Spectral Graph Theory Lecture 1

Introduction

Daniel A. Spielman August 29, 2012

1.1 About these notes

These notes are not necessarily an accurate representation of what happened in class. The notes written before class say what I think I should say. The notes written after class way what I wish I said. My jokes were delivered with perfect timing.

Be skeptical of all statements in these notes that can be made mathematically rigorous.

1.2 First Things

1. Please call me "Dan". If such informality makes you uncomfortable, you can try "Professor Dan". If that fails, I will also answer to "Prof. Spielman".
2. If you are going to take this course, please sign up for it on Classes V2. This is the only way you will get emails like "Problem 3 was false, so you don't have to solve it".
3. This class meets this coming Friday, August 31, but not on Labor Day, which is Monday September 3.

1.3 Introduction

I have three objectives in this lecture: to tell you what the course is about, to help you decide if this course is right for you, and to tell you how the course will work.

As the title suggests, this course is about the eigenvalues and eigenvectors of matrices associated with graphs, and their applications. I will never forget my amazement at learning that combinatorial properties of graphs could be revealed by an examination of the eigenvalues and eigenvectors of their associated matrices. I hope to both convey my amazement to you and to make it feel like common sense. I'm now shocked when any important property of a graph is not revealed by its eigenvalues and eigenvectors.

This class will fundamentally be a math class, but my emphasis is on material that I find useful. I'll present a lot of theorems, a few algorithms, and a bunch of open problems.

You can get a feel for the topics that I'll cover during the semester from looking at the lecture notes from the 2009 version of this course. This year's version will be pretty similiar. As in 2009, I hope

to write notes for every lecture. To help you decide whether or not to take the course, I'll make this lecture fairly technical. The first lecture in 2009 was mostly a sales pitch.

1.4 Background: Graphs

First, we recall that a graph $G = (V, E)$ is specified by its vertex[1] set, V, and edge set E. In an undirected graph, the edge set is a set of unordered pairs of vertices. Unless otherwise specified, all graphs will be undirected, simple (having no loops or multiple edges) and finite. We will sometimes assign weights to edges. These will usually be real numbers. If no weights have been specified, we view all edges as having weight 1. This is an arbitrary choice, and we should remember that it has an impact.

Graphs are typically used to model connections or relations between things, where "things" are vertices. However, I often prefer to think of the edges in a graph as being more important than the vertices. In this case, I may just specify an edge set E, and ignore the ambient vertex set.

Common "natural" examples of graphs are:

- Friendship graphs: people are vertices, edges exist between pairs of people who are friends (assuming the relation is symmetric).
- Network graphs: devices, routers and computers are vertices, edges exist between pairs that are connected.
- Circuit graphs: electronic components, such as transistors, are vertices: edges exist between pairs connected by wires.
- Protein-Protein Interaction graphs: proteins are vertices. Edges exist between pairs that interact. These should really have weights indicating the strength and nature of interaction. Most other graphs should to.

It is much easier to study abstract, mathematically defined graphs. For example,

- The path on n vertices. The vertices are $\{1, \ldots n\}$. The edges are $(i, i+1)$ for $1 \leq i < n$.
- The ring on n vertices. The vertices are $\{1, \ldots n\}$. The edges are all those in the path, plus the edge $(1, n)$.
- The hypercube on 2^k vertices. The vertices are elements of $\{0, 1\}^k$. Edges exist between vertices that differ in only one coordinate.

While we think of the hypercube as an abstract graph, it will also show up in applications. When studying probabilities involving k independent random variables taking values in $\{0, 1\}$, we will find it convenient to identify them with the vertices of the hypercube.

[1] I will use the words "vertex" and "node" interchangably. Sorry about that.

1.5 Matrices for Graphs

Typically, and without loss of generality, we will assume that $V = \{1, \ldots, n\}$. The most natural matrix to associate with a graph G is its adjacency matrix, $\boldsymbol{A}_G$, whose entries $\boldsymbol{A}_G(u, v)$ are given by

$$\boldsymbol{A}_G(u, v) = \begin{cases} 1 & \text{if } (u, v) \in E \\ 0 & \text{otherwise.} \end{cases}$$

While the adjacency matrix is the most natural matrix to associate with a graph, I also find it the least useful. Eigenvalues and eigenvectors are most meaningful when used to understand a natural operator or a natural quadratic form. The adjacency matrix provides neither.

