Graph Theory

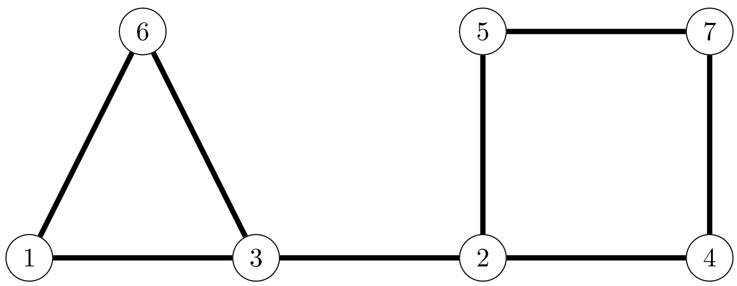
Justin Wyss-Gallifent

July 21, 2021

12.1 Introduction

Let's get started with a simple example.

Example 12.1. Consider this picture which represents seven objects connected to one another:



This picture could represent a computer network, a network of friends, or the lines could represent roads between locations or borders between countries.

The study of structures like these is the heart of *graph theory* and in order to manage large graphs we need linear algebra.

12.2 Basic Definitions

Definition 12.2.0.1. A *graph* is a collection of *vertices* (nodes or points) connected by *edges* (line segments).

Definition 12.2.0.2. A graph is *simple* if has no multiple edges, (meaning two vertices can only be connected by one edge) and no loops (a vertex cannot have an edge connecting it to itself).

Definition 12.2.0.3. A graph is *connected* if it is in one single connected piece.

All the graphs we will look at will be simple connected graphs.

The example in the introduction is then a simple connected graph with seven vertices connected by eight edges.

Definition 12.2.0.4. The *degree* of a vertex is the number of edges connected to the vertex.

Definition 12.2.0.5. For a simple graph G with n vertices the *degree matrix* for G is the $n \times n$ diagonal matrix D such that d_{ii} equals the degree of the i^{th} vertex.

Definition 12.2.0.6. For a simple graph G the *adjacency matrix* is the symmetric matrix A such that a_{ij} equals 1 if vertices i and j are connected by an edge and 0 otherwise.

Definition 12.2.0.7. For a simple graph G the *Laplacian matrix* L is defined by $L = D - A$.

The term *Laplacian matrix* for a graph is actually very general. There are lots of different Laplacian matrices, this one is by far the most common and is technically the *unnormalized graph Laplacian matrix* but since it's the only one we will look at we will simply called it the *Laplacian matrix.*

Example 12.1 Revisited. For the graph given in the introduction we have:

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

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

$$L = D - A = \begin{bmatrix} 2 & 0 & -1 & 0 & 0 & -1 & 0 \\ 0 & 3 & -1 & -1 & -1 & 0 & 0 \\ -1 & -1 & 3 & 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 2 & 0 & 0 & -1 \\ 0 & -1 & 0 & 0 & 2 & 0 & -1 \\ -1 & 0 & -1 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & -1 & -1 & 0 & 2 \end{bmatrix}$$

Both the adjacency matrix and the Laplacian matrix contain all information about the graph and both can be used to analyze the graph.

12.3 Basic Graph Analysis

The adjacency matrix of a graph can give us some interesting facts about that graph.

Definition 12.3.0.1. A *walk* from vertex i to vertex j is an alternating series of connected vertices and edges that starts with vertex i and ends with vertex j. There are no restrictions on repeating edges or vertices.

Theorem 12.3.0.1. If A is the $n \times n$ adjacency matrix of a graph with n vertices then for every integer $k \geq 1$, the ij-entry of A^k equals the number of walks of length k from vertex i to vertex j.

Proof. The proof proceeds by induction.

The $k = 1$ case is clear by definition of A.

Assume that the statement is true for A^k and look at the ij-entry of A^{k+1}. By the definition of matrix multiplication

$$(A^{k+1})_{ij} = (A^k)_{i1}a_{1j} + (A^k)_{i2}a_{2j} + ... + (A^k)_{in}a_{nj}$$

Since $(A^k)_{il}$ equals the number of walks of length k from vertex i to vertex l and $a_{lj} = 1$ iff there is an edge from vertex l to vertex j (and 0 otherwise) it follows that the right side above equals the total number of walks of length $k + 1$ from vertex i to vertex j as desired. $\square$

Example 12.2. Consider the A for the introductory graph. We have:

$$A^5 = \begin{bmatrix} 12 & 8 & 19 & 8 & 8 & 13 & 2 \\ 8 & 2 & 27 & 25 & 25 & 8 & 0 \\ 19 & 27 & 14 & 2 & 2 & 19 & 14 \\ 8 & 25 & 2 & 0 & 0 & 8 & 18 \\ 8 & 25 & 2 & 0 & 0 & 8 & 18 \\ 13 & 8 & 19 & 8 & 8 & 12 & 2 \\ 2 & 0 & 14 & 18 & 18 & 2 & 0 \end{bmatrix}$$

Thus, for example, there are 27 walks of length 5 from vertex 2 to vertex 3 (and from 3 to 2) and there are 13 walks of length 5 from vertex 1 to vertex 6 (and from 6 to 1).

This theorem gives us an interesting use of A^3. First, a definition:

Definition 12.3.0.2. The *trace* of a square matrix M, denoted $\mathrm{tr}(M)$, equals the sum of the entries along the main diagonal.

Then we have the following:

Theorem 12.3.0.2. Thus the number of triangles in a graph equals $\frac{1}{6}\mathrm{tr}(A^3)$.

Proof. A walk of length 3 from a vertex to itself is a triangle, and that triangle actually yields two walks, one in each direction. It follows that if a vertex i is contained in a triangle then $(A^3)_{ii} = 2$. From there we see that $\mathrm{tr}(A^3)$ equals twice the number of vertices contained in triangles. However since each triangle contains three vertices it follows that $\mathrm{tr}(A^3)$ equals six times the number of triangles. $\square$

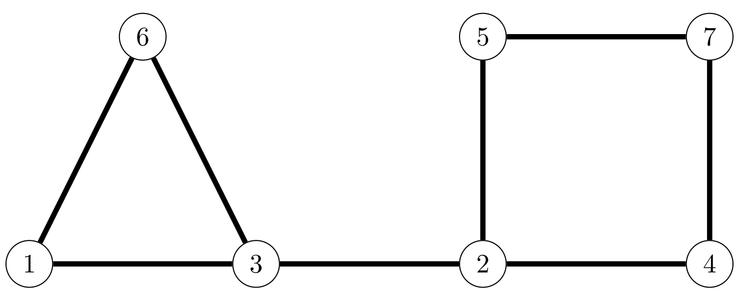
This same approach doesn't work for squares, pentagons, etc. Why not?

12.4 Graph Partitioning

12.4.1 Introduction to Partitioning

Consider the graph from the chapter opening:

Example 12.1 Revisited.



One way we might immediately describe this graph is that it is a square connected to a triangle. What we are doing when we see this is we are breaking the graph into those two subgraphs.

This process, of breaking a graph into two or more subgraphs, has generic uses when analyzing networks.

Consequently what we'd like to know is if there is a way of doing this easily.

In order to investigate we first need some more definitions.

Definition 12.4.1.1. Given a graph G with n vertices $V = \{1, 2, ..., n\}$ For an integer $k \geq 2$ a *k-partition* of G is an partition of the vertices into into k subsets V_1, ..., V_k such that the subsets do not overlap and their union is all of V. We will write $P = (V_1, V_2, ..., V_k)$. A 2-partition is often just called a *partition*.

Example 12.1 Revisited. For example the partition we intuitively saw with our starting graph could be denoted $P = (\{1, 3, 6\}, \{2, 4, 5, 7\})$.

Sometimes we describe a partition by describing which edge(s) would need to be removed in order to disconnect the graph into the resulting pieces. However we're not actually removing the edges, just indicating that they would do the job.

Example 12.1 Revisited. For example we might say that our starting graph's partition could be partitioned by removing the $(2, 3)$ edge.

Definition 12.4.1.2. For a partition $P = (V_1, V_2)$ of a graph G we define the *cut* of P, denoted $\text{cut}(P)$, as the number of edges joining a vertex in V_1 with a vertex in V_2.

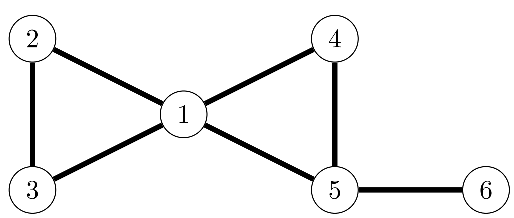
Example 12.1 Revisited. In our opening example we would have $\text{cut(P)} = 1$ because there is only one edge to count, the $(2, 3)$ edge.

So how might we want to partition a graph? One obvious way is:

Definition 12.4.1.3. A *minimum cut* is a partition P of a graph G in a manner that minimizes $\text{cut}(P)$. In other words it's the minimum number of edges we need to remove to partition the graph.

One problem with a minimum cut is that if there is a stray vertex connected to the rest of the graph by one edge then this would be a minimum cut. This tends to leave the subgraphs unbalanced which is somewhat unsatifactory.

Example 12.3. Consider this example:



A minimum cut can be achieved by removing the $(5, 6)$ edge. However the result (the bow-tie on the left and the single vertex on the right) isn't very satisfactory in a balanced sense.

The usual solution to this is to minimize $\text{cut}(P)$ with the added condition that we try to keep the number of vertices in each of the two remaining subgraphs as equal as possible.

In the above example we might remove the $(1, 4)$ and $(1, 5)$ edges which is more edges than just the $(5, 6)$ edge (two instead of one) but gains the advantage that the partition subsets have equal size.

It is this attempt at a balanced approach we will take, attempting to find a partition $P = (V_1, V_2)$ which minimizes $\text{cut}(P)$ while keeping $|V_1| \approx |V_2|$.

12.4.2 Introduction to the Fiedler Method

The Fiedler Method is an easy way to partition a graph. First we will state the method in its most fundamental form and give some simple examples. Lots of questions will remain unanswered.

Next we will look at what the Fiedler method is actually doing. After that we can look at some more complicated examples.

Lastly we will go through a rigorous proof.

The Fiedler Method is named after Miroslav Fiedler, a Czech mathematician, who worked in graph theory and linear algebra. This method was presented by him in 1973.

