جمهورية العراق
وزارة التعليم العالي و البحث العلمي
جامعة بابل / كلية التربية للعلوم الصرفة
قسم الرياضيات

# Matrices in Graph theory

Graduation research Project submitted to the Council of
the Department of Mathematics/ University of Babylon as
part of the requirements for obtaining a Bachelor’s Degree

**إعداد الطالب**

**علاء توفيق وهيم**

**باشراف**

**م.م. إحسان عبد الرحمن**

٢٠٢٦م

١٤٤٧هـ

# شكر وتقدير

الحمد لله الذي وفقني لإتمام هذا البحث، والشكر له أولاً وآخراً.

أتقدم بخالص الشكر والامتنان إلى **عمادة كلية التربية للعلوم الصرفة** وإلى أساتذتي في **قسم الرياضيات** على دعمهم العلمي طوال سنوات الدراسة.

كما أخص بالشكر الجزيل أستاذي الفاضل ومشرفي **إحسان عبد الرحمن** لجهوده الكبيرة وتوجيهاته القيمة التي كانت خير معين لي في إنجاز هذا العمل.

والشكر موصول إلى **عائلتي الغالية** على دعمهم اللامحدود وصبرهم الدائم، وإلى كل من ساعدني بكلمة أو تشجيع للوصول إلى هذه اللحظة.

# إهداء

إلى من كلّلهما الله بالوقار، وألبسهما ثوب الطهر والعفاف.. إلى نبض قلبي وسرّ وجودي، **والديّ العزيزين**؛ تقديراً لتضحياتهما التي لا تُحصى، وحبّهما الذي لا ينضب، وإيمانهما الدائم بي الذي كان الدافع الأكبر لتحقيق هذا النجاح.

إلى **عائلتي الغالية**، سندي ومصدر قوتي، من شاركوني لحظات التعب قبل الفرح، وكان تشجيعهم الدائم لي بمثابة الوقود الذي يدفعني للاستمرار والمواصلة في أصعب الظروف.

إلى **أساتذتي الأفاضل**، ورثة الأنبياء ومنارات العلم، الذين لم يبخلوا عليّ يوماً بعلمٍ أو نصيحة، وأناروا دربي بفيض حكمتهم وصبرهم، فلهم مني كل الود والتقدير.

إلى كل من ألهم رغبتي في التعلم، وشحذ همتي نحو التميز والإبداع، وإلى كل روحٍ مرّت في حياتي وتركت أثراً طيباً دفعني لأكون ما أنا عليه اليوم..

**أهديكم ثمرة جهدي هذا.**

# Abstract

Graph theory is a fundamental area of discrete mathematics with extensive applications in computer science, network analysis, and engineering. One of the most powerful approaches to studying graphs is through algebraic graph theory, which translates structural properties of graphs into matrix operations. This research provides a comprehensive study of various matrix representations specifically for undirected graphs. The study systematically defines and explores six essential matrices: the Adjacency Matrix, Incidence Matrix, Cycle Matrix, Path Matrix, Degree Matrix, and the Laplacian Matrix. For each of these matrices, the research examines its mathematical formulation, core properties, and structural implications. By analyzing and comparing these algebraic tools, this project highlights how they facilitate the extraction of crucial graph characteristics, such as connectivity, node relationships, cycle detection, and overall network topology. Ultimately, this research serves as a foundational guide demonstrating the effectiveness of matrix theory in modeling, analyzing, and solving complex problems within undirected networks.

## • Introduction

Graph Theory has gradually established itself as one of the most influential and versatile branches of modern mathematics, not only because of its theoretical elegance but also due to its remarkable ability to describe relationships that exist in real-world systems. At its most basic level, a graph consists of a collection of vertices (or nodes) connected by edges, forming a structure that may appear simple at first glance. Yet, this simplicity often conceals a deep capacity for representing highly complex interactions, whether they occur in communication networks, transportation systems, social connections, or even within biological and computational environments. What makes graph theory particularly compelling is this balance between simplicity and expressive strength, allowing researchers to model reality in a way that is both structured and meaningful.

However, as graphs grow in size and complexity, relying solely on visual representation or descriptive analysis becomes increasingly insufficient. It is no longer practical to track relationships or extract patterns through diagrams alone, especially when dealing with large-scale networks. At this point, the need for a more systematic and computationally efficient approach becomes evident. This is where matrix representations emerge not merely as a convenience, but as a necessity. By translating graphs into matrices, we effectively convert structural relationships into numerical forms that can be manipulated, analyzed, and interpreted using the tools of linear algebra. This shift from visual intuition to algebraic formulation marks a significant step in deepening our understanding of graph structures.

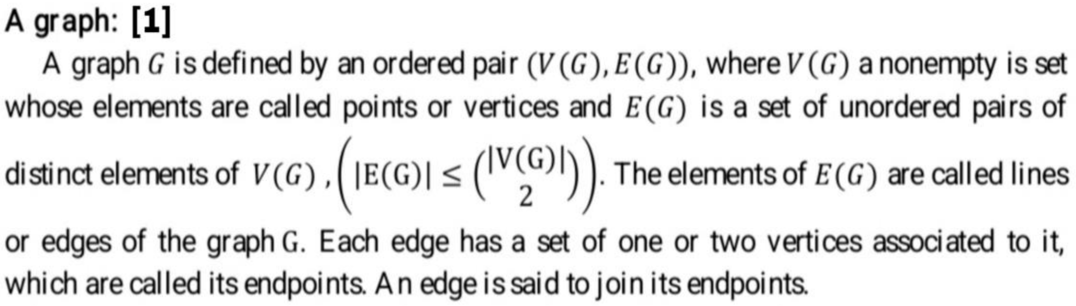
From a computational perspective, matrices offer a level of precision and efficiency that is essential in modern applications. Many fundamental problems—such as determining the shortest path between nodes, identifying cycles, or analyzing connectivity—can be addressed more effectively when graphs are expressed in matrix form. The ability to perform operations algorithmically on matrices makes them particularly valuable in computer science and engineering, where large datasets and complex networks are common. In this sense, matrices do not simply represent graphs; they enable their practical use in solving real-world problems.

It is also important to recognize that there is no single matrix representation that captures all aspects of a graph. Instead, multiple forms exist, each highlighting different structural properties. Some matrices emphasize direct connections between vertices, while others reveal deeper characteristics such as flow, balance, or resistance within the network. Understanding these distinctions is not merely a technical requirement, but a conceptual one, as it allows the researcher to select the most appropriate representation depending on the problem at hand.

