

# *Fiedler's Theory of Spectral Graph Partitioning*

Ayushman Anupam (MDS202411)  
Biswajit Kala (MDS202412)  
Boda Surya Venkata Jyothi Sowmya (MDS202413)

*Supervisor:* Kavita Sutar Deshpande

Data Science  
Chennai Mathematical Institute

# Contents

<b>1</b>	<b>Abstract</b>	<b>3</b>
<b>2</b>	<b>Acknowledgement</b>	<b>4</b>
<b>3</b>	<b>Introduction</b>	<b>5</b>
3.1	Problem Statement . . . . .	5
3.2	Solution Approach . . . . .	6
<b>4</b>	<b>Graph Theory Fundamentals</b>	<b>7</b>
4.1	Definition of a Graph . . . . .	7
4.2	Graph Matrix Examples . . . . .	8
4.3	Eigenvalues and Eigenvectors of the Laplacian Matrix . . . . .	8
<b>5</b>	<b>Fiedler's Spectral Graph Partitioning Method</b>	<b>9</b>
5.1	Procedure . . . . .	9
5.2	Example 1: Six-Node Graph . . . . .	9
5.3	Example 2: Three-Node Path Graph . . . . .	10
<b>6</b>	<b>Standing Wave Analogy for Spectral Partitioning</b>	<b>11</b>
<b>7</b>	<b>Spectral Graph Partitioning: Mathematical Proof</b>	<b>14</b>
<b>8</b>	<b>Summary of what we did and Next Steps</b>	<b>16</b>
<b>9</b>	<b>Mathematical Preliminaries and Theoretical Foundations</b>	<b>17</b>
9.1	Key Definitions . . . . .	17
9.2	Remarks . . . . .	17
9.3	Lemmas . . . . .	17
9.4	Theorems . . . . .	19
<b>10</b>	<b>Fiedler's Theorem of Spectral Graph Partitioning</b>	<b>21</b>
<b>11</b>	<b>Implementation : Examples</b>	<b>24</b>
<b>12</b>	<b>Limitations of Spectral Partitioning</b>	<b>28</b>
<b>13</b>	<b>Further Work After Fiedler's Decomposition</b>	<b>30</b>
<b>14</b>	<b>Conclusion</b>	<b>31</b>
<b>15</b>	<b>References</b>	<b>32</b>

# 1 Abstract

The growing prevalence of graph-structured data in modern applications poses unique challenges in terms of scalability, storage, and analysis. Unlike tabular data, graphs require partitioning strategies that retain structural relationships and connectivity.

This report explores **Fiedler’s Spectral Graph Partitioning** as a principled approach to segmenting large graphs without compromising their intrinsic structure. Rooted in spectral graph theory, the method leverages the Laplacian matrix and its eigenvectors—particularly the Fiedler vector—to identify natural partitions.

Key contributions of this work include:

- A rigorous mathematical treatment of graph Laplacians and their spectral characteristics
- A formal proof of Fiedler’s theorem ensuring connected subgraphs
- Strategies for practical and computationally efficient implementation
- Empirical validation using synthetic and real-world graph datasets

Our findings suggest that Fiedler’s method offers a robust and interpretable framework for graph partitioning. It is particularly suited to applications demanding balanced divisions and preserved connectivity, such as parallel computing, social network analysis, and biological systems modeling.

This report contributes to the field of graph algorithms by providing both theoretical insights and practical considerations for deploying spectral methods in real-world scenarios.

## 2 Acknowledgement

We express our sincere gratitude to all those who supported and guided us throughout this project.

First and foremost, we are deeply thankful to **Kavita Sutar Deshpande**, our project guide, for her valuable insights, continuous support, and encouragement.

We also extend our thanks to the faculty and staff of the **Department of Data Science** at **Chennai Mathematical Institute**, for providing a conducive environment for research and learning.

We are grateful to our peers and friends for their help, motivation, and meaningful discussions during the project.

Finally, we thank our families for their unwavering support and encouragement throughout our academic journey.

## 3 Introduction

### Background

Modern data systems increasingly rely on large-scale graph structures, which present significant challenges in storage, computation, and analysis due to their high connectivity. Unlike tabular data, which can be partitioned straightforwardly, graph data requires careful segmentation to preserve essential structural relationships. Spectral graph theory, and in particular Fiedler's method, offers a mathematically grounded and effective strategy for partitioning such graphs while maintaining their integrity.

### 3.1 Problem Statement

This work focuses on the problem of partitioning a large, connected graph into two smaller subgraphs such that:

- The number of edges crossing between the two subgraphs is minimized,
- Each resulting subgraph remains internally connected,
- The partition is balanced with respect to the number of vertices.

This problem is critical in applications such as distributed computing, parallel processing, and social network analysis.

### Aim and Objectives

**Aim:** To explore and implement Fiedler's Spectral Graph Partitioning method for effective and balanced graph segmentation.

**Objectives:**

- Understand the mathematical foundations of spectral graph theory.
- Analyze the properties of the Laplacian matrix and its eigenvectors.
- Implement Fiedler's method and demonstrate it on representative graphs.
- Establish that the resulting subgraphs are connected.
- Evaluate the performance and potential applications of the method.

### 3.2 Solution Approach

The solution is based on Fiedler's spectral method, which involves the following steps:

1. Construct the Laplacian matrix  $L = D - A$ , where  $D$  is the degree matrix and  $A$  is the adjacency matrix.
2. Compute the eigenvalues and eigenvectors of  $L$ .
3. Use the second-smallest eigenvector (known as the Fiedler vector) to assign vertices to two partitions based on the sign (positive or negative) of their corresponding components.

This approach minimizes edge cuts and promotes connectivity within each subgraph.

### Summary of Key Outcomes

- Developed a complete implementation of Fiedler's spectral partitioning algorithm.
- Demonstrated that the method yields connected and relatively balanced subgraphs.
- Validated theoretical claims using controlled examples with small graphs.
- Discussed the advantages, limitations, and practical relevance of the method in real-world scenarios.

## 4 Graph Theory Fundamentals

### 4.1 Definition of a Graph

A graph  $G = (V, E)$  consists of a set of vertices  $V$  and a set of edges  $E$ , where each edge connects a pair of vertices. In this report, we consider **simple, undirected** graphs—i.e., graphs without multiple edges or loops.

- **Vertices (Nodes):**  $V = \{v_1, v_2, \dots, v_n\}$
- **Edges:** Unordered pairs  $(v_i, v_j) \in E$

Each vertex is labeled with a unique integer for representation and matrix construction.

### Degree of a Vertex

The **degree** of a vertex  $v$ , denoted  $d(v)$ , is the number of edges incident on it. For undirected graphs:

$$d(v) = \text{Number of neighbors of } v$$

### Adjacency Matrix

The **adjacency matrix**  $A$  of a graph  $G$  with  $n$  vertices is an  $n \times n$  matrix where:

$$A_{ij} = \begin{cases} 1 & \text{if } (v_i, v_j) \in E \\ 0 & \text{otherwise} \end{cases}$$

This matrix indicates which vertices are connected by an edge.