12.4.3 Basic Fiedler Method

First, a few definitions and facts about the Laplacian Matrix $L = D - A$. The following are addressed in more detail later but let's just get them out right now.

Fact 12.4.3.1.

If G is a simple connected graph with n vertices and if L is the Laplacian matrix for G then L has n real eigenvalues satisfying

$$0 = \lambda_1 < \lambda_2 \leq \lambda_3 \leq ... \leq \lambda_n$$

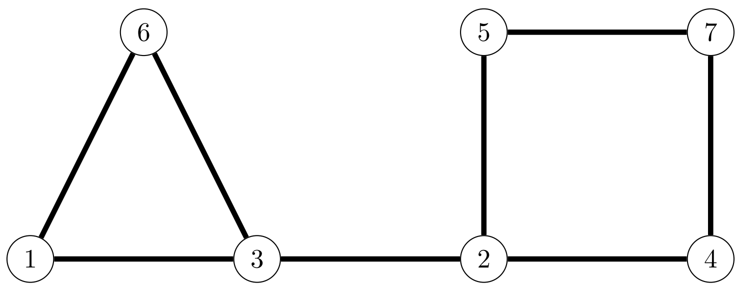
Definition 12.4.3.1. The *Fiedler Value* or the *algebraic connectivity* of a graph is the second smallest eigenvalue of its Laplacian matrix L.

The Fiedler Value gives a measurement as to how well connected the graph is. This value only has meaning when compared to something called the *vertex connectivity* which we won't go into.

Definition 12.4.3.2. A *Fiedler Vector* of a graph is an eigenvector corresponding to the Fiedler Value.

Notice that the eigenspace corresponding to the Fiedler Value may be multidimensional.

Example 12.1 Revisited. In our example:



we saw that:

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

the eigenvalues in order are:

$$0,\ 0.3588,\ 2.0000,\ 2.2763,\ 3.0000,\ 3.5892,\ 4.7757$$

Note that the Fiedler Value is 0.3588. A Fiedler Vector is an eigenvector corresponding to this. Any nonzero multiple of the following unit vector will suffice:

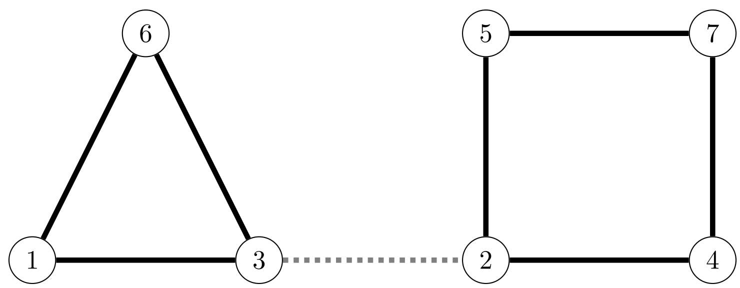
$$\bar{v} = \begin{bmatrix} 0.48 \\ -0.15 \\ 0.31 \\ -0.35 \\ -0.35 \\ 0.48 \\ -0.42 \end{bmatrix}$$

At its most basic, the Fiedler Method basically states that we can achieve a "reasonable" partition into two subgraphs by separating the vertices according to the sign of the values in a Fiedler Vector $\bar{v}$ where each entry corresponds to a vertex. This means we group together the vertices i with $v_i = +$ and we group together the vertices i with $v_i = -$. In the case that $v_i = 0$ we simply have to make a choice.

By "reasonable" we mean that an attempt is made to remove as few edges as possible while keeping the resulting subgraphs of approximately equal size.

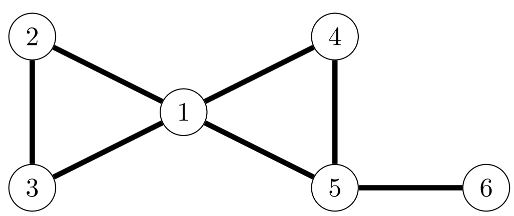
It's worth noting that the Fiedler Method is not perfect, as we'll see, but often the problems that arise can be easily accounted for.

Example 12.1 Revisited. In our example above $v_i = +$ for $i = 1, 3, 6$ and $v_i = -$ for $i = 2, 4, 5, 7$, so that $P = (\{1, 3, 6\}, \{2, 4, 5, 7\})$. This means we separate the vertices accordingly:



This is just what we predicted!

Example 12.4. Let's go back to the bow-tie:



Here we have Laplacian Matrix:

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

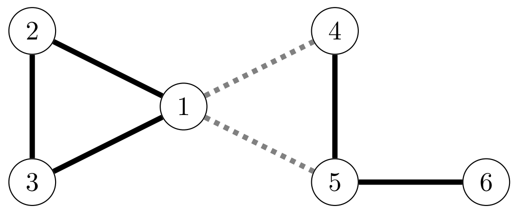
The eigenvalues in order are:

$$0, 0.6314, 1.4738, 3, 3.7877, 5.1071$$

Note that the Fiedler value is 0.6314. A Fiedler Vector is an eigenvector corresponding to this. Any nonzero multiple of the following unit vector will suffice:

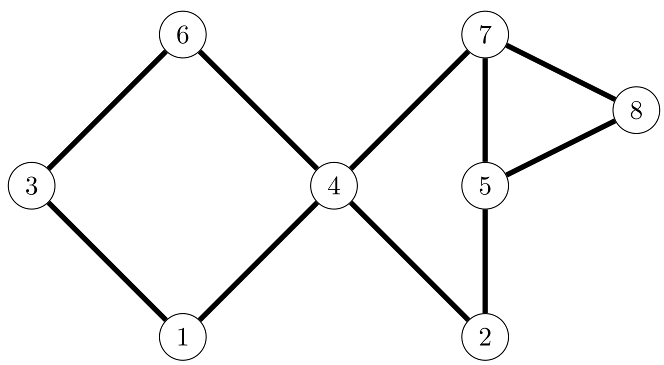
$$\bar{v} = \begin{bmatrix} -0.16 \\ -0.44 \\ -0.44 \\ 0.07 \\ 0.26 \\ 0.71 \end{bmatrix}$$

We separate the vertices accordingly:



This is a more-balanced partition than the minimum cut.

Example 12.5. Consider this graph:



We have

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

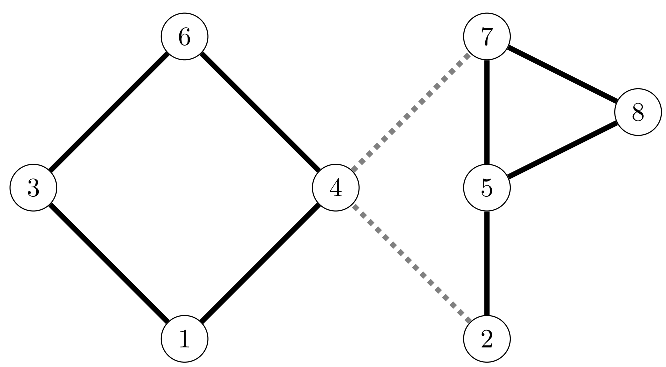
The eigenvalues in order are:

$$0,\ 0.4869,\ 1.6769,\ 2.0000,\ 2.7647,\ 3.4963,\ 4.0000,\ 5.5753$$

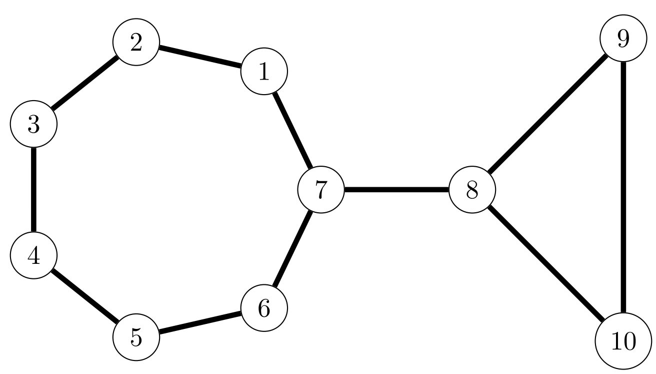
The Fiedler Value is therefore 0.4869. A Fiedler Vector is an eigenvector corresponding to this.

$$\begin{bmatrix} 0.38 \\ -0.20 \\ 0.50 \\ 0.07 \\ -0.38 \\ 0.38 \\ -0.30 \\ -0.44 \end{bmatrix}$$

So a reasonable partition is achieved via $P = (\{1, 3, 4, 6\}, \{2, 5, 7, 8\})$. This requires removing the $(2, 4)$ and $(4, 7)$ edges:



Example 12.6. Consider this graph:



We have

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

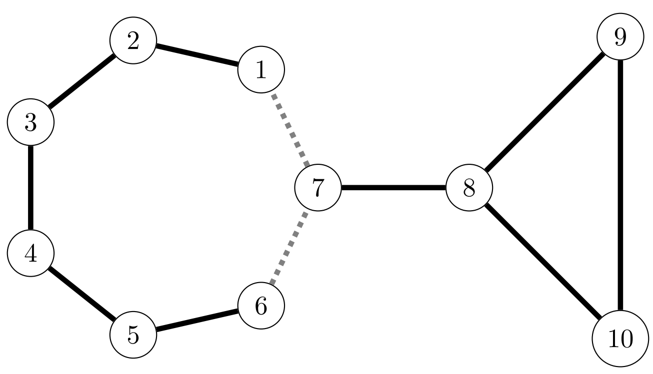
The eigenvalues in order are:

$$0,\ 0.2375,\ 0.7530,\ 1.0000,\ 2.4450,\ 2.5634,\ 3.0000,\ 3.4832,\ 3.8019,\ 4.7159$$

Thus the Fiedler Value is 0.2375. Since this is positive the graph is connected. A Fiedler Vector is an eigenvector corresponding to this. From Matlab:

$$\begin{bmatrix} 0.11 \\ 0.25 \\ 0.33 \\ 0.33 \\ 0.25 \\ 0.11 \\ -0.05 \\ -0.37 \\ -0.49 \\ -0.49 \end{bmatrix}$$

So a reasonable partition is achieved via $P = (\{1,2,3,4,5,6\},\{7,8,9,10\})$. This requires removing the $(1,7)$ and $(6,7)$ edges so $\text{cut}(P) = 2$.



Notice that obtaining a cut of 1 is possible but would leave a much more unbalanced graph. Notice that another partition actually does better than the

Fiedler Method, with $P = (\{1,2,3,4,5\},\{6,7,8,9\})$ having $\text{cut}(P) = 2$ and also $|V_1| = |V_2|$. We'll look at how this relates to the Fiedler Method later.