Furthermore, matrix representations create a meaningful bridge between discrete mathematics and continuous methods. Through this connection, advanced concepts such as eigenvalues and eigenvectors can be applied to graphs, opening the door to more sophisticated areas like spectral graph theory. These approaches have proven to be highly relevant in contemporary fields such as data science, machine learning, and network analysis, where uncovering hidden patterns and structures is of central importance.

In this research, the focus will be placed on exploring these matrix representations in a gradual and structured manner. Beginning with fundamental concepts such as the adjacency matrix, the discussion will extend toward more advanced forms, including the Laplacian matrix and related constructs. The aim is not only to present definitions and properties, but also to develop an intuitive and practical understanding of how these mathematical tools can be used to analyze and interpret graph-based systems.

# Basic Definitions



**A graph: [1]**

A graph $G$ is defined by an ordered pair $(V(G), E(G))$, where $V(G)$ a nonempty is set whose elements are called points or vertices and $E(G)$ is a set of unordered pairs of distinct elements of $V(G)$, $\left(|E(G)| \leq \binom{|V(G)|}{2}\right)$. The elements of $E(G)$ are called lines or edges of the graph G. Each edge has a set of one or two vertices associated to it, which are called its endpoints. An edge is said to join its endpoints.

**2. Loop :[3]** A loop is an edge that connects a vertex to itself.

**3. Multiple Edges :[3]** Multiple edges are two or more edges that connect the same pair of vertices.

**4. Simple Graph :[3]** A simple graph is a graph that has no loops and no multiple edges between any pair of vertices.

**5. Pseudograph :[3]** A non-simple graph is a graph that may contain loops and/or multiple edges between the same pair of vertices.

**6. Symmetry :[2]** A matrix is symmetric if it is equal to its transpose, meaning $a_{ij} = a_{ji}$ for all i and j.

**7. Size :[3]** The size of a graph is the number of edges it contains, usually denoted by |E|.

**8. Order :[3]** The order of a graph is the number of vertices it contains, usually denoted by |V|.

**9. Matrix :[2]** A matrix is a rectangular arrangement of numbers or elements organized in rows and columns.

**10. Main Diagonal :[2]** The main diagonal of a matrix consists of elements where the row index equals the column index ($a_{ii}$).

**11. Off-Diagonal Elements:[2]** Off-diagonal elements are all elements in a matrix that are not on the main diagonal ($a_{ij}$ where $i \neq j$).

**12. Isolated Vertex :[3]** An isolated vertex is a vertex that has no edges connected to it (its degree is zero).

**13. [1]"The degree** of a vertex v in a graph G, denoted by \deg(v), is defined as the number of edges incident with that vertex. It represents the local connectivity of a vertex within the graph structure."

# 1. Adjacency Matrix (A) [4]

The adjacency matrix is one of the most important and widely used representations in graph theory because it provides a clear numerical description of how vertices are connected within a graph. Suppose we have a graph $G = (V, E)$ consisting of a set of vertices $V$ and a set of edges $E$. If the graph contains $n$ vertices, then the adjacency matrix $A$ is defined as a square matrix of size $n \times n$.

Each element in the matrix is denoted by $a_{ij}$, where the indices $i$ and $j$ represent vertices $v_i$ and $v_j$. The value of each entry is defined as follows:

- $a_{ij} = 1$ if there is an edge between $v_i$ and $v_j$
- $a_{ij} = 0$ if there is no edge between $v_i$ and $v_j$

This representation makes it very easy to determine whether two vertices are connected. It also allows graphs to be stored in a structured way that is suitable for computer processing and algorithm design.

## Properties:

1. In undirected graphs, since edges have no direction, the connection between vertices is mutual, so:

- $a_{ij} = a_{ji}$ (the matrix is symmetric)

2. For simple graphs (graphs without loops), no vertex is connected to itself, so:

- $a_{ii} = 0$ for all $i$

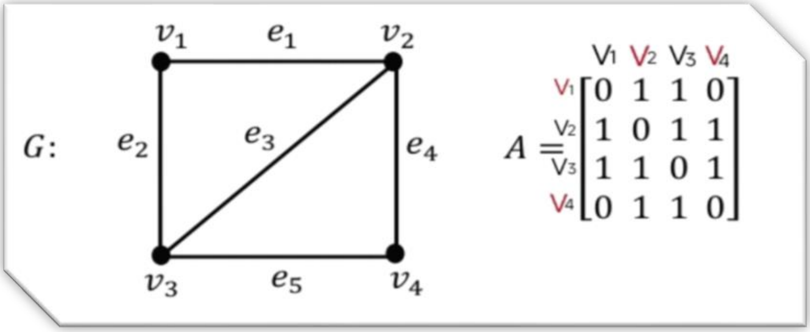
3. In graphs that allow loops or multiple edges:

- $a_{ij}$ can be greater than 1 if there are multiple edges between two vertices
- $a_{ii}$ will be nonzero if a vertex has a loop

The adjacency matrix is very useful in many applications, especially in computer science. It is used in algorithms for graph traversal, shortest path problems, and network analysis. Also, powers of the adjacency matrix can be used to count the number of paths between vertices. Despite its advantages, one limitation is that it requires $n^2$ storage space, which may not be efficient for very large graphs. Nevertheless, it remains a fundamental concept in graph theory.

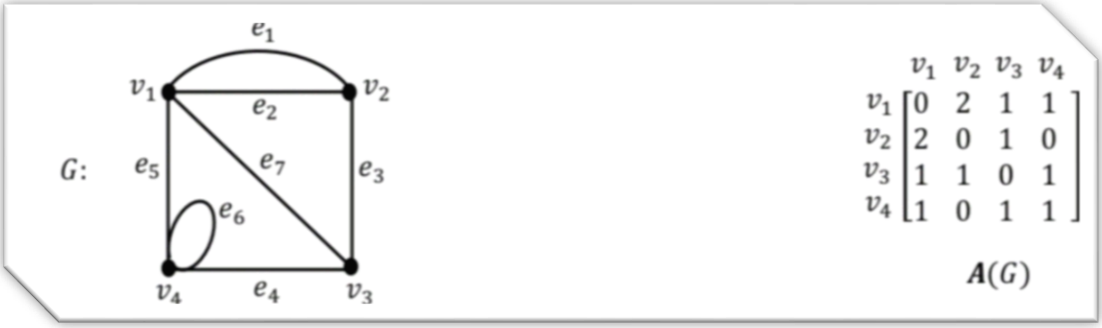
## Examples

### Example 1: Simple Graph



### Example 2: Graph with Multiple Edges and Loop

Adjacency Matrix:



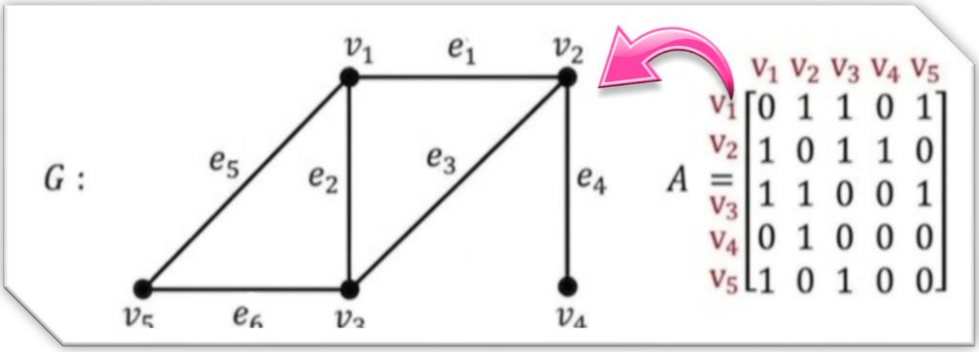
### Example 3: Example: Converting an Adjacency Matrix to a Graph

The matrix A' represents a simple undirected graph with 5 vertices.

Since it is a simple graph, the main diagonal consists of zeros ($a_{ii} = 0$), which indicates that there are no loops in the graph. This means no vertex is connected to itself.

The matrix is symmetric ($A = A^T$), which confirms that the edges have no specific direction. In other words, if there is a connection between $v_i$ and $v_j$, then the same connection exists between $v_j$ and $v_i$.

Each entry with value 1 in the matrix represents a direct edge between two vertices, while each entry with value 0 indicates that there is no connection between those vertices.



### Example 4:

The provided figure illustrates graph G and its corresponding adjacency matrix A. This graph consists of four vertices $\{v_1, v_2, v_3, v_4\}$ and includes multiple edges as well as a loop.

Multiple Edges:

Vertices $v_1$ and $v_2$ are connected by two edges ($e_1$, $e_2$), and vertices $v_1$ and $v_4$ are also connected by two edges ($e_5$, $e_6$). This is reflected in the adjacency matrix by the value 2 at the positions corresponding to ($v_1$, $v_2$) and ($v_1$, $v_4$), indicating more than one edge between these vertices.

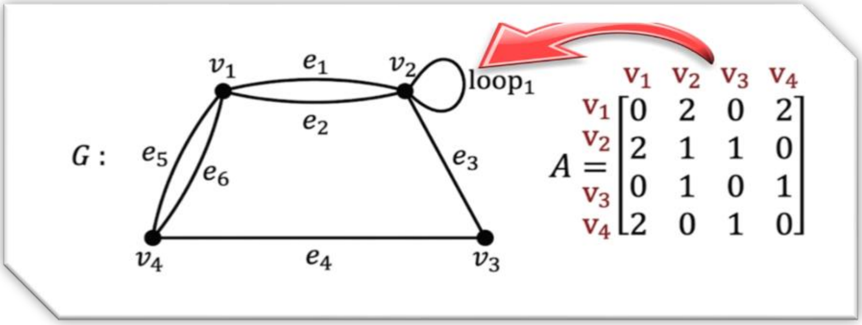
Self-Loops:

There is a loop at vertex $v_2$, which is represented in the matrix by the value 1 at the diagonal entry ($v_2$, $v_2$). This indicates that the vertex is connected to itself.

Symmetry:

Since the graph is undirected, the adjacency matrix is symmetric.

This means that $a_{ij} = a_{ji}$ for all i and j, reflecting that the relationship between vertices is mutual and has no direction.



## 2. Incidence Matrix (B) [5]

The incidence matrix is another fundamental representation in graph theory that focuses on the relationship between vertices and edges rather than connections between vertices directly. Given a graph G = (V, E) with n vertices and m edges, the incidence matrix B is defined as a rectangular matrix of size n × m. Each row corresponds to a vertex, and each column corresponds to an edge.

Each element of the matrix is denoted by $b_{ij}$, where i represents a vertex $v_i$ and j represents an edge $e_j$. The value of $b_{ij}$ is defined as follows:

- $b_{ij} = 1$ if vertex $v_i$ is incident (connected) to edge $e_j$
- $b_{ij} = 0$ otherwise

This means that each column of the matrix describes which vertices are connected by a particular edge. In the case of a simple undirected graph, each edge connects exactly two distinct vertices. Therefore, each column in the incidence matrix contains exactly two entries equal to 1, while all other entries are 0.

### Properties:

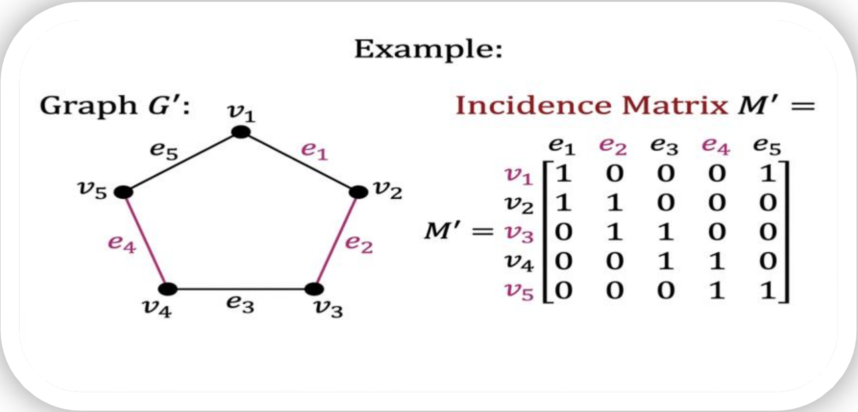
1. The incidence matrix provides a clear representation of how edges are distributed across vertices.
2. Unlike the adjacency matrix, which focuses on vertex-to-vertex relationships, the incidence matrix emphasizes vertex-to-edge relationships.
3. In graphs that include loops, a column corresponding to a loop may contain a single nonzero entry (depending on the convention used).
4. In some definitions, a loop is represented by placing a value of 2 in the corresponding row, since the edge is incident to the same vertex twice.
5. The incidence matrix can represent a wide range of graph types.
6. It is useful in algebraic graph theory for constructing other important matrices, such as the Laplacian matrix.
7. It provides insight into properties like connectivity and can be used to analyze the structure of complex networks

One important aspect of the incidence matrix is that it makes it especially useful in applications such as network flow problems, electrical circuit analysis, and combinatorial optimization.

## Examples

**Example 1:** The Graph (G'): A simple pentagon-shaped graph with 5 vertices ($v_1$ to $v_5$) and 5 edges ($e_1$ to $e_2$).

