Chapter 1

Introduction

Before we start with the subject of this notes we want to show how one actually arrives at large eigenvalue problems in practice. In the following, we restrict ourselves to problems from physics [7, 18, 14] and computer science.

1.1 What makes eigenvalues interesting?

In physics, eigenvalues are usually related to vibrations. Objects like violin strings, drums, bridges, sky scrapers can swing. They do this at certain frequencies. And in some situations they swing so much that they are destroyed. On November 7, 1940, the Tacoma narrows bridge collapsed, less than half a year after its opening. Strong winds excited the bridge so much that the platform in reinforced concrete fell into pieces. A few years ago the London millennium footbridge started wobbling in a way that it had to be closed. The wobbling had been excited by the pedestrians passing the bridge. These are prominent examples of vibrating structures.

But eigenvalues appear in many other places. Electric fields in cyclotrones, a special form of particle accelerators, have to oscillate in a precise manner, in order to accelerate the charged particles that circle around its center. The solutions of the Schrödinger equation from quantum physics and quantum chemistry have solutions that correspond to vibrations of the, say, molecule it models. The eigenvalues correspond to energy levels that molecule can occupy.

Many characteristic quantities in science *are* eigenvalues:

- decay factors,
- frequencies,
- norms of operators (or matrices),
- singular values,
- condition numbers.

In the sequel we give a number of examples that show why computing eigenvalues is important. At the same time we introduce some notation.

1.2 Example 1: The vibrating string

1.2.1 Problem setting

Let us consider a string as displayed in Fig. 1.1. The string is fixed at both ends, at $x = 0$

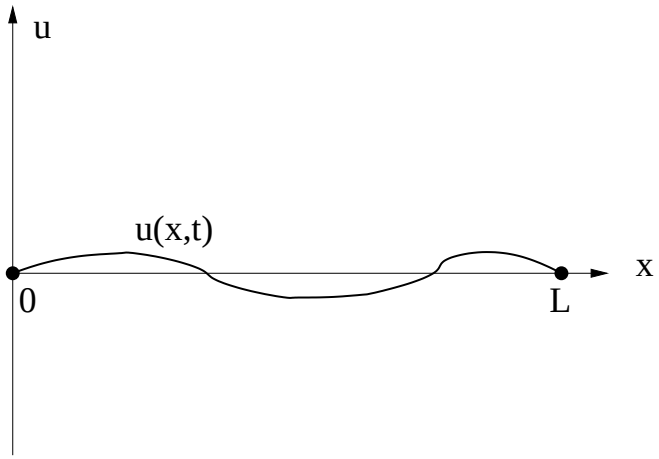


Figure 1.1: A vibrating string fixed at both ends.

and $x = L$. The x-axis coincides with the string's equilibrium position. The displacement of the rest position at x, $0 < x < L$, and time t is denoted by $u(x, t)$.

We will assume that the spatial derivatives of u are not very large:

$$\left|\frac{\partial u}{\partial x}\right| \quad \text{is small.}$$

This assumption entails that we may neglect terms of higher order.

Let $v(x, t)$ be the velocity of the string at position x and at time t. Then the kinetic energy of a string section ds of mass $dm = \rho\, ds$ is given by

$$dT = \frac{1}{2} dm\, v^2 = \frac{1}{2}\rho\, ds \left(\frac{\partial u}{\partial t}\right)^2. \tag{1.1}$$

From Fig. 1.2 we see that $ds^2 = dx^2 + \left(\frac{\partial u}{\partial x}\right)^2 dx^2$ and thus

$$\frac{ds}{dx} = \sqrt{1 + \left(\frac{\partial u}{\partial x}\right)^2} = 1 + \frac{1}{2}\left(\frac{\partial u}{\partial x}\right)^2 + \text{ higher order terms.}$$

Plugging this into (1.1) and omitting also the second order term (leaving just the number 1) gives

$$dT = \frac{\rho\, dx}{2}\left(\frac{\partial u}{\partial t}\right)^2.$$

The kinetic energy of the whole string is obtained by integrating over its length,

$$T = \int_0^L dT(x) = \frac{1}{2}\int_0^L \rho(x)\left(\frac{\partial u}{\partial t}\right)^2 dx$$

The potential energy of the string has two components

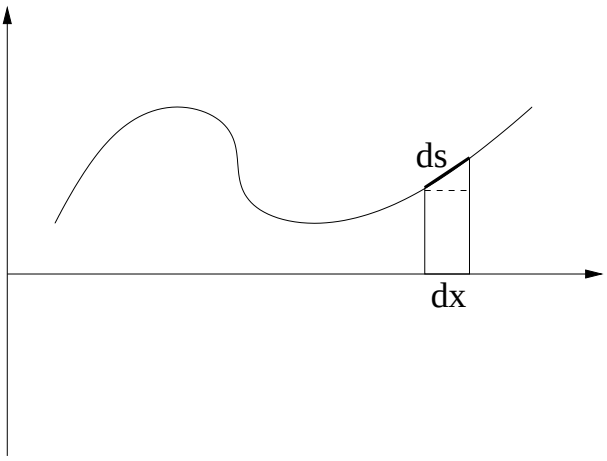


Figure 1.2: A vibrating string, local picture.

1. the stretching times the exerted strain τ,

$$\tau \int_0^L ds - \tau \int_0^L dx = \tau \int_0^L \left(\sqrt{1 + \left(\frac{\partial u}{\partial x} \right)^2} - 1 \right) dx$$
$$= \tau \int_0^L \left(\frac{1}{2} \left(\frac{\partial u}{\partial x} \right)^2 + \text{ higher order terms} \right) dx$$

2. exterior forces of density f,

$$- \int_0^L f u dx.$$

Summing up, the potential energy of the string becomes

$$V = \int_0^L \left(\frac{\tau}{2} \left(\frac{\partial u}{\partial x} \right)^2 - f u \right) dx. \tag{1.2}$$

To consider the motion (vibration) of the string in a certain time interval $t_1 \leq t \leq t_2$ we form the integral

$$I(u) = \int_{t_1}^{t_2} (T - V)\, dt = \frac{1}{2} \int_{t_1}^{t_2} \int_0^L \left[\rho(x) \left(\frac{\partial u}{\partial t} \right)^2 - \tau \left(\frac{\partial u}{\partial x} \right)^2 + 2 f u \right] dx\, dt \tag{1.3}$$

Here functions $u(x,t)$ are admitted that are differentiable with respect to x and t and satisfy the **boundary conditions (BC)** that correspond to the fixing,

$$u(0,t) = u(L,t) = 0, \qquad t_1 \leq t \leq t_2, \tag{1.4}$$

as well as given **initial conditions** and **end conditions**,

$$\begin{array}{l} u(x,t_1) = u_1(x), \\ u(x,t_2) = u_2(x), \end{array} \qquad 0 < x < L. \tag{1.5}$$

According to the **principle of Hamilton** a mechanical system with kinetic energy T and potential energy V behaves in a time interval $t_1 \le t \le t_2$ for given initial and end positions such that

$$I = \int_{t_1}^{t_2} L\,dt, \qquad L = T - V,$$

is minimized.

Let $u(x,t)$ be such that $I(u) \le I(w)$ *for all* w, that satisfy the initial, end, and boundary conditions. Let $w = u + \varepsilon\, v$ with

$$(*) \qquad v(0,t) = v(L,t) = 0, \qquad v(x,t_1) = v(x,t_2) = 0.$$

v is called a *variation*. We now consider $I(u + \varepsilon\, v)$ as a function of ε. Then we have the equivalence

$$I(u) \text{ minimal} \quad \Longleftrightarrow \quad \boxed{\frac{dI}{d\varepsilon}(u) = 0 \text{ for all admitted } v.}$$

Plugging $u + \varepsilon\, v$ into eq. (1.3) we obtain

$$\begin{aligned} I(u+\varepsilon\, v) &= \frac{1}{2}\int_{t_1}^{t_2}\int_{0}^{L}\left[\rho(x)\left(\frac{\partial(u+\varepsilon\, v)}{\partial t}\right)^2 - \tau\left(\frac{\partial(u+\varepsilon\, v)}{\partial x}\right)^2 + 2f(u+\varepsilon\, v)\right] dx\,dt \\ &= I(u) + \varepsilon\int_{t_1}^{t_2}\int_{0}^{L}\left[\rho(x)\frac{\partial u}{\partial t}\frac{\partial v}{\partial t} - \tau\frac{\partial u}{\partial x}\frac{\partial v}{\partial x} + 2fv\right] dx\,dt + \mathcal{O}(\varepsilon^2). \end{aligned} \tag{1.6}$$

Thus, after integration by parts, exploiting the conditions in $(*)$, the equation

$$\frac{\partial I}{\partial \varepsilon} = \int_{t_1}^{t_2}\int_{0}^{L}\left[\rho\frac{\partial^2 u}{\partial t^2} - \tau\frac{\partial^2 u}{\partial x^2} + 2\,f\right] v\,dx\,dt = 0$$

must hold *for all* admissible v. Therefore, the bracketed expression must vanish,

$$-\rho\frac{\partial^2 u}{\partial t^2} + \tau\frac{\partial^2 u}{\partial x^2} = 2\,f. \tag{1.7}$$

This last differential equation is named **Euler-Lagrange equation**.

If the force is proportional to the displacement $u(x,t)$ (like, e.g., in Hooke's law) then we get a differential equation of the form

$$\boxed{\begin{gathered} -\rho(x)\frac{\partial^2 u}{\partial t^2} + \frac{\partial}{\partial x}\left(p(x)\frac{\partial u}{\partial x}\right) + q(x)u(x,t) = 0. \\ u(0,t) = u(1,t) = 0 \end{gathered}} \tag{1.8}$$

which is a special case of the Euler-Lagrange equation (1.7). Here, $\rho(x)$ plays the role of a mass density, $p(x)$ of a locally varying elasticity module. We do not specify initial and end conditions for the moment. Note that there are no *external* forces present in (1.8).

From physics we know that $\rho(x) > 0$ and $p(x) > 0$ for all x. These properties are of importance also from a mathematical view point! For simplicity, we assume that $\rho(x) = 1$.