### Degree Matrix

The **degree matrix**  $D$  is a diagonal matrix of size  $n \times n$ , where each diagonal entry represents the degree of a vertex:

$$D_{ij} = \begin{cases} d(v_i) & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

### Laplacian Matrix

The **Laplacian matrix**  $L$  is defined as:

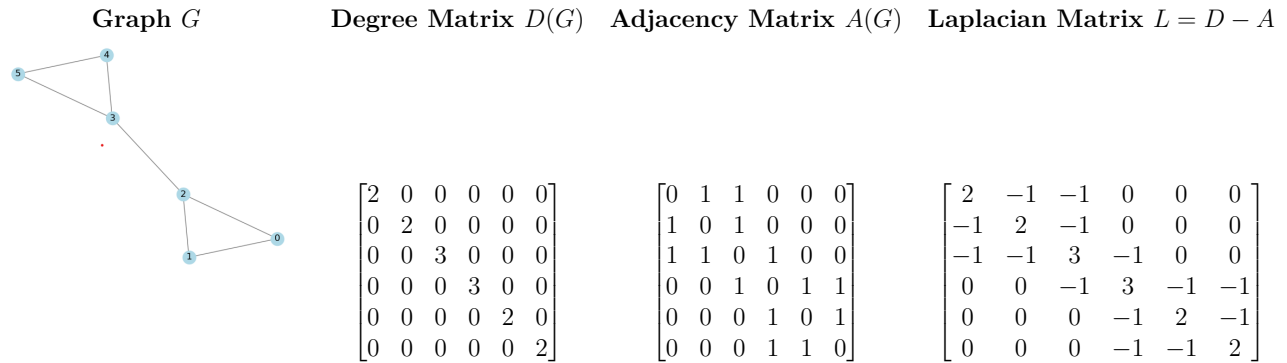
$$L = D - A$$

It is symmetric and positive semi-definite. The smallest eigenvalue of  $L$  is always 0, and its spectral properties are central to many graph algorithms.

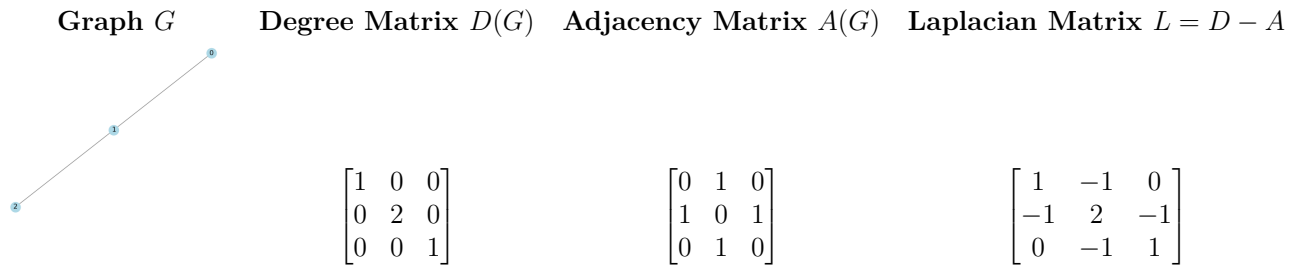
## 4.2 Graph Matrix Examples

Below are some examples for matrices associated to a Graph

### Example 1: Six-Node Graph



### Example 2: Three-Node Path Graph



## 4.3 Eigenvalues and Eigenvectors of the Laplacian Matrix

The eigenvalues and eigenvectors of the Laplacian matrix  $L$  provide critical spectral information about the graph's structure.

- The eigenvalues  $\lambda$  and eigenvectors  $x$  satisfy the equation:

$$Lx = \lambda x$$

- The smallest eigenvalue  $\lambda_1 = 0$ , with the corresponding eigenvector being the constant vector (i.e., all entries equal).
- The second smallest eigenvalue  $\lambda_2$  is known as the **algebraic connectivity** of the graph.
- The corresponding eigenvector  $x_2$  is called the **Fiedler vector**, which captures how to divide the graph into two parts with minimal edge cuts.



## 5 Fiedler's Spectral Graph Partitioning Method

**Fiedler's method** is a spectral graph partitioning technique that uses the eigenstructure of the Laplacian matrix  $L(G)$  to divide a graph into two subgraphs with minimal edge cuts. It leverages the properties of the **Fiedler vector**—the eigenvector corresponding to the second-smallest eigenvalue of  $L$ .

### 5.1 Procedure

To partition a graph using Fiedler's method, follow these steps:

1. Compute the Laplacian matrix  $L(G) = D - A$ , where  $D$  is the degree matrix and  $A$  is the adjacency matrix.
2. Compute the eigenvalues and eigenvectors of  $L(G)$ .
3. Identify the second-smallest eigenvalue  $\lambda_2$ , and extract the corresponding eigenvector  $x_2$  (the **Fiedler vector**).
4. Partition the vertex set  $V$  based on the sign of the entries in  $x_2$ :

If  $x_i < 0$ , assign vertex  $i$  to  $G_1$ ; else assign it to  $G_2$ .

This approach often produces balanced partitions and reflects the underlying structure of the graph.

### 5.2 Example 1: Six-Node Graph

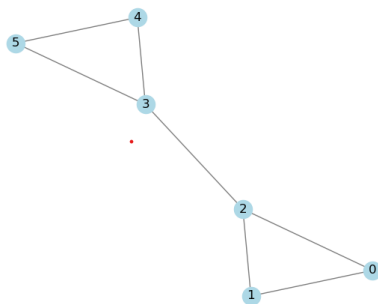


Figure 1: \*  
Input Graph

$$\begin{bmatrix} 0.4647 \\ 0.4647 \\ 0.2610 \\ -0.2610 \\ -0.4647 \\ -0.4647 \end{bmatrix}$$