12.4.4 What are We Wishing For?

Earlier we commented that ideally for a partition $P = (V_1, V_2)$ of a graph G we would like to minimize $\text{cut}(P)$ while keeping $|V_1| \approx |V_2|$.

To formalize this first observe that a partition of a graph G with n vertices can be defined by choosing a vector $\bar{x} \in \mathbb{R}^n$ with each entry $x_i = \pm 1$. Having such a vector we can then create a partition by taking the vertices i with $x_i = +1$ as one subset and the vertices i with $x_i = -1$ as the other subset.

More formally

$$P = (\{i \mid x_i = +1\}, \{i \mid x_i = -1\})$$

Keeping the sizes of the subsets equal amounts to having $\sum_{i=1}^{n} x_i = 0$ and keeping them close amounts to having $\sum_{i=1}^{n} x_i \approx 0$

In what follows, the edge set of a graph only includes each edge once so for example if $(1,2) \in E$ then we don't count $(2,1)$ as different.

Lemma 12.4.4.1. For any partition $P = (V_1, V_2)$ of a graph G with edge set E we have

$$\text{cut}(P) = \frac{1}{4} \sum_{(i,j)\in E} (x_i - x_j)^2$$

Proof. Consider that

$$\begin{aligned}
\sum_{(i,j)\in E} (x_i - x_j)^2 &= \sum_{\substack{(i,j)\in E \\ x_i = -x_j}} (x_i - x_j)^2 + \sum_{\substack{(i,j)\in E \\ x_i = x_j}} (x_i - x_j)^2 \\
&= \sum_{\substack{(i,j)\in E \\ x_i = -x_j}} (\pm 2)^2 + \sum_{\substack{(i,j)\in E \\ x_i = x_j}} (0)^2 \\
&= 4\,\text{cut}(P)
\end{aligned}$$

□

The $\frac{1}{4}$ doesn't matter for minimizing so the goal can be rephrased as trying to minimize $\sum_{(i,j)\in E} (x_i - x_j)^2$ with the conditions that $\sum_{i=1}^{n} x_i \approx 0$. and $x_i = \pm 1$.

Notice that this is computationally intensive and involves checking all possible combinations of the x_i.

For example if the graph has 10 vertices then there are $2^{10} = 1024$ possible $\bar{x}$.
and if the graph has 100 vertices then there are $2^{100} = 1267650600228229401496703205376$ possible $\bar{x}$.

In addition we need to decide how close we want $|V_1| \approx |V_2|$ when looking for a trade-off in minimizing the cut value.

What we do instead is relax the requirement somewhat.

12.4.5 What are We Getting?

Old Goal: Choose $\bar{x}$ to minimize $\sum_{(i,j)\in E}(x_i - x_j)^2$ with the conditions that $\sum_{i=1}^{n} x_i \approx 0$ and $x_i = \pm 1$.

New Goal: Choose $\bar{x}$ to minimize $\sum_{(i,j)\in E}(x_i - x_j)^2$ with the conditions that $\sum_{i=1}^{n} x_i = 0$ and $\sum_{i=1}^{n} x_i^2 = n$.

Notice that the New Goal is a slightly weaker version of the Old Goal. The Old Goal would satisfy the New Goal but not necessarily the reverse.

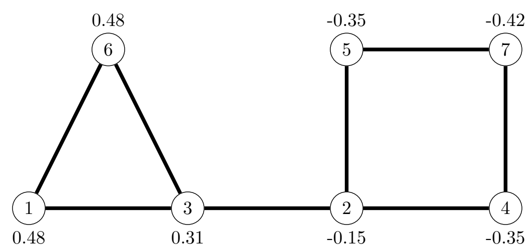
What will the x_i values in this $\bar{x}$ mean? Consider the following observations:

- The first condition makes sure that all the x_i average to 0, meaning that they should be spread out around 0.
- The second condition prevents all the x_i from being too close to 0 and prevents any one x_i from being $\sqrt{n}$ or larger.
- If two vertices i and j are connected by an edge then minimizing the objective means keeping the corresponding x_i and x_j close so that they only contribute a small value to the objective.
- If two vertices i and j are not connected by an edge then the corresponding x_i and x_j can be further apart.
- Expanding on the previous two bullets slightly, suppose there were two numerically distant clusters of x_i values corresponding to two clusters of vertices. If those two clusters of vertices were connected by many edges then this would contribute a large value to the objective function. However since we're minimizing the objective function this will tend to not happen. What this means is that any two numerically distant clusters of x_i values must correspond to weakly connected clusters of vertices.
- Lastly note that from the previous bullets we can see that it's not reasonable to have a few values less than 0 and many values more than 0 (or the reverse) because these would not average out to 0 unless there were a large gap, which can't exist.

The practical upshot of all of this is that all x_i will be spread around the interval $(-\sqrt{n}, \sqrt{n})$ in such a way that clusters of strongly connected vertices will tend to have close x_i values and weakly connected clusters will tend to have distant x_i values.

Thus we can use the clustering of the x_i values to partition the graph. Using the cutoff of 0 is convenient because it is the average but it may not be ideal in certain situations, or it may be just one of many options.

Example 12.7. Here is our original example with the values from the Fiedler vector labeling the vertices.



Notice the largest gap in the value is the break between positives and negatives and is where we partitioned the graph.

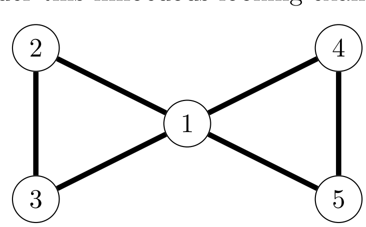
An obvious way to split is to take the vertices corresponding to positive values and those correspoding to negative values and to put 0 in either one group or the other but there are other choices, including splitting at the median.

12.4.6 More and Trickier Examples

We've seen what happens in clear-cut examples but the nature of the Fiedler Vector, whose values indicate a sort of connectedness, lends itself to more than just simple partitions. Now we will look at some examples in which:

- There is a 0 in the Fiedler Vector.
- Repeated values in the Fiedler Vector might yield choices.
- We might choose a k-partition with $k > 2$.
- The eigenspace corresponding to the Fiedler Value has dimension greater than 1.
- The Fiedler Vector can give insight into the graph's structure.
- The Fiedler Vector fails to be helpful at all!

Example 12.8. Consider this innocuous looking example:



The Laplacian matrix is

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

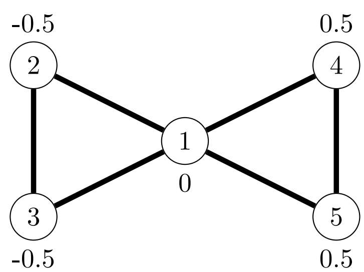
The eigenvalues in order are

$$0,\ 1,\ 3,\ 3,\ 5$$

so the Fiedler Value is 1. A Fiedler Vector is:

$$\begin{bmatrix} 0 \\ -0.5 \\ -0.5 \\ 0.5 \\ 0.5 \end{bmatrix}$$

Here is the graph with the vertices labeled.



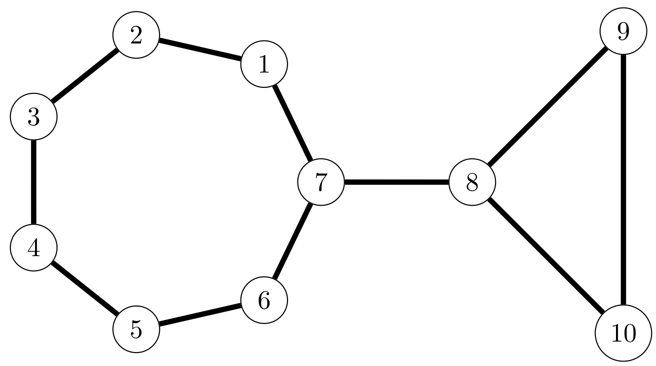
It's clear both from the graph and from the vector that the 1 vertex is difficult to categorize.

Even though the Fiedler Method doesn't explicitly tell us what to do with that vertex the way that the values are spread out makes our options fairly clear. We can either partition as $(\{2,3,1\},\{4,5\})$ or as $(\{2,3\},\{1,4,5\})$.

We can even see this sort of behavior (options!) arising when the Fiedler method does work.

Example 12.6 Revisited.

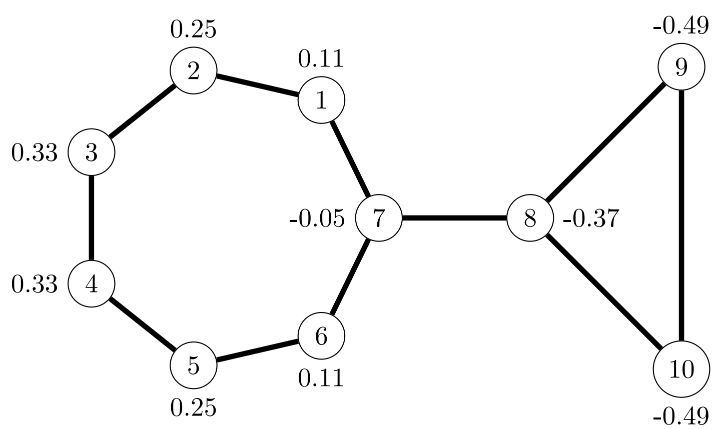
Consider the earlier example:



Here is the Fiedler vector from earlier:

$$\begin{bmatrix} 0.11 \\ 0.25 \\ 0.33 \\ 0.33 \\ 0.25 \\ 0.11 \\ -0.05 \\ -0.37 \\ -0.49 \\ -0.49 \end{bmatrix}$$

Here is the graph with the vertices correspondingly labeled:

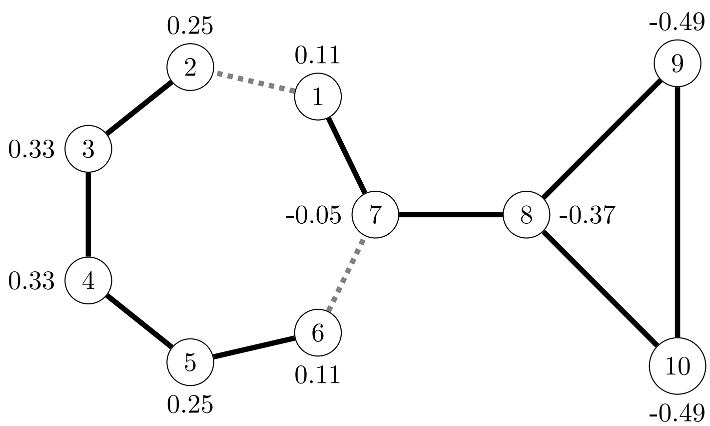


Our choice to separate by the negative and positive values is a classic approach.