1.2.2 The method of separation of variables

For the solution u in (1.8) we make the *ansatz*

$$u(x,t) = v(t)w(x). \tag{1.9}$$

Here, v is a function that depends only on the time t, while w depends only on the spatial variable x. With this ansatz (1.8) becomes

$$v''(t)w(x) - v(t)(p(x)w'(x))' - q(x)v(t)w(x) = 0. \tag{1.10}$$

Now we *separate* the variables depending on t from those depending on x,

$$\frac{v''(t)}{v(t)} = \frac{1}{w(x)}(p(x)w'(x))' + q(x).$$

This equation holds *for any* t and x. We can vary t and x independently of each other without changing the value on each side of the equation. Therefore, each side of the equation must be equal to a constant value. We denote this value by $-\lambda$. Thus, from the left side we obtain the equation

$$-v''(t) = \lambda v(t). \tag{1.11}$$

This equation has the well-known solution $v(t) = a \cdot \cos(\sqrt{\lambda}t) + b \cdot \sin(\sqrt{\lambda}t)$ where $\lambda > 0$ is assumed. The right side of (1.10) gives a so-called **Sturm-Liouville problem**

$$\boxed{-(p(x)w'(x))' + q(x)w(x) = \lambda w(x), \qquad w(0) = w(1) = 0.} \tag{1.12}$$

A value λ for which (1.12) has a *non-trivial* (i.e. nonzero) solution w is called an **eigenvalue**; w is a corresponding **eigenfunction**. It is known that all eigenvalues of (1.12) are positive. By means of our ansatz (1.9) we get

$$u(x,t) = w(x)\left[a \cdot \cos(\sqrt{\lambda}t) + b \cdot \sin(\sqrt{\lambda}t)\right]$$

as a solution of (1.8). It is known that (1.12) has infinitely many real positive eigenvalues $0 < \lambda_1 \le \lambda_2 \le \cdots$, $(\lambda_k \underset{k\to\infty}{\longrightarrow} \infty)$. (1.12) has a non-zero solution, say $w_k(x)$, *only* for these particular values λ_k. Therefore, the general solution of (1.8) has the form

$$u(x,t) = \sum_{k=0}^{\infty} w_k(x)\left[a_k \cdot \cos(\sqrt{\lambda_k}\,t) + b_k \cdot \sin(\sqrt{\lambda_k}\,t)\right]. \tag{1.13}$$

The coefficients a_k and b_k are determined by initial and end conditions. We could, e.g., require that

$$u(x,0) = \sum_{k=0}^{\infty} a_k w_k(x) = u_0(x),$$
$$\frac{\partial u}{\partial t}(x,0) = \sum_{k=0}^{\infty} \sqrt{\lambda_k}\, b_k w_k(x) = u_1(x),$$

where u_0 and u_1 are given functions. It is known that the w_k form an orthogonal basis in the space of square integrable functions $L_2(0,1)$,

$$\int_0^1 w_k(x)w_\ell(x)dx = \gamma_k \delta_{k\ell}.$$

Therefore, it is not difficult to compute the coefficients a_k and b_k,

$$a_k = \int_0^1 u_0(x)w_k(x)dx/\gamma_k, \qquad b_k = \int_0^1 u_1(x)w_k(x)dx/\gamma_k\sqrt{\lambda_k}.$$

In concluding, we see that the difficult problem to solve is the eigenvalue problem (1.12). Knowing the eigenvalues and eigenfunctions the general solution of the time-dependent problem (1.8) is easy to form.

Eq. (1.12) can be solved analytically only in very special situation, e.g., if all coefficients are constants. In general a *numerical method* is needed to solve the Sturm-Liouville problem (1.12).

1.3 Numerical methods for solving 1-dimensional problems

In this section we consider three methods to solve the Sturm-Liouville problem.

1.3.1 Finite differences

We approximate $w(x)$ by its values at the discrete points $x_i = ih$, $h = 1/(n+1)$, $i = 1, \ldots, n$.

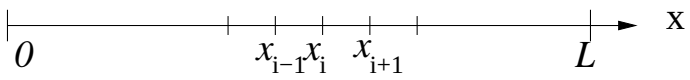


Figure 1.3: Grid points in the interval $(0, L)$.

At point x_i we approximate the derivatives by **finite differences**. We proceed as follows. First we write

$$\frac{d}{dx}g(x_i) \approx \frac{g(x_{i+\frac{1}{2}}) - g(x_{i-\frac{1}{2}})}{h}.$$

For $g = p\frac{dw}{dx}$ we get

$$g(x_{i+\frac{1}{2}}) = p(x_{i+\frac{1}{2}})\frac{w(x_{i+1}) - w(x_i)}{h}$$

and, finally, for $i = 1, \ldots, n$,

$$\begin{aligned} -\frac{d}{dx}\left(p\frac{dw}{dx}(x_i)\right) &\approx -\frac{1}{h}\left[p(x_{i+\frac{1}{2}})\frac{w(x_{i+1}) - w(x_i)}{h} - p(x_{i-\frac{1}{2}})\frac{w(x_i) - w(x_{i-1})}{h}\right] \\ &= \frac{1}{h^2}\left[-p(x_{i-\frac{1}{2}})w_{i-1} + (p(x_{i-\frac{1}{2}}) + p(x_{i+\frac{1}{2}}))w_i - p(x_{i+\frac{1}{2}})w_{i+1}\right]. \end{aligned}$$

Note that at the interval endpoints $w_0 = w_{n+1} = 0$.

We can collect all equations in a matrix equation,

$$\begin{bmatrix} \frac{p(x_{\frac{1}{2}}) + p(x_{\frac{3}{2}})}{h^2} + q(x_1) & -\frac{p(x_{\frac{3}{2}})}{h^2} & & \\ -\frac{p(x_{\frac{3}{2}})}{h^2} & \frac{p(x_{\frac{3}{2}}) + p(x_{\frac{5}{2}})}{h^2} + q(x_2) & -\frac{p(x_{\frac{5}{2}})}{h^2} & \\ & -\frac{p(x_{\frac{5}{2}})}{h^2} & \ddots & \ddots \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_n \end{bmatrix} = \lambda \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_n \end{bmatrix},$$

or, briefly,

$$A\mathbf{w} = \lambda \mathbf{w}. \tag{1.14}$$

By construction, A is symmetric and tridiagonal. One can show that it is positive definite as well. Note that this matrix has just a few nonzeros: out of the n^2 elements of A only $3n-2$ are nonzero. This is an example of a **sparse** matrix.

1.3.2 The finite element method

We write (1.12) in the form

Find a twice differentiable function w with $w(0) = w(1) = 0$ such that

$$\int_0^1 \left[-(p(x)w'(x))' + q(x)w(x) - \lambda w(x)\right] \phi(x)dx = 0$$

for all smooth functions ϕ that satisfy $\phi(0) = \phi(1) = 0$.

To relax the requirements on w we integrate by parts and get the new so-called *weak* or *variational form* of the problem:

Find a differentiable function w with $w(0) = w(1) = 0$ such that

$$\int_0^1 \left[p(x)w(x)'\phi'(x) + q(x)w(x)\phi(x) - \lambda w(x)\phi(x)\right] dx = 0 \tag{1.15}$$

for all differentiable functions ϕ that satisfy $\phi(0) = \phi(1) = 0$.

Remark: Requiring continuous differentiability is too strong and does not lead to a mathematically suitable formulation. In particular, the test functions that will be used below are not differentiable in the classical sense. It is more appropriate to require w and ϕ to be *weakly* differentiable. In terms of Sobolev spaces: $w, \phi \in H_0^1([0,1])$. An introduction to Sobolev spaces is, however, beyond the scope of these notes.

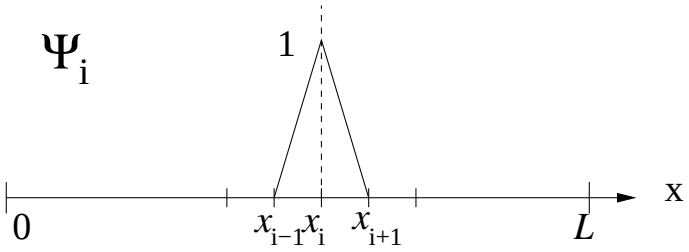


Figure 1.4: A basis function of the finite element space: a hat function.

We now write w as the linear combination

$$w(x) = \sum_{i=1}^{n} \xi_i \, \Psi_i(x), \tag{1.16}$$

where

$$\Psi_i(x) = \left(1 - \frac{|x - x_i|}{h}\right)_+ = \max\{0,\ 1 - \frac{|x - x_i|}{h}\}, \tag{1.17}$$

is the function that is linear in each interval (x_i, x_{i+1}) and satisfies

$$\Psi_i(x_k) = \delta_{ik} := \begin{cases} 1, & i = k, \\ 0, & i \neq k. \end{cases}$$

An example of such a basis function, a so-called *hat function*, is displayed in Fig. 1.4.

We now replace w in (1.15) by the linear combination (1.16), and replace testing 'against all ϕ' by testing against all Ψ_j. In this way (1.15) becomes

$$\int_0^1 \left(-p(x)(\sum_{i=1}^n \xi_i \, \Psi_i'(x))\Psi_j'(x) + (q(x) - \lambda) \sum_{i=1}^n \xi_i \, \Psi_i(x)\Psi_j(x) \right) dx, \quad \text{for all } j,$$

or,

$$\boxed{\sum_{i=1}^n \xi_i \int_0^1 \left(p(x)\Psi_i'(x)\Psi_j'(x) + (q(x) - \lambda)\Psi_i(x)\Psi_j(x) \right) dx = 0, \quad \text{for all } j.} \tag{1.18}$$

These last equations are called the **Rayleigh–Ritz–Galerkin** equations. Unknown are the n values ξ_i and the eigenvalue λ. In matrix notation (1.18) becomes

$$A\mathbf{x} = \lambda M \mathbf{x} \tag{1.19}$$

with

$$a_{ij} = \int_0^1 \left(p(x)\Psi_i'\Psi_j' + q(x)\Psi_i\Psi_j \right) dx \quad \text{and} \quad m_{ij} = \int_0^1 \Psi_i \Psi_j \, dx$$

For the specific case $p(x) = 1 + x$ and $q(x) = 1$ we get

$$a_{kk} = \int_{(k-1)h}^{kh} \left[(1+x)\frac{1}{h^2} + \left(\frac{x - (k-1)h}{h} \right)^2 \right] dx + \int_{kh}^{(k+1)h} \left[(1+x)\frac{1}{h^2} + \left(\frac{(k+1)h - x}{h} \right)^2 \right] dx = 2(n+1+k) + \frac{2}{3}\frac{1}{n+1}$$

$$a_{k,k+1} = \int_{kh}^{(k+1)h} \left[(1+x)\frac{1}{h^2} + \frac{(k+1)h - x}{h} \cdot \frac{x - kh}{h} \right] dx = -n - \frac{3}{2} - k + \frac{1}{6}\frac{1}{n+1}$$

In the same way we get

$$M = \frac{1}{6(n+1)} \begin{bmatrix} 4 & 1 & & \\ 1 & 4 & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & 4 \end{bmatrix}$$

Notice that both matrices A and M are symmetric tridiagonal and positive definite.

1.3.3 Global functions

Formally we proceed as with the finite element method, i.e., we solve equation (1.18). But now we choose the $\Psi_k(x)$ to be functions with global support[1]. We could, e.g., set

$$\Psi_k(x) = \sin k\pi x,$$

[1] The support of a function f is the set of arguments x for which $f(x) \neq 0$.

functions that are differentiable and satisfy the homogeneous boundary conditions. The Ψ_k are eigenfunctions of the nearby problem $-u''(x) = \lambda u(x)$, $u(0) = u(1) = 0$ corresponding to the eigenvalue $k^2\pi^2$. The elements of matrix A are given by

$$\begin{aligned} a_{kk} &= \int_0^1 \left[(1+x)k^2\pi^2 \cos^2 k\pi x + \sin^2 k\pi x\right]\, dx = \frac{3}{4}k^2\pi^2 + \frac{1}{2}, \\ a_{kj} &= \int_0^1 \left[(1+x)kj\pi^2 \cos k\pi x \cos j\pi x + \sin k\pi x \sin j\pi x\right]\, dx \\ &= \frac{kj(k^2+j^2)((-1)^{k+j}-1)}{(k^2-j^2)^2}, \quad k \neq j. \end{aligned}$$

1.3.4 A numerical comparison

We consider the above 1-dimensional eigenvalue problem

$$-((1+x)w'(x))' + w(x) = \lambda w(x), \qquad w(0) = w(1) = 0, \tag{1.20}$$

and solve it with the finite difference and finite element methods as well as with the global functions method. The results are given in Table 1.1.

Clearly the global function method is the most powerful of them all. With 80 basis functions the eigenvalues all come right. The convergence rate is exponential.

With the finite difference and finite element methods the eigenvalues exhibit quadratic convergence rates. If the mesh width h is reduced by a factor of $q = 2$, the error in the eigenvalues is reduced by the factor $q^2 = 4$. There exist higher order finite elements and higher order finite difference stencils [11, 6].