Figure 2: \*  
Fiedler Vector

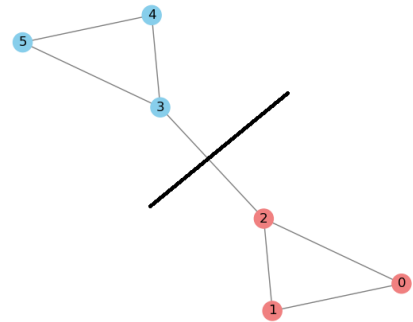


Figure 3: \*  
Partitioned Graph

### 5.3 Example 2: Three-Node Path Graph

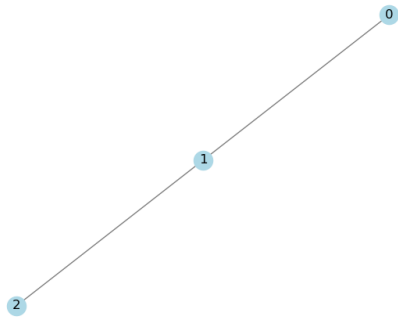


Figure 4: \*  
**Input Graph**

$$\begin{bmatrix} -0.7071 \\ 0.0001 \\ 0.7071 \end{bmatrix}$$

Figure 5: \*  
**Fiedler Vector**

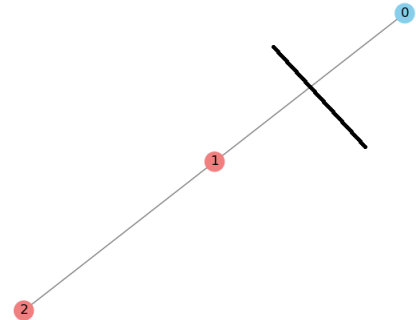


Figure 6: \*  
**Partitioned Graph**

### Remarks

While the Fiedler vector provides a natural way to split a graph, the resulting partition is not guaranteed to:

- Minimize the total number of edge cuts.
- Preserve connectivity within each subgraph  $G_1$  and  $G_2$ .

Despite this, it is often effective in practice and widely used in applications such as clustering, image segmentation, and load balancing.

## 6 Standing Wave Analogy for Spectral Partitioning

The connection between Fiedler’s spectral partitioning and the theory of vibrating strings provides intuitive insight into why the second eigenvector of the Laplacian—the Fiedler vector—is effective for graph partitioning.

### Discrete String Model as a Graph

Consider a string of length  $L$  fixed at both ends ( $x_0 = 0, x_{n+1} = L$ ), discretized into  $n + 1$  equal segments:

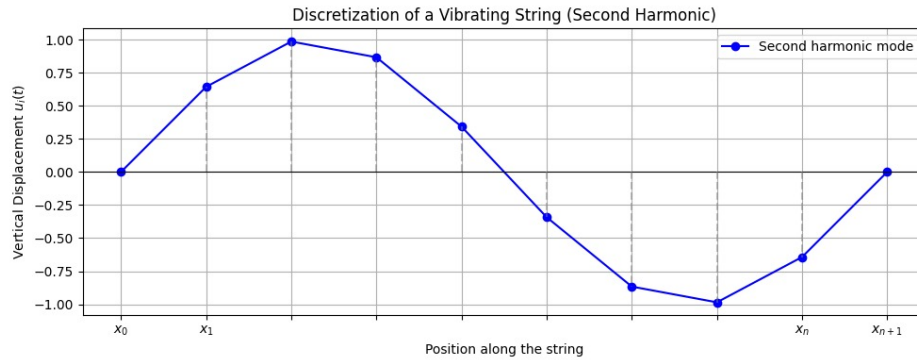


Figure 7: Discretized vibrating string showing  $n$  interior points (vertices) connected as a chain graph. The second vibration mode (dashed) naturally partitions the system.

This system can be modeled as a *chain graph*, where:

- Each interior point  $x_i = i \cdot h$  ( $i = 1, \dots, n$ ) is a vertex.
- Edges connect adjacent points.
- Segment length  $h = L/(n + 1)$  defines spacing.

As  $n \rightarrow \infty$  and  $h \rightarrow 0$ , this model approximates a continuous string.

### Wave Dynamics and the Graph Laplacian

The vertical displacements  $u_i(t)$  of the discrete string follow a finite-difference version of the wave equation. The system satisfies:

- Boundary conditions:  $u_0(t) = u_{n+1}(t) = 0$
- Displacement vector:  $\mathbf{u}(t) = [u_1(t), \dots, u_n(t)]^T$

Using **Second Derivative Approximation via Taylor Expansion**:

$$\begin{aligned} u(x+h) &= u(x) + hu'(x) + \frac{h^2}{2}u''(x) + \frac{h^3}{6}u'''(x) + \mathcal{O}(h^4) \\ u(x-h) &= u(x) - hu'(x) + \frac{h^2}{2}u''(x) - \frac{h^3}{6}u'''(x) + \mathcal{O}(h^4) \end{aligned}$$

Adding both:

$$u(x+h) + u(x-h) = 2u(x) + h^2u''(x) + \mathcal{O}(h^4) \Rightarrow u''(x) \approx \frac{u(x+h) - 2u(x) + u(x-h)}{h^2}$$

This finite difference **leads to a discrete Laplacian Matrix** which is exactly Laplacian matrix of Chain Graph except first and last vertices.

$$\frac{d^2u_i}{dx^2} \approx \frac{u_{i-1} - 2u_i + u_{i+1}}{h^2} \Rightarrow L = \frac{1}{h^2} \begin{bmatrix} -2 & 1 & & \\ 1 & -2 & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & -2 \end{bmatrix}$$

So, the discrete wave equation becomes:

$$\frac{d^2\mathbf{u}}{dt^2} = \frac{c^2}{h^2}L\mathbf{u} \Rightarrow L\mathbf{v} = -\frac{\omega^2h^2}{c^2}\mathbf{v}$$

This is an eigenvalue problem:

$$L\mathbf{u} = \lambda\mathbf{u}, \quad \lambda = -\frac{\omega^2h^2}{c^2}$$

For sinusoidal eigenfunctions such as  $f^r = \sin\left(\frac{r\pi x}{L}\right)$ , we have:

$$\lambda = -\frac{4\pi^2h^2}{c^2}f^r$$

## Eigenvalues of the Discrete Laplacian

For a chain graph with  $n$  vertices, the eigenvalues of  $L$  are:

$$\lambda_k = 2\left(1 - \cos\left(\frac{k\pi}{n+1}\right)\right), \quad k = 1, 2, \dots, n$$

The second smallest eigenvalue ( $\lambda_2$ ) is:

$$\lambda_2 = 2\left(1 - \cos\left(\frac{2\pi}{n+1}\right)\right)$$

The corresponding eigenvector (Fiedler vector) changes sign once and is used to split the graph into two parts.

We solve for the second natural frequency  $f_2$  using:

$$f_2 = \frac{c}{2\pi h} \sqrt{2 \left( 1 - \cos \left( \frac{2\pi}{n+1} \right) \right)}$$

Substitute  $h = \frac{L}{n+1}$ :

$$f_2 = \frac{c(n+1)}{2\pi L} \sqrt{2 \left( 1 - \cos \left( \frac{2\pi}{n+1} \right) \right)}$$

As  $n \rightarrow \infty$ , apply the Taylor approximation:

$$\cos \left( \frac{2\pi}{n+1} \right) \approx 1 - \frac{2\pi^2}{(n+1)^2} \quad \Rightarrow \quad 2 \left( 1 - \cos \left( \frac{2\pi}{n+1} \right) \right) \approx \frac{4\pi^2}{(n+1)^2}$$

So,

$$f_2 \approx \frac{c(n+1)}{2\pi L} \cdot \frac{2\pi}{n+1} = \frac{c}{L}$$

**Conclusion:** The second vibration mode has frequency  $f_2 = \frac{c}{L}$ , which corresponds to the second eigenvalue of the discrete Laplacian and naturally partitions the system.