The most natural operator associated with a graph G is probably its diffusion operator. To construct this, let $\boldsymbol{D}_G$ be the diagonal matrix in which $\boldsymbol{D}_G(u, u)$ is the degree of vertex u. We will usually write $\boldsymbol{d}(u)$ for the degree of vertex u. In the case of a weighted graph, this degree will be the sum of the weights of the edges attached to the vertex u.

We then set

$$\boldsymbol{W}_G = \boldsymbol{D}_G^{-1} \boldsymbol{A}_G.$$

Of course, when the graph is *regular*, that is when every vertex has the same degree, $\boldsymbol{W}_G$ is merely a rescaling of $\boldsymbol{A}_G$. I think this is why researchers got away with studying the adjacency matrix for so long.

This matrix describes the diffusion of stuff among the vertices of a graph. Imagine a process in which each vertex can contain some amount of stuff. At each time step, the stuff at a vertex will be uniformly distributed to its neighbors. None of the stuff that was at a vertex remains at the vertex, but stuff can come from other vertices.

Formally, we use a vector $\boldsymbol{p} \in \mathbb{R}^V$ to indicate how much stuff is at each vertex, with $\boldsymbol{p}(u)$ being the amount of stuff at vertex u. When describing diffusion, I will treat $\boldsymbol{p}$ as a row vector. After one time step, the distribution of stuff at each vertex will be $\boldsymbol{p}\,\boldsymbol{W}_G$. To see this, first consider the case in which $\boldsymbol{p}$ is an elementary unit vector. That is, there exists some vertex u for which $\boldsymbol{p}(u) = 1$, and for every other vertex v, $\boldsymbol{p}(v) = 0$. The vector $\boldsymbol{p}\boldsymbol{D}_G^{-1}$ is has the value $1/\boldsymbol{d}(u)$ at vertex u, and is zero everywhere else. So, the vector $\boldsymbol{p}\boldsymbol{D}_G^{-1} A_G$ has value $1/\boldsymbol{d}(u)$ at every vertex v that is a neighbor of u, and zero everwhere else. If this is not immediately obvious, think about it until it is.

A few lectures from now we will see that spectral theory provides a good understanding of what happens when one iteratively applies a linear operator like $\boldsymbol{W}_G$.

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

$$\boldsymbol{L}_G \stackrel{\text{def}}{=} \boldsymbol{D}_G - \boldsymbol{A}_G.$$

Given a function on the vertices, $\boldsymbol{x} \in \mathbb{R}^V$, the Laplacian quadratic form is

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

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.

1.6 Background: Spectral Theory

I now review the highlights of the spectral theory for symmetric matrices. Almost all of the matrices we consider in this course will be symmetric or will be similar[2] to symmetric matrices.

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

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

If $\boldsymbol{M}$ is an n-by-n symmetric matrix, then there exist n mutually orthnogonal unit vectors $\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_n$ and numbers $\lambda_1, \ldots, \lambda_n$ 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 might not be orthogonal. In fact, if M is not symmetric, then its eigenvalues and eigenvalues might be the wrong thing to look at.

Before I go on to explain why, I should first remind you 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. Generally, the eigenvectors of a given eigenvalue are only determined up to an orthonogonal transformation.

The equation (1.2) defining eigenvalues and eigenvectors does not give much of a hint as to why the eigenvalues and eigenvectors should have combinatorial significance. Fortunately, the eigenvalues and eigenvectors of symmetric matrices have many characterizations. The ones that will be most useful to us will come from optimization problems. In particular, they arise when maximizing or minimizing the Rayleigh quotient with respect to a matrix $\boldsymbol{M}$.

Definition 1.6.1. *The Rayleigh quotient of a vector $\boldsymbol{x}$ with respect to a matrix $\boldsymbol{M}$ is the ratio*

$$\frac{\boldsymbol{x}^T\boldsymbol{M}\boldsymbol{x}}{\boldsymbol{x}^T\boldsymbol{x}}.$$