1.4 Example 2: The heat equation

The instationary temperature distribution $u(\mathbf{x}, t)$ in an insulated container satisfies the equations

$$\begin{aligned} \frac{\partial u(\mathbf{x},t)}{\partial t} - \Delta u(\mathbf{x},t) &= 0, && \mathbf{x} \in \Omega,\ t > 0, \\ \frac{\partial u(\mathbf{x},t)}{\partial n} &= 0, && \mathbf{x} \in \partial\Omega,\ t > 0, \\ u(\mathbf{x},0) &= u_0(\mathbf{x}), && \mathbf{x} \in \Omega. \end{aligned} \tag{1.21}$$

Here Ω is a 3-dimensional domain[2] with boundary $\partial\Omega$. $u_0(\mathbf{x}), \mathbf{x} = (x_1, x_2, x_3)^T \in \mathbb{R}^3$, is a given bounded, sufficiently smooth function.

$$\Delta u = \sum \frac{\partial^2 u}{\partial {x_i}^2} \tag{1.22}$$

is called the *Laplace operator* and $\frac{\partial u}{\partial n}$ denotes the derivative of u in direction of the outer normal vector $\mathbf{n}$. To solve the heat equation the **method of separation of variables** is employed. We write u in the form

$$u(\mathbf{x}, t) = v(t)w(\mathbf{x}). \tag{1.23}$$

[2] In the sequel we understand a domain to be bounded and simply connected.

Finite difference method				
k	$\lambda_k(n=10)$	$\lambda_k(n=20)$	$\lambda_k(n=40)$	$\lambda_k(n=80)$
1	15.245	15.312	15.331	15.336
2	56.918	58.048	58.367	58.451
3	122.489	128.181	129.804	130.236
4	206.419	224.091	229.211	230.580
5	301.499	343.555	355.986	359.327
6	399.367	483.791	509.358	516.276
7	492.026	641.501	688.398	701.185
8	578.707	812.933	892.016	913.767
9	672.960	993.925	1118.969	1153.691
10	794.370	1179.947	1367.869	1420.585

Finite element method				
k	$\lambda_k(n=10)$	$\lambda_k(n=20)$	$\lambda_k(n=40)$	$\lambda_k(n=80)$
1	15.447	15.367	15.345	15.340
2	60.140	58.932	58.599	58.511
3	138.788	132.657	130.979	130.537
4	257.814	238.236	232.923	231.531
5	426.223	378.080	365.047	361.648
6	654.377	555.340	528.148	521.091
7	949.544	773.918	723.207	710.105
8	1305.720	1038.433	951.392	928.983
9	1702.024	1354.106	1214.066	1178.064
10	2180.159	1726.473	1512.784	1457.733

Global function method				
k	$\lambda_k(n=10)$	$\lambda_k(n=20)$	$\lambda_k(n=40)$	$\lambda_k(n=80)$
1	15.338	15.338	15.338	15.338
2	58.482	58.480	58.480	58.480
3	130.389	130.386	130.386	130.386
4	231.065	231.054	231.053	231.053
5	360.511	360.484	360.483	360.483
6	518.804	518.676	518.674	518.674
7	706.134	705.631	705.628	705.628
8	924.960	921.351	921.344	921.344
9	1186.674	1165.832	1165.823	1165.822
10	1577.340	1439.083	1439.063	1439.063

Table 1.1: Numerical solutions of problem (1.20)

If a constant λ can be found such that

$$\boxed{\begin{aligned} &\Delta w(\mathbf{x}) + \lambda w(\mathbf{x}) = 0, \quad w(\mathbf{x}) \neq 0, && \mathbf{x} \text{ in } \Omega, \\ &\frac{\partial w(\mathbf{x})}{\partial n} = 0, && \mathbf{x} \text{ on } \partial\Omega, \end{aligned}} \tag{1.24}$$

then the product $u = vw$ is a solution of (1.21) if and only if

$$\frac{dv(t)}{dt} + \lambda v(t) = 0, \tag{1.25}$$

the solution of which has the form $a \cdot \exp(-\lambda t)$. By separating variables, the problem (1.21) is divided in two subproblems that are hopefully easier to solve. A value λ, for which (1.24) has a *nontrivial* (i.e. a nonzero) solution is called an *eigenvalue*; w then is called a *corresponding eigenfunction*.

If λ_n is an eigenvalue of problem (1.24) with corresponding eigenfunction w_n, then

$$e^{-\lambda_n t} w_n(\mathbf{x})$$

is a solution of the first two equations in (1.21). It is known that equation (1.24) has *infinitely many* real eigenvalues $0 \leq \lambda_1 \leq \lambda_2 \leq \cdots$, that tend to infinity, $\lambda_n \longrightarrow \infty$ as $n \to \infty$. Multiple eigenvalues are counted according to their multiplicity. An arbitrary bounded piecewise continuous function can be represented as a linear combination of the eigenfunctions $w_1, w_2, \ldots$ Therefore, the solution of (1.21) can be written in the form

$$u(\mathbf{x}, t) = \sum_{n=1}^{\infty} c_n e^{-\lambda_n t} w_n(\mathbf{x}), \tag{1.26}$$

where the coefficients c_n are determined such that

$$u_0(\mathbf{x}) = \sum_{n=1}^{\infty} c_n w_n(\mathbf{x}). \tag{1.27}$$

The smallest eigenvalue of (1.24) is $\lambda_1 = 0$ with $w_1 = 1$ and $\lambda_2 > 0$. Therefore we see from (1.26) that

$$u(\mathbf{x}, t) \underset{t \to \infty}{\longrightarrow} c_1. \tag{1.28}$$

Thus, in the limit (i.e., as t goes to infinity), the temperature will be constant in the whole container. The convergence rate towards this equilibrium is determined by the smallest *positive* eigenvalue λ_2 of (1.24):

$$\begin{aligned} \|u(\mathbf{x}, t) - c_1\| = \|\sum_{n=2}^{\infty} c_n e^{-\lambda_n t} w_n(\mathbf{x})\| &\leq \sum_{n=2}^{\infty} |e^{-\lambda_n t}| \|c_n w_n(\mathbf{x})\| \\ &\leq e^{-\lambda_2 t} \sum_{n=2}^{\infty} \|c_n w_n(\mathbf{x})\| \leq e^{-\lambda_2 t} \|u_0(\mathbf{x})\|. \end{aligned}$$

Here we have assumed that the value of the constant function $w_1(\mathbf{x})$ is set to unity.

1.5 Example 3: The wave equation

The air pressure $u(\mathbf{x}, t)$ in a volume with acoustically "hard" walls satisfies the equations

$$\frac{\partial^2 u(\mathbf{x},t)}{\partial t^2} - \Delta u(\mathbf{x},t) = 0, \qquad \mathbf{x} \in \Omega,\ t > 0, \tag{1.29}$$

$$\frac{\partial u(\mathbf{x},t)}{\partial n} = 0, \qquad \mathbf{x} \in \partial\Omega,\ t > 0, \tag{1.30}$$

$$u(\mathbf{x},0) = u_0(\mathbf{x}), \qquad \mathbf{x} \in \Omega, \tag{1.31}$$

$$\frac{\partial u(\mathbf{x},0)}{\partial t} = u_1(\mathbf{x}), \qquad \mathbf{x} \in \Omega. \tag{1.32}$$

Sound propagates with speed $-\nabla u$, along the (negative) gradient from high to low pressure.

To solve the wave equation we proceed as with the heat equation in section 1.4: separation of u according to (1.23) leads again to equation (1.24) but now together with

$$\frac{d^2 v(t)}{dt^2} + \lambda v(t) = 0. \tag{1.33}$$

We know this equation from the analysis of the vibrating string, see (1.11). From there we know that the general solution of the wave equation has the form

$$u(x,t) = \sum_{k=0}^{\infty} w_k(x) \left[A_k \cdot \cos(\sqrt{\lambda_k}\, t) + B_k \cdot \sin(\sqrt{\lambda_k}\, t) \right]. \tag{1.13}$$

where the w_k, $k = 1, 2, \ldots$, are the eigenfunctions of the eigenvalue problem (1.24). The coefficients a_k and b_k are determined by (1.31) and (1.32).

If a harmonic oscillation is forced on the system, an *inhomogeneous* problem

$$\frac{\partial^2 u(\mathbf{x},t)}{\partial t^2} - \Delta u(\mathbf{x},t) = f(\mathbf{x},t), \tag{1.34}$$

is obtained. The boundary and initial conditions are taken from (1.29)–(1.32). This problem can be solved by expanding u and f in the eigenfunctions $w_n(\mathbf{x})$,

$$\begin{aligned} u(\mathbf{x},t) &:= \sum_{n=1}^{\infty} \tilde{v}_n(t) w_n(\mathbf{x}), \\ f(\mathbf{x},t) &:= \sum_{n=1}^{\infty} \phi_n(t) w_n(\mathbf{x}). \end{aligned} \tag{1.35}$$

With this approach, $\tilde{v}_n$ has to satisfy equation

$$\frac{d^2 \tilde{v}_n}{dt^2} + \lambda_n \tilde{v}_n = \phi_n(t). \tag{1.36}$$

If $\phi_n(t) = a_n \sin \omega t$, then the solution becomes

$$\tilde{v}_n = A_n \cos \sqrt{\lambda_n} t + B_n \sin \sqrt{\lambda_n} t + \frac{1}{\lambda_n - \omega^2} a_n \sin \omega t. \tag{1.37}$$

A_n and B_n are real constants that are determined by the initial conditions. If ω gets close to $\sqrt{\lambda_n}$, then the last term can be very large. In the limit, if $\omega = \sqrt{\lambda_n}$, $\tilde{v}_n$ gets the form

$$\tilde{v}_n = A_n \cos\sqrt{\lambda_n}t + B_n \sin\sqrt{\lambda_n}t - (a_n/2\omega)t\cos\omega t, \tag{1.38}$$

in which case, $\tilde{v}_n$ is not bounded in time anymore. This phenomenon is called *resonance*. Often resonance is not desirable; it may, e.g., mean the blow up of some structure. In order to prevent resonances eigenvalues have to be known. Possible remedies are changing the domain (the structure) or parameters (the materials).

Remark 1.1. Vibrating membranes satisfy the wave equation, too. In general the boundary conditions are different from (1.30). If the membrane (of a drum) is fixed at its boundary, the condition

$$u(\mathbf{x}, t) = 0 \tag{1.39}$$

is imposed. These boundary conditions are called *Dirichlet boundary conditions*. The boundary conditions in (1.21) and (1.30) are called *Neumann boundary conditions*. Combinations of these two can occur. □

1.6 Numerical methods for solving the Laplace eigenvalue problem in 2D

In this section we again consider the eigenvalue problem

$$-\Delta u(\mathbf{x}) = \lambda u(\mathbf{x}), \qquad \mathbf{x} \in \Omega, \tag{1.40}$$

with the more general boundary conditions

$$u(\mathbf{x}) = 0, \qquad \mathbf{x} \in \Gamma_1 \subset \partial\Omega, \tag{1.41}$$

$$\frac{\partial u}{\partial n}(\mathbf{x}) + \alpha(\mathbf{x})u(\mathbf{x}) = 0, \qquad \mathbf{x} \in \Gamma_2 \subset \partial\Omega. \tag{1.42}$$

Here, Γ_1 and Γ_2 are *disjoint* subsets of $\partial\Omega$ with $\Gamma_1 \cup \Gamma_2 = \partial\Omega$. We restrict ourselves in the following on *two-dimensional* domains and write (x, y) instead of (x_1, x_2).