It contains no loops because every edge connects two distinct vertices.



$$M' = \begin{array}{c|ccccc} & e_1 & e_2 & e_3 & e_4 & e_5 \\ \hline v_1 & 1 & 0 & 0 & 0 & 1 \\ v_2 & 1 & 1 & 0 & 0 & 0 \\ v_3 & 0 & 1 & 1 & 0 & 0 \\ v_4 & 0 & 0 & 1 & 1 & 0 \\ v_5 & 0 & 0 & 0 & 1 & 1 \end{array}$$

### Example 2: Graph Including a Loop (Incidence Matrix)

Loop ($e_5$):

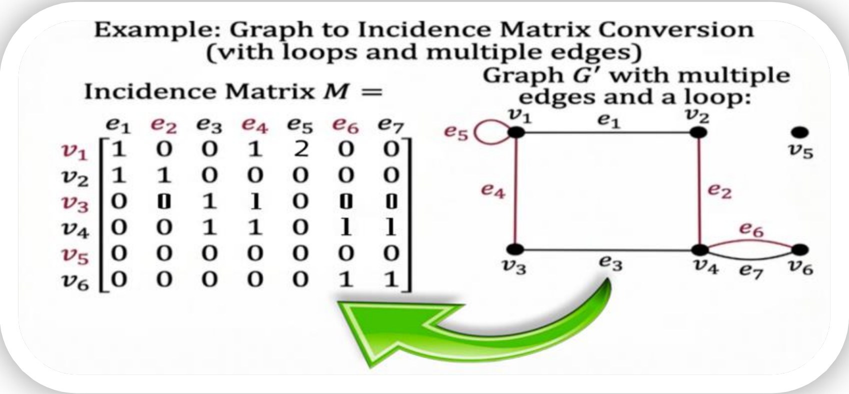
A self-loop is represented in the matrix by the value 2 in the row corresponding to vertex $v_1$. This indicates that the edge both starts and ends at the same vertex, so it is counted twice in the incidence representation.

Multiple Edges ($e_6$, $e_7$):

These edges are represented by identical columns in the matrix, showing that both edges connect the same pair of vertices, namely $v_4$ and $v_6$. This reflects the presence of more than one edge between the same two vertices.

Isolated Vertex ($v_5$):

This vertex is represented by a row consisting entirely of zeros, indicating that it is not incident to any edge in the graph. In other words, $v_5$ has no connections with any other vertex.



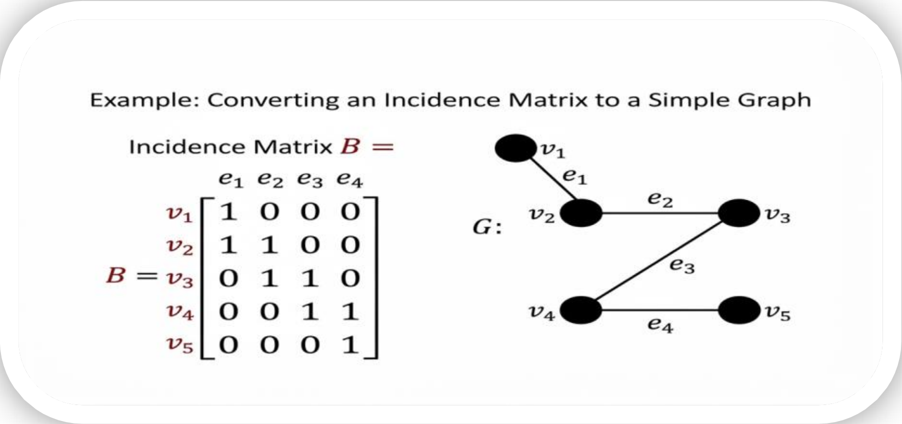
## Example 3: Matrix to Graph: Converting an Incidence Matrix to a Simple Graph

This example demonstrates the mapping of an incidence matrix into a simple graph.

In this matrix, each column represents an edge ($e_n$) and contains exactly two entries equal to 1. These two values indicate the two vertices ($v_n$) that the edge connects.

The absence of any row containing the value 2 confirms that there are no loops in the graph. Additionally, the absence of any column containing only a single value of 1 indicates that every edge connects exactly two distinct vertices.

These properties confirm that the graph is simple, meaning it contains no loops and no irregular edge structures.

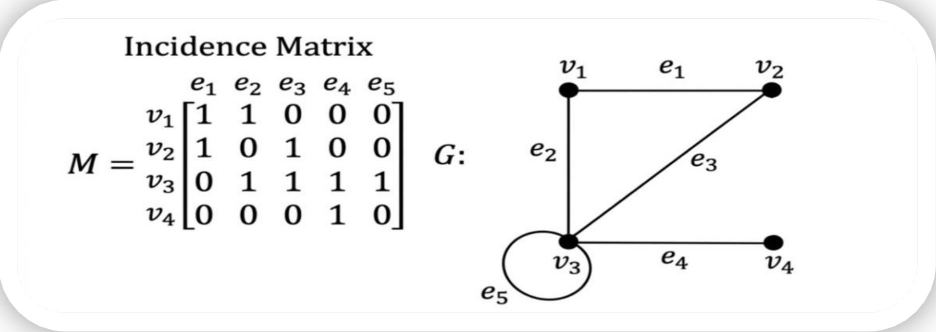


### Example 4: Graph to Matrix

This example illustrates the conversion of an incidence matrix M into a graph G. In this representation, each column corresponds to an edge, while each row corresponds to a vertex.

The Loop Representation ($e_5$):

In an incidence matrix, a self-loop can be identified when a column contains a single value of 1 (or sometimes the value 2 depending on the convention) in one row, while all other entries in that column are zero.



Incidence Matrix

$$M = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{ccccc} e_1 & e_2 & e_3 & e_4 & e_5 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

## 3. Cycle Matrix (B(G)) [6]

The cycle matrix is an important representation in graph theory that focuses on cycles within a graph rather than just connections between vertices or edges. A cycle in a graph is a closed path in which no vertices (except the starting and ending vertex) are repeated. The cycle matrix provides a structured way to represent all such cycles in a graph using a matrix form.

Given a graph G with m edges and q independent cycles, the cycle matrix B(G) is defined as a matrix of size q × m. Each row corresponds to a cycle, and each column corresponds to an edge in the graph. Each entry in the matrix is denoted by $b_{ij}$, where i represents the i-th cycle and j represents the j-th edge.

The values of the matrix are defined as follows:

- $b_{ij} = 1$ if the j-th edge is part of the i-th cycle
- $b_{ij} = 0$ if the j-th edge is not part of the i-th cycle

This means that each row describes which edges are included in a particular cycle.

