1.2.3 A quadratic form

The most natural quadratic form associated with a graph is defined in terms of its Laplacian matrix,

$$\boldsymbol{L}_G \overset{\text{def}}{=} \boldsymbol{D}_G - \boldsymbol{M}_G.$$

Given a function on the vertices, $\boldsymbol{x} \in \mathbb{R}^V$, the Laplacian quadratic form of a weighted graph in which edge (a, b) has weight $w_{a,b} > 0$ is

$$\boldsymbol{x}^T \boldsymbol{L}_G \boldsymbol{x} = \sum_{(a,b) \in E} w_{a,b} (\boldsymbol{x}(a) - \boldsymbol{x}(b))^2. \tag{1.1}$$

This form measures the smoothness of the function $\boldsymbol{x}$. It will be small if the function $\boldsymbol{x}$ does not jump too much over any edge.

We use the notation $\boldsymbol{x}(a)$ to denote the coordinate of vector $\boldsymbol{x}$ corresponding to vertex a. Other people often use subscripts for this, like $\boldsymbol{x}_a$. We usually use subscripts to name vectors.

There are many possible definitions of Laplacians with negative edge weights. So, we will only define them when we need them.

1.3 Spectral Theory

We now review the highlights of the spectral theory for symmetric matrices. Almost all of the matrices we consider will be symmetric or will be similar[5] to symmetric matrices.

We recall that a vector $\boldsymbol{\psi}$ is an eigenvector of a matrix $\boldsymbol{M}$ with eigenvalue λ if

$$\boldsymbol{M}\boldsymbol{\psi} = \lambda \boldsymbol{\psi}. \tag{1.2}$$

That is, λ is an eigenvalue if and only if $\lambda \boldsymbol{I} - \boldsymbol{M}$ is a singular matrix. Thus, the eigenvalues are the roots of the characteristic polynomial of $\boldsymbol{M}$:

$$\det(x\boldsymbol{I} - \boldsymbol{M}).$$

Theorem 1.3.1. *[The Spectral Theorem] If $\boldsymbol{M}$ is an n-by-n, real, symmetric matrix, then there exist real numbers $\lambda_1, \ldots, \lambda_n$ and n mutually orthogonal unit vectors $\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_n$ and such that $\boldsymbol{\psi}_i$ is an eigenvector of $\boldsymbol{M}$ of eigenvalue λ_i, for each i.*

This is the great fact about symmetric matrices. If the matrix is not symmetric, it might not have n eigenvalues. And, even if it has n eigenvalues, their eigenvectors will not be orthogonal[6]. If $\boldsymbol{M}$ is not symmetric, its eigenvalues and eigenvalues might be the wrong thing to study.

[5] A matrix $\boldsymbol{M}$ is similar to a matrix $\boldsymbol{B}$ if there is a non-singular matrix $\boldsymbol{X}$ such that $\boldsymbol{X}^{-1}\boldsymbol{M}\boldsymbol{X} = \boldsymbol{B}$. In this case, $\boldsymbol{M}$ and $\boldsymbol{B}$ have the same eigenvalues. See the exercises at the end of this section.

[6] You can prove that if the eigenvectors are orthogonal, then the matrix is symmetric.

Recall that the eigenvectors are not uniquely determined, although the eigenvalues are. If $\boldsymbol{\psi}$ is an eigenvector, then $-\boldsymbol{\psi}$ is as well. Some eigenvalues can be repeated. If $\lambda_i = \lambda_{i+1}$, then $\boldsymbol{\psi}_i + \boldsymbol{\psi}_{i+1}$ will also be an eigenvector of eigenvalue λ_i. The eigenvectors of a given eigenvalue are only determined up to an orthogonal transformation.

Definition 1.3.2. *A matrix is positive definite if it is symmetric and all of its eigenvalues are positive. It is positive semidefinite if it is symmetric and all of its eigenvalues are nonnegative.*

Fact 1.3.3. *The Laplacian matrix of a graph is positive semidefinite.*

Proof. Let $\boldsymbol{\psi}$ be a unit eigenvector of $\boldsymbol{L}$ of eigenvalue λ. Then,

$$\boldsymbol{\psi}^T \boldsymbol{L} \boldsymbol{\psi} = \boldsymbol{\psi}^T \lambda \boldsymbol{\psi} = \lambda = \sum_{(a,b) \in E} w_{a,b} (\boldsymbol{\psi}(a) - \boldsymbol{\psi}(b))^2 \geq 0.$$

□

We always number the eigenvalues of the Laplacian from smallest to largest. Thus, $\lambda_1 = 0$. We will refer to λ_2, and in general λ_k for small k, as *low-frequency* eigenvalues. λ_n is a *high-frequency* eigenvalue. We will see why soon.