In general it is not possible to solve a problem of the form (1.40)–(1.42) exactly (analytically). Therefore, one has to resort to numerical approximations. Because we cannot compute with infinitely many variables we have to construct a finite-dimensional eigenvalue problem that represents the given problem as well as possible, i.e., that yields good approximations for the desired eigenvalues and eigenvectors. Since finite-dimensional eigenvalue problem only have a finite number of eigenvalues one cannot expect to get good approximations for all eigenvalues of (1.40)–(1.42).

Two methods for the discretization of eigenvalue problems of the form (1.40)–(1.42) are the *Finite Difference Method* [11, 16, 9] and the *Finite Element Method (FEM)* [6, 15, 8]. We briefly introduce these methods in the following subsections.

1.6.1 The finite difference method

In this section we just want to mediate some impression what the finite difference method is about. Therefore we assume for simplicity that the domain Ω is a square with sides of

length 1: $\Omega = (0,1) \times (0,1)$. We consider the eigenvalue problem

$$\begin{aligned} -\Delta u(x,y) &= \lambda u(x,y), && 0 < x,y < 1 \\ u(0,y) = u(1,y) &= u(x,0) = 0, && 0 < x,y < 1, \\ \frac{\partial u}{\partial n}(x,1) &= 0, && 0 < x < 1. \end{aligned} \tag{1.43}$$

This eigenvalue problem occurs in the computation of eigenfrequencies and eigenmodes of a homogeneous quadratic membrane with three fixed and one free side. It can be solved analytically by separation of the two spatial variables x and y. The eigenvalues are

$$\lambda_{k,l} = \left(k^2 + \frac{(2l-1)^2}{4}\right)\pi^2, \quad k,l \in \mathbb{N},$$

and the corresponding eigenfunctions are

$$u_{k,l}(x,y) = \sin k\pi x \sin \frac{2l-1}{2}\pi y.$$

In the finite difference method one proceeds by defining a rectangular grid with grid points $(x_i, y_j), 0 \le i,j \le N$. The coordinates of the grid points are

$$(x_i, y_j) = (ih, jh), \qquad h = 1/N.$$

By a Taylor expansion one can show that for sufficiently smooth functions u

$$\begin{aligned} -\Delta u(x,y) = \frac{1}{h^2}&(4u(x,y) - u(x-h,y) - u(x+h,y) - u(x,y-h) - u(x,y+h)) \\ &+ O(h^2). \end{aligned}$$

It is therefore straightforward to replace the differential equation $-\Delta u(x,y) = \lambda u(x,y)$ by a difference equation at the interior grid points

$$4u_{i,j} - u_{i-1,j} - u_{i+1,j} - u_{i,j-1} - u_{i,j+1} = \lambda h^2 u_{i,j}, \quad 0 < i,j < N. \tag{1.44}$$

We consider the unknown variables $u_{i,j}$ as approximations of the eigenfunctions at the grid points (i,j):

$$u_{i,j} \approx u(x_i, x_j). \tag{1.45}$$

The Dirichlet boundary conditions are replaced by the equations

$$u_{i,0} = u_{i,N} = u_{0,i}, \qquad 0 < i < N. \tag{1.46}$$

At the points at the upper boundary of Ω we first take the difference equation (1.44)

$$4u_{i,N} - u_{i-1,N} - u_{i+1,N} - u_{i,N-1} - u_{i,N+1} = \lambda h^2 u_{i,N}, \quad 0 \le i \le N. \tag{1.47}$$

The value $u_{i,N+1}$ corresponds to a grid point *outside* of the domain! However the Neumann boundary conditions suggest to reflect the domain at the upper boundary and to extend the eigenfunction symmetrically beyond the boundary. This procedure leads to the equation $u_{i,N+1} = u_{i,N-1}$. Plugging this into (1.47) and multiplying the new equation by the factor $1/2$ gives

$$2u_{i,N} - \frac{1}{2}u_{i-1,N} - \frac{1}{2}u_{i+1,N} - u_{i,N-1} = \frac{1}{2}\lambda h^2 u_{i,N}, \quad 0 < i < N. \tag{1.48}$$

In summary, from (1.44) and (1.48), taking into account that (1.46) we get the matrix equation

$$
\begin{pmatrix}
4 & -1 & 0 & -1 & & & & & & & & \\
-1 & 4 & -1 & 0 & -1 & & & & & & & \\
0 & -1 & 4 & 0 & 0 & -1 & & & & & & \\
-1 & 0 & 0 & 4 & -1 & 0 & -1 & & & & & \\
 & -1 & 0 & -1 & 4 & -1 & 0 & -1 & & & & \\
 & & -1 & 0 & -1 & 4 & 0 & 0 & -1 & & & \\
 & & & -1 & 0 & 0 & 4 & -1 & 0 & -1 & & \\
 & & & & -1 & 0 & -1 & 4 & -1 & 0 & -1 & \\
 & & & & & -1 & 0 & -1 & 4 & 0 & 0 & -1 \\
 & & & & & & -1 & 0 & 0 & 2 & -\frac{1}{2} & 0 \\
 & & & & & & & -1 & 0 & -\frac{1}{2} & 2 & -\frac{1}{2} \\
 & & & & & & & & -1 & 0 & -\frac{1}{2} & 2
\end{pmatrix}
\begin{pmatrix} u_{1,1} \\ u_{1,2} \\ u_{1,3} \\ u_{2,1} \\ u_{2,2} \\ u_{2,3} \\ u_{3,1} \\ u_{3,2} \\ u_{3,3} \\ u_{4,1} \\ u_{4,2} \\ u_{4,3} \end{pmatrix}
\tag{1.49}
$$

$$
= \lambda h^2
\begin{pmatrix}
1 & & & & & & & & & & & \\
 & 1 & & & & & & & & & & \\
 & & 1 & & & & & & & & & \\
 & & & 1 & & & & & & & & \\
 & & & & 1 & & & & & & & \\
 & & & & & 1 & & & & & & \\
 & & & & & & 1 & & & & & \\
 & & & & & & & 1 & & & & \\
 & & & & & & & & \frac{1}{2} & & & \\
 & & & & & & & & & \frac{1}{2} & & \\
 & & & & & & & & & & \frac{1}{2}
\end{pmatrix}
\begin{pmatrix} u_{1,1} \\ u_{1,2} \\ u_{1,3} \\ u_{2,1} \\ u_{2,2} \\ u_{2,3} \\ u_{3,1} \\ u_{3,2} \\ u_{3,3} \\ u_{4,1} \\ u_{4,2} \\ u_{4,3} \end{pmatrix}.
$$

For arbitrary $N > 1$ we define

$$
\mathbf{u}_i := \begin{pmatrix} u_{i,1} \\ u_{i,2} \\ \vdots \\ u_{i,N-1} \end{pmatrix} \in \mathbb{R}^{N-1},
$$

$$
T := \begin{pmatrix}
4 & -1 & & \\
-1 & 4 & \ddots & \\
 & \ddots & \ddots & -1 \\
 & & -1 & 4
\end{pmatrix} \in \mathbb{R}^{(N-1)\times(N-1)},
$$

$$
I := \begin{pmatrix}
1 & & & \\
 & 1 & & \\
 & & \ddots & \\
 & & & 1
\end{pmatrix} \in \mathbb{R}^{(N-1)\times(N-1)}.
$$

In this way we obtain from (1.44), (1.46), (1.48) the discrete eigenvalue problem

$$\begin{pmatrix} T & -I & & \\ -I & T & \ddots & \\ & \ddots & \ddots & -I \\ & & -I & \frac{1}{2}T \end{pmatrix} \begin{pmatrix} \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_3 \\ \mathbf{u}_4 \end{pmatrix} = \lambda h^2 \begin{pmatrix} I & & \\ & \ddots & \\ & & I \\ & & & \frac{1}{2}I \end{pmatrix} \begin{pmatrix} \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_{N-1} \\ \mathbf{u}_N \end{pmatrix} \tag{1.50}$$

of size $N \times (N-1)$. This is a **matrix eigenvalue problem** of the form

$$A\mathbf{x} = \lambda M\mathbf{x}, \tag{1.51}$$

where A and M are **symmetric** and M additionally is **positive definite**. If M is the identity matrix, we call (1.51) a *special* and otherwise a *generalized* eigenvalue problem. In these lecture notes we deal with numerical methods, to solve eigenvalue problems like these.

In the case (1.50) it is easy to obtain a special (symmetric) eigenvalue problem by a simple transformation: By left multiplication by

$$\begin{pmatrix} I & & & \\ & I & & \\ & & I & \\ & & & \sqrt{2}I \end{pmatrix}$$

we obtain from (1.50)

$$\begin{pmatrix} T & -I & & \\ -I & T & -I & \\ & -I & T & -\sqrt{2}I \\ & & -\sqrt{2}I & T \end{pmatrix} \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \mathbf{u}_3 \\ \frac{1}{\sqrt{2}}\mathbf{u}_4 \end{pmatrix} = \lambda h^2 \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \mathbf{u}_3 \\ \frac{1}{\sqrt{2}}\mathbf{u}_4 \end{pmatrix}. \tag{1.52}$$

A property common to matrices obtained by the finite difference method are its **sparsity**. Sparse matrices have only very few nonzero elements.

In real-world applications domains often cannot be covered easily by a rectangular grid. In this situation and if boundary conditions are complicated the method of finite differences can be difficult to implement. Because of this the finite element method is often the method of choice.

Nevertheless, problems that are posed on rectangular grids can be solved very efficiently. Therefore, tricks are used to deal with irregular boundaries. The solution of the problem may be extended artificially beyond the boundary, see e.g. [1, 17, 9]. Similar techiques, so-called *immersed boundary conditions* are applied at (irregular) interfaces where, e.g., equations or parameters change [11].

1.6.2 The finite element method (FEM)

Let $(\lambda, u) \in \mathbb{R} \times V$ be an eigenpair of problem (1.40)–(1.42). Then

$$\int_\Omega (\Delta u + \lambda u) v \, dx \, dy = 0, \quad \forall v \in V, \tag{1.53}$$

where V is vector space of bounded twice differentiable functions that satisfy the boundary conditions (1.41)–(1.42). By partial integration (Green's formula) this becomes

$$\int_\Omega \nabla u \nabla v \, dx \, dy + \int_{\Gamma_2} \alpha \, u \, v \, ds = \lambda \int_\Omega u \, v \, dx \, dy, \quad \forall v \in V, \tag{1.54}$$

or

$$a(u,v) = (u,v), \qquad \forall v \in V \tag{1.55}$$

where

$$a(u,v) = \int_\Omega \nabla u \, \nabla v \, dx \, dy + \int_{\Gamma_2} \alpha \, u \, v \, ds, \quad \text{and} \quad (u,v) = \int_\Omega u \, v \, dx \, dy.$$

We complete the space V with respect to the Sobolev norm [8, 3]

$$\sqrt{\int_\Omega (u^2 + |\nabla u|^2) \, dx \, dy}$$

to become a Hilbert space H [3, 19]. H is the space of quadratic integrable functions with quadratic integrable first derivatives that satisfy the Dirichlet boundary conditions (1.41)

$$u(x,y) = 0, \qquad (x,y) \in \Gamma_1.$$

(Functions in H in general do not satisfy the so-called *natural* boundary conditions (1.42).) One can show [19] that the eigenvalue problem (1.40)–(1.42) is equivalent with the eigenvalue problem

$$\begin{array}{l} \text{Find } (\lambda, u) \in \mathbb{R} \times H \text{ such that} \\ a(u,v) = \lambda (u,v) \quad \forall v \in H. \end{array} \tag{1.56}$$

(The essential point is to show that the eigenfunctions of (1.56) are elements of V.)