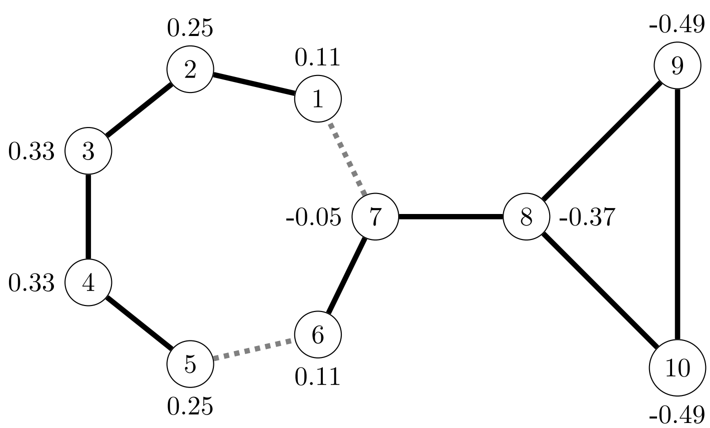
However other approaches may arise. Here is the same Fiedler Vector we saw before except with the vector entries placed in increasing order (equal values chosen arbitrarily) with the vertex number (that is, the vector index) labeling each.

Vertex	Entry
9	-0.49
10	-0.49
8	-0.37
7	-0.05
1	0.11
6	0.11
2	0.25
5	0.25
3	0.33
4	0.33

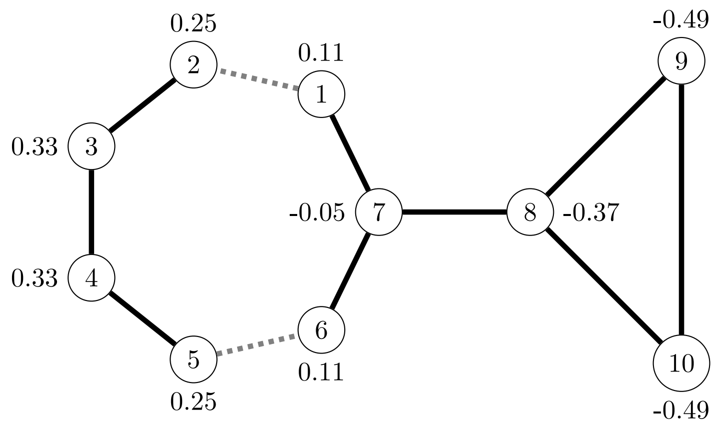
Another alternative would be to take the smallest half of the entries. Since 0.11 appears twice we could split those two up, meaning we could take vertex 1 with the first half and overtex 6 with the second half, or the reverse, giving either: $(\{9, 10, 8, 7, 1\}, \{6, 2, 5, 3, 4\})$



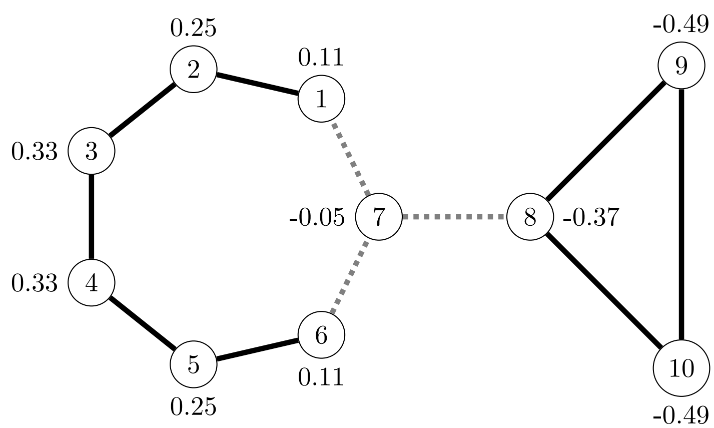
or $(\{9, 10, 8, 7, 6\}, \{1, 2, 5, 3, 4\})$



Or we could take both 1 and 6 with the first half, giving $(\{9, 10, 8, 7, 1, 6\}, \{2, 5, 3, 4\})$

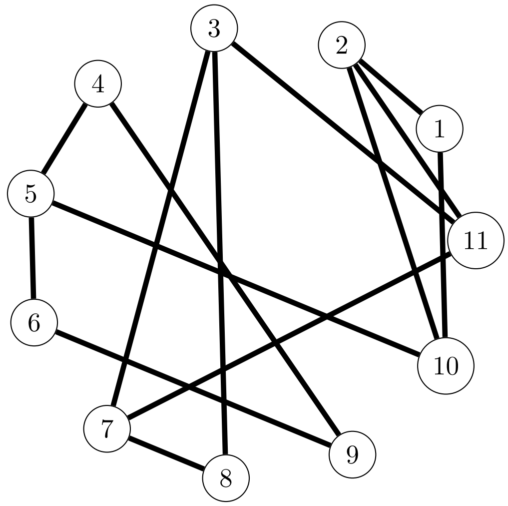


Or we could argue that since the 7 vertex has value very close to 0 perhaps it should just be left alone. Then we would partition into more than two subgraphs, giving $(\{9, 10, 8\}, \{7\}, \{1, 6, 2, 5, 3, 4\})$.



The Fiedler Vector can also help us figure out the structure of a graph which is not given in an obvious way.

Example 12.9. Consider this example:



The Laplacian matrix is not shown but the eigenvalues are

$$0.0000, 0.1483, 0.5858, 2.0000, 2.2170, 2.3820, 3.4142, 3.6913, 4.0000, 4.6180, 4.9434$$

and so the Fiedler Value is 0.1483.

The Fiedler Vector is:

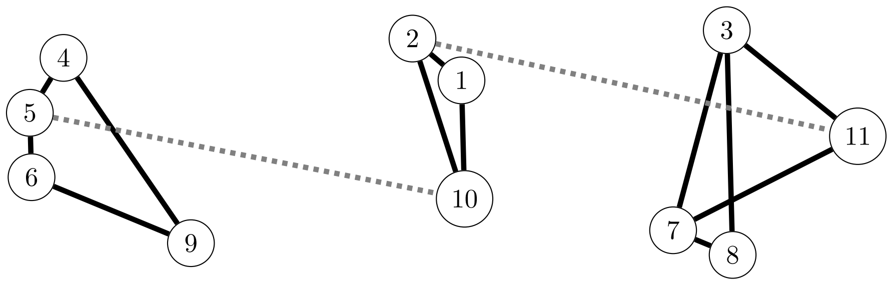
$$\begin{bmatrix} 0.0000 \\ 0.0726 \\ 0.3626 \\ -0.3626 \\ -0.2798 \\ -0.3626 \\ 0.3626 \\ 0.3917 \\ -0.3917 \\ -0.0726 \\ 0.2798 \end{bmatrix}$$

Sorted with the corresponding vertex numbers:

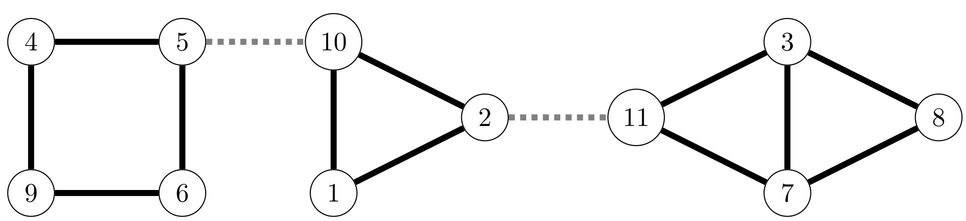
Vertex	Entry
9	-0.3917
6	-0.3626
4	-0.3626
5	-0.2798
10	-0.0726
1	0
2	0.0726
11	0.2798
7	0.3626
3	0.3626
8	0.3917

The values here are basically divided into three groupings according to the vertices ($\{9, 4, 6, 5\}, \{10, 1, 2\}, \{11, 3, 7, 8\}$) so it might make sense to partition the graph into three subgraphs.

Here is the graph redrawn with those groupings separated and with the cut edges as dotted lines. Basically I dragged the first group left and the third group right from the original graph. The underlying structure becomes much more clear now!

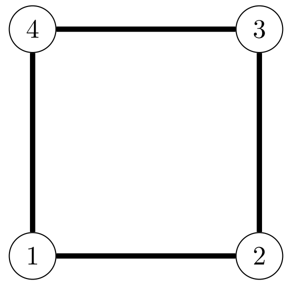


If we clean it up a bit:



Here's a particularly messy example which looks so nice at the start.

Example 12.10. Consider the simple square:



The Laplacian matrix for this graph is:

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

The eigenvalues are $\{0, 2, 2, 4\}$ so the Fiedler Value has multiplicity 2 and hence has a two dimensional subspace. This subspace is spanned by the two correponding vectors:

$$\begin{bmatrix} 0.7071 \\ 0 \\ -0.7071 \\ 0 \end{bmatrix} \text{ and } \begin{bmatrix} 0 \\ 0.7071 \\ 0 \\ -0.7071 \end{bmatrix}$$

However this gives lots of confusing options:

If we use the first vector then vertices 1 and 3 are separate but depending on what we do with vertices 2 and 4 we could get either $(\{1\}, \{2, 3, 4\})$, $(\{1, 2\}, \{3, 4\})$, $(\{1, 4\}, \{2, 3\})$ or $(\{1, 2, 4\}, \{3\})$.

If we use the second vector then vertices 2 and 4 are separate but depending on what we do with vertices 1 and 3 we could get either $(\{2\}, \{1, 3, 4\})$, $(\{1, 2\}, \{3, 4\})$, $(\{2, 3\}, \{1, 4\})$ or $(\{1, 2, 3\}, \{4\})$.