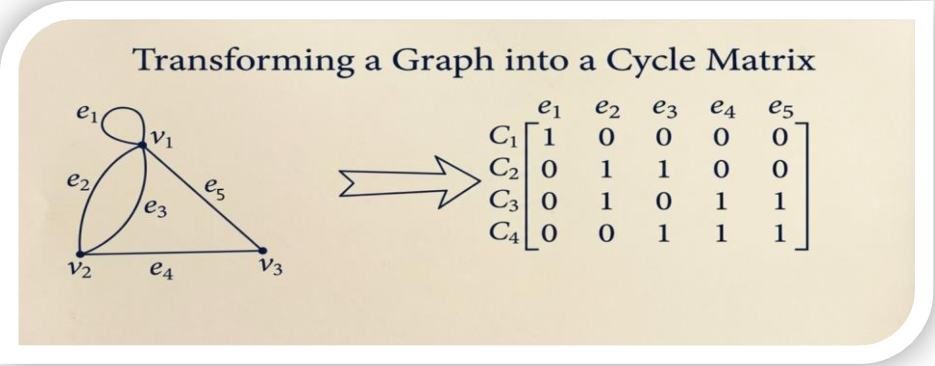
## Properties:

1. The cycle matrix is especially useful in studying the internal structure ofgraphs, particularly in identifying independent cycles and understanding how they are formed.
2. If a column consists entirely of zeros, then the corresponding edge does not belong to any cycle.
3. Such edges are typically called bridges or cut-edges, meaning their removal would increase the number of connected components of the graph.
4. The cycle matrix helps distinguish between edges that are part of cycles and those that are not.
5. It is widely used in applications such as electrical network analysis, where cycles correspond to loops in circuits.
6. It is also useful in topology and combinatorics for studying graph connectivity and structure.
7. The number of independent cycles in a graph is related to its structure and can be determined using known formulas involving the number of vertices and edges.

# Examples

### Example 1:

The figure illustrates the transformation of a graph into its corresponding cycle matrix. In this representation, the rows correspond to the graph's cycles (C), and the columns represent its edges (e). A matrix entry is 1 if the edge belongs to the specific cycle, and 0 otherwise. This matrix is essential for analyzing the structural properties of the graph."

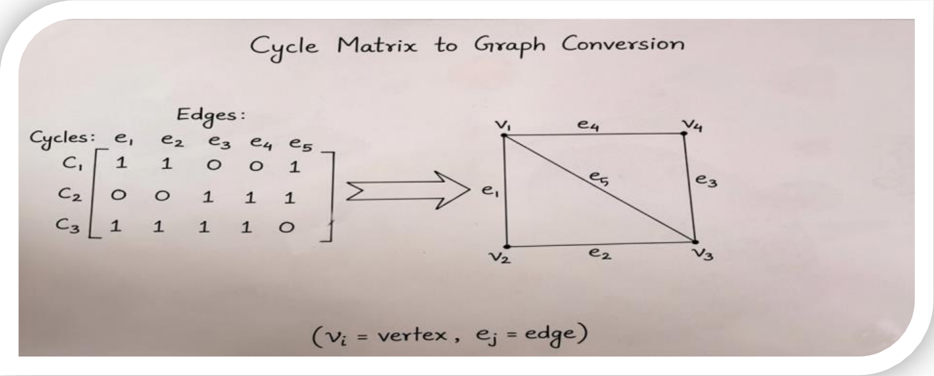


Transforming a Graph into a Cycle Matrix

$$\begin{array}{c} \\ C_1 \\ C_2 \\ C_3 \\ C_4 \end{array}\begin{array}{c} \begin{array}{ccccc} e_1 & e_2 & e_3 & e_4 & e_5 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix} \end{array}$$

Example 2:

This figure demonstrates the mapping between a cycle matrix and its corresponding undirected graph. The rows of the matrix represent the fundamental cycles ($C_1$, $C_2$, $C_3$), while the columns correspond to the graph's edges ($e_1$ to $e_5$). A binary value of 1 indicates the presence of a specific edge within a given cycle, whereas a 0 indicates its absence. This conversion highlights the direct relationship between algebraic graph representations and their geometric visual structures.



Cycle Matrix to Graph Conversion

$$\begin{array}{c|ccccc} \text{Cycles:} & e_1 & e_2 & e_3 & e_4 & e_5 \\ \hline C_1 & 1 & 1 & 0 & 0 & 1 \\ C_2 & 0 & 0 & 1 & 1 & 1 \\ C_3 & 1 & 1 & 1 & 1 & 0 \end{array}$$

($v_i$ = vertex , $e_j$ = edge)

## 4. Path Matrix (P(u, v)) [7]

The path matrix is a specialized matrix in graph theory that focuses on representing all possible paths between two specific vertices in a graph. Unlike other matrices such as the adjacency or incidence matrix, which describe global relationships in the graph, the path matrix is defined with respect to a particular pair of vertices, usually denoted as u and v. This makes it a more focused and analytical tool for studying connectivity and path structures between selected nodes.

Given a graph G = (V, E), suppose we are interested in all possible paths between two vertices u and v. Let there be k distinct paths between these two vertices. The path matrix P(u, v) is then defined as a matrix of size k × m, where m is the number of edges in the graph. Each row corresponds to a specific path from u to v, and each column corresponds to an edge in the graph.

Each entry in the matrix is denoted by $p_{ij}$, where:

- i represents the i-th path between u and v
- j represents the j-th edge in the graph
- The value of each entry is defined as:
- $p_{ij} = 1$ if the edge $e_j$ is included in the i-th path
- $p_{ij} = 0$ otherwise

This means that each row describes which edges are used in a particular path from u to v. By examining the matrix, we can easily compare different paths and identify common or distinct edges among them.

## Properties:

1. If a column in the matrix contains only ones, the corresponding edge is present in every possible path between u and v.
2. Such an edge is critical because its removal would disconnect u from v (similar to bridges or cut edges).
3. The path matrix depends on the choice of vertices u and v, so different vertex pairs produce different path matrices.
4. It is particularly useful in network analysis, reliability studies, and optimization problems.
5. It can identify critical connections in communication networks to ensure connectivity between nodes.
6. It is also useful in transportation and routing problems, where multiple paths between two points need to be analyzed and compared.
7. The matrix allows easy comparison of different paths to identify common or distinct edges.