The Rayleigh–Ritz–Galerkin method

In the Rayleigh–Ritz–Galerkin method one proceeds as follows: A set of *linearly independent* functions

$$\phi_1(x,y), \cdots, \phi_n(x,y) \in H, \tag{1.57}$$

are chosen. These functions span a *subspace* S of H. Then, problem (1.56) is solved where H is replaced by S.

$$\boxed{\begin{array}{l} \text{Find } (\lambda, u) \in \mathbb{R} \times S \text{ such that} \\ a(u,v) = \lambda (u,v) \quad \forall v \in S. \end{array}} \tag{1.58}$$

With the Ritz ansatz [15]

$$u = \sum_{i=1}^{n} x_i \phi_i, \tag{1.59}$$

equation (1.58) becomes

$$\begin{array}{l} \text{Find } (\lambda, \mathbf{x}) \in \mathbb{R} \times \mathbb{R}^n \text{ such that} \\ \displaystyle\sum_{i=1}^{n} x_i a(\phi_i, v) = \lambda \sum_{i=1}^{n} x_i (\phi_i, v), \quad \forall v \in S. \end{array} \tag{1.60}$$

Eq. (1.60) must hold *for all* $v \in S$, in particular for $v = \phi_1, \cdots, \phi_n$. But since the $\phi_i, 1 \le i \le n$, form a basis of S, equation (1.60) is equivalent with

$$\sum_{i=1}^{n} x_i a(\phi_i, \phi_j) = \lambda \sum_{i=1}^{n} x_i (\phi_i, \phi_j), \quad 1 \le j \le n. \tag{1.61}$$

This is a matrix eigenvalue problem of the form

$$A\mathbf{x} = \lambda M \mathbf{x} \tag{1.62}$$

where

$$\mathbf{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}, \quad A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix}, \quad M = \begin{pmatrix} m_{11} & \cdots & m_{1n} \\ \vdots & \ddots & \vdots \\ m_{n1} & \cdots & m_{nn} \end{pmatrix} \tag{1.63}$$

with

$$a_{ij} = a(\phi_i, \phi_j) = \int_\Omega \nabla\phi_i \, \nabla\phi_j \, dx \, dy + \int_{\Gamma_2} \alpha \, \phi_i \, \phi_j \, ds$$

and

$$m_{ij} = (\phi_i, \phi_j) = \int_\Omega \phi_i \, \phi_j \, dx \, dy.$$

The **finite element method (FEM)** is a *special case* of the Rayleigh–Ritz method. In the FEM the subspace S and in particular the basis $\{\phi_i\}$ is chosen in a particularly clever way. For simplicity we assume that the domain Ω is a simply connected domain with a polygonal boundary, c.f. Fig 1.5. (This means that the boundary is composed entirely of straight line segments.) This domain is now partitioned into triangular subdomains

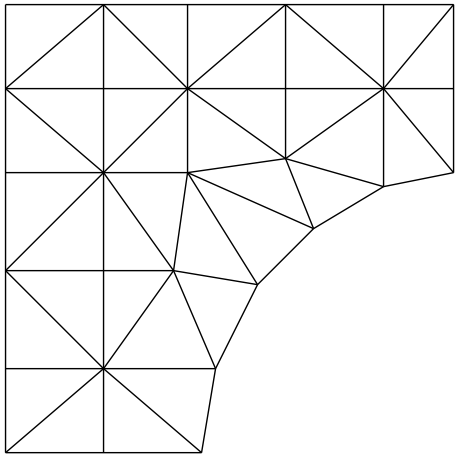


Figure 1.5: Triangulation of a domain Ω

$T_1, \cdots, T_N$, so-called *elements*, such that

$$T_i \cap T_j = \emptyset \quad \text{for all } i \ne j, \quad \text{and} \quad \bigcup_e \overline{T_e} = \overline{\Omega}. \tag{1.64}$$

Finite element spaces for solving (1.40)–(1.42) are typically composed of functions that are *continuous* in Ω and are *polynomials* on the individual subdomains T_e. Such functions

are called *piecewise polynomials*. Notice that this construction provides a subspace of the Hilbert space H but not of V, i.e., the functions in the finite element space are not very smooth and the natural boundary conditions are not satisfied.

An essential issue is the selection of the *basis* of the finite element space S. If $S_1 \subset H$ is the space of continuous, piecewise linear functions (the restriction to T_e is a polynomial of degree 1) then a function in S_1 is uniquely determined by its values at the vertices of the triangles. Let these *nodes*, except those on the boundary portion Γ_1, be numbered from 1 to n, see Fig. 1.6. Let the coordinates of the i-th node be (x_i, y_i). Then $\phi_i(x, y) \in S_1$ is defined by

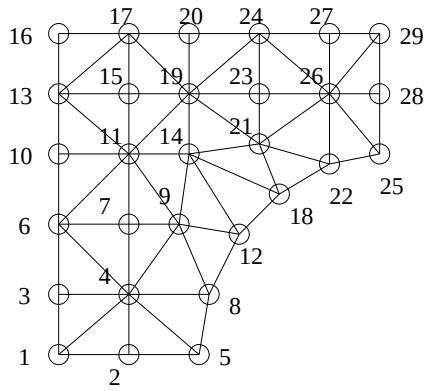


Figure 1.6: Numbering of nodes on Ω (piecewise linear polynomials)

$$\phi_i((x_j, y_j)) := \delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \tag{1.65}$$

A typical basis function ϕ_i is sketched in Figure 1.7.

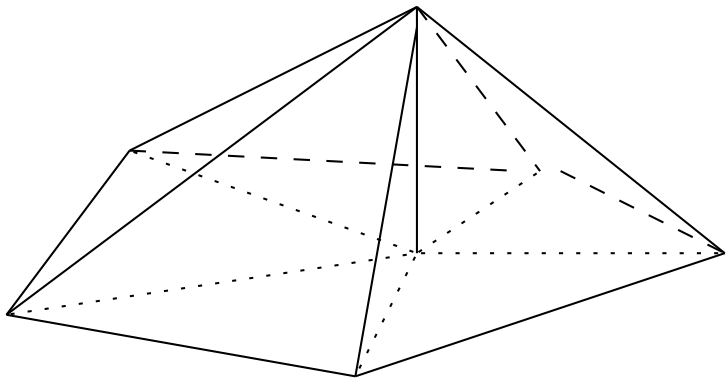


Figure 1.7: A piecewise linear basis function (or hat function)

Another often used finite element element space is $S_2 \subset H$, the space of continuous, *piecewise quadratic* polynomials. These functions are (or can be) uniquely determined by their values at the vertices and edge midpoints of the triangle. The basis functions are defined according to (1.65). There are two kinds of basis functions ϕ_i now, first those that are 1 at a vertex and second those that are 1 at an edge midpoint, cf. Fig. 1.8. One immediately sees that for most $i \neq j$

$$a(\phi_i, \phi_j) = 0, \quad (\phi_i, \phi_j) = 0. \tag{1.66}$$

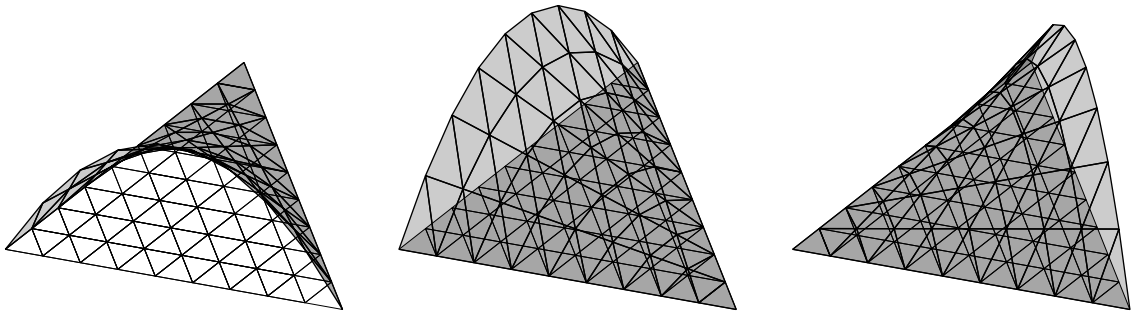


Figure 1.8: The piecewise quadratic basis functions corresponding to the edge midpoints [5]

Therefore the matrices A and M in (1.62)–(1.63) will be **sparse**. The matrix M is positive definite as

$$\mathbf{x}^T M\mathbf{x} = \sum_{i,j=1}^{N} x_i x_j m_{ij} = \sum_{i,j=1}^{N} x_i x_j(\phi_i, \phi_j) = (u,u) > 0, \quad u = \sum_{i=1}^{N} x_i\phi_i \neq 0, \tag{1.67}$$

because the ϕ_i are linearly independent and because $||u|| = \sqrt{(u,u)}$ is a norm. Similarly it is shown that

$$\mathbf{x}^T A\mathbf{x} \geq 0.$$

It is possible to have $\mathbf{x^T A x} = 0$ for a nonzero vector $\mathbf{x}$. This is the case if the constant function $u = 1$ is contained in S. This happens if Neumann boundary conditions $\frac{\partial u}{\partial n} = 0$ are posed on the *whole* boundary $\partial\Omega$. Then,

$$u(x,y) = 1 = \sum_i \phi_i(x,y),$$

i.e., we have $\mathbf{x^T A x} = 0$ for $\mathbf{x} = [1, 1, \ldots, 1]$.

1.6.3 A numerical example

We want to determine the acoustic eigenfrequencies and corresponding modes in the interior of a car. This is of interest in the manufacturing of cars, since an appropriate shape of the form of the interior can suppress the often unpleasant droning of the motor. The problem is three-dimensional, but by separation of variables the problem can be reduced to two dimensions. If rigid, acoustically hard walls are assumed, the mathematical model of the problem is again the Laplace eigenvalue problem (1.24) together with Neumann boundary conditions. The domain is given in Fig. 1.9 where three finite element triangulations are shown with 87 (grid_1), 298 (grid_2), and 1095 (grid_3) vertices (nodes), respectively. The results obtained with piecewise linear polynomials are listed in Table 1.2. From the results we notice the quadratic convergence rate. The smallest eigenvalue is always zero. The corresponding eigenfunction is the constant function. This function can be represented exactly by the finite element spaces, whence its value is correct (up to rounding error).

The fourth eigenfunction of the acoustic vibration problem is displayed in Fig. 1.10. The physical meaning of the function value is the difference of the pressure at a given location to the normal pressure. Large amplitudes thus means that the corresponding noise is very much noticable.