1.4 Some examples

Before we get to any theorems, we will examine evidence that the eigenvalues and eigenvectors of graphs are meaningful by looking at some examples. These were produced in Julia using a Jupyter notebook. You may find the notebook on the book homepage.

1.4.1 Paths

A path graph has vertices $\{1, \ldots, n\}$ and edges $(i, i+1)$ for $1 \leq i < n$. Here is the adjacency matrix of a path graph on 4 vertices.

```
M = path_graph(4)
Matrix(M)
 0.0  1.0  0.0  0.0
 1.0  0.0  1.0  0.0
 0.0  1.0  0.0  1.0
 0.0  0.0  1.0  0.0
```

And, here is its Laplacian matrix

```
Matrix(lap(M))
  1.0  -1.0   0.0   0.0
```

```
 -1.0   2.0  -1.0   0.0
  0.0  -1.0   2.0  -1.0
  0.0   0.0  -1.0   1.0
```

Here are the eigenvalues of a longer path.

```
L = lap(path_graph(10))
E = eigen(Matrix(L))
println(E.values)
```

```
[0.0, 0.097887, 0.381966, 0.824429, 1.38197, 2.0, 2.61803, 3.17557, 3.61803, 3.90211]
```

The eigenvector of the zero-eigenvalue is a constant vector (up to numerical issues):

```
E.vectors[:,1]
```

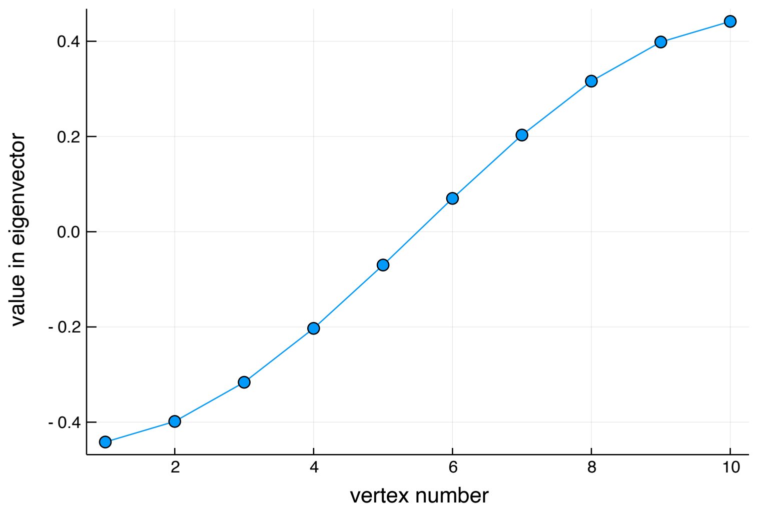
```
 0.31622776601683755
 0.31622776601683716
 0.31622776601683766
 0.3162277660168381
 0.31622776601683855
 0.3162277660168381
 0.3162277660168385
 0.31622776601683805
 0.3162277660168378
 0.3162277660168378
```

The eigenvector of λ_2 is the lowest frequency eigenvector, as we can see that it increases monotonically along the path:

```
v2 = E.vectors[:,2]
```

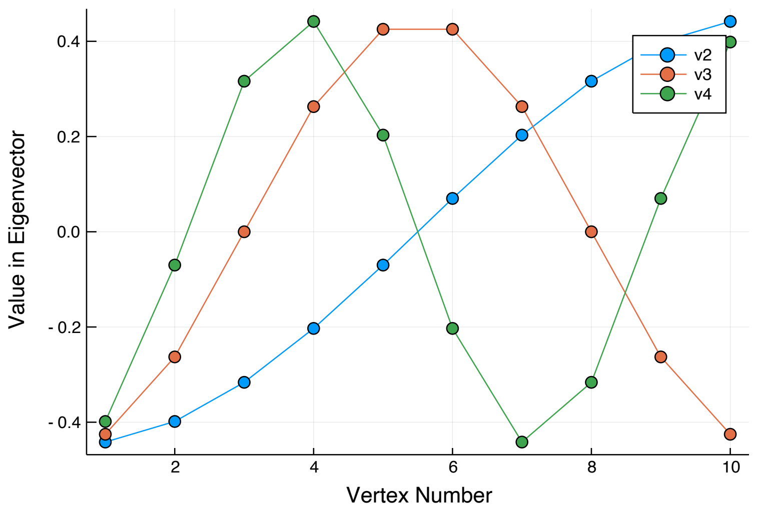
```
 -0.44170765403093937
 -0.39847023129620024
 -0.316227766016838
 -0.20303072371134553
 -0.06995961957075425
  0.06995961957075386
  0.2030307237113457
  0.31622776601683766
  0.3984702312961997
  0.4417076540309382
```