Observe that if $\boldsymbol{\psi}$ is an eigenvector of $\boldsymbol{M}$ of eigenvalue λ, then

$$\frac{\boldsymbol{\psi}^T\boldsymbol{M}\boldsymbol{\psi}}{\boldsymbol{\psi}^T\boldsymbol{\psi}} = \frac{\boldsymbol{\psi}^T\lambda\boldsymbol{\psi}}{\boldsymbol{\psi}^T\boldsymbol{\psi}} = \frac{\lambda\boldsymbol{\psi}^T\boldsymbol{\psi}}{\boldsymbol{\psi}^T\boldsymbol{\psi}} = \lambda.$$

Theorem 1.6.2. *Let $\boldsymbol{M}$ be a symmetric matrix and let $\boldsymbol{x}$ be the non-zero vector that maximizes the Rayleigh quotient with respect to $\boldsymbol{M}$. Then, $\boldsymbol{x}$ is an eigenvector of $\boldsymbol{M}$ with eigenvalue equal to the Rayleigh quotient. Moreover, this eigenvalue is the largest eigenvalue of $\boldsymbol{M}$.*

This theorem is so important that I will show you at least two proofs of it. First, we will assume the spectral theorem.

[2] A matrix M is similar to a matrix A if there is a non-singular matrix B such that $B^{-1}MB = A$. In this case, M and A have the same eigenvalues.

Proof. First, number the eigenvalues of $\boldsymbol{M}$ so that $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$. Since $\{\boldsymbol{\psi}_i\}_i$ is an orthonormal basis, we can write any vector $\boldsymbol{x}$ in this basis as

$$\boldsymbol{x} = \sum_i (\boldsymbol{\psi}_i^T \boldsymbol{x}) \boldsymbol{\psi}_i.$$

If you don't remember this, the easiest way to verify it is to check that this gives the correct inner products with every vector in the basis:

$$\begin{aligned}
\boldsymbol{\psi}_j^T \left(\sum_i (\boldsymbol{\psi}_i^T \boldsymbol{x}) \boldsymbol{\psi}_i \right) &= \left(\sum_i (\boldsymbol{\psi}_i^T \boldsymbol{x}) \boldsymbol{\psi}_j^T \boldsymbol{\psi}_i \right) && \text{(by linearity)} \\
&= (\boldsymbol{\psi}_j^T \boldsymbol{x}) \boldsymbol{\psi}_j^T \boldsymbol{\psi}_j && (\text{as } \boldsymbol{\psi}_j^T \boldsymbol{\psi}_i = 0 \text{ for } i \neq j) \\
&= \boldsymbol{\psi}_j^T \boldsymbol{x} && (\text{as } \boldsymbol{\psi}_j \text{ is a unit vector , } \boldsymbol{\psi}_j^T \boldsymbol{\psi}_j = 1).
\end{aligned}$$

As the multiplication of $\boldsymbol{x}$ by a constant does not change the Rayleigh quotient, we can assume without loss of generality that $\boldsymbol{x}$ is a unit vector. That is,

$$\sum_i (\boldsymbol{\psi}_i^T \boldsymbol{x})^2 = 1.$$

We now evaluate the Rayleigh quotient with respect to $\boldsymbol{x}$:

$$\begin{aligned}
\frac{\boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}} &= \frac{\left(\sum_i (\boldsymbol{\psi}_i^T \boldsymbol{x}) \boldsymbol{\psi}_i \right)^T \boldsymbol{M} \left(\sum_j (\boldsymbol{\psi}_j^T \boldsymbol{x}) \boldsymbol{\psi}_j \right)}{1} \\
&= \left(\sum_i (\boldsymbol{\psi}_i^T \boldsymbol{x}) \boldsymbol{\psi}_i \right)^T \left(\sum_j (\boldsymbol{\psi}_j^T \boldsymbol{x}) \lambda_j \boldsymbol{\psi}_j \right) \\
&= \sum_{i,j} (\boldsymbol{\psi}_i^T \boldsymbol{x})(\boldsymbol{\psi}_j^T \boldsymbol{x}) \lambda_j \boldsymbol{\psi}_i^T \boldsymbol{\psi}_j \\
&= \sum_j (\boldsymbol{\psi}_j^T \boldsymbol{x})(\boldsymbol{\psi}_j^T \boldsymbol{x}) \lambda_j, && \text{as } \boldsymbol{\psi}_i^T \boldsymbol{\psi}_j = 0 \text{ for } i \neq j, \\
&= \sum_j (\boldsymbol{\psi}_j^T \boldsymbol{x})^2 \lambda_j \\
&\leq \lambda_n \sum_j (\boldsymbol{\psi}_j^T \boldsymbol{x})^2 \\
&= \lambda_n.
\end{aligned}$$

