allowed.




$$A(G) = \left[a_{ij}\right]_{n\times n} = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{pmatrix}$$

**Incidence Matrix**: This matrix captures the relationship between vertices and edges. Each row corresponds to a vertex, and each column corresponds to an edge. The entries indicate whether a vertex is incident to an edge. This form is particularly useful in network flow problems and electrical circuit modeling.

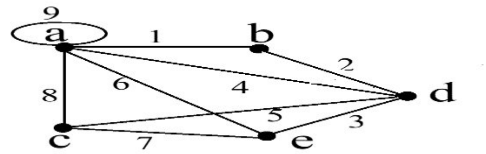
**Incident matrix for simple graph**

$I(G) = \left[\mathbf{a_{ij}}\right]_{\mathbf{n\times e}}$ = 1, if $e_j$ in incident on $v_i$

= 0, if $v_i$ is not an end of $e_j$.

= 2, if $e_j$ is self loop at $v_i$.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue X October 2025




$$
I(G) = \left[a_{ij}\right]_{n\times e} = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array}\begin{array}{c} \begin{array}{ccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 2 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} \end{array}
$$

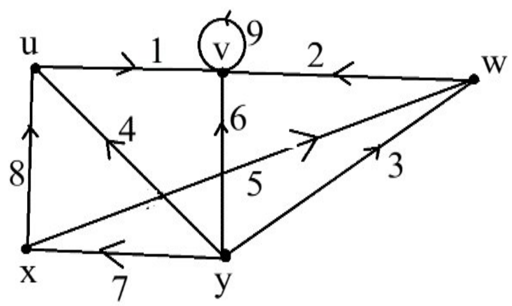
**Incident matrix for digraph**

$I(G) = \left[a_{ij}\right]_{n\times e}$ = 1, if $e_j$ is incident out of $v_i$

= - 1, if $e_j$ is incident into of $v_i$

= 0, if $v_i$ is not an end of $e_j$.

= 2, if $e_j$ is self loop at $v_i$.




$$
I(G) = \left[a_{ij}\right]_{n\times e} = \begin{array}{c} \\ u \\ v \\ w \\ x \\ y \end{array}\begin{array}{c} \begin{array}{ccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 0 \\ -1 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 2 \\ 0 & 1 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & -1 & 0 & 0 \end{bmatrix} \end{array}
$$

The

Incidence Matrix provides a vertex-edge relational view that is particularly useful in algorithmic applications involving flows, matching, or covering problems. It can also be extended to accommodate multi-graphs and directed graphs.

**Laplacian Matrix**: Defined as the difference between the degree matrix and the adjacency matrix, the Laplacian matrix plays a central role in spectral graph theory. It reveals important structural properties of the graph such as connectedness, community structure, and potential flow across the network. Its eigenvalues and eigenvectors are widely used in graph partitioning and clustering algorithms.

In the mathematical field of graph theory, the Laplacian matrix, also called the graph Laplacian, admittance matrix, Kirchhoff matrix, or discrete Laplacian, is a matrix representation of a graph. Named after Pierre-

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue X October 2025

Simon Laplace, the graph Laplacian matrix can be viewed as a matrix form of the negative discrete Laplace operator on a graph approximating the negative continuous Laplacian obtained by the finite difference method.

The Laplacian matrix relates to many functional graph properties. Kirchhoff's theorem can be used to calculate the number of spanning trees for a given graph. The sparsest cut of a graph can be approximated through the Fiedler vector — the eigenvector corresponding to the second smallest eigenvalue of the graph Laplacian — as established by Cheeger's inequality. The spectral decomposition of the Laplacian matrix allows the construction of low-dimensional embeddings that appear in many machine learning applications and determines a spectral layout in graph drawing. Graph-based signal processing is based on the graph Fourier transform that extends the traditional discrete Fourier transform by substituting the standard basis of complex sinusoids for eigenvectors of the Laplacian matrix of a graph corresponding to the signal.

The Laplacian matrix is the easiest to define for a simple graph but more common in applications for an edge-weighted graph, i.e., with weights on its edges — the entries of the graph adjacency matrix. Spectral graph theory relates properties of a graph to a spectrum, i.e., eigenvalues and eigenvectors of matrices associated with the graph, such as its adjacency matrix or Laplacian matrix. Imbalanced weights may undesirably affect the matrix spectrum, leading to the need of normalization — a column/row scaling of the matrix entries — resulting in normalized adjacency and Laplacian matrices.