Let's plot that.



```
plot(v2,marker=5,legend=false)
xlabel!("vertex number")
ylabel!("value in eigenvector")
```

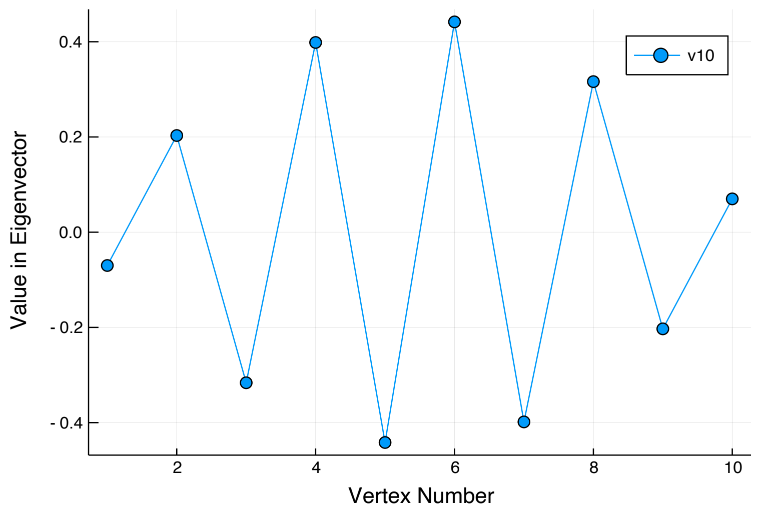
The x-axis is the name/number of the vertex, and the y-axis is the value of the eigenvector at that vertex. Now, let's look at the next few eigenvectors.



```
Plots.plot(E.vectors[:,2],label="v2",marker = 5)
Plots.plot!(E.vectors[:,3],label="v3",marker = 5)
Plots.plot!(E.vectors[:,4],label="v4",marker = 5)
xlabel!("Vertex Number")
ylabel!("Value in Eigenvector")
```

You may now understand why we refer to these as the low-frequency eigenvectors. The curves they trace out resemble the low-frequency modes of vibration of a string. The reason for this is that the path graph can be viewed as a discretization of the string, and its Laplacian matrix is a discretization of the Laplace operator. We will relate the low-frequency eigenvalues to connectivity.

In contrast, the highest frequency eigenvalue alternates positive and negative with every vertex. We will see that the high-frequency eigenvectors may be related to problems of graph coloring and finding independent sets.



```
Plots.plot(E.vectors[:,10],label="v10",marker=5)
xlabel!("Vertex Number")
ylabel!("Value in Eigenvector")
```

1.5 Highlights

We now attempt to motivate this book, and the course on which it is based, by surveying some of its highlights.

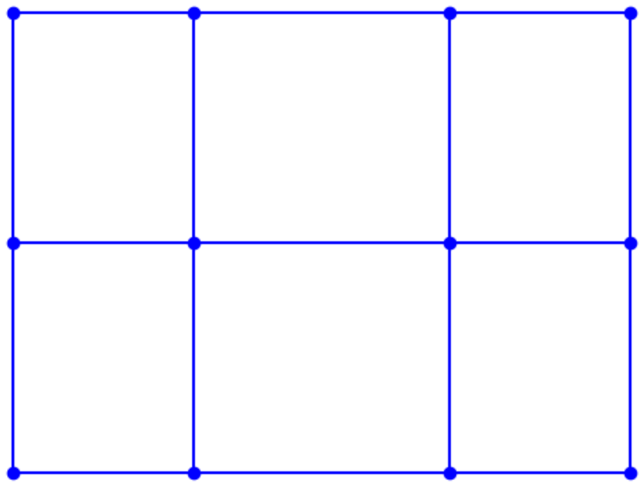
1.5.1 Spectral Graph Drawing

We can often use the low-frequency eigenvalues to obtain a nice drawing of a graph. For example, here is 3-by-4 grid graph, and its first two non-trivial eigenvectors. Looking at them suggests that they might provide nice coordinates for the vertices.

```
M = grid2(3,4)
L = lap(M)
E = eigen(Matrix(L))
V = E.vectors[:,2:3]
```