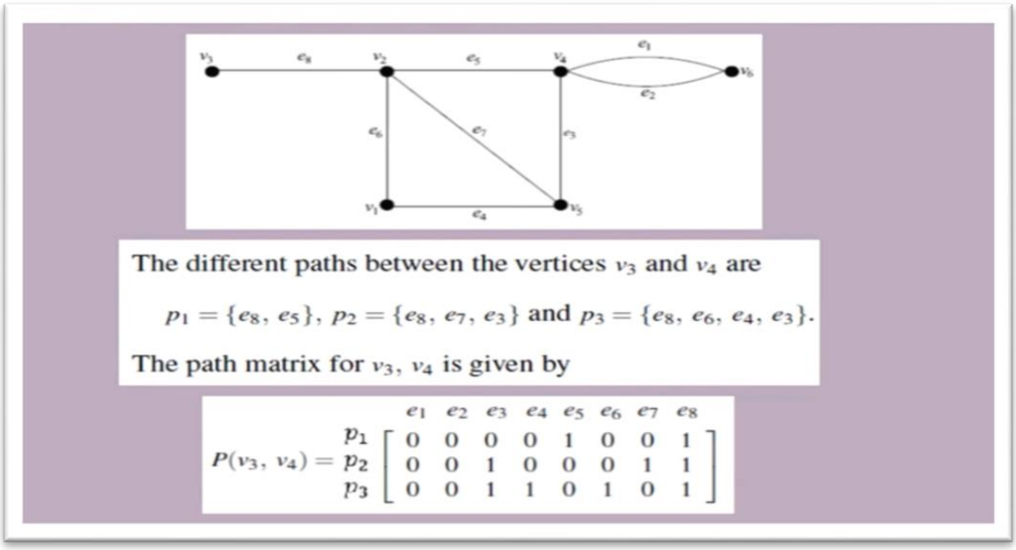
Unlike global matrices, the path matrix is flexible and powerful for localized analysis within a graph.

### • Example : Construction of an Edge-Path Matrix

The figure illustrates the method of representing paths between two specific vertices in an undirected graph using a Path Matrix.

In this example, the objective is to identify all simple paths between vertex $v_3$ and vertex $v_4$.

Three distinct paths were identified: $p_1$, $p_2$, and $p_3$, each consisting of a specific set of edges.



The different paths between the vertices $v_3$ and $v_4$ are

$p_1 = \{e_8, e_5\}$, $p_2 = \{e_8, e_7, e_3\}$ and $p_3 = \{e_8, e_6, e_4, e_3\}$.

The path matrix for $v_3$, $v_4$ is given by

$$P(v_3, v_4) = \begin{array}{c} \\ p_1 \\ p_2 \\ p_3 \end{array} \begin{array}{c} \begin{array}{cccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \end{array}$$

## 5. Degree Matrix (D) [8]

The degree matrix is a fundamental concept in graph theory that captures the degree of each vertex in a graph in a structured matrix form. The degree of a vertex is defined as the number of edges incident to it. For a graph G = (V, E) with n vertices, the degree matrix D is an n × n diagonal matrix, meaning all off-diagonal elements are zero. Each diagonal element $d_{ii}$ corresponds to the degree of vertex $v_i$.

Formally, the entries of the degree matrix are defined as:

- $D_{ii} = \deg(v_i)$, where $\deg(v_i)$ represents the degree of vertex $v_i$
- $D_{ij} = 0$ for all $i \neq j$

This structure provides a compact representation of vertex degrees, which is crucial for many computations in graph theory, particularly when combined with other matrices like the adjacency matrix to form the Laplacian matrix.

### Properties:

1. The sum of all diagonal entries equals twice the number of edges in the graph: $\sum d_{ii} = 2|E|$
2. This is consistent with the handshaking lemma, which states that the sum of the degrees of all vertices in a graph is twice the total number of edges

3. The degree matrix plays an essential role in spectral graph theory, where it is used in eigenvalue computations to study connectivity, graph partitions, and network robustness

In practice, the degree matrix is very simple to construct, especially when the adjacency matrix is known. For undirected graphs without loops, the degree of a vertex $v_i$ can be computed as the sum of the entries in the corresponding row (or column) of the adjacency matrix. In graphs with loops, each loop contributes 2 to the degree of the corresponding vertex. This allows the degree matrix to accurately reflect the connectivity of vertices in any type of graph.

The degree matrix is widely used in combination with the adjacency matrix to form other important matrices that reveal deeper graph properties. For example, the Laplacian matrix L, which is fundamental in network analysis, is defined as $L = D – A$. The diagonal form of the degree matrix ensures that the resulting Laplacian captures both vertex connectivity and edge structure simultaneously.

## Examples

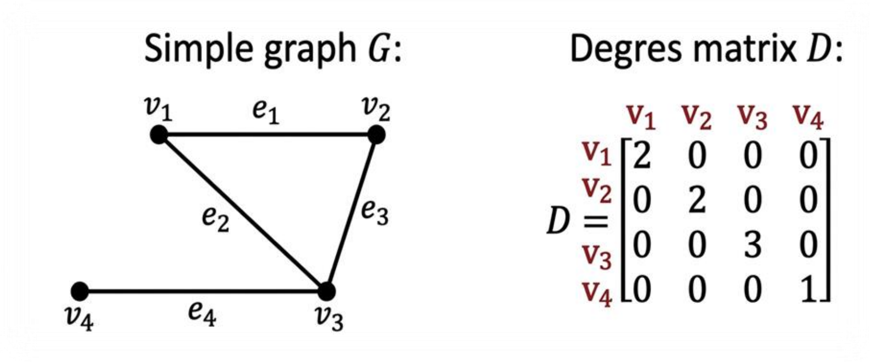
### Example 1:

Degree Matrix of a Simple Graph

This figure represents a simple undirected graph G without loops or multiple edges. It illustrates the construction of the Degree Matrix D, where:

•The diagonal elements represent the degree of each vertex (the number of incident edges).

•All other elements are zeros.



## Example 2:

Degree Matrix for Graphs with Loops and Multiple Edges

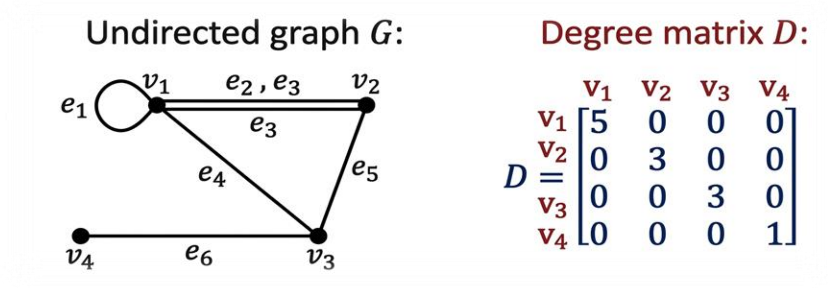
This example illustrates the construction of the Degree Matrix D for an undirected graph G that includes loops and multiple edges. In graph theory, the degree of a vertex is determined by the number of edges incident to it.