Any linear combination using nonzero multiples of both vectors will lead to a Fiedler vector of the form:

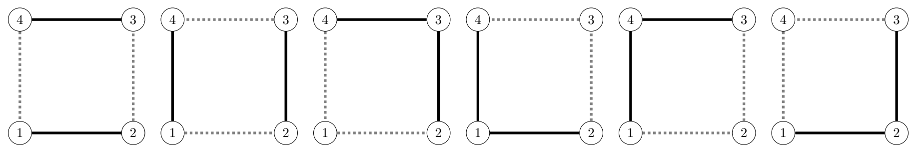
$$\begin{bmatrix} + \\ + \\ - \\ - \end{bmatrix} \text{ or } \begin{bmatrix} + \\ - \\ - \\ + \end{bmatrix} \text{ or } \begin{bmatrix} - \\ + \\ + \\ - \end{bmatrix} \text{ or } \begin{bmatrix} - \\ - \\ + \\ + \end{bmatrix}$$

These yield only the two partitions $(\{1, 2\}, \{3, 4\})$ and $(\{1, 4\}, \{2, 3\})$.

Thus in total there are six possibilities:

$$(\{1, 2\}, \{3, 4\}), (\{1, 4\}, \{2, 3\}), (\{1\}, \{2, 3, 4\}), (\{1, 2, 4\}, \{3\}), (\{2\}, \{1, 3, 4\}),$$
$$(\{1, 2, 3\}, \{4\})$$

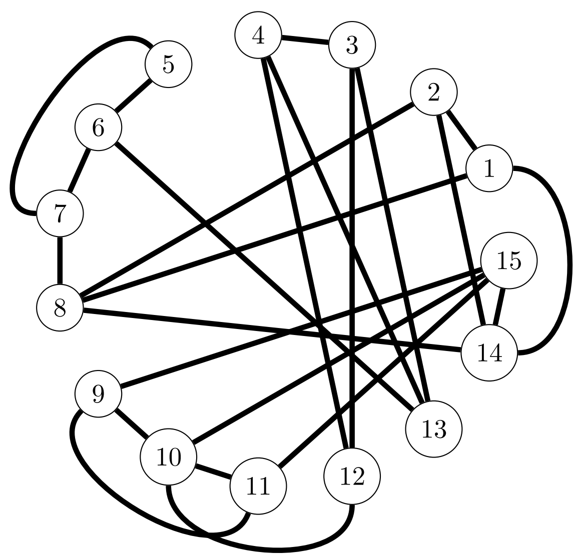
with corresponding pictures:



In this example the Fiedler method can't decide other than ensuring that either 1 and 3 are separate or 2 and 4 are separate, which actually seems reasonable, but beyond that options abound.

Here's an example where the Fiedler vector doesn't do the best job of partitioning the graph.

Example 12.11. Consider the graph:



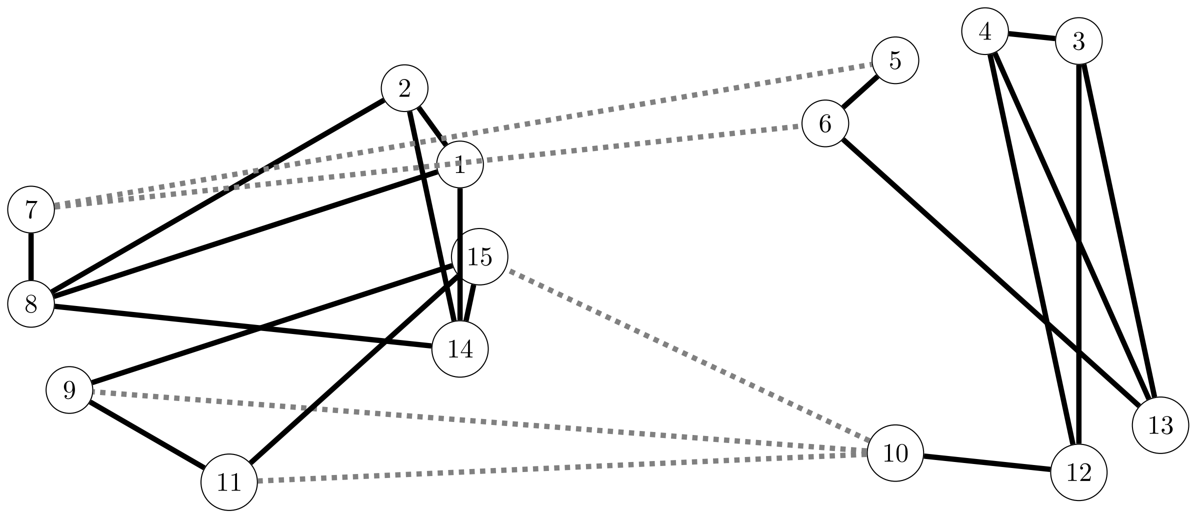
The Fiedler Value is 0.3424 and the Fiedler Vector is:

$$\begin{bmatrix} -0.3778 \\ -0.3778 \\ 0.3819 \\ 0.3819 \\ 0.0187 \\ 0.1027 \\ -0.0718 \\ -0.3121 \\ -0.0177 \\ 0.0520 \\ -0.0177 \\ 0.3070 \\ 0.3261 \\ -0.3142 \\ -0.0813 \end{bmatrix}$$

Sorted with the corresponding vertex numbers:

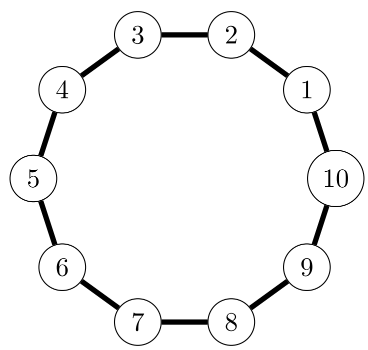
Vertex	Entry
1	-0.3778
2	-0.3778
14	-0.3142
8	-0.3121
15	-0.0813
7	-0.0718
9	-0.0177
11	-0.0177
5	0.0187
10	0.0520
6	0.1027
12	0.3070
13	0.3261
3	0.3819
4	0.3819

The Fiedler Method does a pretty mediocre job of dividing the graph into two subgraphs using $(\{1, 2, 14, 8, 15, 7, 9, 11\}, \{5, 10, 6, 12, 13, 3, 4\})$ as shown here:



12.4.7 Why Might the Fiedler Method Have Issues

The Fiedler vector tends to have issues with graphs in which it's difficult to measure distance between vertices. For example in the cycle:



It's clear that vertices 1 and 6 are far apart but are 1 and 2 close or not? Intuitively they are but by some measurement (around the wrong way) they're not. The mathematics in the Fiedler Method tends to stumble on things like this.

12.4.8 Why Does the Fiedler Vector Do This?

The final thing we need to address is why the Fiedler Method accomplishes our relaxed goal from a mathematical standpoint.

Lemma 12.4.8.1. Let G be a graph with n vertices and let L be its Laplacian matrix. Then L is orthogonally diagonalizable and the eigenvalues are all nonnegative and hence there exists an orthonormal basis of eigenvectors $\bar{v}_1$, $\bar{v}_2$, ..., $\bar{v}_n$ corresponding to eigenvalues $0 \leq \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$.

Proof. Since L is symmetric most of this follows from the Spectral Theorem. Proving that the eigenvalues are all nonnegative takes a bit more work but is omitted. $\square$

From here on whenever we discuss the eigenvalues and eigenvectors of a Laplacian matrix for a graph we'll assume that it is an orthonormal basis from above.

Lemma 12.4.8.2. A graph G is connected iff $\lambda_2 > 0$.

Proof. Omitted. While this is not difficult it takes a bit of time to write down and the proof is largely unrelated to and doesn't provide any insight into how we use it. $\square$

Lemma 12.4.8.3. Let G be a graph with n vertices and let L be its Laplacian matrix. Then we have $\lambda_1 = 0$ and $\bar{v}_1 = \frac{1}{\sqrt{n}}\bar{1}$.

Proof. Since each row of L adds to 0 we have $L\bar{1} = \bar{0}$ and so $L\bar{1} = 0\bar{1}$ and so 0 is an eigenvalue with eigenvector $\bar{1}$ and hence with unit eigenvector $\frac{1}{\sqrt{n}}\bar{1}$. $\square$

Lemma 12.4.8.4. Let G be a graph with n vertices and let L be its Laplacian matrix. For any eigenvalue $\lambda > 0$ of L the entries in any corresponding eigenvector $\bar{v}$ add to 0.

Proof. Let the entries of L be a_{ij}. If $A\bar{v} = \lambda\bar{v}$ then we have

$$\begin{aligned} a_{11}v_1 + a_{12}v_2 + ... + a_{1n}v_n &= \lambda v_1 \\ a_{21}v_1 + a_{22}v_2 + ... + a_{2n}v_n &= \lambda v_2 \\ ... &= ... \\ a_{n1}v_1 + a_{n2}v_2 + ... + a_{nn}v_n &= \lambda v_n \end{aligned}$$

The sum of this system on the left and right yields:

$$(a_{11} + a_{21} + ... + a_{n1})v_1 + (...)v_2 + ... + (...)v_n = \lambda(v_1 + ... + v_n)$$

Since the columns of L sum to zero the left size is zero and hence $v_1 + ... + v_n = 0$. $\square$

Lemma 12.4.8.5. Let G be a graph with n vertices and let L be its Laplacian matrix. If $\bar{x}$ satisfies $\sum_{i=1}^{n} x_i = 0$ then $\bar{x} = \sum_{i=2}^{n} w_i\bar{v}_i$ for appropriate w_i.