So, the Rayleigh quotient is never more than λ_n. We have already established that the value of λ_n is achieved by $\boldsymbol{\psi}_n$. It remains to show that only eigenvectors of eigenvalue λ_n can have such a large Rayleigh quotient. This follows from the observation that the inequality above is tight if and only if $\lambda_i = \lambda_n$ for all i such that $\boldsymbol{\psi}_i^T \boldsymbol{x} \neq 0$. That is, if and only if $\boldsymbol{x}$ is in the span of the eigenvectors with eigenvalue λ_n.

□

You can similarly prove that λ_1 is the minimum possible Rayleigh quotient, and that

$$\lambda_i = \min_{\boldsymbol{x} \perp \boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_{i-1}} \frac{\boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}},$$

and that one can choose the eigenvectors by

$$\boldsymbol{\psi}_i = \arg \min_{\boldsymbol{x} \perp \boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_{i-1}} \frac{\boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}}.$$

I'll now give another proof of Theorem 1.6.2 that does not assume the spectral theorem. In fact, we can use this proof to prove the spectral theorem.

Proof of Theorem 1.6.2. Let $\boldsymbol{x}$ be a non-zero vector that maximizes the Rayleigh quotient. We recall that the gradient of a function at its maximum must be the zero vector. So, let's compute the gradient. I'll do this using "matrix derivative" notation. We first compute

$$\frac{\partial}{\partial \boldsymbol{x}} \boldsymbol{x}^T \boldsymbol{x} = 2\boldsymbol{x}.$$

That is, I am taking the derivative with respect to the elements of the vector $\boldsymbol{x}$ and writing it as a vector. You can similarly compute

$$\frac{\partial}{\partial \boldsymbol{x}} \boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x} = (\boldsymbol{M} \boldsymbol{x}) + (\boldsymbol{x}^T \boldsymbol{M})^T = 2\boldsymbol{M} \boldsymbol{x}.$$

Note that this last computation made essential use of the symmetry of $\boldsymbol{M}$.

Finally, we compute

$$\frac{\partial}{\partial \boldsymbol{x}} \frac{\boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}} = \frac{(\boldsymbol{x}^T \boldsymbol{x})(2\boldsymbol{M} \boldsymbol{x}) - (\boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x})(2\boldsymbol{x})}{(\boldsymbol{x}^T \boldsymbol{x})^2}.$$

In order for this to be zero, we must have

$$(\boldsymbol{x}^T \boldsymbol{x})(2\boldsymbol{M} \boldsymbol{x}) - (\boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x})(2\boldsymbol{x}) = 0,$$

which implies

$$\boldsymbol{M} \boldsymbol{x} = \frac{\boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}} \boldsymbol{x}.$$

That is, if and only if $\boldsymbol{x}$ is an eigenvector of $\boldsymbol{M}$ with eigenvalue equal to its Rayleigh quotient. □

1.7 Eigenvectors of the Laplacian

We will now use this characterization of eigenvalues to reason about the smallest eigenvalues of the Laplacian. We will do this by thinking about the Laplacian quadratic form (1.1). As all of the terms in the sum defining the form are squares, we can see that it is never negative. If we let $\boldsymbol{x}$ be a constant vector, then this form is zero. So, the smallest eigenvalue of the Laplacian is zero.

One can also show that $\lambda_2 > 0$ if and only if the graph is connected. If the graph is disconnected, one can construct at least two orthogonal vectors with eigenvalue zeros: consider vectors that are constant on one component of the graph and zero everywhere else. On the other hand, if the graph is connected then we can show that $\lambda_2 > 0$: Let $\boldsymbol{x}$ be any vector orthogonal to the constant vectors. So, there must be two vertices u and v for which $\boldsymbol{x}(u) \neq \boldsymbol{x}(v)$. As there is a path between these vertices, there must be some edge accross which $\boldsymbol{x}$ changes value. So, the quadratic form will be positive on $\boldsymbol{x}$. (For a more formal proof, look at the notes from my second lecture on 2009)