Key rules applied in this example:

Loops: In an undirected graph, a loop (an edge connecting a vertex to itself) is counted as two towards the degree of that vertex.

Multiple Edges: Each edge between the same pair of vertices is counted individually.

Matrix Structure: The Degree Matrix D is a diagonal matrix where each diagonal entry $d_{\{ii\}}$ represents the degree of vertex $v_i$, while all off-diagonal entries are zero.



## Example 3: Graph to Matrix

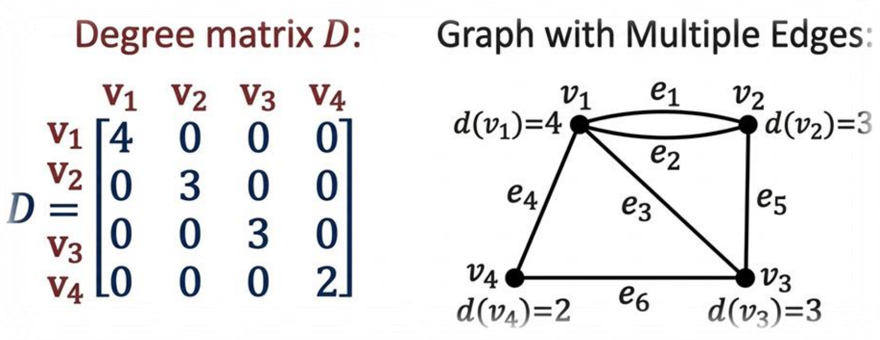
The figure illustrates the degree matrix D for a graph containing multiple edges. In graph theory, the degree matrix is a diagonal matrix that contains the degree of each vertex, meaning the number of edges incident to it.

As shown in the graph, there are four vertices ($v_1$ to $v_4$):

Vertex $v_1$ has a degree of 4 because it is connected to $v_2$ by two edges ($e_1$, $e_2$), to $v_3$ by $e_3$, and to $v_4$ by $e_4$.

The other vertices follow the same logic, where $d(v_2) = 3$, $d(v_3) = 3$, and $d(v_4) = 2$.

In the matrix D, these values are placed along the main diagonal as $d_{ii}$, while all off-diagonal elements are equal to zero. This matrix is essential for calculating the Laplacian matrix, where L = D − A, which is a fundamental tool in algebraic graph theory and network analysis.



## 6. Laplacian Matrix (L = D − A) [9]

The Laplacian matrix is one of the most important matrices in graph theory, providing a bridge between algebraic methods and structural properties of graphs. For a graph G = (V, E) with n vertices, the Laplacian matrix L is defined as the difference between the degree matrix D and the adjacency matrix A:

- L = D − A

Here, D is the diagonal degree matrix, where each diagonal element $d_{ii}$ represents the degree of vertex $v_i$, and A is the adjacency matrix representing the connections between vertices. The Laplacian matrix captures both the degree of each vertex and the connectivity between vertices in a single structure, making it a powerful tool for analyzing graphs.

Each entry of L can be described as follows:

- $l_{ii} = \deg(v_i)$, the degree of vertex $v_i$
- $l_{ij} = -1$ if vertices $v_i$ and $v_j$ are adjacent
- $l_{ij} = 0$ if vertices $v_i$ and $v_j$ are not adjacent and $i \neq j$

## Properties:

1. The sum of each row and each column of L equals zero.
2. The Laplacian matrix reveals fundamental graph characteristics such as connectivity, the number of spanning trees, and the behavior of network flows.
3. The smallest eigenvalue of L is always zero, and its multiplicity corresponds to the number of connected components in the graph.
4. The second smallest eigenvalue, called the algebraic connectivity, indicates how well connected the graph is.
5. Using Kirchhoff's Matrix-Tree Theorem, the determinant of a reduced Laplacian matrix gives the total number of spanning trees.
6. L is a crucial tool in combinatorial optimization, network reliability, and electrical network analysis.
7. By combining information about degrees and adjacency in a single matrix, the Laplacian allows for efficient computational analysis.
8. It forms the foundation for many algorithms in graph theory, physics, and computer science.

# Examples

## Example 1:

The example shows a simple undirected graph G with four vertices $v_1$, $v_2$, $v_3$, $v_4$.

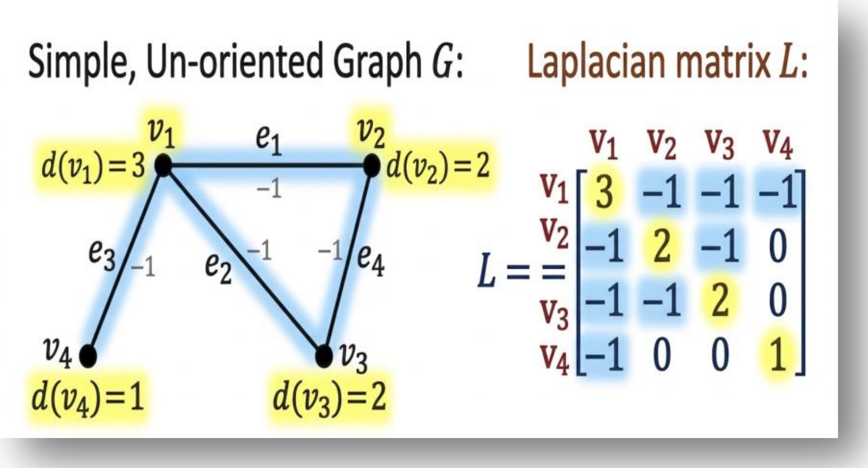
The degrees are $d(v_1) = 3$, $d(v_2) = 2$, $d(v_3) = 2$, and $d(v_4) = 1$.

The graph has four edges:

$e_1 = (v_1, v_2)$, $e_2 = (v_1, v_3)$, $e_3 = (v_1, v_4)$, and $e_4 = (v_2, v_3)$.

Its Laplacian matrix L is a 4 × 4 matrix where each diagonal entry equals the degree of the corresponding vertex, and each off-diagonal entry is −1 if the vertices are adjacent, and 0 otherwise.