Proof. We know that since $\bar{v}_1, \bar{v}_2, ..., \bar{v}_n$ forms a basis for $\mathbb{R}^n$ that for appropriate w_i we may write:

$$\begin{aligned}
\bar{x} &= \sum_{i=1}^{n} w_i \bar{v}_i \\
&= w_1 \bar{v}_1 + \sum_{i=2}^{n} w_i \bar{v}_i \\
&= w_1 \frac{1}{\sqrt{n}} \bar{1} + \sum_{i=2}^{n} w_i \bar{v}_i
\end{aligned}$$

Now then we know that $\bar{x}$ and $\bar{v}_i$ (and hence $w_i \bar{v}_i$) are all in the subspace of $\mathbb{R}^n$ consisting of vectors whose entries add to 0. Consequently the entries of $w_1 \frac{1}{\sqrt{n}} \bar{1}$ must all add to 0 because subspaces are closed under linear combinations. But this implies $w_1 = 0$ as desired. □

Lemma 12.4.8.6. Let G be a graph with n vertices and let L be its Laplacian matrix. If $\bar{x}$ satisfies $\sum\limits_{i=1}^{n} x_i = 0$ then we have

$$\sum_{i=1}^{n} x_i^2 = \sum_{i=2}^{n} w_i^2$$

Proof. Observe that:

$$\begin{aligned}
\sum_{i=1}^{n} x_i^2 &= \bar{x}^T \bar{x} \\
&= \left[\sum_{i=2}^{n} w_i \bar{v}_i \right]^T \left[\sum_{i=2}^{n} w_i \bar{v}_i \right] \\
&= \left[\sum_{i=2}^{n} w_i \bar{v}_i^T \right] \left[\sum_{i=2}^{n} w_i \bar{v}_i \right] \\
&= \sum_{i=2}^{n} \sum_{j=2}^{n} w_i \bar{v}_i^T w_j \bar{v}_j \\
&= \sum_{i=2}^{n} \sum_{j=2}^{n} w_i w_j \bar{v}_i^T \bar{v}_j \\
&= \sum_{i=2}^{n} w_i^2
\end{aligned}$$

□

Lemma 12.4.8.7. Let G be a graph with n vertices and let E be the set of all edges of G. For any vector $\bar{x}$ we have

$$\bar{x}^T A\bar{x} = \sum_{(i,j)\in E} 2x_i x_j$$

Proof. We know that for any $\bar{x}$ by calculation that

$$\bar{x}^T A\bar{x} = \sum_{1\le i\le n, 1\le j\le n} a_{ij} x_i x_j$$

Since $a_{ij} = 1$ iff there is an edge between vertex i and vertex j and 0 otherwise that:

$$\bar{x}^T A\bar{x} = \sum_{(i,j)\in E} 2x_i x_j$$

Here the 2 appears because each pair i, j appears twice in the original sum but we're only counting it once in the set of all edges. □

Lemma 12.4.8.8. Let G be a graph with n vertices, let D be its degree matrix, and let E be the set of all edges of G. For any vector $\bar{x}$ we have

$$\bar{x}^T D\bar{x} = \sum_{(i,j)\in E} (x_i^2 + x_j^2)$$

Proof. Let V be the set of all vertices of G. We know by straightforward calculation that

$$\bar{x}^T D\bar{x} = \sum_{i\in V} d_i x_i^2$$

An alternate way to calculate the degree of any vertex would be to look over the set of all edges and for each edge contribute $+1$ to the degree of each of the two vertices it connects. In order to have the total coefficient of each x_i^2 be the degree of vertex i this means that when we sum over all edges each edge between vertices i and j must contribute $+x_i^2 + x_j^2$ to the total sum.

Thus as desired

$$\bar{x}^T D\bar{x} = \sum_{i\in V} d_i x_i^2 = \sum_{(i,j)\in E} (x_i^2 + x_j^2)$$

□

Lemma 12.4.8.9. Let G be a graph with n vertices, let D be its degree matrix, let E be the set of all edges of G, and let L be its Laplacian matrix. Then for any vector $\bar{x}$ we have

$$\bar{x}^T L\bar{x} = \sum_{(i,j)\in E} (x_i - x_j)^2$$

Proof. We have:

$$\begin{aligned}
\bar{x}^T L\bar{x} &= \bar{x}^T (D - A)\bar{x} \\
&= \bar{x}^T D\bar{x} - \bar{x}^T A\bar{x} \\
&= \sum_{(i,j)\in E} (x_i^2 + x_j^2) - \sum_{(i,j)\in E} 2x_i x_j \\
&= \sum_{(i,j)\in E} (x_i - x_j)^2
\end{aligned}$$

$\square$

Lemma 12.4.8.10. Let G be a graph with n vertices, let E be the set of all edges of G, and let L be its Laplacian matrix. If $\bar{x}$ satisfies $\sum x_i = 0$ then we have:

$$\sum_{(i,j)\in E} (x_i - x_j)^2 = \sum_{i=2}^{n} w_i^2 \lambda_i$$

Proof. Observe that:

$$\begin{aligned}
\sum_{(i,j)\in E}(x_i - x_j)^2 &= \bar{x}^T L \bar{x} \\
&= \left[\sum_{i=2}^{n} w_i \bar{v}_i\right]^T L \left[\sum_{i=2}^{n} w_i \bar{v}_i\right] \\
&= \left[\sum_{i=2}^{n} w_i \bar{v}_i^T\right] L \left[\sum_{i=2}^{n} w_i \bar{v}_i\right] \\
&= \sum_{i=2}^{n}\sum_{j=2}^{n} w_i \bar{v}_i^T L w_j \bar{v}_j \\
&= \sum_{i=2}^{n}\sum_{j=2}^{n} w_i w_j \bar{v}_i^T L \bar{v}_j \\
&= \sum_{i=2}^{n}\sum_{j=2}^{n} w_i w_j \bar{v}_i^T \lambda_j \bar{v}_j \\
&= \sum_{i=2}^{n}\sum_{j=2}^{n} w_i w_j \lambda_j \bar{v}_i^T \bar{v}_j \\
&= \sum_{i=2}^{n} w_i^2 \lambda_i
\end{aligned}$$

□

Theorem 12.4.8.1. The entries in a Fiedler Vector obtain the desired goal.

Proof. The goal is to select $\bar{x}$ which minimizes $\sum\limits_{(i,j)\in E}(x_i - x_j)^2$ with the conditions that $\sum\limits_{i=1}^{n} x_i^2 = n$. and $\sum\limits_{i=1}^{n} x_i = 0$. Accordingly this means we wish to mimize $\sum\limits_{i=2}^{n} w_i^2 \lambda_i$ with the conditions that $\sum\limits_{i=2}^{n} w_i^2 = n$ and $\sum\limits_{i=1}^{n} x_i = 0$.

Given that $\lambda_2 \leq \lambda_3 \leq ... \leq \lambda_n$, this will be accomplished by setting $w_2 = \sqrt{n}$ and $w_3 = ... = w_n = 0$.

From here we get $\bar{x} = \sqrt{n}\bar{v}_2$. Since this $\bar{x}$ is an eigenvector of $\bar{L}$ corresponding to λ_2 the entries add to 0.

This vector is a Fiedler vector. Of course since $\bar{v}_2$ is simply a multiple of this which scales the values, we can use $\bar{v}_2$ itself instead.

□

12.5 Matlab

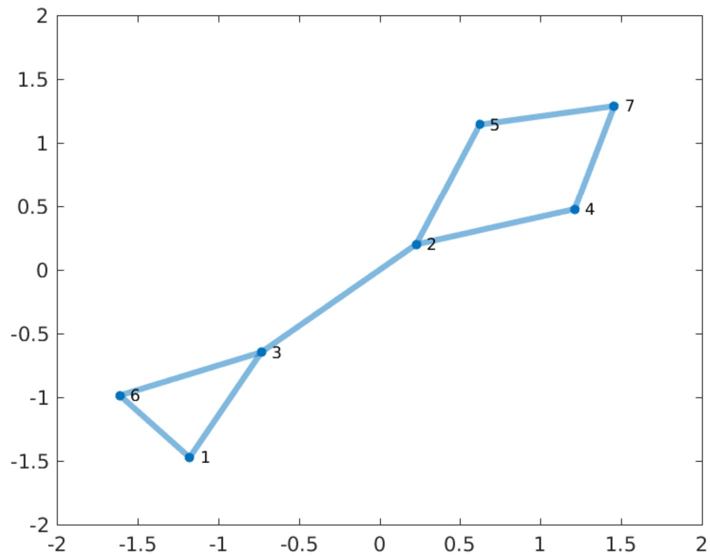
Matlab can plot a graph from the adjacency matrix. It does a pretty reasonable job of arranging the vertices so the graph is comprehensible. First, the following function m-file will create the adjacency matrix for a graph given a matrix of edges and a total number of vertices:

```
function M = createadjacency(v,n)
  % Create the Adjacency Matrix for a graph.
  % Usage:
  % createadjacency([1,2;2,3;1;4],5)
  % Will create a graph with 5 vertices
  % and edges joining 1-2, 2-3 and 1-4.
  M = zeros(n,n);
  for i = 1:length(v)
    M(v(i,1),v(i,2)) = 1;
    M(v(i,2),v(i,1)) = 1;
  end
end
```

In order to plot this graph in Matlab we first create the graph object and then we plot it. Here's the example which started the chapter:

```
>> A = createadjacency([1,3;1,6;3,6;3,2;2,5;5,7;7,4;2,4],7);
>> G = graph(A);
>> plot(G,'LineWidth',3)
```

This produces the image where I've thickened the lines a bit. There's currently no easy way to change the label size in Matlab.



The following function m-file will create the Laplacian matrix for a graph. It's just a slight modification on the one above:

```
function M = createlaplacian(v,n)
  % Create the Laplacian Matrix for a graph.
  % Usage:
  % createlaplacian([1,2;2,3;1;4],5)
  % Will create a graph with 5 vertices
  % and edges joining 1-2, 2-3 and 1-4.
  M = zeros(n,n);
  for i = 1:length(v)
    M(v(i,1),v(i,2)) = -1;
    M(v(i,2),v(i,1)) = -1;
  end
  for i = 1:n
    M(i,i) = -1*sum(M(:,i));
  end
end
```

Then we can find a Fiedler Vector easily. Here we examine the eigenvalues first, notice that the Fieder Value has multiplicity one so we can take any multiple of the corresponding eigenvector, we just look at the eigenvector Matlab gives:

```
>> L = createlaplacian([1,3;1,6;3,6;3,2;2,5;5,7;7,4;2,4],7);
>> [p,d] = eig(L);
>> diag(d)
ans =
   -0.0000
    0.3588
    2.0000
    2.2763
    3.0000
    3.5892
    4.7757
>> p(:,2)
ans =
    0.4801
   -0.1471
    0.3078
   -0.3482
   -0.3482
    0.4801
   -0.4244
```

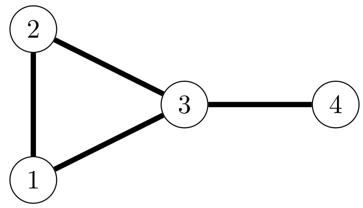
To order this vector and attach the index numbers is easy too:

```
>> L = createlaplacian([1,2;1,10;2,10;2,11;3,7;3,8;3,11;
4,5;4,9;5,6;5,10;6,9;7,8;7,11],11);
>> [p,d] = eig(L);
>> v = p(:,2);
>> sortrows(horzcat(v,[1:size(v)]'))
ans =
   -0.3917    9.0000
   -0.3626    6.0000
   -0.3626    4.0000
   -0.2798    5.0000
   -0.0726   10.0000
    0.0000    1.0000
    0.0726    2.0000
    0.2798   11.0000
    0.3626    7.0000
    0.3626    3.0000
    0.3917    8.0000
```

The `[1:size(v)]` command creates a horizontal vector with entries `1` up to the length of `v`. The `'` does the transpose so it's vertical just like `v`. The `horzcat` command concatenates them horizontally, putting them together. The `sortrows` command sorts each row by the first column.