## Generalization to Arbitrary Graphs

Although derived for a 1D chain, the intuition extends to general graphs:

- The Laplacian  $L = D - A$  generalizes the second derivative operator.
- The Fiedler vector is analogous to the second standing wave mode.
- Spectral partitioning separates the graph at its “weakest” connectivity point, akin to natural vibration nodes.

## Conclusion from above analogy

- The corresponding frequency  $f$  in the standing wave model equals the second harmonic  $\frac{c}{L}$ . In this mode, the particles of the string experience acceleration in opposite directions — half are moving upward, and half downward. This behavior effectively partitions the string into two distinct regions. Analogously, the Fiedler vector partitions a chain graph into two groups of vertices based on the sign of the vector components.
- As previously discussed, the second eigenvector of the graph Laplacian  $L(G)$  naturally partitions a chain graph into two nearly equal halves, with the sign change representing the division.
- This intuition extends to more complex graphs. In particular, for planar graphs that can be visualized geometrically (e.g., as a trampoline or membrane), the second mode of vibration divides the graph into two balanced regions, much like how a trampoline deforms into two lobes during its second vibrational mode.

## 7 Spectral Graph Partitioning: Mathematical Proof

Let  $G = (V, E)$  be an undirected graph with  $n$  vertices. The Laplacian matrix  $L(G)$  is defined as:

$$L = D - A$$

where:

- $D$  is the degree matrix—a diagonal matrix where  $D_{ii} = \deg(v_i)$ ,
- $A$  is the adjacency matrix of the graph.

### Key Properties of the Laplacian

- $L$  is symmetric and positive semi-definite.
- Its smallest eigenvalue  $\lambda_1$  is always zero. and associated eigenvector is the constant vector is  $\mathbf{v}_{\lambda_1} = [1, 1, \dots, 1]^T \in \mathbb{R}^n$ .

### The Optimization Problem Behind Partitioning

With  $\lambda_1 = 0$ , the next informative quantity is the second smallest eigenvalue  $\lambda_2$ , known as the **algebraic connectivity** of the graph.

The corresponding eigenvector  $\mathbf{x} \in \mathbb{R}^n$ , called the Fiedler vector, minimizes the Rayleigh quotient subject to orthogonality to the constant vector:

$$\lambda_2 = \min_{\mathbf{x} \perp \mathbf{1}} \frac{\mathbf{x}^T L \mathbf{x}}{\mathbf{x}^T \mathbf{x}}.$$

Expanding the numerator, we obtain:

$$\mathbf{x}^T L \mathbf{x} = \sum_{i=1}^n D_{ii} x_i^2 - \sum_{(i,j) \in E} 2x_i x_j = \sum_{(i,j) \in E} (x_i - x_j)^2.$$

This formulation shows that the objective encourages neighboring vertices  $i$  and  $j$  to have similar values, promoting smoothness in the assignment of values over the graph.

Since  $\mathbf{x} \perp \mathbf{v}_{\lambda_1}$ , we have:

$$\sum_{i=1}^n x_i = 0.$$

This constraint ensures that the entries of  $\mathbf{x}$  cannot all have the same sign. Consequently, **a natural way to partition the graph is by using the signs of the components**:

$$V = V_- \cup V_+, \quad \text{where } \begin{cases} V_- = \{v_i \in V \mid x_i < 0\}, \\ V_+ = \{v_i \in V \mid x_i \geq 0\}. \end{cases}$$

Thus, the Fiedler vector yields a bipartition of the graph based on the sign structure of its entries. The spectral objective can also be interpreted as:

$$\lambda_2 = \min_{\sum x_i = 0} \sum_{(i,j) \in E} (x_i - x_j)^2.$$

This expression is minimized when adjacent vertices fall on the same side of the partition. When they differ in sign, the squared difference increases the cost—indicating a cut edge.

Therefore, spectral methods inherently **Minimize the number of edges crossing between partitions, and Promote connectivity within partitions.**

### Balanced Subgraphs

While the optimization does not explicitly enforce equal-sized partitions, the zero-mean constraint  $\sum x_i = 0$  often results in approximately balanced subsets. This helps further reduce cross-partition edges, making the cut both small and balanced.

Spectral graph partitioning thus achieves **Low edge-cut partitions, and a Balanced vertex sets.** ■

### Visual Representation

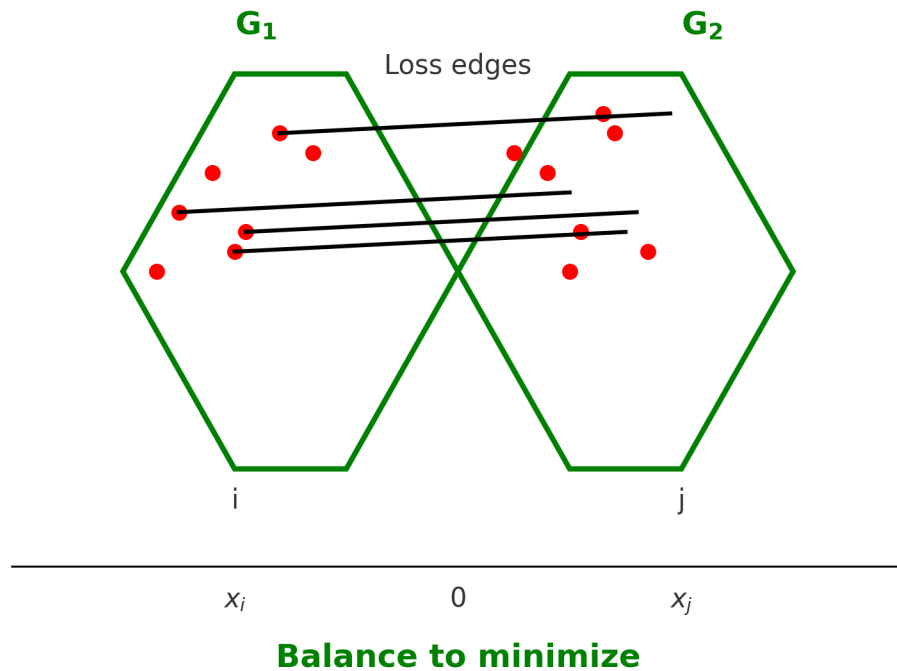


Figure 8: Graph partitioning based on the sign of Fiedler vector components

## 8 Summary of what we did and Next Steps

### Established Results:

- A valid graph partition can be derived from the Fiedler vector.
- This partition minimizes the number of cut edges.
- It tends to produce partitions of approximately equal size.