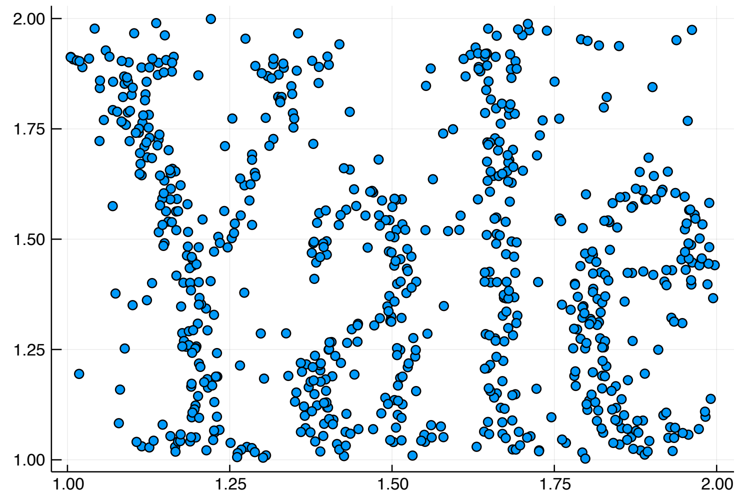
```
 -0.377172   0.353553
 -0.15623    0.353553
  0.15623    0.353553
  0.377172   0.353553
 -0.377172  -1.66533e-16
 -0.15623   -4.16334e-16
  0.15623   -5.82867e-16
  0.377172   2.77556e-16
 -0.377172  -0.353553
 -0.15623   -0.353553
  0.15623   -0.353553
  0.377172  -0.353553
```

In the figure below, we use these eigenvectors to draw the graph. Vertex a be been plotted at coordinates $\boldsymbol{\psi}_2(a), \boldsymbol{\psi}_3(a)$. That is, we use $\boldsymbol{\psi}_2$ to provide a horizontal coordinate for every vertex, and $\boldsymbol{\psi}_3$ to obtain a vertical coordinate. We then draw the edges as straight lines.



```
plot_graph(M,V[:,1],V[:,2])
```

Let's do a fancier example that should convince you something interesting is going on. We begin by generating points by sampling them from the Yale logo.

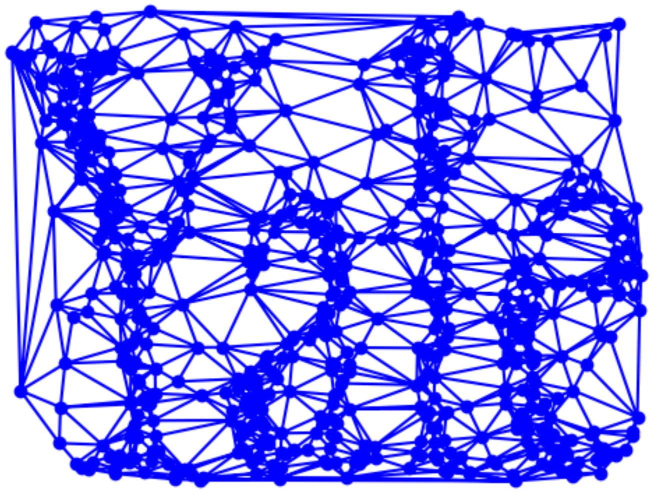


```
@load "yale.jld2"
scatter(xy[:,1],xy[:,2],legend=false)
```

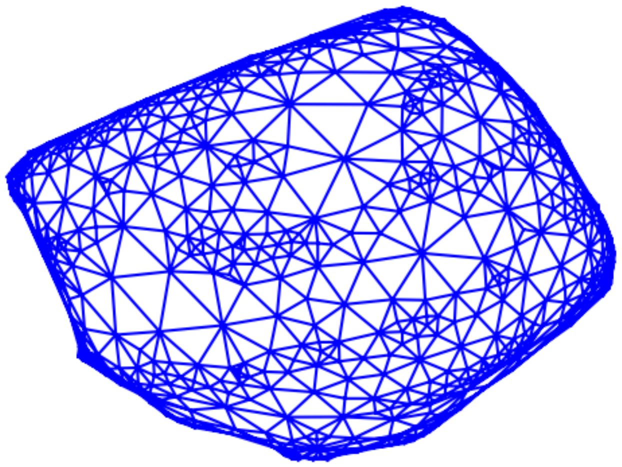
We then construct a graph on the points by forming their Delaunay triangulation[7], and use the edges of the triangles to define a graph on the points.