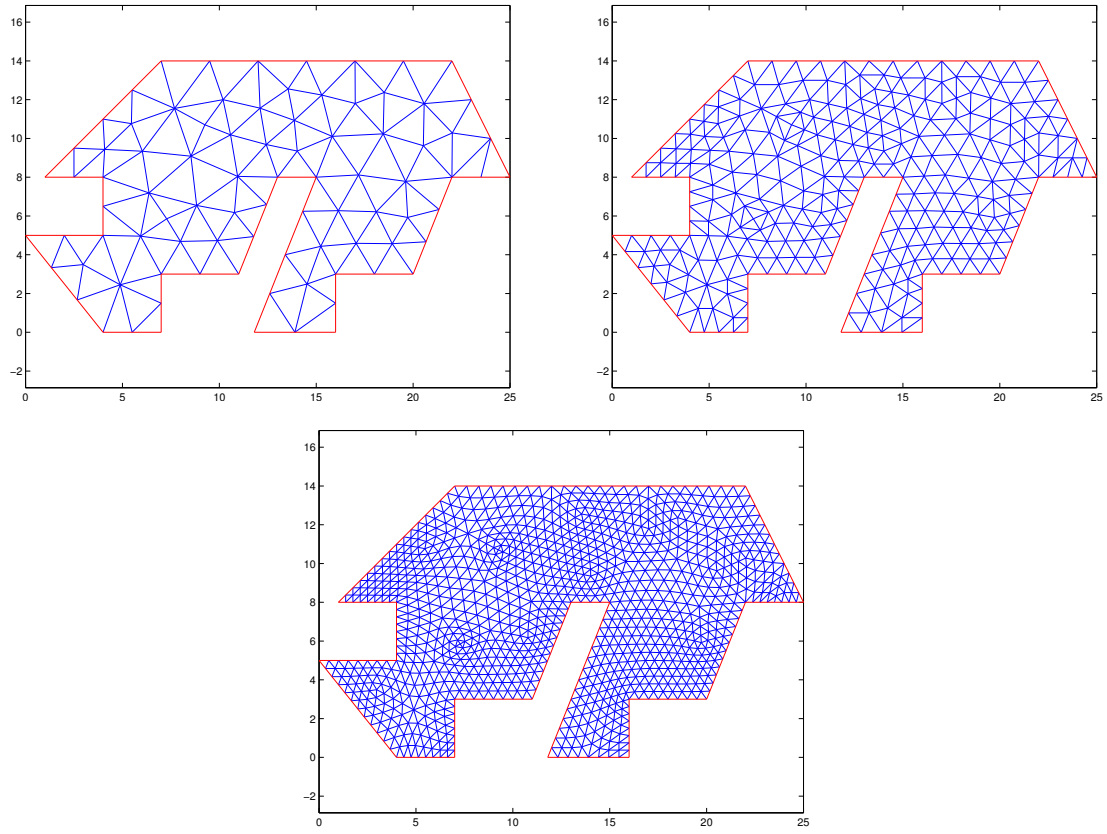


Figure 1.9: Three meshes for the car length cut

1.7 Cavity resonances in particle accelerators

The Maxwell equations in vacuum are given by

$$\begin{aligned}
\mathbf{curl}\,\mathbf{E}(\mathbf{x},t) &= -\frac{\partial \mathbf{B}}{\partial t}(\mathbf{x},t), && \text{(Faraday's law)}\\
\mathbf{curl}\,\mathbf{H}(\mathbf{x},t) &= \frac{\partial \mathbf{D}}{\partial t}(\mathbf{x},t) + \mathbf{j}(\mathbf{x},t), && \text{(Maxwell–Ampère law)}\\
\operatorname{div}\mathbf{D}(\mathbf{x},t) &= \rho(\mathbf{x},t), && \text{(Gauss's law)}\\
\operatorname{div}\mathbf{B}(\mathbf{x},t) &= 0. && \text{(Gauss's law – magnetic)}
\end{aligned}$$

where $\mathbf{E}$ is the electric field intensity, $\mathbf{D}$ is the electric flux density, $\mathbf{H}$ is the magnetic field intensity, $\mathbf{B}$ is the magnetic flux density, $\mathbf{j}$ is the electric current density, and ρ is the electric charge density. Often the "optical" problem is analyzed, i.e. the situation when the cavity is not driven (cold mode), hence $\mathbf{j}$ and ρ are assumed to vanish.

Again by separating variables, i.e. assuming a *time harmonic* behavior of the fields, e.g.,

$$\mathbf{E}(\mathbf{x},t) = \mathbf{e}(\mathbf{x})e^{i\omega t},$$

and by using the constitutive relations

$$\mathbf{D} = \varepsilon\mathbf{E}, \quad \mathbf{B} = \mu\mathbf{H}, \quad \mathbf{j} = \sigma\mathbf{E},$$

Finite element method			
k	$\lambda_k(\text{grid}_1)$	$\lambda_k(\text{grid}_2)$	$\lambda_k(\text{grid}_3)$
1	0.0000	-0.0000	0.0000
2	0.0133	0.0129	0.0127
3	0.0471	0.0451	0.0444
4	0.0603	0.0576	0.0566
5	0.1229	0.1182	0.1166
6	0.1482	0.1402	0.1376
7	0.1569	0.1462	0.1427
8	0.2162	0.2044	0.2010
9	0.2984	0.2787	0.2726
10	0.3255	0.2998	0.2927

Table 1.2: Numerical solutions of acoustic vibration problem

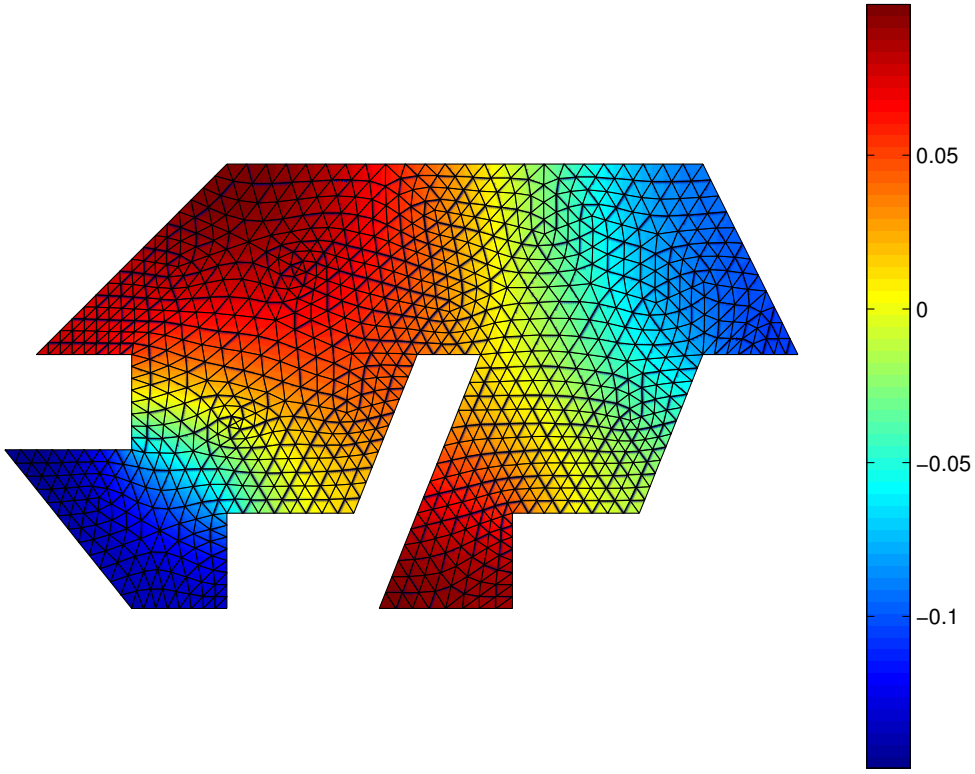


Figure 1.10: Fourth eigenmode of the acoustic vibration problem

one obtains after elimination of the magnetic field intensity the so called **time-harmonic Maxwell equations**

$$\begin{aligned} \mathbf{curl}\,\mu^{-1}\mathbf{curl}\,\mathbf{e}(\mathbf{x}) &= \lambda\,\varepsilon\,\mathbf{e}(\mathbf{x}), & \mathbf{x}\in\Omega,\\ \operatorname{div}\varepsilon\,\mathbf{e}(\mathbf{x}) &= 0, & \mathbf{x}\in\Omega,\\ \mathbf{n}\times\mathbf{e} &= 0, & \mathbf{x}\in\partial\Omega. \end{aligned} \tag{1.68}$$

Here, additionally, the cavity boundary $\partial\Omega$ is assumed to be *perfectly electrically conducting*, i.e. $\mathbf{E}(\mathbf{x},t)\times\mathbf{n}(\mathbf{x})=\mathbf{0}$ for $\mathbf{x}\in\partial\Omega$.

The eigenvalue problem (1.68) is a *constrained eigenvalue problem*. Only functions are taken into account that are divergence-free. This constraint is enforced by Lagrange multipliers. A weak formulation of the problem is then

Find $(\lambda,\mathbf{e},p)\in\mathbb{R}\times H_0(\mathbf{curl};\Omega)\times H_0^1(\Omega)$ *such that* $\mathbf{e}\neq\mathbf{0}$ *and*

(a) $(\mu^{-1}\mathbf{curl}\,\mathbf{e},\mathbf{curl}\,\mathbf{\Psi})+(\mathbf{grad}\,p,\mathbf{\Psi})=\lambda(\varepsilon\,\mathbf{e},\mathbf{\Psi}),\qquad \forall\mathbf{\Psi}\in H_0(\mathbf{curl};\Omega),$

(b) $(\mathbf{e},\mathbf{grad}\,q)=0,\qquad \forall q\in H_0^1(\Omega).$

With the correct finite element discretization this problem turns in a matrix eigenvalue problem of the form

$$\begin{bmatrix} A & C\\ C^T & O\end{bmatrix}\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix}=\lambda\begin{bmatrix} M & O\\ O & O\end{bmatrix}\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix}.$$

The solution of this matrix eigenvalue problem correspond to vibrating electric fields. A possible shape of domain Ω is given in Figure 1.11.

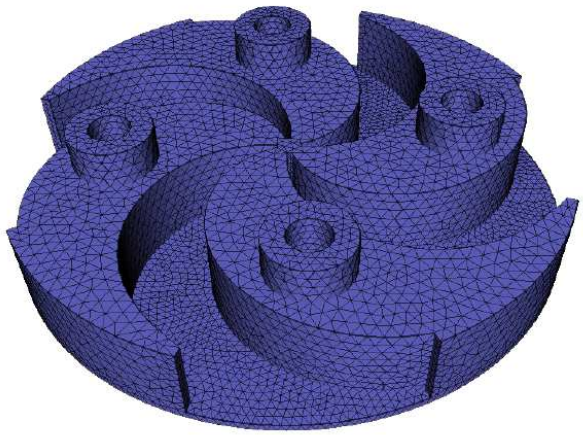


Figure 1.11: Comet cavity of Paul Scherrer Institute

1.8 Spectral clustering[3]

The goal of *clustering* is to group a given set of data points $\mathbf{x}_1,\ldots,\mathbf{x}_n$ into k clusters such that members from the same cluster are (in some sense) close to each other and members from different clusters are (in some sense) well separated from each other.

[3]This section is based on a tutorial by von Luxburg [12]. Thanks to Daniel Kressner for compiling it!

A popular approach to clustering is based on *similarity graphs*. For this purpose, we need to assume some notion of similarity $s(\mathbf{x}_i, \mathbf{x}_j) \geq 0$ between pairs of data points $\mathbf{x}_i$ and $\mathbf{x}_j$. An undirected graph $G = (V, E)$ is constructed such that its vertices correspond to the data points: $V = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$. Two vertices $\mathbf{x}_i, \mathbf{x}_j$ are connected by an edge if the similarity s_{ij} between $\mathbf{x}_i$ and $\mathbf{x}_j$ is sufficiently large. Moreover, a weight $w_{ij} > 0$ is assigned to the edge, depending on s_{ij}. If two vertices are not connected we set $w_{ij} = 0$. The weights are collected into a *weighted adjacency matrix*

$$W = \big(w_{ij}\big)_{i,j=1}^{n}.$$

There are several possibilities to define the weights of the similarity graph associated with a set of data points and a similarity function:

fully connected graph All points with positive similarity are connected with each other and we simply set $w_{ij} = s(\mathbf{x}_i, \mathbf{x}_j)$. Usually, this will only result in reasonable clusters if the similarity function models locality very well. One example of such a similarity function is the Gaussian $s(\mathbf{x}_i, \mathbf{x}_j) = \exp\big(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\big)$, where $\|\mathbf{x}_i - \mathbf{x}_j\|$ is some distance measure (e.g., Euclidean distance) and σ is some parameter controlling how strongly locality is enforced.

k-nearest neighbors Two vertices $\mathbf{x}_i, \mathbf{x}_j$ are connected if $\mathbf{x}_i$ is among the k-nearest neighbors of $\mathbf{x}_j$ or if $\mathbf{x}_j$ is among the k-nearest neighbors of $\mathbf{x}_i$ (in the sense of some distance measure). The weight of the edge between connected vertices $\mathbf{x}_i, \mathbf{x}_j$ is set to the similarity function $s(\mathbf{x}_i, \mathbf{x}_j)$.