### Remaining Question:

We now seek to prove that each subgraph obtained from this spectral partition is **connected**.

This result stems from a foundational theorem by Miroslav Fiedler, which asserts that partitions induced by the second eigenvector yield connected components under certain conditions.

Before presenting Fiedler's formal proof, we will build the necessary groundwork by following a simplified and pedagogical path outlined by James Demmel. This involves introducing supporting definitions and lemmas that collectively establish the connectivity of the resulting subgraphs.

The next section will present these foundational results and then prove the connectivity of the induced subgraphs.



## 9 Mathematical Preliminaries and Theoretical Foundations

### 9.1 Key Definitions

#### Definition 9.1.1: Non-negative Matrix

A matrix  $A$  is said to be non-negative if all of its elements are greater than or equal to zero:

$$a_{ij} \geq 0 \quad \text{for all } i, j.$$

#### Definition 9.1.2: Spectral Radius

The spectral radius of a square matrix  $A$  is the largest absolute value among its eigenvalues:

$$\rho(A) = \max_{1 \leq i \leq n} |\lambda_i|.$$

#### Definition 9.1.3: Positive Definite Matrix

A matrix  $A$  is positive definite if for all non-zero vectors  $x$ , the following holds:

$$x^\top A x > 0.$$

### 9.2 Remarks

#### Remark 9.2.1: Symmetric and Positive-Definite Matrices

A matrix  $A$  is symmetric and positive definite if and only if every eigenvalue of  $A$  is positive. Similarly,  $A$  is symmetric and positive semi-definite if and only if every eigenvalue of  $A$  is non-negative.

### 9.3 Lemmas

#### Lemma 9.3.1

Let  $A$  be an  $n \times n$  symmetric matrix with eigenvalues  $\lambda_1 \leq \dots \leq \lambda_n$ . Then:

$$\lambda_1 = \min_{v \neq 0} \frac{v^\top A v}{v^\top v}, \quad \lambda_n = \max_{v \neq 0} \frac{v^\top A v}{v^\top v}.$$

That is,  $\lambda_1$  and  $\lambda_n$  are the extremal values of the Rayleigh quotient over all non-zero vectors  $v \in \mathbb{R}^n$ .

**Proof.** Let  $v \in \mathbb{R}^n \setminus \{0\}$ . Define the Rayleigh quotient:

$$R(v) = \frac{v^\top A v}{v^\top v}.$$

Let  $y = \frac{v}{\|v\|}$ , so  $R(v) = y^\top A y$  with  $\|y\| = 1$ .

Since  $A$  is symmetric, it is orthogonally diagonalizable:  $A = Q\Lambda Q^\top$ , with  $Q$  orthogonal and  $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$ . Then:

$$y^\top Ay = (Q^\top y)^\top \Lambda (Q^\top y) = \sum_{i=1}^n \lambda_i z_i^2, \quad \text{where } z = Q^\top y \text{ and } \sum z_i^2 = 1.$$

Thus,  $R(v)$  is a convex combination of the eigenvalues:

$$\lambda_1 \leq R(v) \leq \lambda_n.$$

Hence,

$$\min_{v \neq 0} \frac{v^\top Av}{v^\top v} = \lambda_1, \quad \max_{v \neq 0} \frac{v^\top Av}{v^\top v} = \lambda_n.$$

■

### Lemma 9.3.2

Let  $A$  be an  $n \times n$  symmetric matrix. If  $\rho(A) < 1$ , then  $I_n - A$  is nonsingular and:

$$(I_n - A)^{-1} = \sum_{i=0}^{\infty} A^i.$$

**Proof.**

**1. Eigenvalues of  $I_n - A$ :** Since  $A$  is symmetric, its eigenvalues are real and lie in  $(-1, 1)$ . The eigenvalues of  $I_n - A$  are:

$$\lambda_i(I_n - A) = 1 - \lambda_i(A) \in (0, 2),$$

so  $I_n - A$  is positive definite and invertible.

**2. Convergence of the Series:** Diagonalize  $A = Q\Lambda Q^\top$ , then:

$$A^i = Q\Lambda^i Q^\top.$$

As  $|\lambda_i| < 1$ ,  $\Lambda^i \rightarrow 0$ , so the series  $\sum_{i=0}^{\infty} A^i$  converges.

**3. Telescoping Sum:** Let:

$$S_m = \sum_{i=0}^m A^i.$$

Then:

$$(I_n - A)S_m = I_n - A^{m+1}.$$

As  $m \rightarrow \infty$ ,  $A^{m+1} \rightarrow 0$ , so:

$$(I_n - A) \left( \sum_{i=0}^{\infty} A^i \right) = I_n.$$

Therefore,

$$(I_n - A)^{-1} = \sum_{i=0}^{\infty} A^i$$

■

## 9.4 Theorems

### Theorem 9.4.1 (Cauchy Interlacing Theorem)

Let  $A \in \mathbb{R}^{n \times n}$  be a symmetric matrix with eigenvalues ordered as:

$$\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n.$$

Let  $A(i:j, i:j)$  denote the principal submatrix of  $A$ , formed by taking rows and columns  $i$  through  $j$ , so that:

$$A(i:j, i:j) \in \mathbb{R}^{(j-i+1) \times (j-i+1)}.$$

Let the eigenvalues of  $A(i:j, i:j)$  be ordered as:

$$\chi_1 \leq \chi_2 \leq \cdots \leq \chi_{j-i+1}.$$

Then, for any  $k \leq j - i + 1$ , the matrix  $A$  has at least  $k$  eigenvalues less than or equal to  $\chi_k$ ; that is,

$$|\{\lambda_\ell \mid \lambda_\ell \leq \chi_k\}| \geq k.$$

### Theorem 9.4.2

Let  $A \in \mathbb{R}^{n \times n}$  and  $X \in \mathbb{R}^{n \times n}$  be a nonsingular matrix. Then

$$X^\top A X \text{ is symmetric positive definite} \iff A \text{ is symmetric positive definite.}$$

#### Proof ( $\Leftarrow$ direction):

Assume  $A$  is symmetric positive definite, i.e.,  $A = A^\top$  and  $\lambda_{\min}(A) > 0$ .

- **Symmetry:** Since  $A = A^\top$ , we get  $(X^\top A X)^\top = X^\top A^\top X = X^\top A X$ , so  $X^\top A X$  is symmetric.
- **Positive definiteness:** Consider the Rayleigh quotient:

$$\lambda_{\min}(X^\top A X) = \min_{v \neq 0} \frac{v^\top X^\top A X v}{v^\top v}.$$

Let  $y = Xv$ , then  $v^\top X^\top A X v = y^\top A y$ , and using  $v^\top v = \frac{y^\top y}{v^\top X^\top X v / v^\top v}$ , we get:

$$\lambda_{\min}(X^\top A X) \geq \lambda_{\min}(A) \cdot \lambda_{\min}(X^\top X) > 0.$$

Hence,  $X^\top A X$  is positive definite.