Since the vertices came with coordinates, it was easy to draw a nice picture of the graph. But, what if we just knew the graph, and not the coordinates? We could generate coordinates by computing two eigenvectors, and using each as a coordinate. Below, we plot vertex a at position $\boldsymbol{\psi}_2(a), \boldsymbol{\psi}_3(a)$, and again draw the edges as straight lines.

[7]While it does not make sense to cover Delaunay triangulations in this book, they are fascinating and I recommend that you look them up.



```
plot_graph(a,xy[:,1],xy[:,2])
```



```
plot_graph(a, v2,v3, dots=false)
```

That's a great way to draw a graph if you start out knowing nothing about it[8]. Note that the middle of the picture is almost planar, although edges do cross near the boundaries.

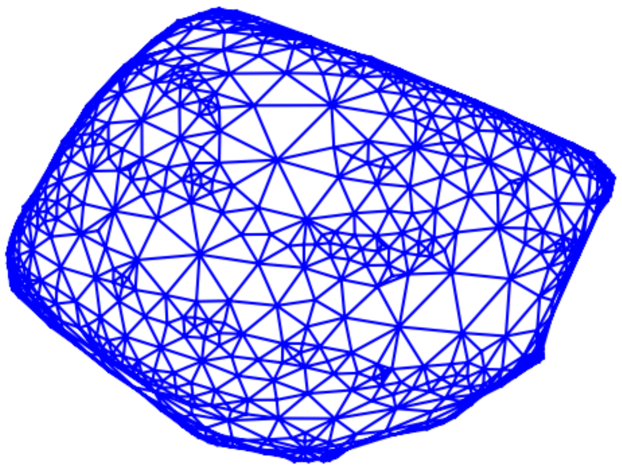
[8]It's the first thing I do whenever I meet a strange graph.

1.5.2 Graph Isomorphism

It is important to note that the eigenvalues do not change if we relabel the vertices. Moreover, if we permute the vertices then the eigenvectors are similarly permuted. That is, if $\boldsymbol{P}$ is a permutation matrix, then

$$\boldsymbol{L}\boldsymbol{\psi} = \lambda\boldsymbol{\psi} \quad \text{if and only if} \quad (\boldsymbol{P}\boldsymbol{L}\boldsymbol{P}^T)(\boldsymbol{P}\boldsymbol{\psi}) = \lambda(\boldsymbol{P}\boldsymbol{\psi}),$$

because $\boldsymbol{P}^T\boldsymbol{P} = \boldsymbol{I}$. To prove it by experiment, let's randomly permute the vertices, and plot the permuted graph.



```
Random.seed!(1)
p = randperm(size(a,1))
M = a[p,p]
E = eigen(Matrix(lap(M)))
V = E.vectors[:,2:3]
plot_graph(M,V[:,1],V[:,2], dots=false)
```

Note that this picture is slightly different from the previous one: it has flipped vertically. That's because eigenvectors are only determined up to signs, and that's only if they have multiplicity 1. This gives us a very powerful heuristic for testing if one graph is a permutation of another (this is the famous "Graph Isomorphism Testing Problem"). First, check if the two graphs have the same sets of eigenvalues. If they don't, then they are not isomorphic. If they do, and the eigenvalues have multiplicity one, then draw the pictures above. If the pictures are the same, up to horizontal or vertical flips, and no vertex is mapped to the same location as another, then by lining up the pictures we can recover the permutation.

As some vertices can map to the same location, this heuristic doesn't always work. We will learn about it to the extent to which it does. In particular, we will see in Chapter 39 that if every eigenvalue of two graphs G and H have multiplicity 1, then we can efficiently test whether or not they are isomorphic.

These algorithms have been extended to handle graph in which the multiplicity of every eigenvalue is bounded by a constant [BGM82]. But, there are graphs in which every non-trivial eigenvalue has large multiplicity. We will learn how to construct and analyze these, as they constitute fundamental examples and counter-examples to many natural conjectures. For example, here are the eigenvalues of a Latin Square Graph on 25 vertices. These are a type of Strongly Regular Graph.