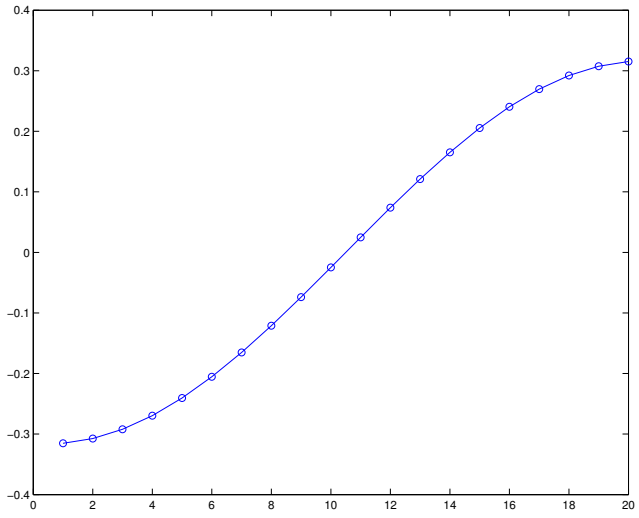
The attempt to make this qualitative statement quantative led Fiedler [**?**] to call λ_2 the "algebraic connectivity of a graph". We will relate λ_2 to how well connected a graph is. In general, it will be interesting to turn qualitative statements like this into quantative ones. For example, we will later see that the smallest eigenvalue of the diffusion matrix is zero if and only if the graph is bipartite. We will then relate the magnitude of this eigenvalue to how far a graph is from being bipartite.

1.8 Examples

Let's start by examining the eigenvectors of the Laplacian of the path graph. I'll do examples in Matlab, so you can follow along if you like.

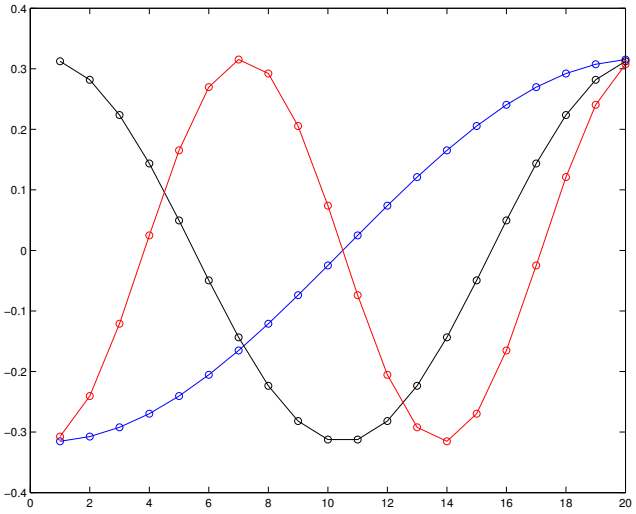
I'll first construct the path graph on 20 vertices, and plot its second eigenvector. The x-axis in this plot gives the number of the vertex. The y-axis is the value of the eigenvector at a vertex.

```
a = diag(ones(1,19),1);
a = a + a';
la = diag(sum(a)) - a;
[v,d] = eig(full(la));
diag(d)
plot(v(:,2));
hold on; plot(v(:,2),'o')
```



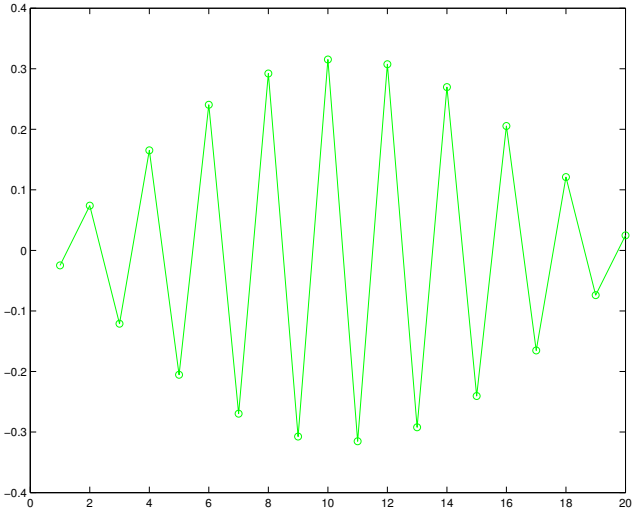
I'll now plot the third and fourth eigenvectors on top of this. You can see that the number of sign changes in the eigenvetors equals its index. The higher the eigenvalue, the more times it oscillates.

```
plot(v(:,3),'k');
plot(v(:,3),'ko');
plot(v(:,4),'r');
plot(v(:,4),'ro');
```