The matrix shown satisfies $L \cdot 1 = 0$, where 1 is a vector of ones, and it is symmetric positive semidefinite, which are standard properties of a Laplacian matrix.



$$L = = \begin{array}{c|cccc} & v_1 & v_2 & v_3 & v_4 \\ \hline v_1 & 3 & -1 & -1 & -1 \\ v_2 & -1 & 2 & -1 & 0 \\ v_3 & -1 & -1 & 2 & 0 \\ v_4 & -1 & 0 & 0 & 1 \end{array}$$

## Example 2:

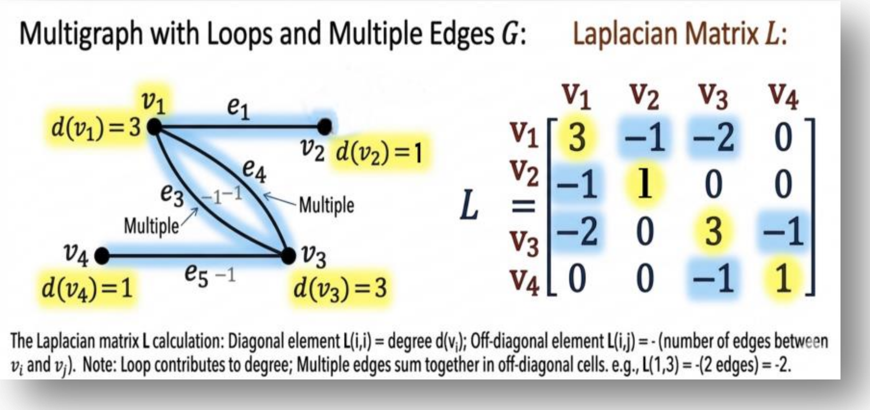
This example presents a multigraph G with multiple edges.

The graph has four vertices, where two parallel edges connect $v_1$ and $v_3$, while the remaining edges are simple.

The degree of each vertex is computed by counting each parallel edge individually.

The Laplacian matrix L is then constructed as follows: each diagonal entry L(i,i) equals the degree of vertex $v_i$, and each off-diagonal entry L(i,j) equals the negative of the number of edges between $v_i$ and $v_j$.

This illustration clearly shows how the Laplacian matrix extends naturally to multigraphs without relying on loops.



$$L = \begin{array}{c|cccc} & v_1 & v_2 & v_3 & v_4 \\ \hline v_1 & 3 & -1 & -2 & 0 \\ v_2 & -1 & 1 & 0 & 0 \\ v_3 & -2 & 0 & 3 & -1 \\ v_4 & 0 & 0 & -1 & 1 \end{array}$$

The Laplacian matrix L calculation: Diagonal element L(i,i) = degree d($v_i$); Off-diagonal element L(i,j) = - (number of edges between $v_i$ and $v_j$). Note: Loop contributes to degree; Multiple edges sum together in off-diagonal cells. e.g., L(1,3) = -(2 edges) = -2.

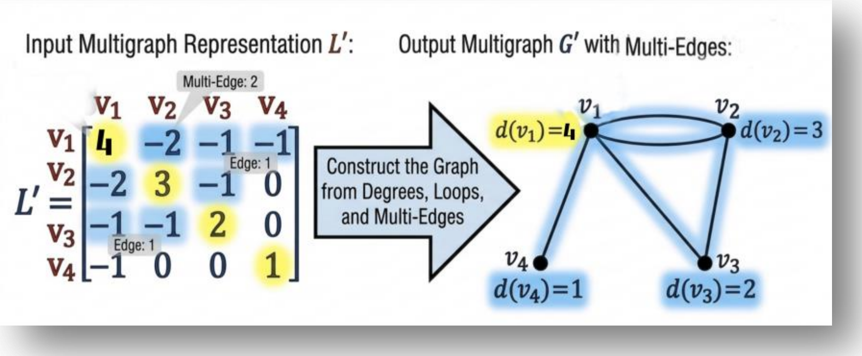
### Example 3: Graph to Matrix

This figure demonstrates the construction of a multigraph G' from its representation matrix L.

The diagonal elements (highlighted in yellow) represent the degree of each vertex $d(v_i)$, which is the total number of edges connected to that vertex.

The off-diagonal elements (highlighted in blue) indicate the negative count of edges between two distinct vertices. For example, the value −2 between $v_1$ and $v_2$ represents a multi-edge (two parallel edges connecting the same vertices).

This example effectively shows how complex graph properties, such as loops and multiple connections, are encoded within a matrix structure.



## • Conclusion

graph theory provides a powerful and flexible mathematical framework for analyzing relationships and structures across a wide range of systems. The use of matrix representations transforms abstract graphs into structured numerical forms that can be easily processed, analyzed, and applied in both theoretical and practical contexts. Through matrices such as the adjacency matrix, incidence matrix, cycle matrix, path matrix, degree matrix, and Laplacian matrix, complex relationships between vertices and edges can be expressed in a clear and systematic way.

These matrix forms not only simplify the representation of graphs but also enable deeper analysis of their properties. Concepts such as connectivity, paths, cycles, and vertex degrees become more accessible when expressed through matrices. In particular, the Laplacian matrix plays a central role in understanding graph structure, revealing important information about connectivity, spanning trees, and network behavior. Similarly, specialized matrices like the path and cycle matrices provide insight into specific structural features, allowing for more focused analysis.

Beyond theoretical importance, these representations have significant practical applications. They are widely used in computer science, network design, electrical engineering, transportation systems, and data analysis. By converting graphs into matrices, many real-world problems can be approached using linear algebra techniques, making it possible to develop efficient algorithms and computational solutions.

Overall, matrix representations of graphs serve as a bridge between discrete mathematics and applied sciences. They provide both clarity and computational power, making them essential tools for understanding and solving complex problems involving networks and interconnected systems.

## • References

**[1].** Bondy, J. A., & Murty, U. S. R. (2008). *Graph Theory*. Springer Science & Business Media.

**[2].** H. Anton and C. Rorres, *Elementary Linear Algebra: Applications Version*, 11th ed. Hoboken, NJ, USA: Wiley, 2013.

**[3].** K. H. Rosen, *Discrete Mathematics and Its Applications*, 8th ed. New York, NY, USA: McGraw-Hill Education, 2018.

**[4].**Douglas West, Introduction to Graph Theory, 2001, 2nd edition, pp. 15–30

**[5].**Gary Chartrand, A First Course in Graph Theory, 2012, 1st edition, pp. 40–55

**[6].**Norman Biggs, Algebraic Graph Theory, 1993, 2nd edition, pp. 10–35

**[7].** Jonathan Gross, Graph Theory and Its Applications, 2013, 2nd edition, pp. 70–95

**[8].** Reinhard Diestel, Graph Theory, 2017, 5th edition, pp. 8–25

**[9].** Fan Chung, Spectral Graph Theory, 1997, pp. 1–20