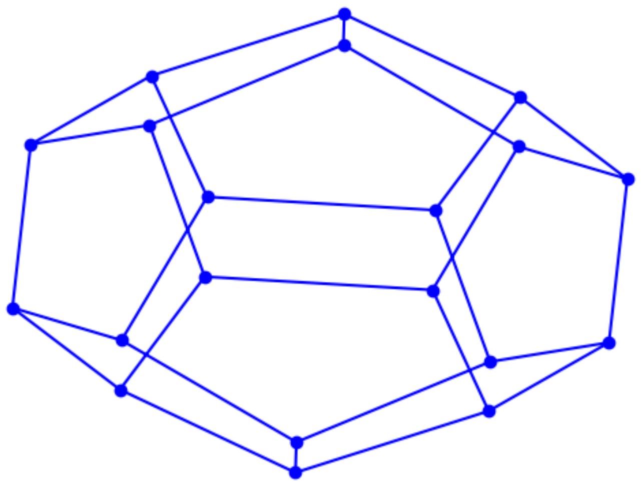
```
M = latin_square_graph(5);
println(eigvals(Matrix(lap(M))))
```

```
[0.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0,
15.0, 15.0, 15.0, 15.0, 15.0, 15.0, 15.0, 15.0, 15.0, 15.0, 15.0, 15.0]
```

All Latin Square Graphs of the same size have the same eigenvalues, whether or not they are isomorphic. We will learn some surprisingly fast (but still not polynomial time) algorithms for checking whether or not Strongly Regular Graphs are isomorphic.

1.5.3 Platonic Solids

Of course, somme graphs are not meant to be drawn in 3 dimensions. For example let's try this with the dodecahedron.

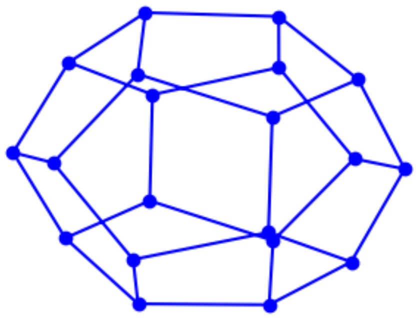


```
M = read_graph("dodec.txt")
spectral_drawing(M)
```

You will notice that this looks like what you would get if you squashed the dodecahedron down to the plane. The reason is that we really shouldn't be drawing this picture in two dimensions: the smallest non-zero eigenvalue of the Laplacian has multiplicity three.

```
E = eigen(Matrix(lap(M)))
println(E.values)
```

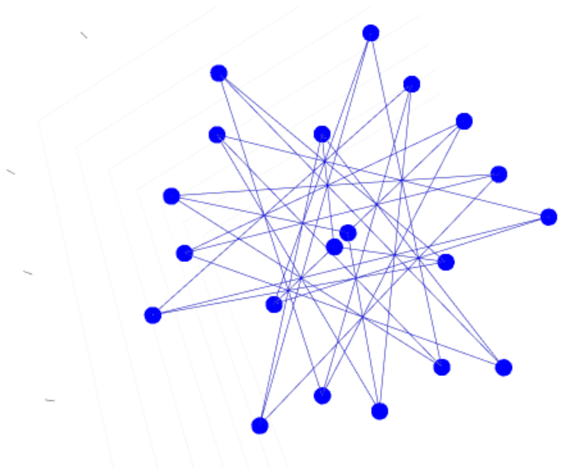
So, we can't reasonably choose just two eigenvectors. We should be choosing three that span the eigenspace. If we do, we would get the canonical representation of the dodecahedron in three dimensions.



```
x = E.vectors[:,2]
y = E.vectors[:,3]
z = E.vectors[:,4]
plot_graph(M, x, y, z; setaxis=false)
```

As you would guess, this happens for all Platonic solids. In fact, if you properly re-weight the edges, it happens for every graph that is the one-skeleton of a convex polytope [Lov01]. Let me state that more concretely. Given a convex polytope in $\mathbb{R}^d$, we can treat its 1-skeleton as a graph on its vertices. There is always a way of assigning positive weights to edges so that the second-smallest Laplacian eigenvalue has multiplicity d, and so that the corresponding eigenspace is spanned by the coordinate vectors of the vertices of the polytope.