ε-neighbors Two vertices $\mathbf{x}_i, \mathbf{x}_j$ are connected if their pairwise distance is smaller than ε for some parameter $\varepsilon > 0$. In this case, the weights are usually chosen uniformly, e.g., $w_{ij} = 1$ if $\mathbf{x}_i, \mathbf{x}_j$ are connected and $w_{ij} = 0$ otherwise.

Assuming that the similarity function is symmetric ($s(\mathbf{x}_i, \mathbf{x}_j) = s(\mathbf{x}_j, \mathbf{x}_i)$ for all $\mathbf{x}_i, \mathbf{x}_j$) all definitions above give rise to a symmetric weight matrix W. In practice, the choice of the most appropriate definition depends – as usual – on the application.

1.8.1 The graph Laplacian

In the following we construct the so called *graph Laplacian*, whose spectral decomposition will later be used to determine clusters. For simplicity, we assume the weight matrix W to be symmetric. The degree of a vertex $\mathbf{x}_i$ is defined as

$$d_i = \sum_{j=1}^{n} w_{ij}. \tag{1.69}$$

In the case of an unweighted graph, the degree d_i amounts to the number of vertices adjacent to v_i (counting also v_i if $w_{ii} = 1$). The degree matrix is defined as

$$D = \operatorname{diag}(d_1, d_2, \ldots, d_n).$$

The *graph Laplacian* is then defined as

$$L = D - W. \tag{1.70}$$

By (1.69), the row sums of L are zero. In other words, $L\mathbf{e} = 0$ with $\mathbf{e}$ the vector of all ones. This implies that 0 is an eigenvalue of L with the associated eigenvector $\mathbf{e}$. Since L

is symmetric all its eigenvalues are real and one can show that 0 is the smallest eigenvalue; hence L is positive semidefinite. It may easily happen that more than one eigenvalue is zero. For example, if the set of vertices can be divided into two subsets $\{\mathbf{x}_1, \ldots, \mathbf{x}_k\}$, $\{\mathbf{x}_{k+1}, \ldots, \mathbf{x}_n\}$, and vertices from one subset are not connected with vertices from the other subset, then

$$L = \begin{pmatrix} L_1 & 0 \\ 0 & L_2 \end{pmatrix},$$

where L_1, L_2 are the Laplacians of the two disconnected components. Thus L has two eigenvectors $\binom{\mathbf{e}}{\mathbf{0}}$ and $\binom{\mathbf{0}}{\mathbf{e}}$ with eigenvalue 0. Of course, any linear combination of these two linearly independent eigenvectors is also an eigenvector of L.

The observation above leads to the basic idea behind spectral graph partitioning: If the vertices of the graph decompose into k connected components $V_1, \ldots, V_k$ there are k zero eigenvalues and the associated invariant subspace is spanned by the vectors

$$\chi_{V_1}, \chi_{V_2}, \ldots, \chi_{V_k}, \tag{1.71}$$

where χ_{V_j} is the indicator vector having a 1 at entry i if $\mathbf{x}_i \in V_j$ and 0 otherwise.

1.8.2 Spectral clustering

On a first sight, it may seem that (1.71) solves the graph clustering problem. One simply computes the eigenvectors belonging to the k zero eigenvalues of the graph Laplacian and the zero structure (1.71) of the eigenvectors can be used to determine the vertices belonging to each component. Each component gives rise to a cluster.

This tempting idea has two flaws. First, one cannot expect the eigenvectors to have the structure (1.71). Any computational method will yield an arbitrary eigenbasis, e.g., arbitrary linear combinations of $\chi_{V_1}, \chi_{V_2}, \ldots, \chi_{V_k}$. In general, the method will compute an orthonormal basis U with

$$U = \big(\mathbf{v}_1, \ldots, \mathbf{v}_k\big)Q, \tag{1.72}$$

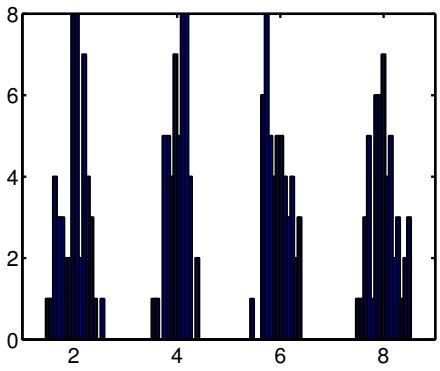
where Q is an arbitrary orthogonal $k \times k$ matrix and $\mathbf{v}_j = \chi_{V_j}/|V_j|$ with the cardinality $|V_j|$ of V_j. Second and more importantly, the goal of graph clustering is not to detect connected components of a graph[4]. Requiring the components to be completely disconnected from each other is too strong and will usually not lead to a meaningful clustering. For example, when using a fully connected similarity graph only one eigenvalue will be zero and the corresponding eigenvector $\mathbf{e}$ yields one component, which is the graph itself! Hence, instead of computing an eigenbasis belonging to zero eigenvalues, one determines an eigenbasis belonging to the k smallest eigenvalues.

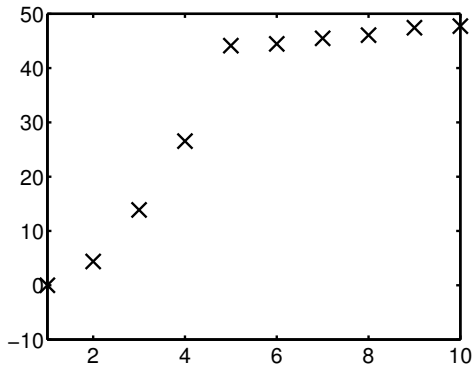
Example 1.1 200 real numbers are generated by superimposing samples from 4 Gaussian distributions with 4 different means:

```
m = 50; randn('state',0);
x = [2+randn(m,1)/4;4+randn(m,1)/4;6+randn(m,1)/4;8+randn(m,1)/4];
```

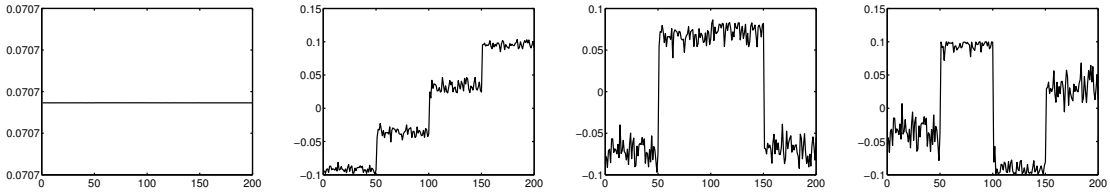
The following two figures show the histogram of the distribution of the entries of $\mathbf{x}$ and the eigenvalues of the graph Laplacian for the fully connected similarity graph with similarity function $s(\mathbf{x}_i, \mathbf{x}_j) = \exp\big(-\frac{|\mathbf{x}_i - \mathbf{x}_j|^2}{2}\big)$:

[4] There are more efficient algorithms for finding connected components, e.g., breadth-first and depth-first search.





As expected, one eigenvalue is (almost) exactly zero. Additionally, the four smallest eigenvalues have a clearly visible gap to the other eigenvalues. The following four figures show the entries of the 4 eigenvectors belonging to the 4 smallest eigenvalues of L:



On the one hand, it is clearly visible that the eigenvectors are well approximated by linear combinations of indicator vectors. On the other hand, none of the eigenvectors is close to an indicator vector itself and hence no immediate conclusion on the clusters is possible.

To solve the issue that the eigenbasis (1.72) may be transformed by an arbitrary orthogonal matrix, we "transpose" the basis and consider the row vectors of U:

$$U^T = \big(u_1, u_2, \ldots, u_n\big), \quad u_i \in \mathbb{R}^k.$$

If U contained indicator vectors then each of the short vectors u_i would be a unit vector e_j for some $1 \leq j \leq k$ (possibly divided by $|V_j|$). In particular, the u_i would separate very well into k different clusters. The latter property does not change if the vectors u_i undergo an orthogonal transformation Q^T. Hence, applying a clustering algorithm to $u_1, \ldots, u_n$ allows us to detect the membership of u_i independent of the orthogonal transformation. The key point is that the short vectors $u_1, \ldots, u_n$ are much better separated than the original data $\mathbf{x}_1, \ldots, \mathbf{x}_n$. Hence, a much simpler algorithm can be used for clustering. One of the most basic algorithms is **k-means clustering**. Initially, this algorithm assigns each u_i randomly[5] to a cluster ℓ with $1 \leq \ell \leq k$ and then iteratively proceeds as follows:

1. Compute cluster centers c_ℓ as cluster means:

$$c_\ell = \sum_{i \text{ in cluster } \ell} u_i \Big/ \sum_{i \text{ in cluster } \ell} 1.$$

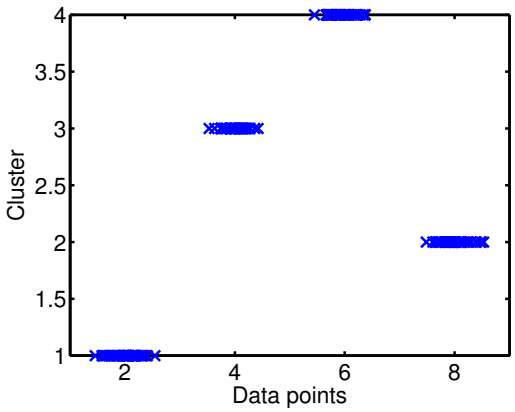
2. Assign each u_i to the cluster with the nearest cluster center.

3. Goto Step 1.

The algorithm is stopped when the assigned clusters do not change in an iteration.

[5]For unlucky choices of random assignments the k-means algorithm may end up with less than k clusters. A simple albeit dissatisfying solution is to restart k-means with a different random assignment.

Example 1.1 (cont'd). The k-means algorithm applied to the eigenbasis from Example 1.1 converges after 2 iterations and results in the following clustering:



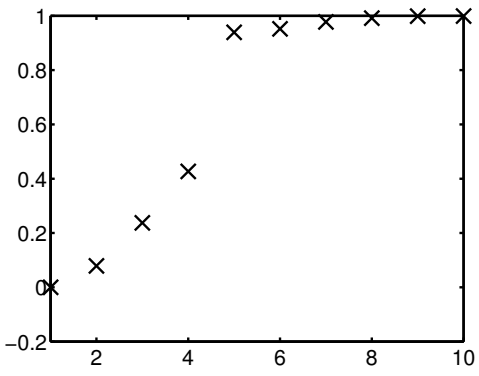
1.8.3 Normalized graph Laplacians

It is sometimes advantageous to use a normalized Laplacian

$$D^{-1}L = I - D^{-1}W \tag{1.73}$$

instead of the standard Laplacians. Equivalently, this means that we compute the eigenvectors belonging to the smallest eigenvalues of the generalized eigenvalue problem $W\mathbf{x} = \lambda D\mathbf{x}$. Alternatively, one may also compute the eigenvalues from the symmetric matrix $D^{-1/2}WD^{-1/2}$ but the eigenvectors need to be adjusted to compensate this transformation.