**Proof (  $\Rightarrow$  direction):**

Assume  $X^\top AX$  is symmetric positive definite.

- **Symmetry:** Since  $(X^\top AX)^\top = X^\top A^\top X = X^\top AX$ , we have  $A = A^\top$ , so  $A$  is symmetric.
- **Positive definiteness:** As  $X^\top AX$  is positive definite and  $X$  is nonsingular, the Rayleigh quotient gives:

$$\lambda_{\min}(X^\top AX) = \lambda_{\min}(A) \cdot \lambda_{\min}(X^\top X) > 0 \Rightarrow \lambda_{\min}(A) > 0.$$

Thus,  $A$  is positive definite. ■

## 10 Fiedler's Theorem of Spectral Graph Partitioning

### Theorem and Proof

**Fiedler's Theorem.** *Let  $G$  be a connected graph and let  $L(G)$  be its Laplacian matrix. Create the subgraphs  $G_1$  and  $G_2$  using Fiedler's method (i.e., partitioning based on the sign of the entries in the Fiedler vector). Then, both  $G_1$  and  $G_2$  are connected.*

### Proof (by contradiction):

- Assume, without loss of generality, that  $G_1$  is disconnected, i.e., it is composed of two connected components.
- Let  $x$  be the eigenvector corresponding to  $\lambda_2$ , the second-smallest eigenvalue of  $L(G)$ , also called the Fiedler vector.
- The Laplacian matrix  $L(G)$  and the vector  $x$  can be written in block form as:

$$L(G) = \begin{bmatrix} L_{1,1} & \mathbf{0} & L_{1,3} \\ \mathbf{0} & L_{2,2} & L_{2,3} \\ L_{1,3}^T & L_{2,3}^T & L_{3,3} \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

- Here:
  - $x$  is the Fiedler vector,
  - $x_1$  and  $x_2$  are positive,
  - $x_3$  is negative.
- $O$  denotes matrices of appropriate sizes with all entries equal to zero.
- Since  $L(G)$  is symmetric, all diagonal blocks (e.g.,  $L_{1,1}$ ,  $L_{2,2}$ ,  $L_{3,3}$ ) are symmetric matrices, and off-diagonal blocks satisfy  $L_{i,j} = L_{j,i}^T$  and are non-positive.
- Each block  $L_{i,j}$  corresponds to the structure of a subgraph or component of  $G$ .
- Since the graph is partitioned based on the signs of entries in the Fiedler vector  $x$ , the vertices corresponding to  $x_1$  and  $x_2$  are assigned to the same subgraph  $G_1$ .
- There are no edges between the vertices corresponding to  $x_1$  and  $x_2$ , since the entries in  $L_{1,2}$  and  $L_{2,1}$  are zero. This implies that  $G_1$  is disconnected, and its components are represented by Laplacian submatrices such as  $L_{1,1}$  and  $L_{2,2}$ .

Let  $x$  be an eigenvector of  $L(G)$  with eigenvalue  $\lambda_2$ , so:

$$L(G)x = \lambda_2 x.$$

Consider the first block  $x_1$  of  $x$ . Multiplying out the corresponding block row:

$$L_{1,1}x_1 + L_{1,3}x_3 = \lambda_2 x_1.$$

Add  $\varepsilon x_1$  to both sides:

$$(\varepsilon I + L_{1,1})x_1 + L_{1,3}x_3 = (\varepsilon + \lambda_2)x_1.$$

Since  $\varepsilon I + L_{1,1}$  is symmetric positive definite, we use Demmel's method and let:

$$\varepsilon I + L_{1,1} = D - N,$$

where:

- $D$  is diagonal,
- $-N$  contains off-diagonal entries,
- $D^{1/2}$  is the square root of  $D$ .

Then:

$$\varepsilon I + L_{1,1} = D^{1/2}(I - M)D^{1/2}, \quad \text{where } M = (D^{1/2})^{-1}N(D^{1/2})^{-1}.$$

Since  $I - M$  is positive definite, so is  $(D - N)$  (From Theorem 9.4.2) and  $M$  is symmetric with eigenvalues in  $(-1, 1)$ , Using **Lemma 9.3.2** we write:

$$Y = (\varepsilon I + L_{1,1})^{-1} = (D^{1/2})^{-1} \left( \sum_{i=0}^{\infty} M^i \right) (D^{1/2})^{-1}.$$

Thus,  $Y$  is a nonnegative matrix.

Multiply the earlier equation by  $Y$ :

$$x_1 + YL_{1,3}x_3 = (\varepsilon + \lambda_2)Yx_1.$$

Multiply both sides by  $x_1^T$ :

$$x_1^T x_1 + x_1^T YL_{1,3}x_3 = (\varepsilon + \lambda_2)x_1^T Yx_1.$$

by rearranging we get:

$$(\varepsilon + \lambda_2) \cdot \frac{x_1^T Yx_1}{x_1^T x_1} = 1 + \frac{x_1^T YL_{1,3}x_3}{x_1^T x_1}.$$

Since  $x_1^T YL_{1,3}x_3 > 0$ , this implies:

$$(\varepsilon + \lambda_2) \cdot \lambda_{n_1}(Y) > 1 \Rightarrow \frac{\varepsilon + \lambda_2}{\varepsilon + \lambda_1(L_{1,1})} > 1.$$

Hence:

$$\lambda_1(L_{1,1}) < \lambda_2.$$

Similarly, we get:

$$\lambda_1(L_{2,2}) < \lambda_2.$$

Now, consider the block matrix:

$$B = \begin{bmatrix} L_{1,1} & 0 \\ 0 & L_{2,2} \end{bmatrix}$$

By the Cauchy Interlacing Theorem(Theorem 9.4.1):

$$\lambda_2(L(G)) \leq \lambda_2(B),$$

but we showed:

$$\lambda_1(L_{1,1}), \lambda_1(L_{2,2}) < \lambda_2 \Rightarrow \lambda_2(B) < \lambda_2(L(G)),$$

a contradiction.

Therefore, the assumption that  $G_1$  is disconnected must be false.

**Hence,  $G_1$  is connected.**

**By symmetry, the same applies to  $G_2$ . Thus, both  $G_1$  and  $G_2$  are connected.** ■

Thus, **Fiedler's Spectral Partitioning** provides a way to divide a graph into two connected subgraphs with desirable properties. Based on our analysis and demonstrations so far, we have established the following:

- The **Fiedler vector** enables effective partitioning of a graph.
- The number of vertices in both subgraphs is approximately balanced.
- The partition minimizes the number of edges cut between the two parts.
- Each resulting subgraph is **internally connected**.

## 11 Implementation : Examples

All implementations were done using Python. The following examples illustrate spectral graph partitioning using the Laplacian matrix and Fiedler vector.