12.6 Exercises

Exercise 12.1. Consider the following graph:

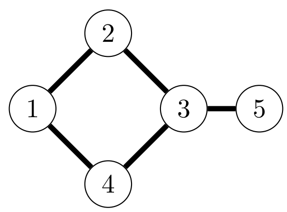


(a) Find the number of walks of length 3 from vertex 1 to vertex 3.

(b) Find the number of walks of length 20 from vertex 2 to vertex 4.

(c) There are no walks of length 3 from vertex 4 to itself. Rather than using A^3, explain intuitively why this is.

Exercise 12.2. Consider the following graph:



(a) Find the number of walks of length 3 from vertex 1 to vertex 2.

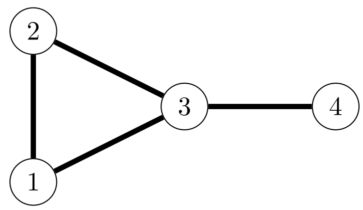
(b) Find the number of walks of length 10 from vertex 2 to vertex 4.

(c) Examine the number of walks of length k from vertex 3 to vertex 5 for various even k. What do you notice? Give an intutive explanation for this.

(d) Examine the number of walks of length k from vertex 2 to vertex 4 for various odd k. What do you notice? Give an intutive explanation for this.

Exercise 12.3. A small theorem in the book shows that the number of triangles in a graph G equals $\frac{1}{6}\text{tr}(A^3)$, where A is the adjacency matrix for G. Why does this not work for squares, etc.? In other words why does the number of squares not equal some multiple of $\text{tr}(A^4)$, why does the number of pentagons not equal some multiple of $\text{tr}(A^5)$, and so on?

Exercise 12.4. Consider the following graph:

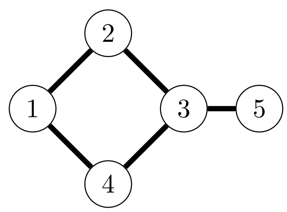


(a) Intuitively how would you partition the graph into two subgraphs in a reasonable manner?

(b) Apply the Fiedler Method to partition the graph.

(c) Do the results match?

Exercise 12.5. Consider the following graph:

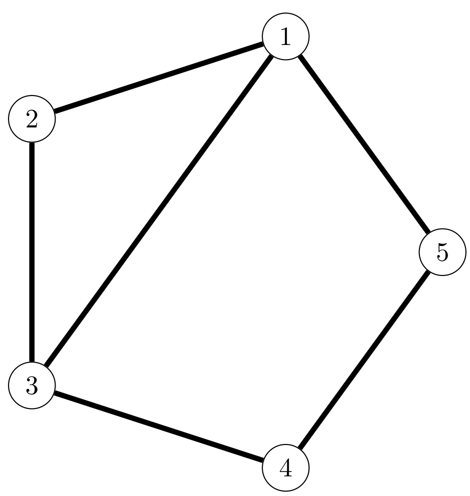


(a) Intuitively how would you partition the graph into two subgraphs in a reasonable manner?

(b) Apply the Fiedler Method to partition the graph.

(c) Do the results match?

Exercise 12.6. Consider the following graph:



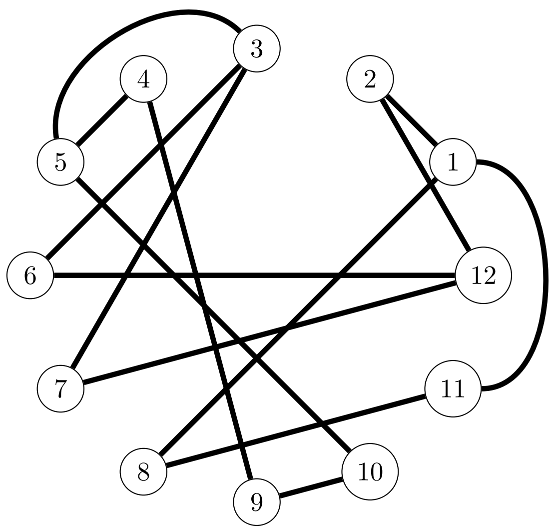
(a) Write down the Laplacian matrix for this graph.

(b) This matrix has eigenvalues 0, 1.3820, 2.3820, 3.6180, 4.6180 with corresponding eigenvectors

$$\begin{bmatrix} -0.447 \\ -0.447 \\ -0.447 \\ -0.447 \\ -0.447 \end{bmatrix}, \begin{bmatrix} -0.195 \\ -0.632 \\ -0.195 \\ 0.512 \\ 0.512 \end{bmatrix}, \begin{bmatrix} 0.372 \\ 0 \\ -0.372 \\ -0.602 \\ 0.602 \end{bmatrix}, \begin{bmatrix} 0.512 \\ -0.632 \\ 0.512 \\ -0.195 \\ -0.195 \end{bmatrix}, \begin{bmatrix} 0.602 \\ 0 \\ -0.602 \\ 0.372 \\ -0.372 \end{bmatrix}$$

Using this, partition the graph with the Fiedler method.

Exercise 12.7. Consider the following graph:



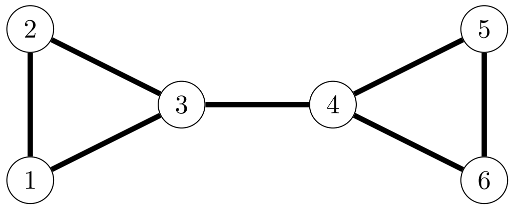
The Fiedler vector is:

$$[0.37, 0.23, -0.12, -0.34, -0.28, -0.04, -0.03, 0.42, -0.36, -0.34, 0.42, 0.06]^T$$

Use this vector to partition the graph into three components and then use this to draw a more understandable picture of the graph.

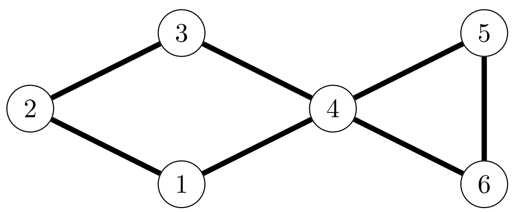
Exercise 12.8. Without doing any calculation match the following graphs with their Fiedler Vectors. Explain your decision.

$G1$ shown here:



$G2$ shown here:

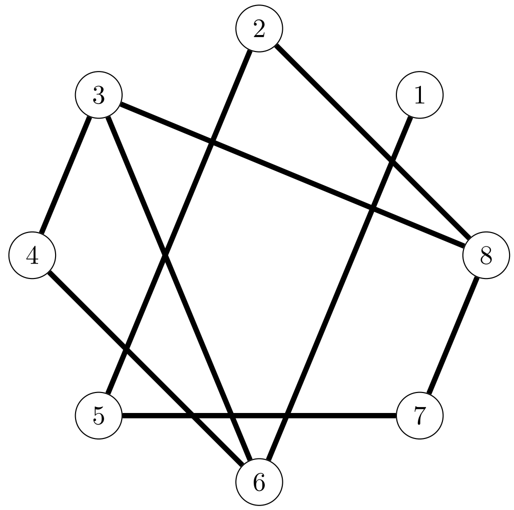
$G3$ shown here:



with:

$$\bar{v} = \begin{bmatrix} -0.56 \\ -0.41 \\ -0.15 \\ 0.15 \\ 0.41 \\ 0.56 \end{bmatrix} \text{ and } \bar{w} = \begin{bmatrix} 0.46 \\ 0.46 \\ 0.26 \\ -0.26 \\ -0.46 \\ -0.46 \end{bmatrix} \text{ and } \bar{x} = \begin{bmatrix} 0.32 \\ 0.51 \\ 0.32 \\ -0.12 \\ -0.51 \\ -0.51 \end{bmatrix}$$

Exercise 12.9. Consider the following graph:

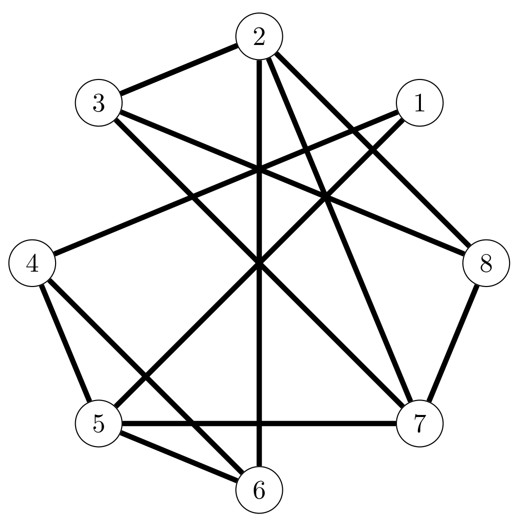


(a) Use the Fiedler Method to partition the graph.

(b) Draw and label the separated components neatly and individually and then indicate with dashed lines the edges that go between them.

(c) From the previous step are there any insights you gain about the structure of the graph?

Exercise 12.10. Consider the following graph:

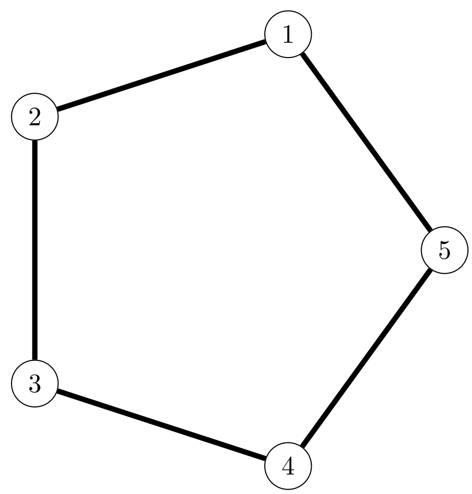


(a) Use the Fiedler Method to partition the graph.

(b) Draw and label the separated components neatly and individually and then indicate with dashed lines the edges that go between them.