Example 1.1 (cont'd). The eigenvalues of the normalized Laplacian for Example 1.1 are shown below:



In comparison to the eigenvalues of the standard Laplacian, the four smallest eigenvalues of the normalized Laplacian are better separated from the rest. Otherwise, the shape of the eigenvectors is similar and the resulting clustering is identical with the one obtained with the standard Laplacian.

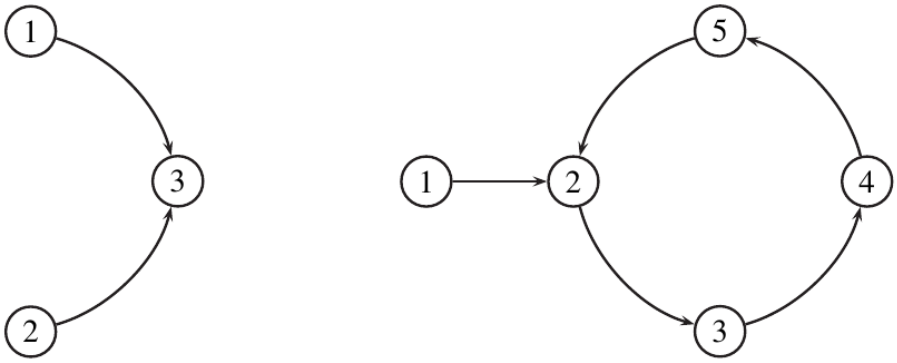


Figure 1.12: Things that can go wrong with the basic model: left is a dangling node, right a terminal strong component featuring a cyclic path. Figures are from [2]

1.9 Google's PageRank[6]

One of the reasons why Google is such an effective search engine is the *PageRank* that determines the importance of a web page [2, 10, 13]. The PageRank is determined entirely by the link structure of the World Wide Web. For any particular query, Google finds the pages on the Web that match that query and lists those pages in the order of their PageRank. Let's imagine a surfer brachiate through pages of the world wide web *randomly* choosing an outgoing link from one page to get to the next. This can lead to dead ends at pages with no outgoing links, or cycles around cliques of interconnected pages. So, every once in a while, the surfer simply chooses a random page from the Web. This theoretical random walk is known as a *Markov chain* or *Markov process*. The limiting probability that an infinitely dedicated random surfer visits any particular page is its PageRank. A page has high rank if other pages with high rank link to it.

Let W be the set of (reachable) web pages and let $n = |W|$. On `WorldWideWebSize.com`[7] it is estimated that Google's index contains around to 47 billion pages.

The elements of the *connectivity matrix* $G \in \mathbb{R}^{n\times n}$ is defined by

$$g_{ij} = \begin{cases} 1 & \text{there is a hyperlink } j \mapsto i, \\ 0 & \text{otherwise.} \end{cases}$$

Clearly, this is an extremely sparse matrix. The number of its nonzero elements `nnz(G)` equals the number of hyperlinks in W. Let r_i and c_j be the row and column sums of G,

$$r_i = \sum_j g_{ij}, \qquad c_j = \sum_i g_{ij}.$$

Then r_i is called the *in-degree* and c_j is called the *out-degree* of the jth page. $c_j = 0$ means a dead end.

In Fig. 1.13 we see the example of a tiny web with just $n = 6$ nodes. The nodes α, β, γ, δ, ρ, σ correspond to labels 1 to 6 in the matrix notation, in this sequence.

[6] Here we closely follow Section 2.11 in Moler's MATLAB introduction [13].
[7] http://www.worldwidewebsize.com/ accessed on Feb. 20, 2018.

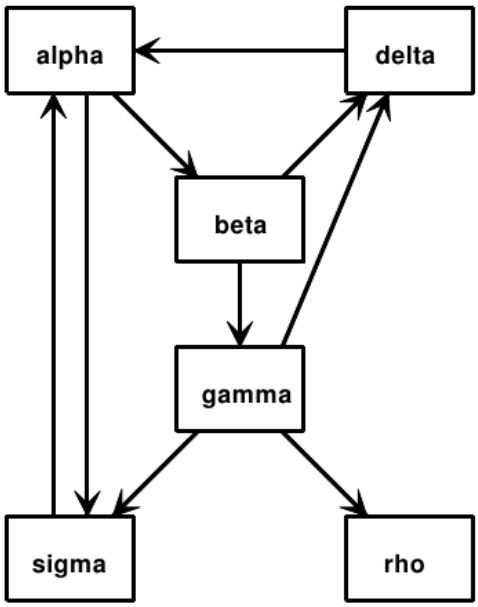


Figure 1.13: A small web with 6 nodes.

Then the connectivity matrix for the small web is given by

$$G = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

Notice the zero 5th column of G. This column corresponds to the dead end at the dangling node ρ.

Let A be the matrix with elements

$$a_{ij} = \begin{cases} g_{ij}/c_j & \text{if } c_j \neq 0 \\ 1/n & \text{if } c_j = 0 \text{ (dead end).} \end{cases}$$

In the small web example above,

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & \frac{1}{6} & 1 \\ \frac{1}{2} & 0 & 0 & 0 & \frac{1}{6} & 0 \\ 0 & \frac{1}{2} & 0 & 0 & \frac{1}{6} & 0 \\ 0 & \frac{1}{2} & \frac{1}{3} & 0 & \frac{1}{6} & 0 \\ 0 & 0 & \frac{1}{3} & 0 & \frac{1}{6} & 0 \\ \frac{1}{2} & 0 & \frac{1}{3} & 0 & \frac{1}{6} & 0 \end{bmatrix}.$$

The entries in A's column j indicate the probabilities of jumping from the jth page to the other pages on the web. Column 3, e.g., tells that starting from node 3 ($= \gamma$) nodes δ, ρ, σ are chosen with equal probability 1/3. Note that we choose any page of the web with equal probability when we land at a dead end.

To not be stuck to much in parts of the web, we follow the links only with probability α. With probability $1-\alpha$ we choose a random page. Therefore, we replace A by the matrix

$$\tilde{A} = \alpha A + (1-\alpha)\mathbf{p}\mathbf{e}^T,$$

where $\mathbf{p}$ is a *personalization* vector and $\mathbf{e} = (1, 1, \ldots, 1)^T$. ($\mathbf{p}$ has nonnegative elements that sum to 1, $\|\mathbf{p}\|_1 = 1$.) Note that $\mathbf{p}$ may have zero entries indicating, e.g., uncongenial, discredited, or discriminated web pages. We assume an innocent web and set $\mathbf{p} = \mathbf{e}/n$. Since $n \approx 50 \cdot 10^9$ in the real WWW, a typical entry of $\mathbf{p}$ is about $2 \cdot 10^{-11}$.

Note that

$$\mathbf{e}^T\tilde{A} = \mathbf{e}^T.$$

So, $1 \in \sigma(A^T) = \sigma(A)$, i.e., 1 is an eigenvalue of A with left eigenvector $\mathbf{e}$. Since the matrix norm

$$\|A\|_1 = \max_{1\le j\le n} \sum_{i=1}^{n} |a_{ij}| = 1,$$

A cannot have an eigenvalue larger than 1 in modulus. The *Perron–Frobenius theorem* for matrices with nonnegative entries states that such matrices have a simple real eigenvalue of largest modulus [4]. Therefore, the eigenvalue 1 is in fact the largest eigenvalue of A. We are not interested in the left eigenvector $\mathbf{e}$ but in the right eigenvector $\mathbf{x}$,

$$\mathbf{x} = \tilde{A}\mathbf{x}.$$

The Perron–Frobenius theory confirms that $\mathbf{x}$ can be chosen such that all its entries are nonnegative. If $\mathbf{x}$ is scaled such that

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

then $\mathbf{x}$ is the state vector of the Markov chain and is Google's PageRank.

The computation of the PageRank amounts to determining the largest eigenvalue and corresponding eigenvector of a matrix. It can be determined by vector iteration. The computation gets easier the smaller the damping factor α is chosen. However, small α means small weight is given to the structure of the web. In [13] a MATLAB routine `pagerankpow.m` is provided to compute the PageRank exploiting the sparsity structure of G.

1.10 Other sources of eigenvalue problems

The selection of applications above may lead to the impression that eigenvalue problems in practice virtually always require the computation of the smallest eigenvalues of a symmetric matrix. This is *not* the case. For example, a linear stability analysis requires the computation of all eigenvalues on or close to the imaginary axis of a nonsymmetric matrix. Computational methods for decoupling the stable/unstable parts of a dynamical system require the computation of all eigenvalues in the left and/or right half of the complex plane. The principal component analysis (PCA), which plays an important role in a large variety of applications, requires the computation of the largest eigenvalues (or rather singular values). As we will see in the following chapters, the region of eigenvalues we are interested in determines the difficulty of the eigenvalue problem to a large extent (along

with the matrix order and structure). It should also guide the choice of algorithm for solving an eigenvalue problem.

Saad [14] discusses further interesting sources of eigenvalue problems like electronic structure calculations, the stability of dynamical systems, or Markov chain models similar as Google's PageRank.

Bibliography

[1] A. Adelmann, P. Arbenz, and Y. Ineichen, *A fast parallel Poisson solver on irregular domains applied to beam dynamics simulations*, J. Comput. Phys., 229 (2010), pp. 4554–4566.

[2] U. M. Ascher and C. Greif, *A First Course in Numerical Methods*, SIAM, Philadelphia, PA, 2011.

[3] O. Axelsson and V. Barker, *Finite Element Solution of Boundary Value Problems*, Academic Press, Orlando, FL, 1984.

[4] A. Berman and R. J.Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, Academic Press, New York, 1979. (Republished by SIAM, Philadelphia, 1994.).

[5] O. Chinellato, *The complex-symmetric Jacobi–Davidson algorithm and its application to the computation of some resonance frequencies of anisotropic lossy axisymmetric cavities*, PhD Thesis No. 16243, ETH Zürich, 2005. (doi:10.3929/ethz-a-005067691).

[6] P. G. Ciarlet, *The Finite Element Method for Elliptic Problems*, North-Holland, Amsterdam, 1978. (Studies in Mathematics and its Applications, 4).

[7] R. Courant and D. Hilbert, *Methoden der Mathematischen Physik*, Springer, Berlin, 1968.

[8] H. C. Elman, D. J. Silvester, and A. J. Wathen, *Finite Elements and Fast Iterative Solvers*, Oxford University Press, Oxford, 2005.

[9] G. E. Forsythe and W. R. Wasow, *Finite-difference methods for partial differential equations*, Wiley, New York, 1960.

[10] A. N. Langville and C. D. Meyer, *Google's pagerank and beyond: the science of search engine rankings*, Princeton University Press, Princeton, N.J., 2006.

[11] R. J. LeVeque, *Finite Difference Methods for Ordinary and Partial Differential Equations*, SIAM, Philadelphia, PA, 2007.

[12] U. von Luxburg, *A tutorial on spectral clustering*, Stat. Comput., 17 (2007), pp. 395–416.

[13] C. B. Moler, *Numerical Computing with* Matlab, SIAM, Philadelphia, PA, 2004.

[14] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, SIAM, Philadelphia, PA, 2011.

[15] H. R. Schwarz, *Methode der finiten Elemente*, Teubner, Stuttgart, 3rd ed., 1991.

[16] ——, *Numerische Mathematik*, Teubner, Stuttgart, 3rd ed. ed., 1993.

[17] G. H. Shortley and R. Weller, *The numerical solution of Laplace's equation*, J. Appl. Phys., 9 (1939), pp. 334–344.

[18] G. Strang, *Introduction to Applied Mathematics*, Wellesley-Cambridge Press, Wellesley, 1986.

[19] H. F. Weinberger, *Variational Methods for Eigenvalue Approximation*, Regional Conference Series in Applied Mathematics 15, SIAM, Philadelphia, PA, 1974.