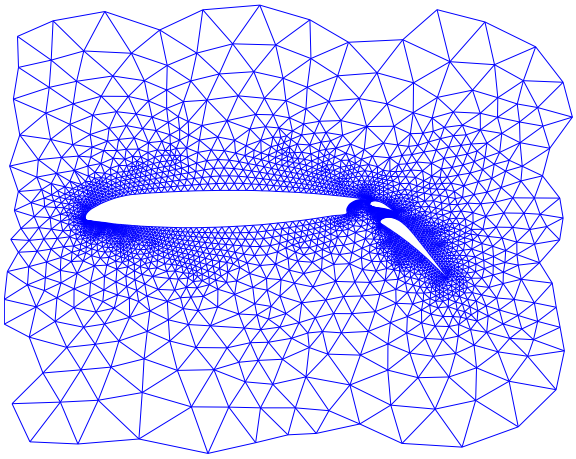
And, here is the eigenvector of highest eigenvalue.

```
clf;
plot(v(:,20),'g');
hold on
plot(v(:,20),'go');
```



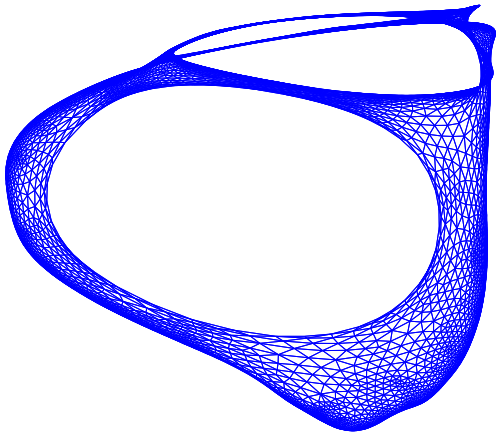
Usually graphs don't come with coordinates. In this case, a good way to try to draw them is to use the eigenvectors to supply coordinates. For example, here is the airfoil graph that comes with Matlab. I'll first draw it using the suggested coordinates. These are given by the variables x and y that come with the file. They indicate where to draw the vertices. The edges are then drawn as straight lines.

```
load airfoil
A = sparse(i,j,1,4253,4253);
A = A + A';
gplot(A,[x,y])
```



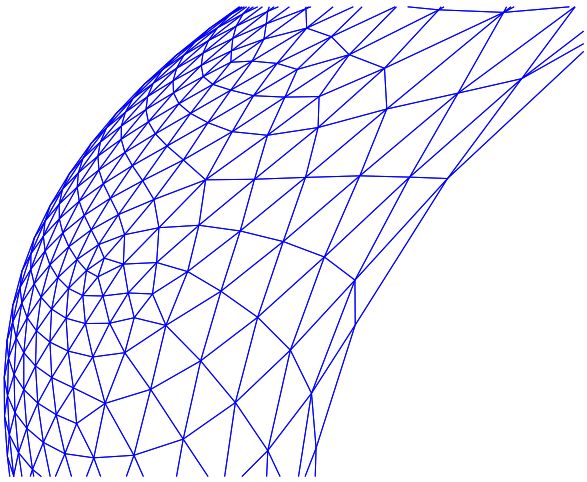
We will now draw it using coordinates from the eigenvectors. I will plot vertex u at point $(\boldsymbol{\psi}_2(u), \boldsymbol{\psi}_3(u))$ I will draw the edges as straight lines between the vertices they connect.

```
L = lap(A);
[v,d] = eigs(L,3,'sa');
gplot(A,v(:,[2 3]))
```



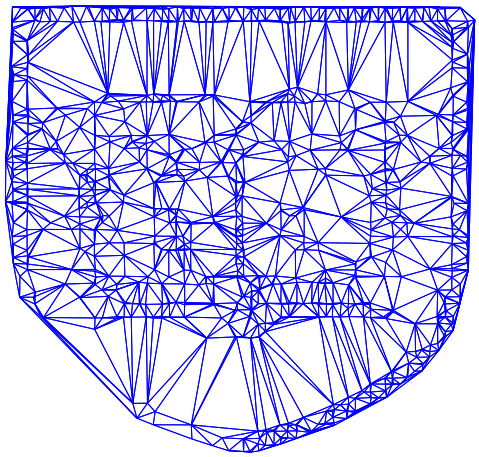
Let's zoom in on a small portion, so that we can see how nice the embedding is.

```
set(gca,'XLim',[-.032, -.024])
set(gca,'YLim',[-.0023, .0052])
```