Given a simple graph GG with n$n$ vertices v1,…,vn$v_1$, $v_2$, $v_3$,……. $v_n$, its Laplacian matrix Ln×nL$n \times n$ is defined element-wise as

- Li,j:={deg(vi)if i=j−1if i≠j and vi is adjacent to vj0otherwise,$L_{ij}$ =

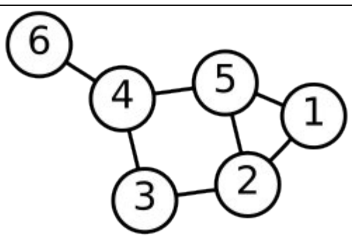
$$\begin{cases} deg(u_i), if\ i = j \\ -1,\ if\ \ i \neq j\ \ and\ u_i\ is\ adjacent\ to\ \ u_j \\ 0, \quad otherwise \end{cases}$$

or equivalently by the matrix

- L=D−A,L = D - A

where D is the degree matrix, and A is the graph's adjacency matrix. Since GG is a simple graph, AA only contains 1s or 0s and its diagonal elements are all 0s.

- Here is a simple example of a labelled, undirected graph and its Laplacian matrix.

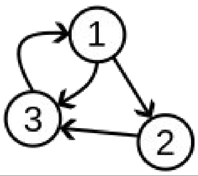
| **Labelled graph** | **D = Degree matrix** | **A = Adjacency matrix** |
|---|---|---|
|  | $\begin{pmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$ (200000030000002000000 3 0 0 0 0 0 0 3 0 0 0 0 0 0 1 ) ( ) | $\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$ |

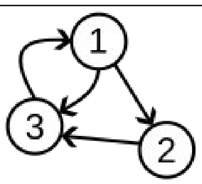
ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue X October 2025

$$\text{Laplacian matrix} = \begin{pmatrix} 2 & -1 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & -1 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & -1 \\ -1 & -1 & 0 & -1 & 3 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \end{pmatrix}$$

We observe for the undirected graph that both the adjacency matrix and the Laplacian matrix are symmetric and that the row- and column-sums of the Laplacian matrix are all zeros (which directly implies that the Laplacian matrix is singular).

For directed graphs, either the indegree or outdegree might be used, depending on the application, as in the following example:

| **Labelled graph** | **Adjacency matrix** | **Out-Degree matrix** | **Out-Degree Laplacian** |
|---|---|---|---|
|  | (011001100) $\begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$ | $\begin{pmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ (200010001) | (2−1−101−1−101) $\begin{pmatrix} 2 & -1 & -1 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \end{pmatrix}$ |

| **Labelled graph** | **Adjacency matrix** | **in-Degree matrix** | **in-Degree Laplacian** |
|---|---|---|---|
|  | (011001100) $\begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$ | $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{pmatrix}$ (200010001) | (2−1−101−1−101) $\begin{pmatrix} 1 & -1 & -1 \\ 0 & 1 & -1 \\ -1 & 0 & 2 \end{pmatrix}$ |

In the directed graph, the adjacency matrix and Laplacian matrix are asymmetric. In its Laplacian matrix, column-sums or row-sums are zero, depending on whether the indegree or outdegree has been used.

The Laplacian Matrix, often used in spectral graph theory, captures essential structural properties of the graph. Its eigenvalues and eigenvectors reveal key characteristics such as the number of connected components, graph connectivity, and potential partitions of the network. It forms the basis for techniques like spectral clustering and graph signal processing**.**

**Applications**

Spectral Clustering: Using Laplacian eigenvectors to identify clusters in data.

Network Analysis: Measuring centrality, community detection.

Machine Learning: Graph neural networks rely heavily on adjacency and Laplacian matrices for feature propagation.

Electrical Engineering**:** Incidence matrices help model circuits and network flows

**Challenges and Future Directions:**

Scalability: Handling large sparse graphs efficiently.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue X October 2025

Dynamic Graphs: Adapting matrix representations as graphs evolve.

Approximation Techniques: For computationally intensive eigenvalue problems.

Integration with Machine Learning**:** Enhancing interpretability and performance.

## CONCLUSION

Matrix representations provide a powerful bridge between discrete graph structures and continuous linear algebra tools. Their rich mathematical properties facilitate a wide range of applications from classical graph algorithms to modern data science. Continued research on optimizing and extending these representations promises further advances in graph analysis.

## REFERENCES

1. Chung, F. R. K. (1997). Spectral Graph Theory. American Mathematical Society.
2. Diestel, R. (2017). Graph Theory (5th ed.). Springer.
3. Newman, M. E. J. (2010). Networks: An Introduction. Oxford University Press.
4. Kipf, T. N., & Welling, M. (2017). Semi-Supervised Classification with Graph Convolutional Networks. ICLR.