(c) From the previous step are there any insights you gain about the structure of the graph?

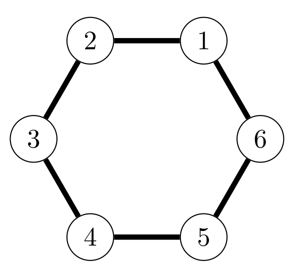
Exercise 12.11. Consider the following graph:



(a) Write down the Laplacian Matrix for the graph.

(b) The Fiedler Vectors span a two-dimensional subspace. Analyze all possible partitions which result. Be methodical.

Exercise 12.12. Consider the following graph:



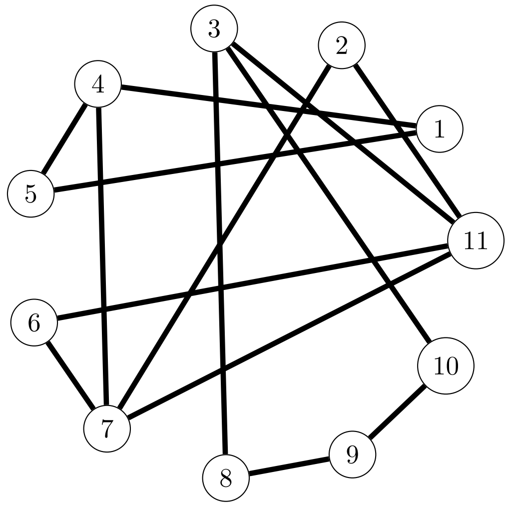
(a) The Fiedler Value has a two-dimensional corresponding eigenspace. Find a basis $\{\bar{v}_1, \bar{v}_2\}$ for the set of all Fiedler Vectors

(b) Any nonzeo linear combination of $\bar{v}_1$ and $\bar{v}_2$ will give a reasonable partition using the Fiedler Method. Experiment to see how many different partitions you can find.

(c) Suppose $\bar{v} = c_1\bar{v}_1 + c_2\bar{v}_2$ for constants c_1, c_2. Assuming neither of the 1-entry and 4-entry of v are 0 explain why vertices 1 and 4 will be in different subgraphs. Repeat for the 2-entry and 5-entry and for the 3-entry and 6-entry.

Exercise 12.13. Let L_n be the Laplacian Matrix for the complete graph K_n (the graph with n vertices with edges between all pairs). By testing various values of n make an educated guess about the eigenvalues of L_n for any n.

Exercise 12.14. Consider the following graph:



(a) Find a Fielder vector.

(b) The values in this Fiedler Vector, when sorted, can be grouped into three separated subsets. Do so.

(c) Use this grouping to partition the graph into three subgraphs.

(d) Draw and label the separated components neatly and individually and then indicate with dashed lines the edges that go between them.

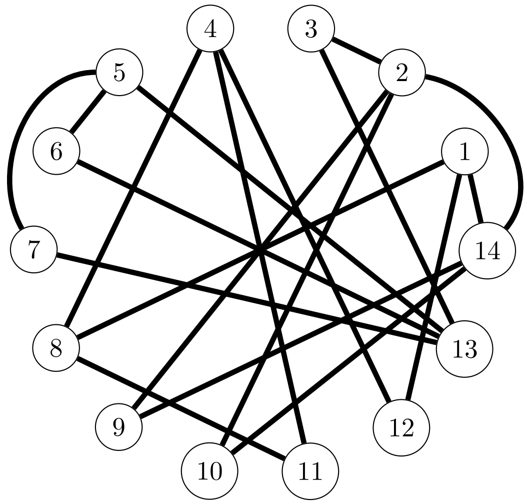
(e) From the previous step are there any insights you gain about the structure of the graph?

Exercise 12.15. Consider the graph with n vertices:

(a) Before doing any calculation what do you think the outcome of the Fiedler Method might be? You may need cases. Justify informally.

(b) Check your hypothesis with a few values of n.

Exercise 12.16. Consider the following graph:



(a) Find a Fiedler vector.

(b) Separate the values into three groups.

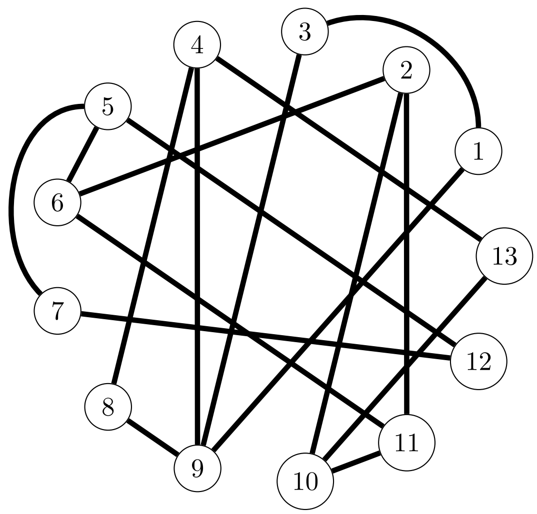
(c) Use these groups to partition the graph into three subgraphs.

(d) Draw and label the separated components neatly and individually and then indicate with dashed lines the edges that go between them.

(e) Which vertex seems like the most critical and why?

(f) From the previous step are there any insights you gain about the structure of the graph?

Exercise 12.17. Suppose the following graph shows all of the people in a small part of a social network. An edge connecting two people indicates that they are friends.



(a) Use the Fiedler Method to identify the group which is most strongly connected to Person 10.

(b) Why is this method ineffective in terms of providing a reasonable answer to (a)? Hint: Is anyone missing from your answer to (a) that is probably important to Person 10?

Exercise 12.18. Suppose a small LAN (local area network) consists of ten computers connected as follows:

- C1 is connected to C8, C9, C10
- C2 is connected to C5, C6.
- C3 is connected to C4, C9.
- C4 is connected to C3, C9.
- C5 is connected to C2, C7.
- C6 is connected to C2, C7, C8.
- C7 is connected to C5, C6.
- C8 is connected to C1, C6, C10.
- C9 is connected to C1, C3, C4.
- C10 is connected to C1, C8.

(a) Write down the Laplacian Matrix for this graph. You don't need to draw the graph!

(b) Find the Fiedler Vector and re-order the entries in increasing order.

(c) Partition the graph into some obvious number of subgraphs.

(d) Draw each of the subgraphs neatly and then use dashed lines to represent the edges that go between them.

(e) From the previous step are there any insights you gain about the structure of the graph?

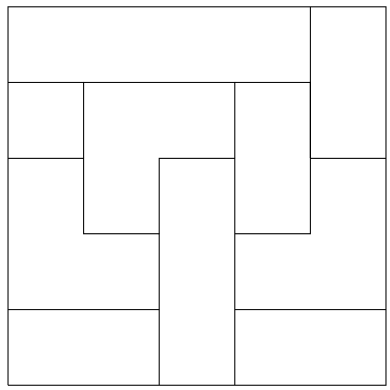
Exercise 12.19. Suppose a small LAN (local area network) consists of ten computers connected as follows:

- C1 is connected to C6.
- C2 is connected to C3, C7, C8 and C9.
- C3 is connected to C2, C4 and C7.
- C4 is connected to C3, C5, C6 and C7.
- C5 is connected to C4 and C6.
- C6 is connected to C1, C4 and C5.
- C7 is connected to C2, C3, C4 and C9.
- C8 is connected to C2, C9 and C10.
- C9 is connected to C2, C7 and C8.
- C10 is connected to C8.

(a) If a network technician wishes to divided these into two groups in order to connect two backup power supplies how should this be done using the Fiedler Method?

(b) If we define the most important links as those that would be removed using the Fiedler Method what are the most important links in this network?

Exercise 12.20. The following is a simplified map of some countries. What we'd like to do is divide the countries into two subsets in a way that tries to balance the number of countries in each subset while minimizing the number of border crossings between subsets.



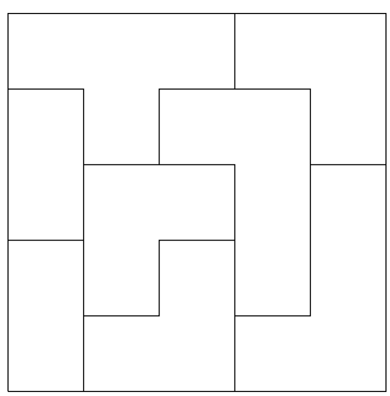
(a) Create a graph from this map by assigning a vertex for each country and connecting two vertices by an edge if the two countries share a border.

(b) Use the Fiedler Method to partition the graph.

(c) Explain in terms of the map what the Fiedler Method has attempted to do.

(d) Shade one subset of the countries in accordance with the result.

Exercise 12.21. The following is a simplified map of some countries. What we'd like to do is divide the countries into two subsets in a way that tries to balance the number of countries in each subset while minimizing the number of border crossings between subsets.



(a) Create a graph from this map by assigning a vertex for each country and connecting two vertices by an edge if the two countries share a border.

(b) Use the Fiedler Method to partition the graph.

(c) Explain in terms of the map what the Fiedler Method has attempted to do.

(d) Shade one subset of the countries in accordance with the result.

Exercise 12.22. A class of ten students needs to be split up. The goal is to get two groups of size as close as possible while minimizing the number of friendships that must be broken up. If the friendships are given in the following table use the Fiedler method to split up the class. How many friendships must be broken up? Draw the two resulting friend networks and indicate with dotted lines where the broken friendships are.

	Austin	Beth	Charlie	Dana	Erik	Fiona	Greg	Helen	Ian	Justin
Austin		✓	✓							✓
Beth	✓				✓					✓
Charlie	✓			✓			✓		✓	
Dana			✓		✓				✓	
Erik		✓		✓		✓		✓		✓
Fiona					✓			✓		
Greg			✓					✓		
Helen					✓	✓	✓		✓	
Ian			✓	✓				✓		✓
Justin	✓	✓			✓				✓	

Exercise 12.23. Pick an area of the world which is geographically divided into at least ten areas. These could be countries, states, counties, anything. Use the Fiedler Method to partition the area. Show the graph, relevant calculations, and a resulting map with the regions colored in two separate colors.

Exercise 12.24. The Fiedler method attempts to do two things with regards to the way it partitions the graph. What are those two things?

Exercise 12.25. What is happening when the Fiedler value turns out to have two or more linearly independent vectors associated to it? Give an example of a graph for which you believe that this would be the case and provide a basic and intutive explanation of why you believe this would be the case.