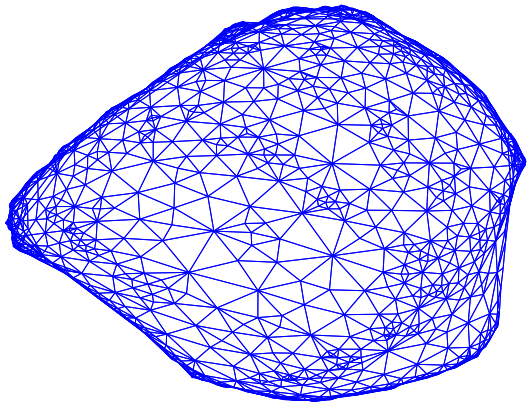
Let's do a more interesting example. I generated the following graph by sampling pixels in the Yale Coat of Arms with probability proportional to their darkness. Those pixels became vertices. I then generated the Delaunay triangulation of these points to form the following graph:

```
load yaleShieldBig
gplot(A,xy)
```



Here is the drawing that I get by using the second and third eigenvectors of the Laplacian.

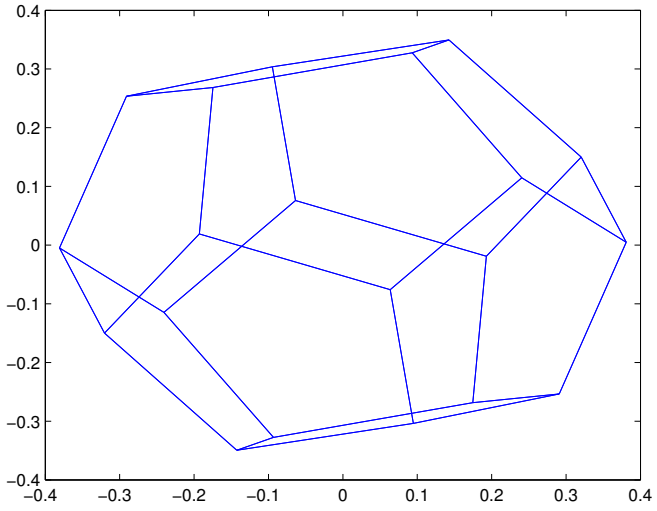
```
L = lap(A);
[v,d] = eigs(L, 3, 'sa');
gplot(A,v(:,[2 3]))
```



That's a great way to draw a graph if you start out knowing nothing about it. It's the first thing I do whenever I meet a strange graph. Note that the middle of the picture is almost planar, although edges do cross near the boundaries.

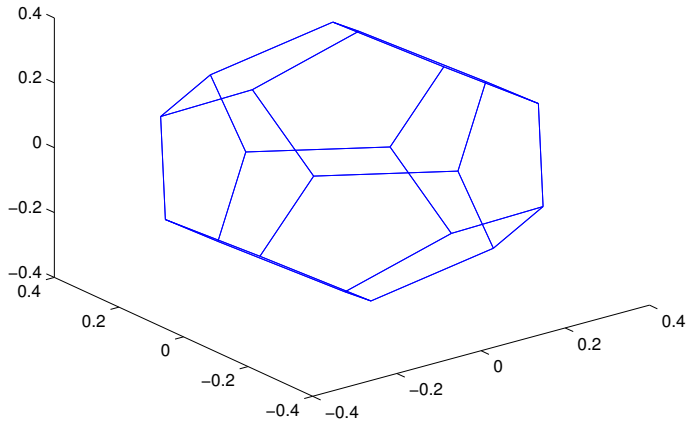
Finally, let's look at a spectral embedding of the edges of the dodecahedron.

```
load dodec
L = lap(A);
[v,d] = eigs(L,3,'sa');
gplot(A,v(:,[2 3]))
```



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. 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.

```
[v,d] = eigs(L,4,'sa');
gplot3(A,v(:,2:4))
```



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.

1.9 Mechanics

I expect to continue to produce lecture notes for every lecture. There will be around 6 problem sets during the semester. You will be allowed to work on them in small groups. There will be no tests or exams.