We finish this section by contemplating an image of the high-frequency eigenvectors of the dodecahedron. This code plots them in three dimensions, although we can only print them in two. Observe that vertices are approximately opposite their neighbors.



```
x = E.vectors[:,20]
y = E.vectors[:,19]
z = E.vectors[:,18]
plot_graph(M, x, y, z; setaxis=false);
```

1.5.4 The Fiedler Value

The second-smallest eigenvalue of the Laplacian matrix of a graph is zero if and only if the graph is disconnected. If G is disconnected, then we can partition it into two graphs G_1 and G_2 with no edges between them, and then write

$$\boldsymbol{L}_G = \begin{pmatrix} \boldsymbol{L}_{G_1} & 0 \\ 0 & \boldsymbol{L}_{G_2} \end{pmatrix}.$$

As the eigenvalues of $\boldsymbol{L}_G$ are the union, with multiplicity, of the eigenvalues of $\boldsymbol{L}_{G_1}$ and $\boldsymbol{L}_{G_2}$ we see that $\boldsymbol{L}_G$ inherits a zero eigenvalue from each. Conversely, if G is connected then we can show that the only vectors $\boldsymbol{x}$ for which $\boldsymbol{x}^T \boldsymbol{L}_G \boldsymbol{x} = 0$ are the constant vectors. If $\boldsymbol{x}$ is not constant and G is connected then there must be an edge (a, b) for which $\boldsymbol{x}(a) \neq \boldsymbol{x}(b)$. And, this edge will contribute a positive term to the sum (1.1).

Fiedler suggested that we make this qualitative observation quantitative and think of λ_2 as a measure of how well connected the graph is. For this reason, he called it the "Algebraic Connectivity" of a graph, and we call it the "Fiedler value".

Fiedler [Fie73] that the further λ_2 is from 0, the better connected the graph is. In Chapter 21 we will prove ultimate extension of this result: Cheeger's inequality.

In short, we say that a graph is poorly connected if one can cut off many vertices by removing only a few edges. We measure how poorly connected it is by the ratio of these quantities (almost). Cheeger's inequality gives a tight connection between this ratio and λ_2. If λ_2 is small, then for some t, the set of vertices

$$S_i \overset{\text{def}}{=} \{i : \boldsymbol{\psi}_2(i) < t\}$$

may be removed by cutting much less than $|S_i|$ edges. This spectral graph partitioning heuristic has proved very successful in practice.

In general, it will be interesting to turn qualitative statements like this into quantitative ones. For example, the smallest eigenvalue of the diffusion matrix is zero if and only if the graph is bipartite. One can relate the magnitude of this eigenvalue to how far a graph is from being bipartite [Tre09].

1.5.5 Bounding Eigenvalues

We will often be interested in the magnitudes of certain eigenvalues. For this reason, we will learn multiple techniques for proving bounds on eigenvalues. The most prominent of these will be proofs by test vectors and proofs by comparison with simpler graphs.

1.5.6 Planar Graphs

We will prove that graphs that can be drawn nicely must have small Fiedler value, and we will prove very tight results for planar graphs.

We will also see how to use the graph Laplacian to draw planar graphs: Tutte [Tut63] that if one reasonably fixes the locations of the vertices on a face of a planar graph and then lets the others settle into the positions obtained by treating the edges as springs, then one obtains a planar drawing of the graph!

1.5.7 Random Walks on Graphs

Spectral graph theory is one of the main tools we use for analyzing random walks on graphs. We will devote a few chapters to this theory, connect it to Cheeger's inequality, and use tools developed to study random walks to derive a fascinating proof of Cheeger's inequality.

1.5.8 Expanders

We will be particularly interested in graphs that are very well connected. These are called *expanders*. Roughly speaking, expanders are sparse graphs (say a number of edges linear in the number of vertices), in which λ_2 is bounded away from zero by a constant. They are among the most important examples of graphs, and play a prominent role in Theoretical Computer Science.

Expander graphs have numerous applications. We will see how to use random walks on expander graphs to construct pseudo-random generators *about which one can actually prove something.* We will also use them to construct good error-correcting codes.

Error-correcting codes and expander graphs are both fundamental objects of study in the field of Extremal Combinatorics and are extremely useful. We will also use error-correcting codes to construct crude expander graphs. In Chapter 30 we will see a simple construction of good expanders. The best expanders are the Ramanujan graphs. These were first constructed by