### Example 1

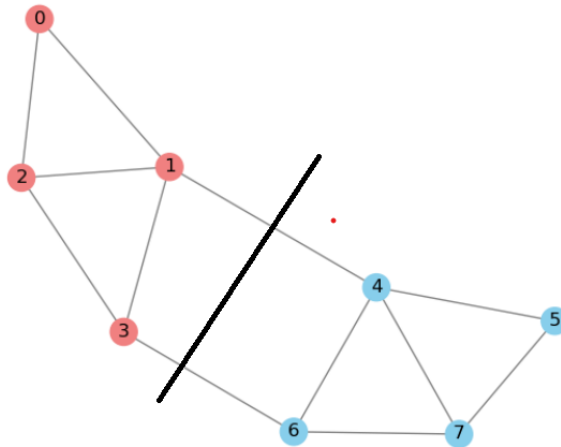


Figure 9: Spectral Partitioned Graph for Example 1

**Laplacian Matrix  $L$ :** max width=

$$\begin{bmatrix} 2 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\ -1 & 4 & -1 & -1 & -1 & 0 & 0 & 0 \\ 0 & -1 & 3 & -1 & 0 & 0 & 0 & -1 \\ -1 & -1 & -1 & 3 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 4 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & 3 & -1 \\ 0 & 0 & -1 & 0 & -1 & 0 & -1 & 3 \end{bmatrix}$$

**Fiedler Vector:**

$$[0.4886, 0.2503, 0.1953, 0.4006, -0.2503, -0.4886, -0.4006, -0.1953]$$

**Lost Edges between  $G_1$  and  $G_2$ :** 2



## Example 2

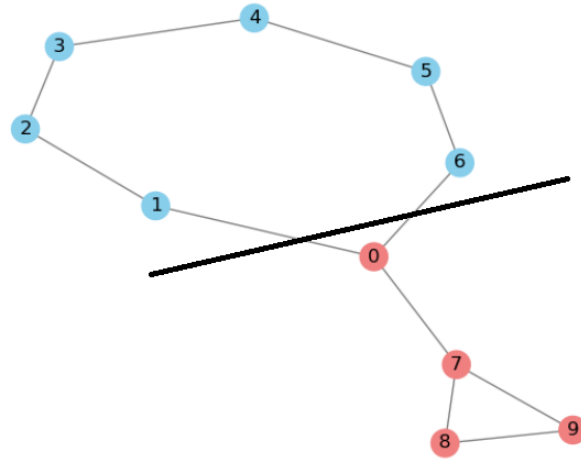


Figure 10: Spectral Partitioned Graph for Example 2

**Laplacian Matrix  $L$ :** max width=

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

**Fiedler Vector:**

$$[0.0521, -0.1147, -0.2543, -0.3335, -0.3335, -0.2543, -0.1147, 0.3734, 0.4897, 0.4897]$$

**Lost Edges between  $G_1$  and  $G_2$ :** 2

### Example 3

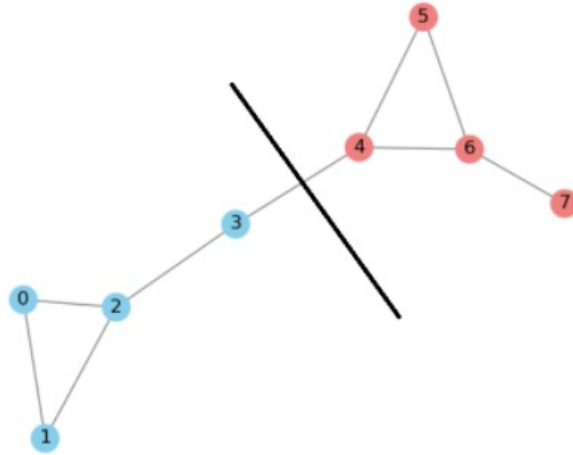


Figure 11: Spectral Partitioned Graph for Example 3

Laplacian Matrix  $L$ :

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

Fiedler Vector:

$$[-0.4486, -0.4486, -0.3507, -0.0782, 0.2114, 0.3154, 0.3507, 0.4486]$$

Lost Edges between  $G_1$  and  $G_2$ : 1

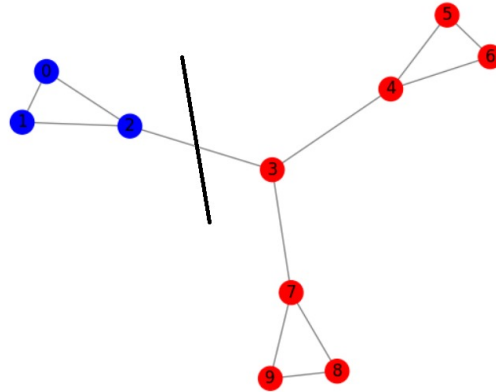
**Example 4**

Figure 12: Spectral Partitioned Graph for Example 4

**Laplacian Matrix  $L$ :**

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

**Fiedler Vector:**

$$[-0.5127, -0.5127, -0.3753, 0.0000, 0.1877, 0.2564, 0.2564, 0.1877, 0.2564, 0.2564]$$

**Lost Edges between  $G_1$  and  $G_2$ : 1**

## 12 Limitations of Spectral Partitioning

Although powerful, spectral partitioning has limitations:

### Problematic Cases

- **Star Graph:** A central node connected to many peripheral nodes. Removing the central edge disconnects the entire graph.
- Graphs with highly skewed degree distributions.
- Graphs with non-uniform edge density.

### Reason for Failure

In such cases, the second smallest eigenvalue (Fiedler value) and its eigenvector may not lead to a meaningful or balanced partition.

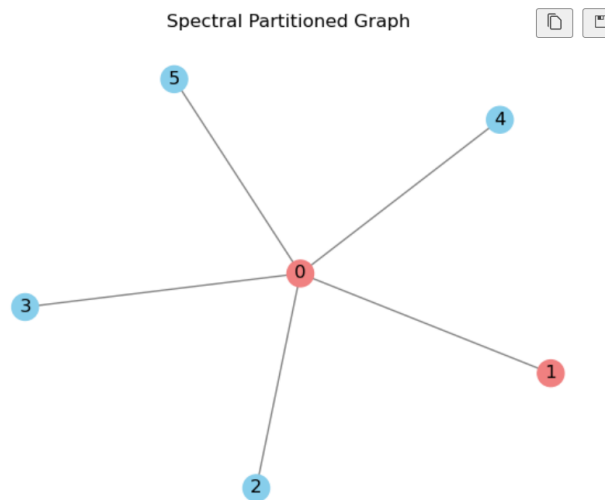


Figure 13: Star Graph Example

## Industrial Applications

Spectral partitioning techniques are widely used across various industries due to their ability to identify natural groupings or structures in complex data. Below are some key applications:

- **Community Detection:** Spectral partitioning is used in social network analysis to identify communities or groups of nodes (individuals or organizations) with similar relationships or behaviors. This is crucial for understanding the structure of social networks, detecting influential individuals, and finding tight-knit groups on platforms like Facebook or Twitter.
- **Image Segmentation:** In computer vision, spectral partitioning is employed to divide an image into distinct regions with similar properties such as color, texture, or intensity. This technique aids in object recognition, medical imaging, and autonomous vehicle navigation by identifying and separating relevant image regions.
- **Web Graph Analysis:** The web is represented as a graph with pages as nodes and hyperlinks as edges. Spectral partitioning techniques help analyze and understand the structure of the internet, aiding in tasks like link prediction, web community detection, and improving search engine optimization (SEO).
- **Biological Networks:** In bioinformatics, spectral partitioning is applied to gene or protein interaction networks to uncover functional modules or sub-networks. This helps researchers identify new biological pathways, drug targets, and understand cellular processes at a molecular level.
- **VLSI Design:** In very-large-scale integration (VLSI) design, spectral partitioning is used to optimize the layout of circuits on a chip. It minimizes the interconnections between blocks, which results in reduced power consumption, improved chip performance, and faster signal transmission in microprocessors and integrated circuits.
- **Parallel Computing:** Spectral partitioning is applied in parallel computing to distribute computational tasks across multiple processors with minimal inter-processor communication. This leads to efficient parallel algorithms and reduces bottlenecks in large-scale scientific simulations or high-performance computing (HPC) environments.
- **Recommendation Systems:** In e-commerce and streaming platforms, spectral partitioning helps create personalized recommendations by grouping users with similar preferences. This improves user experience by providing more relevant content, products, or media based on previous behavior.

## 13 Further Work After Fiedler's Decomposition

Following Fiedler's decomposition, there are several advanced techniques and methods that can enhance the application of spectral partitioning. These include:

- **Multiway Spectral Clustering:** Instead of just performing a binary partition, multiple eigenvectors can be used to partition a graph into  $k$  clusters. This is particularly useful for problems that involve categorizing data into more than two groups, such as customer segmentation or image categorization.
- **Graph Coarsening:** To improve scalability and reduce computational complexity, graph coarsening techniques merge or collapse nodes and edges into supernodes, simplifying the original graph. This approach helps handle large-scale graphs by working with progressively simpler approximations.
- **Dynamic Graph Analysis:** Spectral methods can be applied to dynamic or evolving graphs, which change over time. By combining anomaly detection techniques with spectral partitioning, it is possible to monitor shifts in graph structure, such as detecting emerging communities or identifying unusual behavior in social networks or financial markets.
- **Spectral Embedding:** Spectral embedding techniques use multiple eigenvectors to map graph nodes into a low-dimensional space (2D or 3D). This visualization aids in understanding complex graph structures, revealing underlying patterns, or providing insights into clustering and community detection in high-dimensional data.
- **Graph Signal Processing:** In this advanced technique, the Fiedler vector is treated as a low-frequency signal. It is used for filtering or denoising graph signals, allowing for the extraction of relevant information from noisy or incomplete data. This is particularly valuable in areas like sensor networks or telecommunications.

## 14 Conclusion

In this paper, we explored the theory and practical application of **Spectral Graph Partitioning** using the **Fiedler vector**. Spectral graph partitioning is a powerful technique for clustering nodes based on graph structure and connectivity. By constructing the graph Laplacian  $L(G)$  from the adjacency and degree matrices, we captured the underlying relationships crucial for effective partitioning. This process is foundational in understanding the behavior of complex networks.

A central aspect of spectral partitioning is the second smallest eigenvalue (Fiedler value) and its corresponding eigenvector (Fiedler vector). These components are key for dividing the graph into meaningful clusters, minimizing the edge cut between them. We demonstrated how this technique effectively splits the graph, showing its utility in various applications like community detection and segmentation.

The examples provided validated the method's effectiveness, with visually interpretable results. Each partitioned graph was analyzed, and the outcomes from spectral partitioning aligned with intuitive graph structure understanding. The use of visual and matrix-based illustrations helped clarify the algorithm's logic, making the underlying process easier to follow.

Beyond the basic partitioning, further research into spectral methods opens new avenues. These include **multiway spectral clustering**, using multiple eigenvectors for partitioning into more than two clusters, and **graph coarsening** to simplify larger graphs for scalability. Additionally, **dynamic graph analysis** combining anomaly detection with spectral methods can help track graph changes over time, such as in social networks.

While spectral partitioning is effective in many cases, it has limitations. Graphs with skewed degree distributions, star-like structures, or non-uniform edge densities can challenge the method. In these cases, the second smallest eigenvalue and eigenvector may not provide meaningful partitions, requiring additional methods or refinements.

Spectral partitioning has found wide industrial applications, including **community detection** in social networks, **image segmentation** in computer vision, **VLSI design** for chip layout optimization, and **biological networks** for identifying gene/protein interaction modules. These applications demonstrate the versatility of spectral methods and their ability to uncover hidden patterns.

This work is only the beginning. With the increasing complexity of graphs, further advancements in spectral partitioning are expected. Improvements in algorithmic efficiency and new applications will continue to expand the possibilities for graph analysis in diverse fields.

## 15 References

- Luxburg, U. von. *A Tutorial on Spectral Clustering*. *Statistics and Computing*, 17(4), 2007. A foundational paper explaining the mathematical principles and practical applications of spectral clustering. Available at: <https://doi.org/10.1007/s11222-007-9033-z>
- Fiedler, M. *Algebraic Connectivity of Graphs*. *Czechoslovak Mathematical Journal*, 23(98), 1973. The original work introducing the Fiedler vector, central to spectral graph theory and partitioning.
- Demmel, J. *Graph Partitioning, Part 1*. University of California, Berkeley. Lecture notes covering spectral methods and practical implementation. Available at: <http://www.cs.berkeley.edu/~demmel/cs267/lecture18/lecture18.html>
- Demmel, J. *Graph Partitioning, Part 2*. University of California, Berkeley. Continuation of the above lecture series with further insights. Available at: <http://www.cs.berkeley.edu/~demmel/cs267/lecture20/lecture20.html>
- *Fundamentals of Graph Theory* – GeeksforGeeks. A beginner-friendly introduction to graph theory concepts essential for understanding spectral techniques. Available at: <https://www.geeksforgeeks.org/fundamentals-of-graph-theory/>
- *Spectral Graph Partitioning – Finding a Partition* (YouTube). A visual and intuitive explanation of spectral partitioning with the Fiedler vector. Available at: <https://www.youtube.com/watch?v=siCPjpUtEOA&t=1s>
- **Supplementary Materials:** All relevant code and resources are available at: <https://github.com/AyushmanGHub/Fiedlers-Spectral-Graph-Partitioning-Paper/>